The Systems Journal

ICL

Special Topic – *The Millennium Programme*

Volume 13 Issue 1 *Autumn 1998 Edition*

ICL Systems Journal

All correspondence and papers to be considered for publication should be addressed to the Editor.

The views expressed in the papers are those of the authors and do not necessarily represent ICL policy.

Published twice a year by Group Technical Directorate, ICL, Kidsgrove.

1999 subscription rates (including postage & packing):

	UK and Europe	Rest of World
Annual subscription	£72	$120
Single issues	£43	$72

ISSN 1364-310X

ERRATUM

The date at the foot of the pages i to iv should be Autumn 1998 not November 1997

ICL Systems Journal

Volume 13 Issue 1

Contents

Front cover: Millennium Programme *Xtra*server. See the paper, "Trimetra *Xtra*server," in this issue.

Editorial

This issue of the ICL Systems Journal introduces three new features. Firstly, the format has been changed to A4 in line with most other similar journals. This change will provide greater flexibility in the layout and make it easier to accommodate detailed colour diagrams and photographs.

Secondly, guest editorials have been introduced from leading figures in the computing field, not necessarily directly linked to ICL. In view of ICL's partnership with Microsoft, the Editor invited Roger Needham, who now leads the Microsoft Laboratory in Cambridge, to write the first one. Roger is a busy man these days, dividing his time between being a Pro-Vice Chancellor of Cambridge University, running the Microsoft Laboratory and cruising at thirty five thousand feet over the Arctic!

Thirdly, future publications of the Journal will be known as the Spring and Autumn issues in order to match better ICL activities, such as the Annual Innovation Awards which usually lead to papers for the Journal.

Finally, this issue has, as a special topic, ICL's Millennium Programme. There are five papers in this issue concerned with various aspects of "Millennium" and more will be published in future issues.

V.A.J. Maller

Guest Editorial

In Cambridge there has been for a very long time a major centre of computer research, and recently Microsoft did as other companies have done before it and set up a research laboratory in the city. Why?

Firstly, why does Microsoft do research at all? A short answer is that a small company can get its agenda by looking around at the ambient world of its subject but a big company can't—it has to stay ahead, following its own agenda. That agenda comes from knowing what can be done, and how, before the next guy. The agenda comes from research. Microsoft started research at Redmond seven years or so ago and were soon delighted with what they had done. Although the research programme was genuinely that, and not a product development operation, after six years every major product had been influenced for the good. The response was to plan to multiply the research effort by three—which would involve hiring some 300 people. Not easy, but Microsoft are a tenacious lot!

It did however quickly become apparent that not every good computing researcher wanted to move to Redmond, Washington. After some discussion, with which I was naturally not involved, they decided to open a Laboratory in Europe—and Cambridge was one of the more obvious places to consider, for the same sort of reasons that attracted others. So here we are, established in central Cambridge three minutes from the University Department, getting on for half the way towards our target of 40 researchers by mid-99. I asked Rick Rashid, Vice-President for Research what he wanted us to do; his response was that I presumably knew what sort of business Microsoft were in; I should hire the best people there were and have them do what they were good at. The Chief Technology Officer, Nathan Myhrvold, was present and added that if every project I started succeeded, I had failed. Given that I was old enough at 62 that the concept of career risk didn't exist it was an easy decision to accept the proposition. So we are looking for the best people to hire, and gently putting together a team.

We currently see our areas of activity as being reasonably diverse. We are active in programming language theory—a type of research directed towards improving the process of applying computers by making it easier to design and implement large programs. We engage in statistical learning theory and pattern recognition—key to such techniques as hand-written input. We would dearly like to solve the problem of making computers pleasant to read long documents with. Today you can't write without a computer and you can't read with one! We intend to make progress with understanding large networks of computers—learning to treat computers as a collective, albeit a somewhat fallible one. We shall tackle with energy the management, indexing, and retrieval of all kinds of digital information. But the over-riding requirement is excellent people. When they started research at Redmond they knew they wanted to do graphics—but didn't start for three years because they couldn't find a good enough leader. Now they are world class and unexcelled. We are determined to have a first-class outfit and to be a real help to the Company.

Being valuable to the Company is a serious challenge because we are so far way. Partly we can tackle it through the help of research colleagues at Redmond, and partly it has to be done by personal contact, which, for all the magic of electronic communication, is good for the airlines. We are already working with two Redmond groups outside research, which can't be bad.

Roger Needham

An Architecture for Commercial On-line Internet Services

Nigel Dyer

ICL, Bracknell, UK

Abstract

ICL has entered a three year agreement with the BBC Worldwide (the commercial element of the British Broadcasting Corporation) to build the skills, knowledge and experience of providing and supporting the technology for on-line media publishing into the consumer Internet marketplace.

The partnership is developing a new brand, 'beeb @ the BBC', which exploits the existing BBC brand and is focused on delivering against a medium term target for investment return target by harnessing the opportunity provided by advertising, subscription and e-commerce revenues for on-line content.

Eighteen months into the agreement and twelve months after the initial launch of web content and services, this paper outlines the significant technology elements which underpin this commercial venture.

1. Introduction

The importance of the Internet as a publishing medium and, in the future, as a platform for electronic trading, is widely publicized and recognized by organizations from large to small.

Both ICL and the BBC are optimistic that the Internet market will continue its rapid growth—a growth which far outstrips the market acceptance of television and radio in their early days—and that the medium will deliver sustainable revenue streams for major players.

The revenue streams in the short term will accrue from:

- subscriptions to web content (including pay-to-play games)
- advertising
- an on-line BBC Shop (offering CDs, videos and books).

In the medium to long term, additional revenues can be sought through:

- licensing content to third parties
- international distribution, particularly in North America
- commerce integrated with content
- premium on-line services.

In addition, the business partnership may offer to consumers a branded internet access capability which would exploit ICL's existing network infrastructure.

The creation of a successful business in this new marketplace is dependent upon bringing together the creative talent, brands and intellectual property of the BBC with the technical expertise, system development and integration skills of ICL.

For ICL this project is an opportunity to gain commercial and technical experience as well as expertise in the fields of engineering, marketing and supporting on-line services through ICL's network, thus positioning itself as a provider of services to the media industry. For the BBC it is an opportunity to develop new on-line products and services, drawing on its consumer magazines, consumer publishing businesses and the BBC's TV and radio programme archives, as well as enabling the editorial staff to expand their creative skills in the new medium.

The partnership was agreed in August 1996, to run until the end of 1999. The first six months of the project saw the development of the major elements of the infrastructure to support the production and delivery of multimedia content. The first product launches, a limited trial BBC Shop (www.bbcshop.com) and an on-line Fantasy Formula One subscription game (www.ff1. beeb.com), were launched in December 1996 and February 1997 respectively, followed by a limited service launch in June 1997 which included Sport (www.sport.beeb.com) and the Radio Times (www.rtguide.beeb.com). Since then, new and enhanced products have been launched every month at www.beeb. com.

Audited in March 1998 for its traffic figures by ABC (the Audit Bureau of Circulations, www.abc.org.uk), the BBC as a whole recorded 66.7 million page deliveries with beeb @ the BBC delivering 12.2 million. This places beeb @ the BBC as one of the top five leading entertainment web sites in the UK.

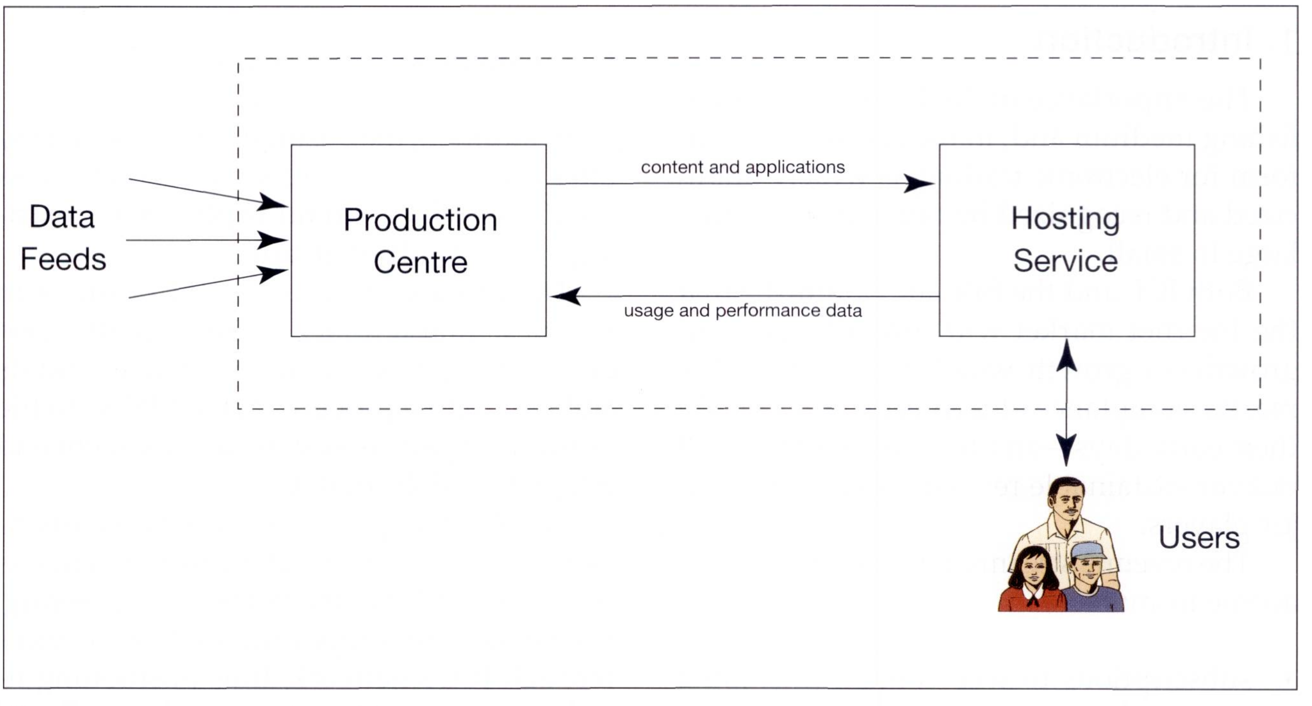

Figure 1: Top Level Architecture

2. Objectives

From its concept, beeb @ the BBC (hereinafter referred to as *beeb*) has set out to be entertaining, fun and engaging with its audience. The uniqueness and depth of potential BBC material for a commercial on-line service is second to none. Its access to personalities and brands from both TV and Radio programmes as well as its highly successful magazine publishing business is unparalleled.

The design impact and immediacy of material is critical in gaining audience loyalty. Production teams must be able to edit a story or feature rapidly and publish it on the web without the knowledge of HTML and underlying IT systems.

To support the goals of the business, we have established the technology platform under a 'best of breed' principle. That is, we have identified and procured the best and most appropriate technology which could be identified to meet our specific requirements. Obviously, in some areas we have built the software needed since no off-the-shelf components exist.

We have designed the infrastructure to support thousands of concurrent end-users, with the ability to scale the services to meet the demands of even more, whilst ensuring that we are able to deliver good performance and high availability.

3. Architecture

Figure 1 illustrates the top-level structure of *beeb* with the key dataflows.

The *Production Centre* has been built on a BBC site (the Television Centre in London) where a total business operation exists within the facilities provided. Overall business management functions determine the priorities of the investment, including marketing spend, and the operation includes commercial and legal affairs and advertising sales. Production and development staff create and maintain content and applications which are loaded on to the *Hosting Service* for access by *Users*.

Data Feeds is the collection, conversion and storage of text, stills, video and audio material which may be used by production staff within the content services. The Data Feeds may come from within the BBC, e.g. Ceefax, and the BBC digital media archive, or from a third-party, e.g. Broadcast Data Services, who supply all television and radio programme listings data. Feeds may be automatic or on-demand.

The *Hosting Service* consists of two server farms on which reside a number of IT systems. Two server farms were chosen to ensure high availability of services and to provide for disaster recovery. The systems on each server farm are used for final integration and performance testing as well as serving requests from *Users*.

Users encompass anyone with Internet access. A user may be a subscriber to a service who is known, authenticated and verified by a credit card debit and who may access a service by username and password, e.g. *beeb* runs a Fantasy Formula One pay-game at www.ff1.beeb.com. A user may be registered in that he or she has provided a basic level of non-verified information about themselves in order to be presented with personalized information, e.g. a postcode is required before regional television and radio listings can be made available in customized content (www.rtguide.beeb.com). Alternatively, a user may be anonymous and may only access a restricted set of content which cannot be personalized by geography or individual preference. Users access the internet through their Internet Service Provider, primarily with Netscape Navigator (2 or above) or Internet Explorer (3 or above) worldwide web browsers, or derivatives thereof. No assumptions are made with respect to availability of particular browser plug-ins nor a Java Virtual Machine (JVM) although the gradual trend is for users to have such capabilities which enables *beeb* to exploit the features and facilities within the offered services.

4. Production Centre

The Production Centre is at the very heart of the *beeb* operation. The key functions and business processes are executed here.

The strategic vision is guided by the Creative Director, Marketing, Advertising, Commercial Affairs, Production and Technology.

Content for individual areas of the web site (known as webzines—magazines for the WWW), application services and associated content related activities are taken from concept to reality by producers. Researchers, editors, journalists and design staff create the actual web page components and finished products.

An applications development team provides the key applications and the technical integration of them with the content. A technical support team develops and maintains the Production Centre IT infrastructure, provides essential services to support business functions, maintains deployed systems and applications and provides problem and change management services.

The Content Production System provides a collaborative environment where production staff can easily create, share, store, preview and release content to the Hosting Service in a controlled way.

A Multimedia Studio provides high-end systems with specialized facilities to enable production staff to capture, convert, edit and store multimedia data for use within the web site.

A desktop platform includes both Macintosh and Windows PC workstations with standard office software including web browsing, electronic mail, as well as meeting and resource scheduling. In most cases, software must be cross-platform. Specialized tools are provided for the appropriate roles:

Designers: graphics, templates, web page/HTML, animation, format conversion editors

Developers: code development, compilers, debuggers, configuration management, test

Advertising Sales and Marketing: contact management, finance.

4.1.1 Network

The BBC control the network up to and including the firewalls and ensure that systems are maintained in accordance with the BBC Information Security Policy. Dial-up access is provided for home and off-site users via the Remote Access Server (Shiva LanRover) which is configured to provide static dial-back after user authentication. This satisfies the BBC's policy for dial-up access since the system calls out on a different line.

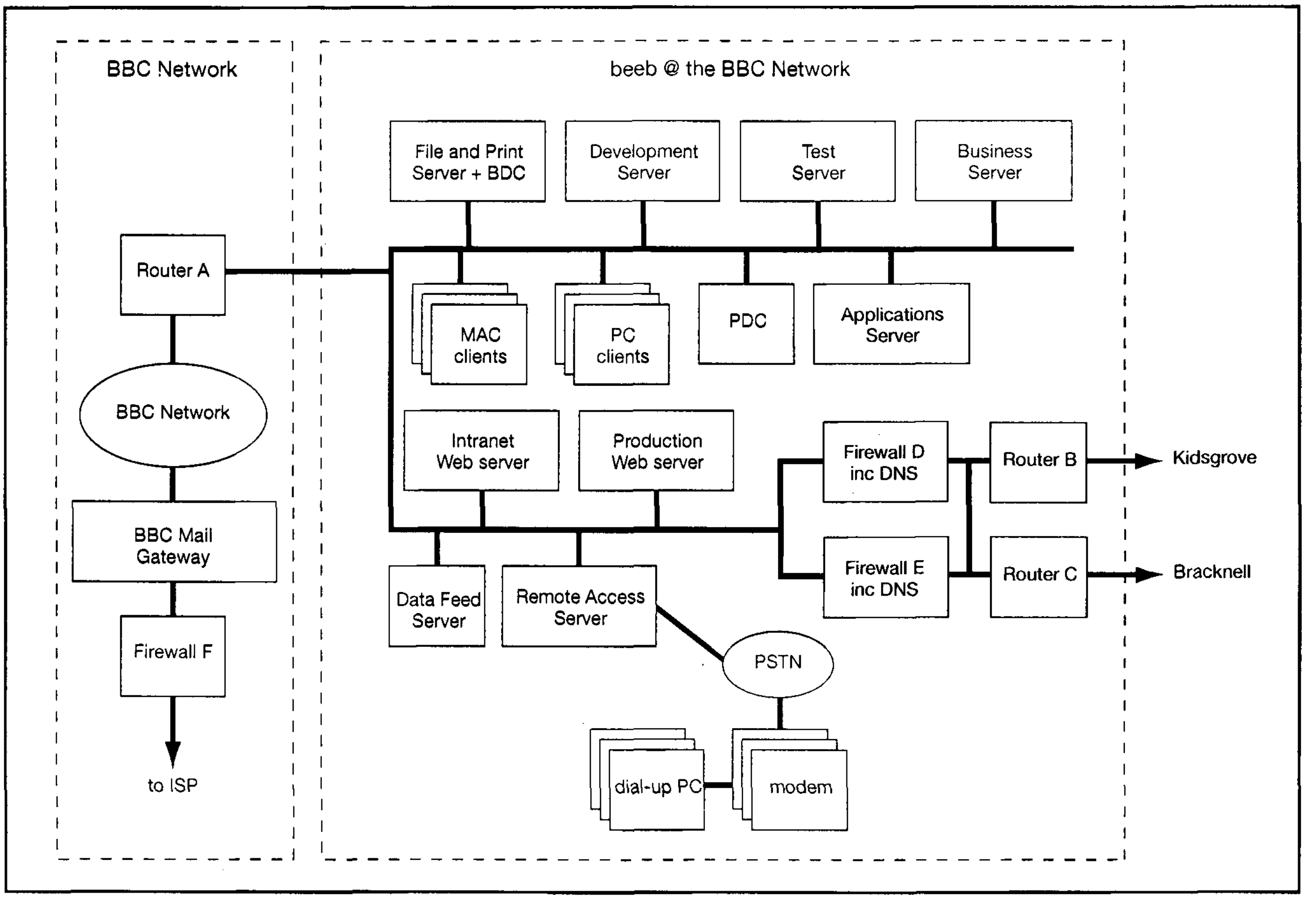

Figure 2: Production Centre Technical Architecture

4.1 Technical Architecture

The Production Centre Technical Architecture, shown in Figure 2, provides the IT infrastructure to enable:

- data feeds
- production of content
- deployment of content from the production centre to the hosting service
- e-mail and remote access services for *beeb* personnel
- Internet and BBC intranet access
- protection of systems from attack.

Email services are provided:

1. using the BBC compatible Microsoft Mail post office which connects into the BBC mail services (firstname.lastname@bbc.co.uk) and routes out to the Internet through the BBC's gateway
2. using the Netscape Mail Server hosted at Kidsgrove running a POP3 (Post Office Protocol) server and accessed by any POP3 compliant client (firstname.lastname@beeb.com).

Class C IP addresses allocated by the BBC are used by all systems. However, all servers (except for Feeds) use a specific range within the subnet and are filtered by the router between *beeb* and the rest of the BBC.

Protection for both the BBC and *beeb* from the Internet is provided by the dualled firewalls D & E. These firewalls have one *beeb* address and one ICL*net*2 address.

Independence between the BBC and *beeb* parts of the network is achieved by placing filters in router A to allow only the PC and Macintosh systems within *beeb* to get across the router and access the BBC network—*beeb* servers are blocked (except for the Feed Server).

The dualled routers B and C have ICL*net*2 addresses.

4.1.2 Data Feeds

Data Feeds enter the Production Centre either from other parts of the BBC (e.g. Ceefax or Radio 5 Live) or from a third-party (e.g. the Press Association). The data has multiple input formats and is of multiple data types (text, picture, audio, video). A Multimedia Design Studio contains the systems and tools to capture, convert, edit and store the data in a format usable by *beeb*.

Some data requires checking and manually integrating with editorial input to provide, for example, a picture with a story for the Top Gear (www.topgear.beeb.com) webzine. Other data may be automatically identified, parsed and deployed to the hosting service with minimal or no human intervention, e.g. the latest cricket score (www.scorewatch.beeb.com). Data feeds will typically be either merged with a predefined template to produce a page of static HTML, or loaded into a database for attention by an editor or journalist.

The Feed Server is allocated an IP address to enable file transfers via FTP from the BBC.

4.1.3 Systems

The clients systems are a mix of Apple Macintosh and Intel (Windows 95 and Windows NT Workstation 4.0) Personal Computers. Low-end client systems are configured to enable testing the site against other configurations (such as Windows For Workgroups on a 486 processor). Comparative testing is also undertaken by using dial-up access to common UK Internet Service Providers to access *beeb*.

The servers are mainly Intel systems running Windows NT 4.0, while SUN SPARC systems are used for code development, testing and production. A Primary Domain Controller manages secure access to the Production Centre network and resources. Peripherals for printing and scanning are configured as shared resources attached to Windows NT servers.

Server storage is allocated to Production Teams and managed by Quota Manager.

Shared applications are loaded from an office Applications Server. An Intranet is maintained to share information and provide access to a document library.

A Digital Studio provides tools to enable the capture, conversion and storage of multimedia data from the RF ring main, e.g. frame capture, for use by production teams into the web content. It includes Digital Audio Tape (DAT) and mini disc players, Hi-8 recorders, FM tuners, television systems, equipment for transferring between multiple tape formats and tools for creating Quicktime movies and RealAudio/RealVideo files.

4.2 Production Process

The production process begins with a brief which scopes the idea and opportunity or need for the service or facility. The brief can be content-focused or technology-focused. A response to the brief which outlines the options and costed proposal is reviewed and allocated resources on business approval. The virtual project team—a mixture of technology and media personnel—is thereafter empowered to manage its deliverables against the plan.

Rapid prototyping methods and tools are used for content-related application development while traditional software engineering methodologies are employed where infrastructure and enabling facilities are required. Designers, researchers and editors incrementally build the system on the Test server, integrating with applications built and tested on the Development server by the applications team. These servers mirror as closely as possible the delivery systems on the Hosting Service. The Development server will also include some development facilities such as program development, compilers and debugging tools.

The project team completes the integration, testing and revision cycle and then mirrors the applications and content on to the Hosting Service. The new content is made accessible to users by i) incorporating a new URL into the service by updating the Primary Domain Name Server and ii) updating the overall *beeb* navigation scheme, i.e. adding a new webzine to a site-wide navigation toolbar.

4.2.1 Production Tools

There are mainly four types of user within the Production Centre who use a mixture of common and bespoke tools to produce content:

- Designers who produce HTML templates, graphics and animations (such as Flash or Shockwave) will use tools such as those provided by Adobe and Macromedia.
- Editors and journalists who use word processing facilities and bespoke tools to reference, for example, related articles. Such tools normally integrate with a database environment and only expose HTML tags which are relevant to the journalist, e.g. <b> and <p> for bold and paragraph.
- Multimedia researchers who will capture, edit and transform audio and video from BBC archived material. This can be commissioned externally and forms the basis of a multimedia content store.
- Application programmers who will use professional programming environments to provide Java applets, Javascript and CGI programs to integrate into the finished HTML.

Producers and Production Assistants make use of standard (Microsoft) office tools including email facilities, fax, scanners and printers.

4.3 Business Services

4.3.1 Rights and Media Management

Rights Management is provided within a media management system—a digital asset library which provides the metadata for recording copyright, licence, usage, termination and royalty payment obligation information for individual objects.

4.3.2 Traffic Measurement

A Critical Success Factor for a commercial venture such as *beeb* is understanding the different aspects of the web visitor's experience and the relative popularity of various parts of the site to:

- enhance content and navigation in order to maximize a user's interest and length of stay
- present audited traffic figures to potential advertisers
- use as a basis for investment decisions.

Critical information to assess the effectiveness of the site includes:

- number of views per page (page impressions)
- number of unique visits/visitors
- average time spent per page
- number of pages viewed per visit
- paths which visitors take through the site
- any error pages which are served by the web server
- effectiveness of advertising (measured by "click-throughs" from the advertisement).

The requests for pages are written to web server logs which are segmented on a daily basis. Overnight scheduled batch jobs rotate the logs on each web server within the Hosting Service and compress them. They are then transferred into the Production Centre, decompressed, and loaded into a database on a business server. Applying filters ensures that *beeb*'s own surfing and search engine software robots do not count towards daily traffic (the exclusion list is defined by ABC, www.abc.org.uk). The compressed logs constitute many hundreds of megabytes of files.

A reporting package—Microsoft Usage Analyst (a component of Site Server)—is used to generate the reports of usage by individual website.

4.3.3 Advertising Management

An advertising management system allows advertising campaigns to be established for various areas of the content. The delivery system enables advertisements to be rotated and spread across a date range, with daily reporting on the number served and the quantity of "click-throughs"—the number of times an end-user has clicked on the advertisement to be taken to the advertiser's site. In the future, advertisements may be targeted according to the usage profile of the user. This profile can be built up either by careful analysis of the web server logs, which shows the trail each user has taken through the site, or via some form of explicit registration.

4.3.4 Marketing

In addition to using output from the traffic measurement system, Marketing are responsible for advertising new content sites and services. Use of electronic mailshots to support specific sales or awareness campaigns to users, who have provided personal details within the site, are also offered, as well as Audience research and on-line feedback.

4.4 Deployment To Live Servers

The deployment process between the Production Centre and live services is a two-stage activity:

1. Deployment through the BBC firewalls to a Staging Server within the Hosting Service
2. Checks, archive and deployment of the appropriate content from the Staging Server to one or more physical servers within the Hosting Service.

This enables:

- the Production Centre systems and personnel from needing to know where particular services are hosted
- services to be migrated between different physical servers for effective load sharing without the need to change systems within the Production Centre
- services to be hosted from more than one physical server
- a secure record of the deployed content to be made and archived within the Hosting Service (a legal and contractual requirement).

5. Hosting Service

The *beeb* Hosting Service consists of three elements:

- Server farms
- Interconnection network
- Internet connections.

5.1 Architecture

Figure 3 (over the page) shows the network and server architecture for *beeb's* Hosting Service.

The network consists of three major elements: the ICL*net*2 backbone provided by BT as part of their SMDS (Switched Multimegabit Data Service) infrastructure (this network connects Bracknell, Kidsgrove and the BBC Television Centre), the interconnection to the Internet via the LINX exchange (London INternet eXchange at Docklands Telehouse) and the physical web server farms.

The architecture of the infrastructure has been designed to provide a high speed and highly available communications network. Dual lines have been installed between each location, with each line entering via an independent routing system, so that if a line fails a backup is always available to carry the traffic.

The physical server farms each contain multiple servers and use router technology to connect them via a point-to-point link with

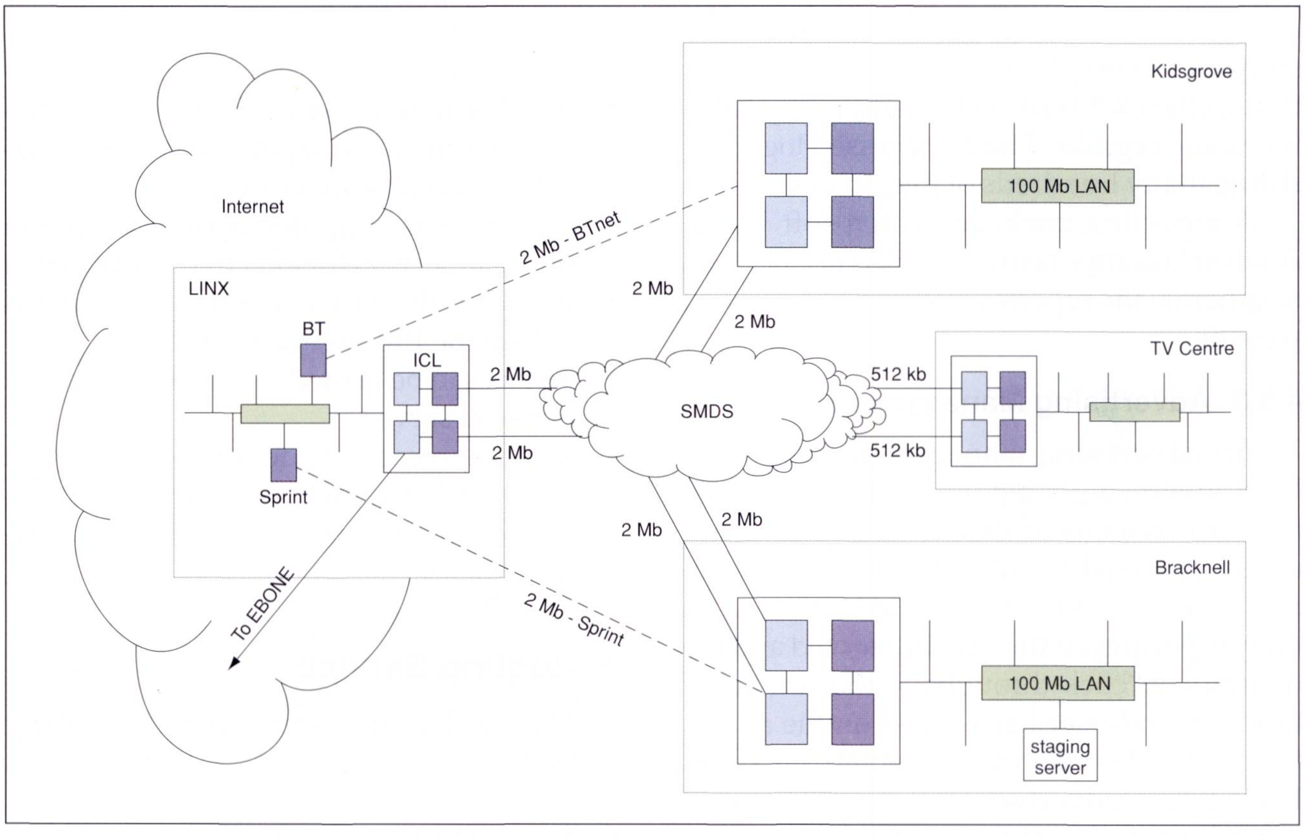

Figure 3: Hosting Service Architecture

the BT SMDS network. The Local Area Network at each web server farm provides FDDI 100Mb/sec. performance.

Redundant network links, terminal servers and routers are provided to enable resilience for high availability and load balancing.

ICL is now a 'Tier 1' Internet Service Provider and is able to place capacity directly into LINX via SMDS. This gives ICL greater control of the network and the flexibility to upgrade its capacity quickly. ICL can also remove its dependency on BTnet and Sprint.

The servers are a mixture of SUN Sparc/Solaris systems and Fujitsu Intel/Windows NT systems. This dual architecture strategy enables *beeb* to select, deploy and maintain any application on the preferred system, without encountering any delays imposed by any third party who may need to undertake platform ports. Such a delay could have an impact on *beeb's* advantage for rapid exploitation for which there is typically a very small window. Furthermore, it enables ICL to maintain and grow its skills in both platforms and assess the appropriate serving environment for each set of specific application requirements.

5.2 Staging Server

The Staging Server is located within the Hosting Service and is the only interface from systems within the Production Centre—essentially it is the only exit point for Producers to populate elements of the web site.

When content has been produced, tested and approved, the content elements are assembled, translated (where required, for example, to provide unique filenames) and delivered to the staging server. The content is then sent automatically (either immediately or deferred) to the appropriate Hosting Service server(s). Normally a final test will be undertaken before a directive is issued to make the content live (again, via the Staging Server).

The deployment mechanism between the Staging Server and the Hosting Service servers consists of bespoke systems integrated by ICL to automate fully the process of extracting new content from the staging server and deploying it to the appropriate servers for delivery to users. This bespoke development was necessary as there were no known products in the marketplace to meet *beeb's* requirements.

5.3 Content and Application Server

SUN SPARC systems contain the web content—all HTML, GIFs, CGI programs, Java, and Shockwave etc.. Programs and code which require data from a database make a call to one of the database servers. This architecture enables Content and Database servers to be managed and scaled independently of each other.

5.4 Database Server

Dual-processor SPARC systems host a replicated Oracle database for scalability and load balancing. The database contains content for the webzines, e.g. new and used car prices (www.topgear.beeb.com), as well as user registration data. Other information includes the data posted by users within *beeb's* forum product (beebForum – www.forums.beeb.com) and the backend advertising management system.

5.5 Audio/Video Server

A single processor INTEL system hosts a RealServer server for audio and video streaming. Due to network limitations and quality of service for the end-user, *beeb* is currently employing video streaming content very sparingly.

5.6 Forum and Chat Servers

Single processor INTEL systems host the Forum and Chat services (see section 6.5).

5.7 BBC Shop

A BBC Shop (www.bbcshop.com), where end-users can browse and purchase BBC branded goods is provided within the Hosting Service. The shop is integrated with the content services, e.g. at Comedy Zone a user can navigate to related comedy videos in the shop (www.comedyzone.beeb.com).

Products can be bought securely on-line (via SSL) by credit card from a catalogue. The system processes orders and payments and integrates with a third-party fulfilment agent, who arranges delivery of the products from stock ordered and warehoused from BBC Worldwide. A telephone help and customer care service is provided to handle all user queries arising from the shopping service.

The BBC Shop is a bespoke development at present but it is likely that it will move to a standard e-commerce platform, such as Microsoft Site Server, in due course.

6. Application Architectures

Early web HTML content was served from simple flat files—text, pictures and other multimedia elements. All but the largest and most complex sites still predominantly use this style today. The introduction of the CGI standard (Common Gateway Interface) enabled some interaction with the end-users by allowing them to submit data within forms. This facilitated the delivery of content according to a user's supplied preferences.

Additional programming logic at the client browser has further enabled a high degree of interactivity and graphical animation tools, and proprietary plug-ins, such as Macromedia Shockwave, have started to bring the web to life. Intelligent web pages with embedded applets are beginning to interact with both the user and remote services to deliver a highly tailored experience.

beeb has developed and delivered webzines which cover the spectrum of application architectures: simple flat sites through relational database driven content sites to an object-oriented approach with ICL COMMANDS.

6.1 Static and Dynamic Content

The World Wide Web is primarily concerned with publishing information in a common format (HTML) which can be accessed via common protocols (HTTP) by client devices driven by users, irrespective of the data location.

Traditional web sites consist of published information which is formatted into pages and stored separately as HTML files. Whenever a page changes, all other pages, which refer to it or its content, also need to be changed. This is acceptable for small sites (less than 100 pages) which do not change frequently, e.g. 10 pages changed once per day. However, any form of personalization becomes difficult. In particular, if sophisticated scripting is used to provide the user with a high-end experience, which is possible with the most recent browsers, (flavours of dynamic HTML are supported in Netscape Navigator 4 and Internet Explorer 4) then the browser type and level must be 'sniffed' from data contained within the HTTP request header and the server must determine which file to return to the client (this can be seen on the homepage at www.beeb.com). Nevertheless, static publishing should always be used wherever feasible since it offers the highest potential for scalability and maximum performance as pages will be cached by proxy servers. The information can also be mirrored and dual-hosted for resilience at an affordable cost.

Dynamic web sites such as RT Guide (www.rtguide.beeb.com) are designed to work with rapidly changing information and to enable information to be viewed differently by each user based on the user's profile. Typically a CGI program acts as the template. Instead of requesting a file of HTML, the hyperlink on the web page causes the web server to run the CGI program. The spawned process extracts the relevant data from a database and present it back to the browser.

6.2 Database Integration

Extensions to a scripting language like PERL, which is used to develop CGI programs, enable CGIs to integrate with databases in a portable and non-proprietary way. Figure 4 shows the architecture.

The DBI (DataBase Interface) provides an abstraction layer which can be used to implement a Database Driver (DBD) for a specific data store, similar to ODBC. The DBD layer translates DBI calls into a form that can be understood by a database. This approach has the benefits of openness—CGI is an industry standard, which is portable, widely available and implemented in web servers—Perl and the DBD/DBI database abstraction libraries are also widely available across multiple platforms with drivers for all the leading database systems. However, the approach also has a penalty—scalability.

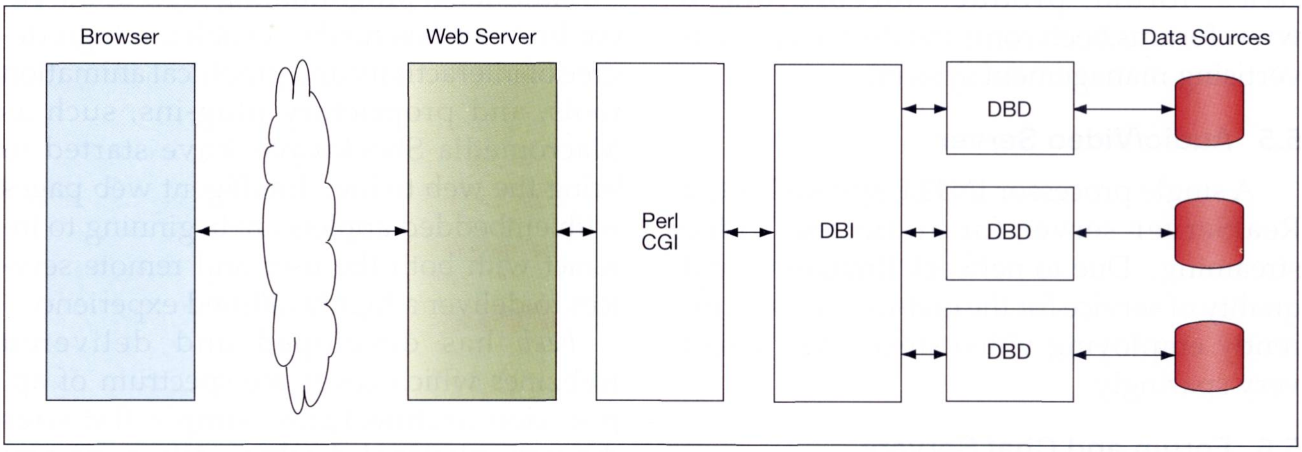

Figure 4: Perl CGI with DBD/DBI Extensions

6.3 Application Scalability

Although the CGI is an open and widely implemented standard it has a number of drawbacks:

- A separate new CGI process is spawned for every request, creating overheads from the operating system. Starting a new process for each request eliminates any potential efficiencies of persistent data

- A rapid CGI development language, such as Perl, incurs a high overhead since it runs as interpreted code (although compilers do exist). This incurs a large memory overhead
- If the CGI makes a connection to a database then the connection is lost when the CGI program exits and the next time the same process is invoked it must establish connection to the database again—typically a very high overhead for most database management systems.

At peak times of the day when most traffic comes to a web site it is possible to run out of system resources due to the inherent inability to scale the CGI.

Web server vendors have developed APIs—NSAPI from Netscape and ISAPI from Microsoft. Applications linked into the server API are likely to be much faster than CGI equivalents. The CGI startup and initialization overhead is removed and the application is persistent across requests. However:

- the web server APIs are proprietary to each vendor and they are difficult to write and maintain
- applications must be written in a language supported by the vendor (typically C or C++)
- there is no process isolation—the application runs in the same address space as the web server, so it can crash or impact both other applications and the web server itself.

There are a number of approaches to dealing with the issue of application scalability, some of which can be used in combination to provide the appropriate solution for a specific requirement.

6.3.1 Web Server Application Programming Interfaces

In response to the performance problems for CGI, major vendors have developed APIs for their servers—NSAPI from Netscape and ISAPI from Microsoft.

Applications linked into the server API are likely to be much faster than CGI programs. The CGI startup/initialization problem is improved, because the application runs in the server process and is persistent across requests. Web server APIs also offer more functionality than CGI.

However, web server APIs sacrifice all of CGI's benefits. Vendor APIs have the following problems:

- web server APIs introduce a steep learning curve, with high implementation and maintenance costs
- applications have to be written in a language supported by the vendor API (usually C/C++). Perl, the most popular language for CGI programs, cannot be used with any existing vendor API
- no process isolation—since the applications run in the server's address space, a bug in an application can corrupt the core server (or another application's data)
- proprietary—coding an application to a particular API locks the program into a particular vendor's server. Moving to a different server requires re-implementing the application
- tie-in to server architecture—API applications have to share the same architecture as the server. If the web server is multi-threaded, the application has to be thread-safe. If the web server has single-threaded processes, multi-threaded applications do not gain any performance advantage. Also, when the vendor changes the server's architecture, the API will usually have to change and applications will have to be adapted or rewritten.

6.3.2 FastCGI

FastCGI is an extension of CGI which eliminates CGI drawbacks and provides the high performance of web server APIs, while remaining highly compatible with the existing CGI applications.

FastCGI is conceptually very similar to CGI but with two major differences:

1. FastCGI processes are persistent—after finishing a request, they wait for a new request instead of closing and releasing system resources
2. Instead of using operating system environment variables and pipes, the FastCGI protocol multiplexes the environment information, standard input, output and error over a single full-duplex connection. This allows FastCGI programs to run on remote machines, using TCP connections between the web server and the FastCGI application.

The advantages include:

- Performance—FastCGI processes are persistent—they are reused to handle multiple requests
- Easy migration from CGI (see below). The FastCGI application library simplifies the migration of existing CGI applications. Applications built with the application library can also run as CGI programs for backward compatibility with old web servers
- Language independence—Like CGI, FastCGI applications can be written in any language, not just languages supported by the vendor API
- Process isolation—FastCGI applications run in separate, isolated processes. A buggy FastCGI application cannot crash or corrupt the core server or other applications, nor can a malicious FastCGI application steal any secrets (such as session keys for encryption) from the web server
- It is non-proprietary—FastCGI is available for Apache and NCSA servers, as well as for commercial servers from Microsoft and Netscape.
- Support for distributed computing—FastCGI provides the ability to run applications remotely, which is useful for distributing the load across multiple systems.

FastCGI applications must do their own housekeeping to maintain a clean environment because they stay active. They must also make sure that they carefully manage persistent information.

Benchmarks have shown that FastCGI performance is comparable to serving static files, and significantly better than CGI (clearly showing the high overhead arising from process creation) with up to a five-fold performance advantage over CGI, due mostly to savings from not having to create and initialize new processes for each request.

The FastCGI solution selected for *beeb* is Fast.Serv from North American Media Engines/Fast Engines.

6.3.3 Application Servers

Proprietary application serving environments are beginning to emerge to address the issues surrounding the development and runtime aspects of distributed applications on the internet. Examples are:

- Netscape Application Server (formerly Kiva from Kivasoft)
- NetDynamics (recently acquired by SUN Microsystems)
- Oracle Application Server
- SilverStream.

These typically offer an integrated development environment and application serving system for application logic written in Java or C/C^{++}. They can manage high volumes of transactions with many different backend databases by maintaining a pool of database connections. Most will work with multiple platforms and web servers and promise a high performance by full load balancing or application partitioning and state persistence within user sessions—something which is difficult to achieve with CGI due to its temporary existence. Tools to deploy the applications and manage the operational environment are normally provided.

6.3.4 Load Balancing

A final method for providing application scalability is load balancing. The challenge is how to ensure the technology can load-balance requests across two or more servers, while avoiding any servers which are unavailable, so that server systems can support peaks and high growth in traffic.

Basic methods can be employed by using the round-robin Domain Naming System (DNS) mechanism whereby requests are routed to the different physical servers in a cyclic manner. Each request simply goes to the next server as defined statically in the DNS, regardless of status. Each system is a mirror with respect to web content (HTML files, graphics etc.)

This technique is simple, but it ignores the differences in requests. The load-balancing DNS attempts to handle these differences, but it can be fooled by a server that is idle for the wrong reasons. If a web server's application or hardware subsystem fails, the load-balancing DNS keeps sending it requests.

Dynamic load balancing must take account of the current load on any system. A dynamic web load balancer performs periodic measurements by monitoring and querying each web server to determine which server is the most appropriate to service the next request. The information is stored in local dynamic tables, which are constantly updated, so all requests are guaranteed to receive the quickest response time possible.

The TCP/IP requests will be distributed to a configurable hierarchy of physical servers. A weighted balancing selection method provides the ability to assign a higher priority to systems which are capable of supporting the load, e.g. a system with a faster processor.

This approach removes the single point of failure, since requests are not sent to a server which is down. In addition to improving performance, the technique also provides for very high levels of availability and it also becomes very easy to take server hardware off-line for maintenance.

The dynamic load balancer is positioned between the Internet and the web servers. The dynamic load balancer connects to the internet router and the internal LAN using two separate network segments (typically Ethernet). It acts as a fast regulating valve between the Internet and the pool of servers.

The dynamic load balancer uses a virtual IP address to communicate with the router, masking the IP addresses of the individual servers. Only the virtual address is advertised to the Internet community, so the dynamic load balancer also acts as a safety net.

The other network segment connects to a hub or switch with a pool of multiple physical servers attached. Each web site server has a unique IP address. The dynamic load balancer knows them all and follows rules that can be configured to determine the routing of each incoming packet.

A number of potential product solutions for dynamic load balancing have been identified by *beeb* and will provide a fully load balanced architecture with replicated applications and content mirroring for high availability.

Wide area load balancing between web server farms can be achieved by products such as the Cisco DistributedDirector which uses routing table intelligence in the network

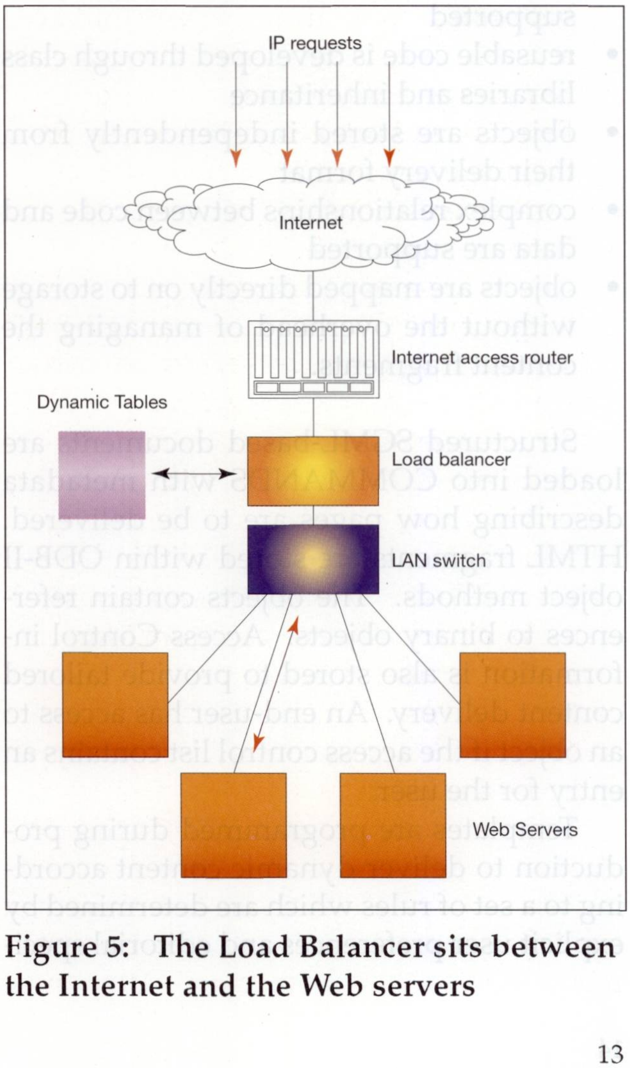

Figure 5: The Load Balancer sits between the Internet and the Web servers

infrastructure to route requests to a given server farm based on client-to-server topological proximity or client-to-server link latency (amongst others).

6.4 COMMANDS

The trend from static to intelligent web page delivery can be seen on many of the web's major sites. Also, the trend towards object-oriented architectures is starting to appear within the internet world, for example, with the standardization of the Document Object Model at the client.

ICL COMMANDS (COntent ManageMent ANd Distribution System) is an object-oriented system which uses the Fujitsu ODB-II Object Database to store material for web delivery. The use of COMMANDS within *beeb* has been to provide a highly personalized and dynamic sports webzine.

The advantages of using COMMANDS for content storage and delivery are:

- arbitrarily complex data types are supported
- reusable code is developed through class libraries and inheritance
- objects are stored independently from their delivery format
- complex relationships between code and data are supported
- objects are mapped directly on to storage without the overhead of managing the content fragments.

Structured SGML-based documents are loaded into COMMANDS with metadata describing how pages are to be delivered. HTML fragments are stored within ODB-II object methods. The objects contain references to binary objects. Access Control information is also stored to provide tailored content delivery. An end-user has access to an object if the access control list contains an entry for the user.

Templates are programmed during production to deliver dynamic content according to a set of rules which are determined by explicit user preferences and editorial specification. Web pages are then generated on request for each user/session according to the rules within the template.

The interface between the web server and COMMANDS is CGI. The CGI does very little processing itself—it merely formulates an instruction to the appropriate COMMANDS process.

COMMANDS logging enables site navigation and page usage information to be accessed for each user session.

It also provides many facilities which support the original objectives of *beeb* but have not been exploited:

- user authentication (all *beeb* users are 'guest' and are logged into the system transparently)
- session management and security
- premium content delivered to pay-only subscribers
- individual page tariffing to enable pay-per-page view billing
- page and object metering to support measurement activities.

The content personlization capabilities of the technology, enabled by dynamic database access, have been found to be too sophisticated for the current web audience requirements and production aspirations. It has led to a bottleneck in creating and maintaining a database session for each on-line user which has degraded performance and scalability. The short term future use of COMMANDS within *beeb* will be as a content store and production system for automatically generating complete web sites which are then served as static files. This has the benefit of automatic hypertext link management and rapid template redesign.

The forward path for the object database ODB-II will be heavily influenced by Computer Associates who have licensed the system from Fujitsu to produce the Jasmine object solution. There are currently no plans to continue development of ODB-II and COMMANDS within Fujitsu or ICL.

6.5 Community Applications

Chat events enable users to enter into real-time conversations with other individuals or groups.

It is one of the Internet's most popular applications—it has proven to be an effective tool for attracting customers to Internet sites and services. For example, following a chat with a star personality *beeb* experiences a general uplift in traffic across its content services.

Sites such as *beeb* that offer chat services enrich their content, encourage customer visitation and develop customer loyalty by fostering community. *beeb's* services to date offer auditorium-style moderated chat events that support large audiences and enable end users to send questions to a facilitator who filters the questions to a star personality and then shares his or her response with the entire on-line audience. The beebChat service is located at www.chat.beeb.com.

beeb's chat solution is based upon ROOMS from Acuity Corp. (formerly I-Chat).

Forums, like chat, enable users to enter into discussions with other individuals or groups on different topics. Unlike chat, forums are not immediate and allow more considered questions, answers and comments to be made (like Newsgroups). Forums provide the medium by which communities of interest can be created around our content products—television, comedy, motoring, travel, music, science etc.. The beebForum service is located at www.forums.beeb.com.

The beebForum solution is based upon bespoke development and integration.

7. The Future

beeb monitors the market closely to assess when and where the next changes will occur for maximum commercial gain. Technologies and services such as XML, multicasting, P3P, Dynamic HTML, ATM, ADSL, Web-TV, Interactive TV and palmtop/personal internet-connected devices are tracked and are all likely to have an impact on *beeb* at some stage in its growth.

A programme of "technology-refresh" will constantly assess new and enhanced products to meet our technology standards. Use of the BT SMDS network is expected to be replaced by the ICL ATM backbone with connectivity into the global network during 1999. Use of Java on the server will become more prevalent, supported by a standard application serving environment. Where possible, bespoke developments will be replaced with off-the-shelf solutions.

We will focus on engineering the key system qualities into the overall technical architecture with specific focus on *availability* and *performance* through database replication, application mirroring and dynamic load balancing. Enhanced resilience will be provided by removing major single points of failure.

beeb products and services will continue to be enhanced. The BBC Shop will migrate to a standard non-bespoke platform where the full BBC Worldwide product catalogue of four thousand items can be offered.

Integration of future payment systems, such as e-cash, and standards, such as SET, will be provided and enhanced integration with content will facilitate context-based commerce.

In order to be successful we have identified the need to integrate our services with those of others through partnerships. Cross-licensing content with third-parties for distribution is a priority, as is enhancing our webzines to enable our users to participate in transactions through electronic commerce.

Alliances, such as the strategic partnership between the BBC and Discovery, will provide new opportunities for *beeb*. Similarly the emergence of the Digital TV market and the opportunity to provide content for it will not be ignored.

Doubtlessly our learning will continue—the rapid pace of change in the technology, the dynamics of the internet marketplace and the opportunity for *beeb* will continue to create new perspectives for the project and partnership.

8. Conclusions

The Internet is rapidly evolving into a mass medium. The explosive user-base growth in both the home and business environments is matched by an emerging wealth of content sources. The BBC and ICL are bringing together complementary skills to exploit the opportunities which this provides.

We believe that *beeb* has the right technical architecture in place to support highly available content services and applications which deliver the best performance of any solutions available on the web today, with the ability to encompass the rapid technological change inherent with the internet.

In partnership with the BBC, ICL's technical skills and solutions will contribute to *beeb*'s success as a commercial service.

Acknowledgements

The contribution of BBC Worldwide, who retain joint ownership with ICL of the Intellectual Property within the project, is gratefully acknowledged.

References

beeb	www.beeb.com
Community Applications	www.acuity.com www.proxicom.com www.allaire.com
Advertising	www.accipiter.com www.netgravity.com
Traffic Measurement	www.microsoft.com/siteserver www.netgenesis.com www.everyware.com www.abc.org.uk
Media Management	www.canto.com www.vicom.ca
Load Balancing	www.cisco.com www.rndnetworks.com www.hydraweb.com www.alteon.com
FastCGI	www.name.net www.fastcgi.com www.fastserv.com
Commands	www.icl.com/commands
Database	www.oracle.com www.cai.com/products/jasmine.htm
Application Servers	home.netscape.com/comprod/server_central/kiva/index.html www.kivasoft.com www.netdynamics.com www.silverstream.com www.oracle.com/products/asd/asdhome.html
Audio/Video Streaming	www.real.com www.emblaze.com.

Biography

Nigel Dyer joined ICL in 1982 after graduating from Leeds University with BSc Combined Honours in Computational Science and Mathematics.

Following a variety of technical roles he is now a Lead Technical Design Authority (TDA) within Electronic Business Services. He is currently the Technical Design Authority of 'beeb @ the BBC'. He is also an ICL Distinguished Engineer.

The Enterprise Datacentre—ICL's "Millennium" Programme

B.J. Procter

ICL High Performance Systems, Manchester, UK

Abstract

In recent years there has been a marked trend towards a re-centralization of the many business services on which an enterprise relies. This return to the concept of a datacentre aims to overcome the disadvantages of de-centralisation without losing the benefits that inspired it in the first place. The new datacentre is, therefore, quite different from the old, monolithic operation built round a mainframe. This new datacentre supplies a diverse set of inter-linked Information Services, employing an *estate* of many platforms to support its various services. However, merely constructing an *estate* from multiple independent systems fails to alleviate many of the disadvantages of the decentralized approach. The Millennium programme, in ICL's High Performance Systems Division, has produced a new vision of the datacentre *estate* and is now delivering an evolving series of products that progressively realize this vision. This paper provides an overview of the vision and an introduction to the programme.

1. Introduction

This paper provides an overview of ICL High Performance Systems Division's (HPSD) Millennium Programme. It is intended to serve as an introduction and to provide a context for other papers appearing in the Systems Journal which will examine key parts of Millennium in more detail.

Within ICL, HPSD is focused on datacentre computing and specifically on the hardware and software infrastructure that supports the central core of medium/large-scale business application services. The nature of these systems places great emphasis on issues of scalability, availability, manageability and the protection of investment over a long period (decades!).

Through HPSD, ICL have developed a vision of datacentre computing at the end of the 1990s and in the early years of the twenty first century. The vision, which is itself evolving, drives an ongoing product and capability programme. The programme has already delivered the first members of a planned series of products marketed under the "Trimetra" brand. The vision and the programme have the umbrella name Millennium. (It is worth mentioning that this name is sometimes used as though it were the name of a specific product. This is not the preferred usage but is a convenient shorthand for the "virtual" target.)

The paper begins by discussing current and expected trends in the usage of datacentre systems over the next few years. Key IT trends over the same period are then examined and, having identified requirements and technology trends, the conclusions reached—expressed in the Millennium vision—and the target architecture are discussed. Millennium is a programme of progressive delivery of products and capabilities, and the next section outlines the overall roadmap and the progress to date. Finally there are some conclusions.

2. Datacentre Trends

Twenty years ago most enterprises ran all their computing services from a datacentre. The datacentre typically contained a single mainframe (or perhaps a small number). Datacentre staff wrote all the applications and ran the systems. Most early applications were concerned with the support of the core business processes of the enterprise (often termed "Line of Business" applications).

In the early 1980s, "Open Systems" appeared. Based on one of the flavours of Unix, they offered lower purchase costs than mainframes and greatly expanded the application market through the growth of Independent Software Vendors (ISV).

The growing use of IT continued to concentrate on core processes—often encapsulating functional divisions in the organization in application systems (and databases). The Gartner Group [Schulte, 1997] called these application systems "stovepipes". Each tends to be self-contained, with little linkage between them.

Core processes tend to change very slowly. In many cases, existing application systems and their databases continue in active use, and will continue into the foreseeable future. The ability to provide long term protection of investment in applications and data is a fundamental datacentre requirement.

The range of applications has expanded enormously since these early beginnings. Their availability continues to be a major factor in responding quickly to changes in the business environment, in containing IT costs and in improving quality. The range and quality of business applications is fuelled by a market that industry analyst International Data Corp has been quoted as valuing at $50Bn in 1997 [Computergram, 1998]. Nowadays, many enterprises are looking to the Microsoft Windows NT environment for the richest portfolio for their new and future applications.

As with bespoke applications, once an application becomes established in an organization, its continual support becomes a requirement. For example, the original platform may become obsolete or be outgrown, but the need for the application continues.

During the late 1980s and the early 1990s a combination of business pressures, organizational thinking and technology trends led to radical organizational changes. Many enterprises moved toward a flatter organization, with minimalist central departments and semi-autonomous operating divisions. Reflecting this trend, IT services were also devolved, and operating divisions were set free to choose their own systems and applications. They were able to bypass the backlog built up by the overloaded datacentres.

With the passage of time, experience with devolved organizations has exposed a number of IT issues:

- The hidden costs of running and maintaining a local IT operation were often not fully appreciated in the decision to go local and the expected savings were therefore not obtained. The cost of sustaining the local operation became burdensome to the division
- Shortage of professional IT skills: a small IT operation cannot offer IT career paths and finds it difficult to recruit and retain IT professionals. Without a professionally managed IT operation the business is at risk
- Running local IT diverts resources and management attention from their prime function
- Where local IT operations have been allowed to make their own choices, fragmentation of the IT operations across the enterprise occurs. Individual divisions make different application and database choices for the same function. It becomes more difficult to mechanise enterprise-wide business flows. More isolated pockets of IT are created (more "stovepipes" in the earlier analogy)
- Inflexibility in changing the organization: the problem of merging incompatible IT systems that has bedevilled inter-enterprise mergers now occurs within the enterprise.

The above issues are today leading towards a trend to consolidate IT services back into a datacentre type of operation in which scarce skills can be concentrated and costs shared. However, the operation is typically organized differently from the old centrally funded datacentre. It is more like a Facility Management operation (and indeed may be contracted out to a FM supplier). Whether outsourced or insourced, the service provider must tightly control costs and satisfy service level agreements with users.

Competitive pressures, regulatory changes, shifting alliances and new ways of interacting with customers and suppliers are creating a revolution in the business environment. Radical changes to business practises may be required to survive. IT is the only possible enabler for much of this change. For the datacentre, the changes mean a much-expanded range of applications with more integration between them:

- Service desks and other customer facing operations require a customer-centric (as opposed to functional) view of the operations of the organization
- In the drive to improve the profitability and effectiveness of operations, data warehousing, data mining and decision support techniques are employed for business tuning and customer tracking
- Intranet, Internet and Extranet are used to share and disseminate information internally and externally
- The use of electronic commerce for customer access and supply-chain management is growing
- Application use is not restricted to trained operatives using dedicated terminals, but may be from desks throughout (or even outside) an organization, using a variety of terminals and connection methods (including intermittently connected mobile users)
- More and more links between application systems are being put in place—both intra and inter enterprise.

These trends are not simply mechanising traditional ways of doing business, but are enabling a fundamental shift in the way enterprises (and society) operate.

In summary, the effect of the above trends is to increase the diversity and the multiplicity of applications, middleware, databases, operating systems, processors and storage deployed in the datacentre. Diversity and multiplicity create their own issues for the datacentre operation:

- Dealing with multiple suppliers
- The operational cost and skills required to manage an estate of multiple systems—compounded when the estate contains systems of different types
- User access, data sharing and application interworking between different systems, especially where they are of different types
- The performance, availability, security and integrity aspects of business services, especially where they straddle systems.

3. Technology trends

This section looks at technology trends relevant to the datacentre. It includes a brief look at trends that affect the *type* of application software employed as well as trends affecting platform architecture.

3.1 Multi-tier architectures

The monolithic architectural style typical of early mainframe based applications has been progressively superseded, first by 2-tier and more recently by multi-tier client/server architectures. For low/moderate scale applications, 2-tier architecture simplifies implementation. However, it does not scale well in either complexity or throughput. Furthermore, it lacks the flexibility to integrate with other applications or to deal with the variety of clients that might be encountered (for example "Wintel" PCs, non-Wintel thin clients, browsers, mobile users...).

Multi-tier architectures restore this flexibility. The general use of multi-tier applications is expected to increase because:

- They provide the functionality needed to support the newer ways of working such as application interworking and a variety of clients and access routes
- They fit in well within modern software development methodologies such as "componentization" and modular, object-oriented architectures
- The application developer is shielded from much of the underlying complexity

by the latest generation of middleware and application development tools.

In a datacentre context, multi-tier architectures imply:

- More application tiers in the datacentre
- By their very nature a trend towards distributed implementation
- The encouragement of application modularity, facilitating the inter-linking of applications (see below).

3.2 Applications and Interworking

Packaged applications are available and widely used for many generic business functions (e.g. Oracle Financials and the various Enterprise Resource Planning packages). Whilst providing comprehensive functionality, they have not been easy to integrate with other applications (analysts attribute 30% of the cost of installing an ERP package to integration).

Meanwhile, past investment in applications for core business functions provides the least risk, lowest cost way of supporting them into the future. The issue is how to link the business processes and information encapsulated in these legacy "stovepipes" with the new applications and user access requirements outlined above.

A wide variety of techniques, interfaces, products and toolkits have appeared. Each tends to focus on a specific aspect of application linking. They can be broadly grouped under three headings:

- Adding new front-ends to existing applications
- Application interworking to allow the business function embedded in an application to be invoked by another application
- Data sharing and cross-platform data access.

Typically, a mix of these tools will be required to cope with different situations.

3.3 Management of datacentre systems

System management products can provide much-needed assistance to the provider of datacentre services in managing operational costs, skills requirement and the provision of required service levels. The need for automated assistance rises rapidly as the number of systems within a datacentre increases. However, a piecemeal, system-by-system approach to management can add as much to the problem as to the solution.

Enterprise management frameworks (e.g. CA Unicenter, HP Openview, IBM Tivoli TME) take a holistic view, providing integrated management of applications, databases, servers, networks and desktops across an entire enterprise. However, a full-scale enterprise-wide implementation is a large investment (money, skills and time). Furthermore, infrastructure projects are typically difficult to fund and organize. As a result, in spite of their benefits, the general take-up of frameworks remains low. For managing just the datacentre, an enterprise framework can be overkill.

Aside from the enterprise framework products, there are a plethora of suppliers and products, each focussed on particular parts of the enterprise network and/or on particular aspects of the system management process. A pragmatic approach is to adopt a small selection of these products for use on all systems throughout the datacentre. A well-supported industry initiative—Web-Based Enterprise Management (WBEM)—is pulling together a set of standards to harmonize communication protocols and information models. These will improve the ability of the management products to access/control a wide range of subject systems, and enable the integration of management processes employing different tools.

3.4 Operating Systems

3.4.1 Mainframes

Mainframe systems such as OpenVME and OS390 have traditionally been used for "mission critical" services in the datacentre.

Whilst proving excellent for "Line of Business" transactional and batch systems, they have not attracted ISV support to the extent of UNIX and latterly NT systems and are not usually selected for newer applications, such as data warehousing or web serving.

3.4.2 Unix

Over the last ten years, Unix systems have matured to deliver acceptable levels of enterprise scalability, availability and manageability. Compared with mainframe based systems they offer lower purchase and software licenses prices (but have higher people costs).

Crucially, the leading Unix brands have sold in sufficient volume to attract ISV's and offer comprehensive software portfolios. A problem with Unix is the large number of varieties. There have been some withdrawals lately, a trend expected to continue with the move to 64-bit Operating Systems. HP-UX, SCO UnixWare, Sunsoft Solaris and Compaq Digital Unix (Bravo) may be the only credible survivors for platforms based on 64-bit Intel processors (with IBM's AIX as a late entrant).

3.4.3 Microsoft Windows NT

With the introduction of the Enterprise Edition of Windows NT4, Microsoft put down a marker as a serious contender in the Enterprise space. Microsoft brings a breadth of vision and investment together with the marketing muscle to maintain a consistent architecture throughout the space in which it operates.

The vision extends to most aspects of enterprise operations, covering conventional transactional applications, web and electronic commerce, information handling, support for desktop and mobile users, and data warehousing. Microsoft is steadily evolving a comprehensive architectural underpinning to enterprise operations. The phrase "Windows NT" usually implies much more than just the Operating System (OS) itself.

At the base is a set of architectural capabilities, sometimes starting as add-ons, but tending to migrate into the OS. They provide the basic models of scalability, distribution, availability, security and integrity. For example:

- Component Model COM and DCOM
- Transaction and message handling
- Active Directory
- Integrated security
- Availability and performance clustering.

On this base are built a set of "Back-Office servers", such as the SQL Server database system and the servers for particular types of application (e.g. Exchange Server, Commercial Internet Server, Terminal Server...).

A notable aspect of Microsoft architecture is its scalability model. In general servers are dedicated to particular applications (rather than mainframe style mixed workloads). Microsoft emphasize distribution as a scalability mechanism:

- Spreading the components of an application across multiple servers
- Replicating application instances and distributing the load between them.

An implication of this scalability model is a significant rise in the number of separate servers. The number is further enlarged when they are clustered to improve availability. A large numbers of NT servers in a datacentre (several dozen) raises new issues of scalability—such as manageability, consolidation of processing and storage (disc and tape) resources and configuration flexibility (the ability to re-deploy resources as needs change).

3.5 Processors and Memory

Microprocessors and memory exhibit a sustained doubling of performance/capacity approximately every 18 months. Microprocessor performance improvements come from three sources: raw silicon, the internal central processor (CPU) architecture and the overall processing sub-system architecture.

The first two improvements tend to emerge first in plug compatible CPU chips that can be used to upgrade existing systems. As speeds rise, the surrounding processing subsystem becomes a bottleneck and must change. Typically this occurs every 2–3 chip iterations (18 months–2 years). Server upgrade may not then be possible, especially with commodity servers. At much longer intervals come improvements that affect software, and we are now in the window of moves from 32-bit to 64-bit architectures.

Intel has dominated the desktop and small server microprocessor scene for years. With the availability of the 200MHz Pentium Pro in 1996, Intel had a product that allowed it to move aggressively into the datacentre space. An average performance growth of some 50% p.a. leading to the 450MHz Xeon processor in 1998 and the introduction of 64-bit architecture around the year 2000 will keep Intel in the first rank. The company, moreover, is actively working with the software industry to ensure the availability of 64-bit software and full backward binary compatibility for existing 32-bit software.

3.6 Systems Interconnect

Mainframe systems use proprietary networking technology to interconnect processors and I/O subsystems. For example, OpenVME systems have SMARTfibre and IBM mainframes have Escon. These networks form an integral part of the overall system architecture and its capabilities. They support:

- High performance I/O
- Reliability features (for example the ability to configure resilient routes that can be automatically invoked)
- Secure partitioning of a set of interconnected subsystems into separate systems
- Disaster tolerance: the ability to site devices remotely (e.g. SMARTfibre can extend to 40Km).

By contrast, Unix and NT systems commonly use the SCSI standard interface to connect storage devices such as disc and tape. Whilst suitable for small configurations, the limitations of SCSI become apparent when trying to interconnect more than a few units (as is required, for example, for resilient systems).

Fibre Channel is an industry development to overcome SCSI limitations. It is a serial interface giving improvements over SCSI in throughput, connectivity and distance. A simple loop topology (FC-AL), supports moderately large systems—for example 50-100 discs can be connected through a single FC-AL, with a second providing full route resilience. FC-AL technology is becoming mature now. Fibre Channel switched fabric will extend to the largest installations, but is still immature. Fibre-optic Fibre Channel links allows connection to remote sites. Fibre Channel is close to delivering the features formerly offered by proprietary mainframe interconnect, but with the crucial advantage that it is an industry standard.

A newly emerging standard (Virtual Interface—VI) concerns efficient application to application connection. It is intended to improve the scalability of distributed application architectures (such as Microsoft above). There are related developments in low-latency system networks (sometimes known as System Area Network or SAN).

3.7 Storage Subsystems

Co-incidentally, disc technology is advancing at a similar rate to silicon, with a doubling in capacity about every 18 months. RAID has become mature technology and a terabyte of reliable storage in late 1998 needs 50-70 discs and can fit comfortably into a single 2 metre high rack.

With multiple servers in use, there is a benefit in consolidating all storage into a single pool. Consolidation offers:

- Single point of management for all storage
- Back-up and archiving provided centrally
- Spare capacity in the pool which allows a rapid response to changes in any of the connected systems
- The costs of RAID and of standby equipment to boost availability can be amortized over many systems

- The provision and management of remote mirrors (for disaster protection) can be done once only.

The storage partitioning and management facilities to support consolidation have been pioneered by high-value storage suppliers, and especially by EMC^2 who are the industry leader in this field. Facilities are gradually extending to mid-range systems such as Clariion and StorageWorks. Fibre Channel provides the means of networking multiple storage and processing subsystems (even when they are widely separated).

Consolidation and other storage functionality built into the storage system by suppliers such as EMC^2 gives rise to terms such as "Storage Centric" and "Networked Storage". Storage functionality can be used to contribute directly to an overall business solution e.g.:

- High availability of data using multiple data plexes, even when they are not supported by an OS
- Remote data plexes (e.g. for disaster tolerance), even where servers or OS do not support them.
- Instant "snapshot" copies of a dataset (e.g. for back-up or transfer to another application) without penalizing throughput
- "Instantaneous" movement of data between applications and some forms of dynamic data sharing.

Tape systems continue to advance at a similar rate to disc systems in capacity and data rate. However, capacity grows much faster than speed, making the time to backup/archive a growing issue. Incremental backup, multiple data streams (and hence tape drives), and selective management strategies are all used to reduce times.

There are good reasons for consolidating tape backup and archive rather than using a traditional per-server approach. Multiple tape drives to reduce backup times are expensive to provide on a per-server basis. As the size of the installation grows, the physical management of tape media becomes a nightmare. Consolidating tape systems makes it cost effective to invest in a robot library whose value is then enjoyed by a number of systems. In fact, tape consolidation, pioneered by industry leader StorageTek, has been used in large datacentres for many years. Recent developments have reduced the cost of shared robot libraries to a point where they are effective for modest scale datacentres.

3.8 Communications Networks

The availability of low cost, high bandwidth communication networks are a vital enabler without which the datacentre concept could not exist. Many factors contribute to the emergence of networks having the capacity to readily support today's datacentre requirements and the potential to accommodate future needs:

- Wide area communications (WAN) [Cochrane, 1998]:
  - Basic communication technologies: particularly digital transmission / switching and fibre optics
  - A highly competitive open market
  - Exponential growth in traffic
- LAN:
  - Ubiquitous use
  - Emergence of a new industry sector supplying low cost LAN components
  - Convergence of technologies between LAN and WAN
- A dominant protocol (TCP/IP)
- The growth of the Internet and the emergence of Internet Service Providers as a new class of low-cost service supplier.

4. Millennium Vision & Architecture

4.1 Developing the vision

The Millennium Vision of the enterprise datacentre has been developed to reflect the trends in datacentre computing and in technology. It is not static, but continues to evolve with the requirements of datacentre users and as the technology and the IT industry progress.

The view of datacentre requirements has been developed and is refined in a number of ways:

- ICL experience with OpenVME customers and their requirements and concerns
- Commissioned analysis and published information from leading market research groups and industry consultants
- Interviews conducted by ICL with OpenVME customers and others
- In depth DILO (Day-In-the-Life-Of) walkthroughs with users
- Interviews and discussion groups conducted anonymously with ICL and non-ICL users as participants
- Feedback from customer events
- Experience from early Trimetra customers.

4.2 The vision of the datacentre

To summarise the previous section on Datacentre Trends: The enterprise datacentre is a (re-)centralization of the main information services in the enterprise. It supports a broad portfolio of application systems built up over the years, continually expanding as new systems are added to address new business needs. It has to deal with the issues of managing multiplicity and diversity, of guaranteeing and improving service to business users, and of protecting and developing corporate assets.

The Millennium vision for enterprise datacentres is ICL's response to the requirements underlying these trends. The Millennium vision explicitly recognizes the requirements for multiplicity and heterogeneity in the datacentre and aims to supply solutions and aids that tackle the issues mentioned above. In summary, the main strands of the vision are:

- The datacentre is, and will continue to be, an "estate" of multiple systems and mixed platform types, supporting a diverse set of applications. Millennium offers a consistent, flexible way of consolidating multiple platforms of the same or different types
- Managing and operating the estate efficiently is an issue. Bringing the management (and support) together and using a common set of management tools contributes to reducing costs, making best use of scarce skills and improving services to the business users
- There is an ongoing requirement to interlink applications, share data, and extend existing applications with new access routes and user interfaces. There is no "magic" overall solution and each case must be analysed to determine the appropriate solution. Millennium offers an architectural framework for integration, an integration toolkit and associated services
- To serve the business needs of the organization, the enterprise datacentre requires an ever-diversifying range of applications (recent additions include, for example, web-delivered multimedia services, data warehousing, and electronic commerce). ICL believes these new needs are increasingly met by combining enterprise quality platforms with software solutions based on Microsoft Windows NT. SCO UnixWare is available for customers needing Unix
- OpenVME customers have applications and databases that remain critical to their businesses now and for the foreseeable future. The provision of a software environment to allow OpenVME to run on industry-standard hardware provides these customers with the assurance of its continuing availability on up-to-date, competitive hardware
- ICL believes the present and future needs of datacentres are best served by hardware and software technologies that are in the computing mainstream and benefit from widespread investment and support across the industry. To source these technologies, ICL has forged working alliances with leading suppliers of hardware and software subsystems
- Datacentres require "enterprise-class" systems. All aspects must be capable of scaling to the capacity and throughput required by enterprise-wide solutions, whilst remaining manageable and able to accom-

modate change. They must provide the availability and security essential to "mission critical" applications and data

- Enterprises depend on their datacentre systems. ICL, through its High Performance Systems Division, has more than twenty five years of experience in developing, supplying and supporting mainframe systems for customers. In Millennium, building on hardware and software from our technology alliance partners, ICL's large system experience is harnessed to devise, integrate, validate and support enterprise-quality systems and solutions.

4.3 The target Millennium architecture

Figure 1 illustrates the target Millennium architecture. It is termed "target" architecture because it will not be achieved all at once. Millennium is a multi-year programme with product and capability releases that progress to the target. The description in this section is about the target architecture. Section 5 discusses the strategic roadmap and the achievements to date.

The five "starbursts" enumerate the OPEN*framework* [ICL, 1993] qualities. They are discussed at the end of this section. The four rectangles around the periphery identify OPEN*framework* perspectives (i.e. views of Millennium from different standpoints). They are discussed later.

The central part of the diagram illustrates the functional components. Each vertical column is a discrete instance of a software system—each with its own software "stack" from OS to application. The three columns across the "front" of the diagram show the different personalities of software supported by Millennium: Microsoft Windows NT, OpenVME, and SCO UnixWare. A single Millennium system is capable of simultaneously supporting one, two or three of these personalities. Columns going into the "depth" of the diagram (rather less obvious in the figure, but a key feature nonetheless) indicate the ability of a Millennium system to support multiple *instances* of a personality (or even of more than one personality).

In order for a software stack to execute, it must be associated with an appropriate set of hardware resources. A Millennium system running multiple stacks is similar to a collection of independent servers (possibly

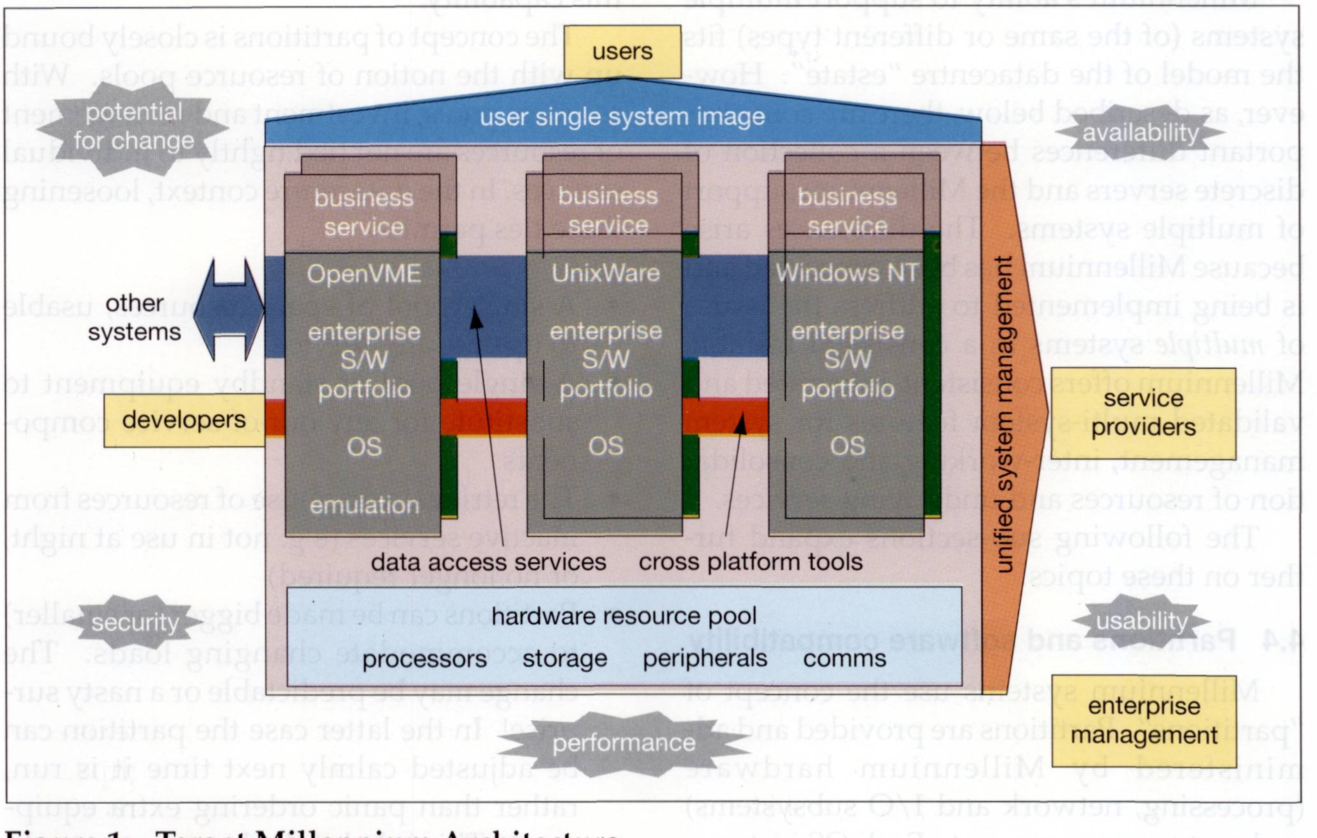

Figure 1: Target Millennium Architecture

of different types) in a conventional datacentre estate. A software stack may be an independent application system—typically providing a business service. Alternatively, a number of stacks may co-operate to provide a business service. The latter is particularly appropriate in Windows NT applications where it is common to distribute a single application service amongst several servers.

Millennium systems are based on industry-leading hardware and software; selected, integrated and validated by ICL. Intel processors in an enterprise-quality processing subsystem are teamed with a range of leading storage subsystems and intra-system networks. Enterprise systems are a focus of ICL's alliance with Microsoft, and the two companies are working together to deliver robust, scalable systems around Microsoft Windows NT. Through its High Performance Systems Division, ICL participates with other companies in a SCO programme to extend the enterprise features of SCO UnixWare, and works with leading middleware and system management vendors to provide fully integrated and validated software stacks.

Millennium's ability to support multiple systems (of the same or different types) fits the model of the datacentre "estate". However, as described below, there are some important differences between a collection of discrete servers and the Millennium support of multiple systems. The differences arise because Millennium has been conceived and is being implemented to address the issues of *multiple* systems in a consistent fashion. Millennium offers consistent, integrated and validated multi-system features for system management, inter-working and consolidation of resources and underlying services.

The following sub-sections expand further on these topics.

4.4 Partitions and software compatibility

Millennium systems use the concept of "partitions". Partitions are provided and administered by Millennium hardware (processing, network and I/O subsystems) and system management. Each OS instance operates in a separate partition that insulates it from all other OSs. To the OS, a partition behaves just like an independent conventional system, allowing Millennium systems to be certified for compatibility with NT and UnixWare. A Millennium stack can support any server software that is approved for the relevant OS and Intel processors.

The OpenVME stack contains an extra component—an emulation package that makes the partition behave just like traditional proprietary hardware. The OpenVME OS, middleware, customer applications and databases run unchanged.

4.5 Partitions and resource pools give configuration flexibility

Whereas the boundaries of a normal system are rigidly determined by the physical configuration of the platform, the boundaries of Millennium partitions are easily moveable. If, for example, the load on a particular business service increases (e.g. due to a seasonal upturn in business) the allocation of resources to the partition can be increased to match. Systems that offer "server farms" constructed from separate servers do not have this capability.

The concept of partitions is closely bound up with the notion of resource pools. With resource pools, investment and management of resources are not tied tightly to individual services. In the datacentre context, loosening these ties permits:

- A single pool of spare resources, usable to up-rate any service
- A single pool of standby equipment to substitute for any out-of-service components
- The retrieval and re-use of resources from inactive services (e.g. not in use at night, or no longer required)
- Partitions can be made bigger (or smaller) to accommodate changing loads. The change may be predictable or a nasty surprise! In the latter case the partition can be adjusted calmly next time it is run, rather than panic ordering extra equipment. This is helpful where a new appli-

cation service is being installed. Both its resource consumption and the level of user demand may be poorly understood. Resources can be quickly adjusted (up or down) in the light of experience, making it all less of a gamble
- Centralizing resources and their management to help reduce support costs. For example, centralizing backup on to a robot tape library cuts out manual handling, reducing operational costs and eliminating a potential source of dangerous errors.

The use of partitions and resource pools contributes to improving three distinct datacentre "measurables" compared with a conventional datacentre operating separate systems:

- Datacentre investment:
  - Lower costs: Pooling can reduce the total datacentre hardware requirement
  - Protection of investment: Pooling makes it easier to re-deploy hardware no longer required by any service
- Operational costs/skills: lower people costs and better use of skills
- Meeting Service Level Agreements to users:
  - Substituting for component outages
  - Faster correction of mismatches between the load on a service and the resources available to it.

4.6 Selection and integration of platform subsystems

An early decision in the "Millennium" programme was the adoption of Intel's 32-bit (IA32) and the later 64-bit (IA64) architectures as the target processing architectures. The choice was made on the basis of a strong product line, suitability for enterprise use, and the opportunities provided by the hardware and software component marketplace around "Wintel" architecture.

Rather than develop its own subsystems to compete in an already crowded marketplace, ICL has chosen to select the subsystems it considers best suited to the datacentre requirement and to forge close OEM partnerships. "Millennium" customers (and OEM suppliers) benefit from greater volumes.

The strategy allows ICL to focus its resources on the system and multi-system aspects of the datacentre. In the enterprise market segment, High Performance Systems Division operates as ICL's "one-stop-shop" between customer requirements and the datacentre platform aspects of the business solution.

The "shop" terminology is not really correct because a working, scalable, reliable etc. system does not magically appear from a basket of components—even when the right components are chosen. Selecting, configuring and integrating subsystems and other hardware and software components is a major activity in its own right. Full system validation and characterization are essential to assure customers that *their* configurations will work and meet *their* requirements. This is an area in which ICL adds value, building on more than twenty five years experience of mainframe systems.

4.7 Securing the future of OpenVME platforms

Prior to Millennium, OpenVME systems have always depended on a proprietary hardware architecture for both processing and I/O, for example the SX system [Eaton et al., 1990]. This approach has been the only way to deliver the required performance.

Technology trends have opened another route to running OpenVME applications:

- Microprocessor performance has consistently grown faster than mainframes and is expected to continue to do so (~50% p.a. compared with ~30% p.a.)
- The development of software emulation technology (which makes one type of system mimic another).

A low-end OpenVME platform using these techniques has already been delivered. It uses an ICL developed emulation package and runs on an Intel processor subsystem [Brightwell, 1998]. Further evolution of the microprocessor and the emulation package will eventually allow the entire OpenVME range to be delivered on the same hardware used for Windows NT and UnixWare.

OpenVME customers have applications that are critical to their business today and for the foreseeable future. Exploiting industry investment in platforms allows OpenVME customers to ride the same technology escalator as is enjoyed by "Wintel" users. At the same time, the replacement of proprietary hardware by industry-standard hardware assures them of the future availability of OpenVME systems to support their business needs. Using the same hardware base as the other personalities in "Millennium" allows the benefits of partitions and resource pools to be shared by OpenVME systems.

4.8 Efficient management and operation

Datacentres exist to provide business services in support of the enterprise. Their effectiveness is measured in terms of the quality of the services they provide (increasingly formalized into Service Level Agreements, SLAs) against the cost of running the operation.

As the datacentre grows in terms of the number and diversity of systems within it, the operational complexity increases. To alleviate this problem, at minimum:

- The management of each type of task should be brought together to a single point
- The same tool should be used for managing a task across the whole datacentre estate.

Unfortunately, even achieving these aims across heterogeneous systems is not straightforward. Furthermore, although mainframes such as OpenVME have traditionally included comprehensive management facilities, they are not the same as the tools for Unix or NT systems.

While Enterprise Systems Management Frameworks provide a solution, their cost and complexity is considered overkill for the datacentre task alone. However, when a framework is in use elsewhere, Millennium is capable of being managed by it.

Millennium offers an integrated set of management tools, selected from leading products, able to operate across all system personalities and instances in one or more Millennium platforms. In the case of OpenVME, the tools supplement its own inbuilt management, allowing day-to-day operations to use the same tools and interfaces as NT and UnixWare. Tools are provided to manage resource pools and their partitioning into system instances.

As with frameworks, it is recognized that individual enterprises may already have a commitment to specific management tools. Accordingly "Millennium" platforms also support industry-standard management interfaces, making them manageable by alternative tools of a customer's choice.

4.9 Causeways between islands

For almost as many years as IT has been in use, there have been complaints of "islands of automation" or "islands of data". In fact, few business application systems are completely isolated. When a new system is added, there is usually a need to link it into some existing systems. In turn, further systems added later may need to be linked to it. It is a never-ending requirement!

A related problem is the need to provide modern user interfaces on existing applications, or to make them accessible via new routes and client types. For example, re-furbishing a legacy "green-screen" application with a modern GUI and opening it up to access from a browser over an Intranet. It is often the case that the application is to be used by a different type of user and there may be a need for additional business rules between the user and the old application. This problem is related to the previous group because the new "front-end" must be linked to the existing application.

Inter-linking application systems and information is difficult. Analysts suggest that some 30% of the cost of putting in a new system is attributable to linking into existing systems.

There is a wide range of tools available to solve different types of inter-linking problems, which can be grouped into broad categories. Consider a client C and two application systems Ax,Ay, each containing their own data Dx,Dy; see Figure 2.

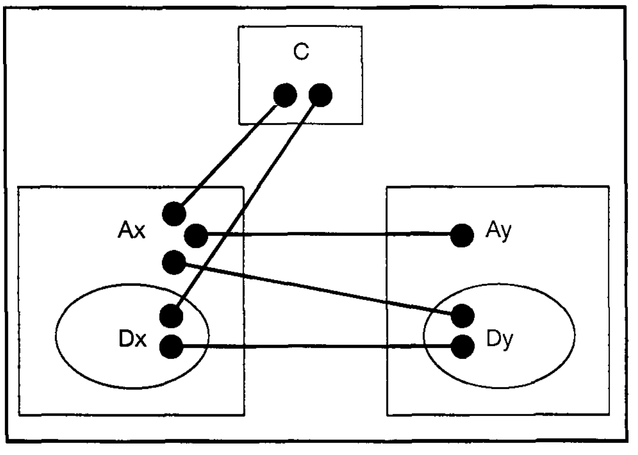

Figure 2: Client and Application Systems Links

The links most commonly used (in no particular order) are:

1. C to An — The client invokes an application; e.g. the modernization of an application interface in the example above
2. C to Dn — The client accesses data from an application system; e.g. to query the content of an order in an orders database
3. Ax to Ay — An application invokes operations in another; e.g. the business process supported by Ax needs to invoke the business process supported by Av, for example, a sales order process queries availability from a stock control process and makes an allocation to a customer
4. Ax to Dy — An application makes use of the data in another; e.g. an order fulfilment application needs customer address information held in the database of a sales application
5. Dx to Dy — Information is transferred between databases; e.g. from a sales database to a data warehouse for subsequent analysis.

Recognizing that each requirement must be treated on its own merits, the Millennium approach is a "toolkit" of inter-working products covering the categories above and the three s/w personalities supported by Millennium (Windows NT, OpenVME, UnixWare). ICL backs up the toolkits with services to assist customers with the selection and implementation of an appropriate solution.

4.10 "Millennium" from different viewpoints

This section takes a brief look at Millennium in the datacentre from the perspectives of different "stakeholders".

4.10.1 User View

The application integration and front-ending tools offered with Millennium allow the construction of integrated user views across different services hosted on Millennium. A data-access tool allows users of desktop applications to query databases anywhere in a Millennium system, including data held in OpenVME IDMS databases—truly "any data, anywhere". Single sign-on allows controlled user access to application services hosted on Millennium, simplifying the user interface to multiple services.

4.10.2 Application Developer View

Application developers and integrators have the benefit of the application interworking and data access tools.

There are also a set of tools which can be used across all personalities. For example, the Natstar application development environment and MicroFocus Cobol language provide a common way of developing applications for any of the personalities, including OpenVME.

4.10.3 Service Provider View

In this context, the "Service Provider" is the operator of the datacentre.

"Millennium" offers a one-stop-shop datacentre estate, already integrated and validated. It relieves service providers of the cost and risk of putting the components together themselves.

Integrated multi-platform management reduces operational cost and the need for specialist skills. Together with resource pools and flexible partitioning it helps service providers to improve their indexes of *service level* versus *equipment cost*, whilst improving their service to business users.

4.10.4 Enterprise Management View

This view considers the wider impact of IT on the enterprise and on its strategic capability.

In some sense, this view "closes the circle" on the Millennium vision. The various capabilities of Millennium were inspired by a vision that the centralized datacentre offers strategic benefit to the enterprise. Enterprise management is concerned with getting the enterprise into a better shape and positioning it to maximise its effectiveness now and in the future. The Millennium datacentre approach assists with:

- Making operational costs visible and helping to reduce them
- Enabling business and organizational change; e.g. by "re-plumbing" around the applications that support core business processes
- Protecting, exploiting and developing assets, particularly the core business processes, information and skills and the investment in IT infrastructure represented by the datacentre
- providing re-assurance that OpenVME will continue to meet future needs for enterprises using OpenVME applications, due to the Millennium strategy of hosting OpenVME on industry-standard platforms.

4.11 Millennium Qualities

The OPEN*framework* qualities look at the attributes of any system. Although they all have some relevance to the datacentre, three are key and are discussed here.

4.11.1 Availability

The "RAS" attributes (reliability, availability and serviceability) are key to the datacentre operation. Services provided from the datacentre are often "mission critical"—if the service stops, the business stops.

Millennium RAS can be considered at three levels:

- The individual subsystem/components
- System and multi-system level
- ICL and supplier processes.

The main subsystem/components of interest are the three different personalities of OS (Microsoft Windows NT, OpenVME, and SCO UnixWare), and the major hardware subsystems (processing, disc storage, tape libraries). Although important at the solution level, communications networks and the equipment connected via them are outside the scope of the "Millennium" programme.

Microsoft Windows NT and SCO UnixWare have somewhat similar approaches to achieving high system availability, although details differ considerably. The Millennium architecture is neutral to internal OS RAS features (although of course the system benefits from them). OS features that are synergistic with Millennium are:

- Support of the RAS features in Millennium subsystems
- Ability to survive many hardware failures by using alternative resources (e.g. bypassing failures in I/O paths or communications networks). These features can exploit the rich configuration of a typical Millennium multi-system
- Support for automatic fail-over within a cluster configuration. A feature (cluster manager) within the OS in which members of a "cluster" of systems monitor each other and, on detection of the failure of a server in the cluster, re-start the failed OS and application instances on an alterna-

tive server. Millennium provides flexible and efficient ways of configuring fail-over cluster systems.

ICL's OpenVME OS has proved itself over more than two decades of use in mission critical situations. The move to industry-standard hardware further extends its resilience and availability capabilities, particularly in the areas of storage system resilience, fail-over cluster systems and disaster tolerance.

The hardware subsystems used in Millennium are chosen for their enterprise qualities, RAS (reliability, availability and serviceability) being a key quality. All subsystems include resilience features that contain common failures so that operations continue without apparent error. Repair and upgrade operations can be done on live systems to minimize downtime. Fault diagnosis, automatic "phone home" to service centres, and problem management combine to ensure quick diagnosis and effective repair.

RAS features are brought together at the system and multi-system level:

- Combining features at the system level creates powerful new RAS capabilities. For example, an application that does not support on-line backup may incur lengthy downtime while its files are backed up. The downtime can be virtually eliminated by "instantaneously" dropping off a storage mirror plex and backing up off-line
- Fail-over is a powerful RAS technique for getting a service up and running again after a hardware failure. The common two-way cluster solution requires all resources to be duplicated. Scaled up to multiple systems, this becomes an expensive solution. "Millennium" RAS and multi-system features can be used to implement efficient and flexible "N+1" fail-over arrangements across a group of services
- Some customers need to be protected from "disasters" at a datacentre. Millennium systems support a range of data and service protection options—from remote siting of tape libraries to full-scale disaster tolerant configurations.

Several of these features have already been employed in the ICL Enterprise Exchange Server—a large-scale implementation of Microsoft Exchange capable of scaling to 50K users. This was developed by ICL in conjunction with Microsoft, EMC^2 and StorageTek for delivery in 1998. An important aspect of scaling-up the implementation was the containment of service downtime (whether caused by failures or by housekeeping activities such as backing up multi-terabyte data). Exploitation of the features mentioned above was instrumental in meeting RAS targets.

ICL has been supplying and supporting systems into mission critical customer applications for more than twenty five years. It understands both the technical features required and, just as critically, the processes needed to supply and support customers with demanding business requirements. The product validation and release methods, the professional services and the customer support infrastructure used for Millennium today are based on practices refined over many years with OpenVME customers.

4.11.2 Performance

This attribute includes the scalability, predictability and assessment of all aspects of performance.

Millennium supports scalability at three levels:

- Datacentre level: the multi-system capability of Millennium makes it easy to add new applications in their own partitions
- Distributed application level: Millennium is ideal for increasing application throughput by spreading it over several servers (a favoured approach for Microsoft Windows NT applications)
- Single system instance level: the flexible partitioning of Millennium allows adjustment of any of the resources within a partition. Uniquely, this includes the ability to vary easily the size of a multi-proces-

sor and the amount of memory allocated to a partition.

System performance must be predictable and "well behaved" under varying loads. This is achieved through the careful selection of subsystems, system configurations, performance modelling and thorough system testing.

Instrumentation is included and performance analysis tools are available to assist the diagnosis and correction of any application performance problems arising in operational use. ICL offers services to assist customers in these tasks.

4.11.3 Potential for change

Requirements for services from the datacentre do not remain constant for long. There is a need for new services, for changed loads on existing services and the orderly closing down of old services. The datacentre requires to juggle these constantly changing needs, while maintaining its service level agreements to its users and containing its costs and investment requirements.

The flexibility that Millennium offers to accommodate changing workloads has received a positive response from users who are wrestling with these issues.

The application inter-working and data access tools help to adapt existing applications and data to changing business requirements.

The quickening rate of change of the underlying technologies faces datacentres with another set of problems. Traditionally, datacentre hardware and software have changed slowly (perhaps in a 3–5 year cycle). Furthermore, there has been a high degree of backward compatibility, allowing graceful evolution of installed systems. The cycle rate is now much faster (~12 months), and "commodity" products are replaced, rather than evolve. For Millennium, the selection, integration and validation of hardware and software components all take account of the need to protect past investment and to enable systems to evolve incrementally.

5. Roadmap and Status

The overall Millennium strategy roadmap is illustrated in Figure 3. The left side of the figure shows the position prior to the Millennium strategy. The needs of OpenVME

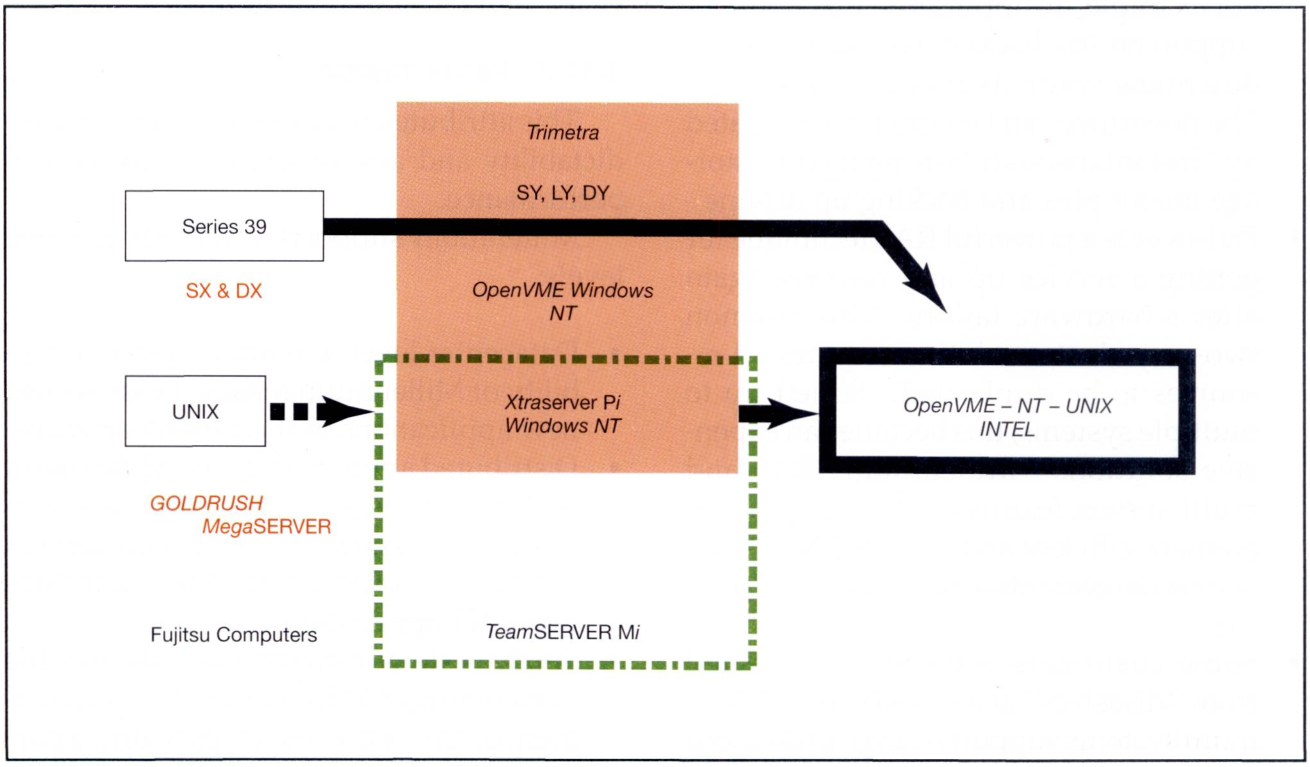

Figure 3: Millennium Roadmap

customers and Unix customers were supplied by completely different ranges of systems.

The strategy provides a progressive convergence of technology and system capabilities from this disjoint approach to a fully integrated range.

5.1 The first steps

The first products in the Millennium programme, indicated by the red area in Figure 3, were delivered in 1997. They comprised two ranges of systems under the overall "Trimetra" brand. The Trimetra "Y" systems were intended for customers who wanted OpenVME. The Trimetra Xtraserver was for customers or applications where there was no OpenVME requirement. Apart from the OpenVME personality, the Y series and Xtraserver Trimetras offer access to the same software portfolio through the use of the same OS options, in each case running on Intel platforms.

Trimetra Y systems consist of an OpenVME element and a Windows NT or UnixWare element [Martin & Stewart, 1998]. The two elements are bundled into a single sales package. The first systems delivered were the top-end OpenVME SY models and the mid-range LY models. In these systems, the OpenVME processing module uses a recently developed CMOS custom processor [Allt et al., 1997]. All the "Y" series Trimetras thus far retain OpenVME's proprietary I/O network and I/O controllers. The basic Windows NT / UnixWare element is a 4-way Intel Pentium Pro based on a product from Fujitsu Computers' TeamSERVER range. The OpenVME and NT / UnixWare elements are physically integrated into a suite of cabinets and are managed and serviced through a common set of interfaces.

Later during 1997 the Trimetra Xtraserver Pi [Messham, 1998] joined the Y Trimetras. This was an enterprise class Intel based system, scalable from 2–12 processors and available with either Windows NT or UnixWare.

Another significant step in the delivery of the Millennium programme occurred in Q2 1998 with the delivery of the first "DY" system. In many respects DY resembles a smaller version of LY but with one crucial difference. DY is the *first* machine to deliver full support of OpenVME OS and applications without needing a custom processor. In DY, OpenVME processing is accomplished by an emulation package running on an industry-standard Intel-based subsystem.

Also during 1998 there were important software releases providing tools and capabilities that span the regimes (OpenVME and Windows NT or UnixWare):

- A package of integrated management tools
- Software development environment and tools
- Application inter-working and data access.

During 1998 the range of SY configurations is being extended with larger single and multi-node systems.

5.2 Looking forward

The deliveries in 1997/8 put down the first steps on the Millennium roadmap. Over the next few years subsequent releases will maintain and extend it:

- New systems will incorporate the latest generation technology when they reach the enterprise-class systems market (usually 3–6 months after their first appearance in the volume market). "Technology refresh" applies to hardware and software; e.g. Intel processors, discs, Fibre Channel network etc., and later releases of software, e.g. Windows NT5.
- 64-bit systems will become available, extending the range of today's 32-bit systems. Millennium expects to be able to offer hardware upgradeability and backward and forward software compatibility between 32-bit and 64-bit systems
- Flexible partitioning will replace permanently partitioned systems (other than for the continuing use of SY processors in high-end OpenVME, see below)

- Integrated management capabilities will be further extended according to customer requirements
- The application interworking and data access tools and services will be expanded to support customer needs, "fast pipes" will provide high speed interconnection between application systems within Millennium
- The OpenVME emulation package, introduced with DY, will be extended in two dimensions
  - Intel processor improvements combined with the development of the emulation package will extend their reach, first to the mid-range and eventually to the most powerful OpenVME configurations
  - OpenVME I/O will be emulated on the same I/O systems used for Windows NT/UnixWare, bypassing the need for proprietary OpenVME I/O controllers.

5.3 Millennium and Microsoft-based enterprise solutions

Enterprise solutions are one of the key themes of the ICL/Microsoft alliance. Within ICL, High Performance Systems Division is responsible for the Millennium programme and is positioned as the focal point for the engineering of enterprise-level Windows NT platform solutions. As discussed earlier, NT-based solutions typically employ multi-server architecture for scaling and for high availability. The Millennium architecture is specifically designed as a multi-server architecture. Previous discussion in this paper has outlined the benefits of consolidating on Millennium compared with the use of many discrete servers. Millennium neatly complements Microsoft-based enterprise solutions, particularly in the areas of:

- Scability and flexibility
- RAS
- Integrated management.

In conjunction with Microsoft and other partners, ICL's High Performance Systems Division is taking the lead in creating and validating enterprise-scale solutions around Windows NT and Millennium. The first fruit of this work is the large-scale configuration of Microsoft Exchange (mentioned earlier), capable of scaling to 50,000 users, which will be delivered this year.

6. Conclusions

This paper has examined the business and technology trends that motivate the re-centralization of IT services and resources in a datacentre-type operation. ICL High Performance Systems Division has researched these trends with users and consultants. The result is the Millennium vision and the associated programme of deliverables. The first steps in the programme were taken in 1997 and further steps in the programme are planned over the next few years.

The reception given by users to the announcement of the Millennium programme and to the delivery of the first Trimetra products was encouraging. There is no doubt that it is addressing real needs.

The ICL/Microsoft Alliance has enterprise computing as one of its areas of focus. The Millennium datacentre estate combined with Microsoft's extensive product range provides an excellent framework for enterprise-quality solutions. The first product will be delivered this year and more will follow.

Acknowledgements

All trademarks are acknowledged.

My thanks to all my former colleagues in ICL High Performance Systems Division who have built up its reputation over the years and who have developed and are delivering the Millennium programme.

Bibliography

ALLT, G., DESYLLAS, P., DUXBURY, M., HUGHES, K., LO, K., LYSONS, J.S.M., and ROSE, P.V., "The SY Node Design", ICL Systems Journal, Vol 12, Issue 1, 1997.

BRIGHTWELL, A.E., "DY and Emulation of OpenVME on Intel Hardware", ICL Systems Journal, Vol 13, Issue 1, 1998

COCHRANE, P., "From Copper to Glass: the Right Idea, Decisions and Investment at the Right Time...", the Computer Journal, published by the Oxford University Press for the British Computer Society, Vol. 40, number 1, 1997

COMPUTERGRAM INTERNATIONAL, quoting International Data Corporation report, cgram-3504 Sept. 1998

EATON, J.R., ALLT, G., and HUGHES, K., "The SX Node Architecture" ICL Technical Journal, Vol 7, Issue 2, 1990

ICL, "OPENframework—The Systems Architecture: An Introduction," Prentice Hall, 1993.

MARTIN, C, STEWART, C.P., "Trimetra UNS," ICL Systems Journal, Vol 13, Issue 1, 1998

MESSHAM, D.K, "Trimetra Xtraserver," ICL Systems Journal, Vol 13, Issue 1, 1998

SCHULTE, R., "Architecture and Planning for Modern Application Styles", Gartner Group Research Report (available to Gartner Group subscribers), April 1997

Biography

Brian Procter joined the Computer Department of EMI Electronics in 1958. During the 1960s the companies in which he worked were progressively taken over as the fragmented British Data Processing industry consolidated to become ICL.

After his original days on the EMIDEC 1100, he worked on the design of many of the smaller ICT 1900 Series machines, eventually managing the group developing them. The group went on to develop the lower end of the successor ICL 2900 Series.

In 1976, all ICL system developments were combined on one site in Manchester. At this time Brian moved to a technology role, becoming one of a small group of System Technology managers who steered the technical evolution of the 2900 Series into the award-winning Series 39 in the 1980s.

In the late 1980s Brian became involved in advanced development projects in parallel computing. He was the technical manager of an ICL led consortium project within the UK Alvey Programme, and subsequently of a European consortium within the Esprit Programme. This led to the ICL Goldrush MegaSERVER product used in data-warehouse and large-scale interactive multimedia delivery.

Prior to his retirement in 1998 Brian was the Chief Architect of ICL High Performance Systems Division, working on the overall architecture and technical strategy of the Millennium programme and the initial round of Trimetra products.

Brian is an ICL Fellow Emeritus.

Trimetra DY and the Emulation of OpenVME on Intel Hardware

Andrew Brightwell

ICL High Performance Systems, Manchester, UK

Abstract

Trimetra DY is the first OpenVME system running on Intel hardware and a key step towards ICL's Millennium vision for datacentre systems. This paper describes how OpenVME is emulated on Intel hardware, what this means now and in the future, as well as the background to the Daisy project whose task was to develop Trimetra DY.

1. Introduction

Data centres were traditionally focussed on the support of the core business processes of the enterprise (often termed "Line of Business" applications). IT support for modern business practises calls for a much expanded range of applications. ICL's Millennium vision is of data centres supporting multiple services running on multiple servers of the same or different types (OpenVME, Windows NT, and UnixWare), integrated together to support data sharing and application interworking, and all operated and supported by a single management image [Procter, 1998].

The Trimetra range marked the first deliveries in the Millennium programme. Here OpenVME runs alongside, and is integrated with, Windows NT and/or UnixWare. The OpenVME portion of Trimetra is referred to as the OpenVME Subsystem (OVS); the Windows NT and UnixWare portions are referred to as the UnixWare/NT Subsystem (UNS). The customer chooses the operating systems, and with UNS the number of modules needed, that are most appropriate to the applications to be run and the business to be conducted. This arrangement is shown in Figure 1.

SY and LY, the large and mid-sized members of the Trimetra range, use an ICL designed, Fujitsu produced CMOS processor for their OVS and an Intel processor module from Fujitsu for their UNS. DY, the smallest member of the Trimetra range, uses Intel processor modules from Fujitsu for its OVS and UNS.

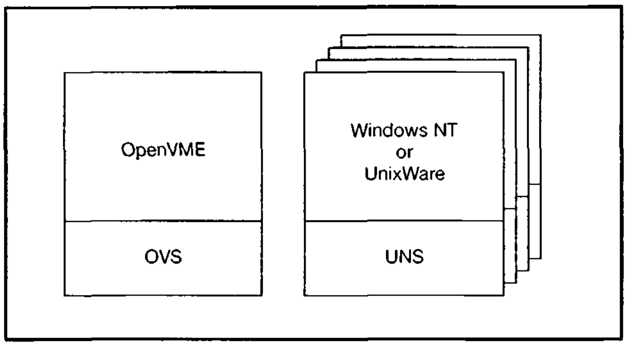

Figure 1: A Trimetra System

The CMOS processor used on SY and LY was, with its Picode, specifically designed to run OpenVME [Allt et al., 1997]. Intel processors were not designed to run OpenVME, so DY used emulation to run the same OpenVME, applications, and data as SY, LY, and Series 39. Just as important was DY maintaining all the RAS (reliability, availability, and serviceability) features expected of enterprise systems.

The Trimetra range is completed with XtraServer offering larger, scalable Windows NT and UnixWare systems and services for the enterprise.

2. Emulation—history and background

Emulation, the imitation of one machine by another, has a long history both in ICL and the IT industry as a whole. It has been used for a number of purposes.

From the 1960s onwards a common use of emulation was, and still is, simulation of the design of a new machine.

With the increasing complexity of computers, the risks associated with designing a new machine led to the idea of emulating the new computer (the subject) first on an existing machine (the host). This meant that before the new machine was built, different designs could be tried, faults in the proposed design identified, and the behaviour of the new machine explored.

A recent example is the simulation of the SY Picode before the SY hardware was switched on, as shown in Figure 2 [Allt et al, 1997].

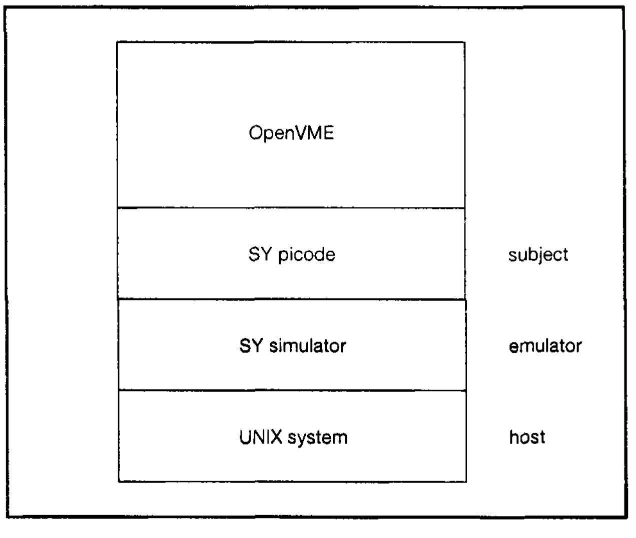

Figure 2: Simulation of SY Picode

The simulator allowed the proposed Picode to be tested on a UNIX system programmed to emulate the SY hardware and its environment. OpenVME was run on top of the Picode test bed up to the point where it required an 'OPER' terminal, i.e. the operator could login to OpenVME. Faults in the Picode and in the alterations to OpenVME needed for SY were removed, and experience gained in how OpenVME behaved in the SY multi-OCP environment.

Another use of emulation is imitation of one machine by another because it is a more cost-effective approach, i.e. it is cheaper or quicker to develop and deliver to the customer. Two historic examples are the 2903 and the Leo III.

In the early 1970s ICL developed MICOS 1 as a base for emulation of other machine architectures. MICOS 1 was a 32-bit word machine with a large number of immediately addressable registers and some features to aid emulation of specific machine architectures.

Its best-known use was in the 2903. Here an emulator, i.e. software running on the MICOS 1, made it look just like a 1903A, complete with an integrated I/O and disc system. This is shown in Figure 3. A specific feature of MICOS 1 made conversion between the 1900 architecture's 24-bit word/6-bit character and the underlying 32-bit word/8-bit byte of MICOS 1 easy and efficient.

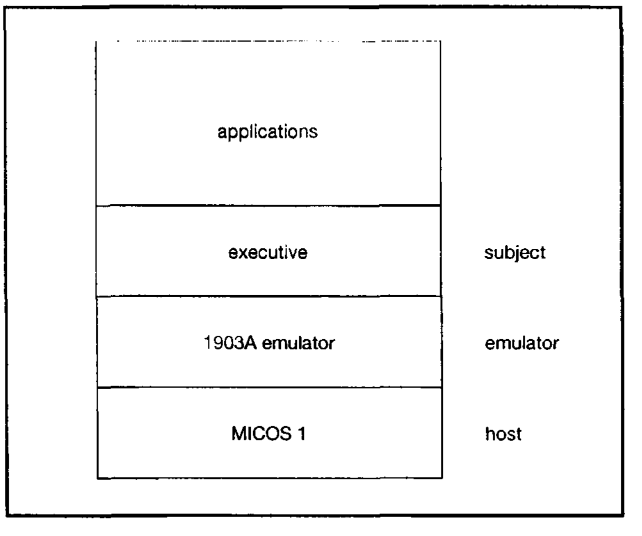

Figure 3: Emulation of 1900 with 2903

MICOS 1 was also used with a different emulator to simulate the 2900 architecture for a proposed small 2900 Series machine that was never marketed. MICOS 1 also formed the basis of several successful 2900 Series communications controllers, the first of which was an emulation of an earlier hardware-based communications controller.

The second historic example of emulation of an existing machine was a request in the late 1970s by a major ICL customer for a 'new' Leo III to run the customer's old, business critical applications. A batch of 2900 Series machines were reprogrammed to emulate the much older Leo III architecture and so run this customer's applications. The significance here is that the 2900 Series was not designed to emulate architectures such as the Leo III.

With simulation of a new design, the functional behaviour of both the emulator and the subject machine being emulated are critical to success. Performance is generally less important; it is more effective to use several host machines, each running the same simulator with each investigating different aspects of the new design. Coverage of the new machine's architecture is often selective. For example, it may omit or simplify some error paths, as it may be more effective to do these tests on the real machine with its real errors. The simulator is only used internally within the development organization and is not exposed to a live customer environment. Its lifetime is limited, since the real machine will supersede the simulation.

Where emulation is intended to run live customer applications, the relative significance of these factors changes. Critical to success are: good, predictable performance; accurate and complete emulation of the subject with nothing left out; a product lifetime consistent with what is expected of live systems and a system that can be supported in the field. This combination of factors makes emulation of live machines a more challenging task than simulation of a new design (although from an engineer's perspective building something new and getting it to work as intended is always interesting).

3. Background to DY

The *Surrey* project set out to develop a CMOS processor that now forms the basis of the SY and LY members of the Trimetra range. One of the Surrey project's objectives was to cover a very large power range with a modular and cost-effective OpenVME node and in this the project was successful [Allt et al., 1997]. During the same period, another project used the significant advances in modular disc subsystems to produce the SMARTarray T-300 disc systems to complement SY.

It was apparent that the I/O and disc subsystems intended for SY would not scale down to match the smaller SY nodes. The result would have been too many boxes, too much floor space, a complex installation process, and so too great a cost. Investigation showed that, with some modest developments, a complete OpenVME system could be fitted inside the same cabinet as the SY processor and so LY was born.

An SY node has up to 4 processors (OCPs), 4 memory cards (MSUs), 16 I/O couplers (IODBs), 2 internode couplers (INDBs), and 3 power cards. High power nodes need all these OCPs, memory, I/O couplers, etc. but lower power nodes do not. A node of LY's power needs only 1 processor, 1 memory card, 8 I/O couplers, and 2 power cards (for resilience). LY fitted its I/O subsystem and disc subsystems into the remaining unused space, and ran them from the SY cabinet's now spare power and environmental control capacity. Figure 4 shows the arrangement of the LY OVS; for clarity the UNS has been omitted.

The LY picture was completed with a new remote support system to give remote diagnostic access from a single point to the SY

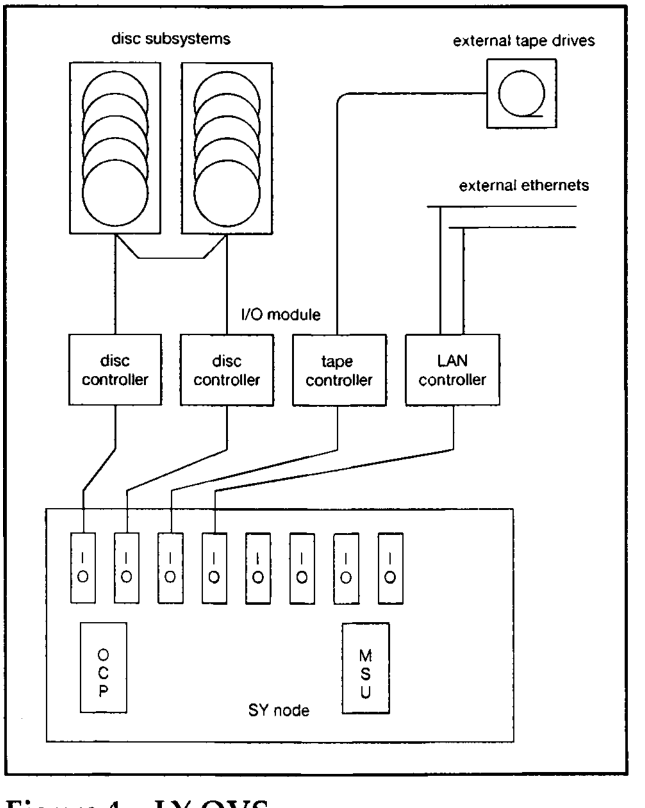

Figure 4: LY OVS

processor, I/O subsystem (controllers) and the disc subsystem. This used parts of the SY power system and software from an existing diagnostic concentrator all mounted on a new board called the Common Support Unit (CSU).

The integration of the SY node, I/O and disc subsystems, and the support system into one cabinet, with remote access to everything, meant LY was now an effective solution for mid-sized OpenVME systems. The Trimetra concept was completed with a UNS (an Intel processor module from Fujitsu running Windows NT or UnixWare with its disc subsystems) mounted in an adjacent cabinet.

During this period the Millennium Architecture was being developed—ICL's innovative vision for future datacentre systems based on industry standard components, such as Intel processors. A small group had started work with OpenVME emulation on workstations. When it became apparent that LY could replace only some of the Series 39 DX and older DM1 systems, then emulation of OpenVME on an industry standard platform (i.e. the host hardware and its intimate software) became a serious prospect. So the Daisy project to deliver DY was born.

4. OpenVME Architecture

The OpenVME architecture has a number of significant features to ensure its efficiency, resilience, security and flexibility. The main features, with particular emphasis on those relevant to emulation, are briefly described here. The full architecture has been described, in more detail, in various publications [The Architecture of OpenVME 55480/001].

It is useful to consider the OpenVME architecture's main features in two groupings:

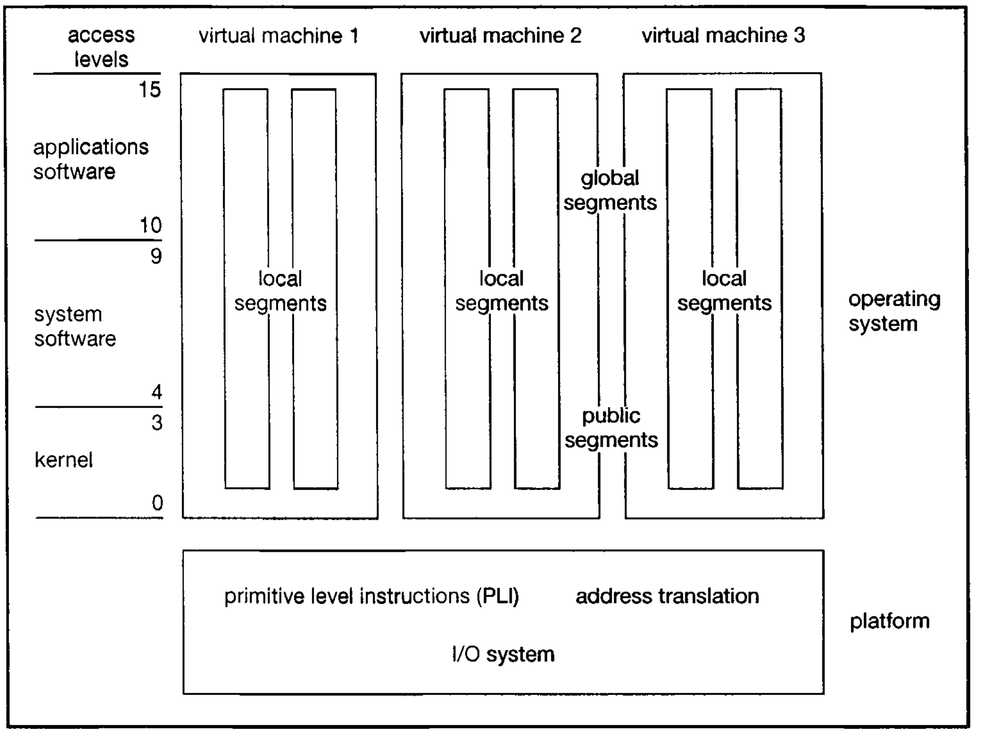

Figure 5: OpenVME Architecture

the operating system (OS) and the platform. The main OS features are Virtual Machines, Virtual Store, and Access Levels. The main platform features are Primitive Level Instructions (PLI), Address Translation and the I/O system, as shown in Figure 5.

A Virtual Machine (VM) is an independent, totally protected environment for each user. A VM has, subject to permission, access to all the underlying OpenVME platform's resources. A VM cannot 'see' or be affected by any other VM, unless this is explicitly agreed. For example, the VMs of a TP Service share information about the service they are providing, but it is only the permitted information that is shared and nothing else.

OpenVME has a Virtual Store to isolate the users from the constraints of the OpenVME platform's real store and this Virtual Store can be bigger or smaller than the real store. The Virtual Store consists of various types of segments. Local Segments contain code or data that is local (and private) to one VM, and cannot be shared with any other VM; they are used by application software (e.g. user programs) and system software. Global Segments are similar to local segments but are shared by two or more VMs; they are used by system software (e.g. IDMS and TPMS), and can be used by applications. Public Segments contain code or data common to all VMs in the system; they are used by the Kernel and system software.

Segments are made up of pages, which, at present, are 1Kbyte in size. The Kernel moves pages into or out of the OpenVME platform's real store as demand dictates and space becomes available.

Access to any code or data in the system, by either OpenVME or user-written software, is strictly controlled through the Access Levels. OpenVME has 16 Access Levels, divided between Kernel, system software and application software. Each segment has separate read, write and execute permissions. For instance, user-written code cannot read or write public data (the user's Access Level is too high for the public segment), and data cannot be executed as code (the segment has no execute permission set).

Each virtual store access must also check that the required page in the segment is actually present in the store. If the page is not currently available then a Virtual Store Interrupt (VSI) is raised and OpenVME suspends the VM at this Access Level until the page becomes available. The page is fetched by a Kernel subsystem running in this VM's lower Access Levels which remain unsuspended.

The underlying OpenVME platform supports, and is part of, the OpenVME architecture. Its main features are PLI, address translation, and I/O.

Primitive Level Instructions (PLI)—i.e. OpenVME's order-code—are a well defined set of instructions and registers available to all the software in the system. OpenVME is a stack-based architecture, so most instructions (e.g. Load, Add, and Store) affect only the stack (a special segment in each VM) and the registers (each VM has its own registers). Other frequently occurring instructions read or write data elsewhere in the VM's virtual store, i.e. local, global or public segments.

Certain instructions must first be checked for permission to execute in the current context, for example, System Call (i.e. check that this user is allowed to call this routine from this part of the system at this time) and Primitive Procedures (i.e. access to features inside the underlying OpenVME platform itself such as generate an interrupt or do an I/O transfer).

Address Translation is the process of checking that the type of virtual store access being requested is allowed and then turning the virtual address into the OpenVME platform's real store address. Address translation first checks the VM's current Access Level against the segment's read, write, and execute permissions. It then turns the VM's Virtual Store Address—i.e. segment number and offset in segment—into the real address of the relevant page in the OpenVME platform's real store and checks that the page is available.

Adjacent or nearby instructions often access the same virtual address. So the OpenVME platform provides architectural support for caching recently used virtual addresses, and the controls to flush or invalidate the caches when this is needed.

OpenVME is an 'in-process' architecture, i.e. everything happens in the context of a VM; there is nothing 'outside' the set of VMs making up an OpenVME session. This is significant as I/O (and other system) requests take place inside the VM initiating the transfer. The I/O system has direct access to the data in the user's virtual store area. There is no need for a context switch or intermediate buffering. Incoming transfers carry identification information which allows the interrupt to be routed to the target VM. A few interrupts, such as communications transfers, which require the message to be partially decoded before the target VM can be correctly identified, are routed via a special VM—the Nodal Target Virtual Machine (NTVM)—reserved for this purpose.

The efficient implementation of the individual instructions that make up PLI, particularly the more frequently occurring cases, and fast address translation, without compromise to security or integrity, are critical to the performance of an OpenVME platform.

5. Emulation of OpenVME

It is now possible to look at emulation of OpenVME on systems such as DY in more detail.

OpenVME and its applications run on an OpenVME emulator. The OpenVME emulator consists of four major software components: OCP, HSC, DNS and PSE. The emulator runs on a host platform consisting of two major components: the Host Operating System (OS) and the Host Hardware, see Figure 6.

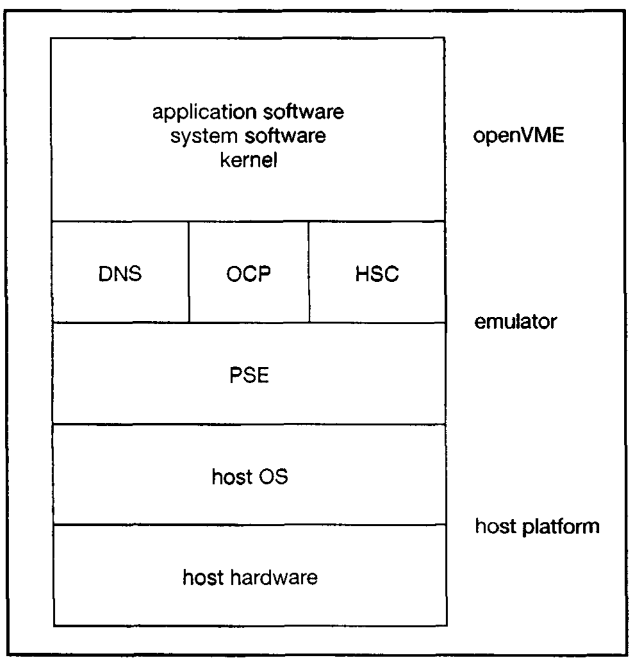

Figure 6: Emulator and its Components

The OCP component of the emulator is functionally equivalent to the SY CMOS OCP and its Picode. Interpretation of OpenVME's PLI (order-code) and address translation form the major part of OCP.

The interpreter fetches each PLI instruction in turn; decodes the instruction type and its operands; checks the instruction is permitted to execute in this context; fetches any data needed from virtual store; carries out the appropriate operation and returns any data needed to the virtual store. The interpreter then does the same for the next OpenVME instruction. Some instructions have a more complex structure but the principle is the same.

Whenever the interpreter needs to access OpenVME's virtual store to fetch code or data, or to store data it must perform address translation, including the Access Level checks.

Interpretation has been used for a long time. MICOS 1 used it for emulating the 1900. The BASIC language is interpreted on most platforms (one particular BASIC interpreter, fitting into 4Kbytes, gave rise to a vast new software business). Java is a more recent example. Interpretation has the advantage that each instruction is treated independently, making the interpreter simpler to write and debug, and relatively compact.

OpenVME has some features that make interpretation faster and more efficient than it at first appears. Code is execute-only (i.e. instructions cannot write to code areas nor modify their address). The interpreter need only translate the virtual address and check the execute permissions once for each page of code (or after a successful jump), not for each instruction. Stack segments have some special properties that similarly reduce the number of checks needed on each access without compromising OpenVME's security or integrity. The emulator also caches recently used addresses as anticipated by the architecture.

The HSC component of the emulator and the DY I/O Coupler are responsible for input/output. They have the same interface to OpenVME as SY. The DY I/O Coupler acts as a 'bridge' between the Intel processor module's I/O bus (PCI) and the DY integral I/O LAN (similar to SMARTfibre). Several I/O Couplers are needed; there is one instance of the HSC component for each physical DY I/O Coupler.

All OpenVME nodes have a Node Support Computer (NSC). This carries out the initial program load (IPL) needed to start running OpenVME when the power is turned on or on request; interfaces the node to OpenVME and provides remote diagnostic access (VISA) to the node. The NSC's capability has improved with each new generation. DY's NSC is the DNS component of the emulator.

PSE provides the common functions and environment for the OCP, HSC and DNS components. A key decision was to design into PSE an abstract platform interface that isolated the rest of the emulator from the details of the underlying host platform (i.e. the host OS and hardware).

6. The DY Platform

The choice of the host platform for the emulator was critical to the development of DY. It is a major factor in the Trimetra concept of offering a choice of operating systems—NT, UnixWare and OpenVME—on a common platform.

The original research on OpenVME emulation was done with UNIX on SPARC workstations. This was a natural choice during the early, investigative phase as the engineers had suitable workstations on their desks and UNIX already had the appropriate tools needed for these studies.

UNIX, like OpenVME, pages information into and out of physical memory according to the demands made on virtual store by the system and its application. The actual paging is invisible to the applications and the users. The UNIX and OpenVME's paging mechanisms have very different approaches to what is paged into or out of the store and to the ways of scheduling the transfers. The resulting conflicts would make optimizing the interpreter difficult and, more importantly, its performance unpredictable and this would be unacceptable. There were also concerns about managing a UNIX system underlying OpenVME, as this would mean yet another interface.

OpenVME's I/O system operates in a real-time environment. Again the problem of scheduling conflicts means UNIX is not really suitable as a host OS for the emulator.

VxWorks is a real-time operating system widely used in several parts of the IT industry for real time and embedded systems. It was already in use in other parts of the Trimetra system and so was chosen as the host OS for the DY emulator.

The choice of host hardware for DY was more complex. Apart from the usual concerns about good quality hardware from reliable suppliers who would continue to deliver and support it, there were other important factors.

The interpreter's efficiency, i.e. how it performs, is influenced by the hardware hosting the emulator. Host hardware with a large cache, combined with careful layout of the interpreter in its memory, will increase the cache-hit rate, so speeding up the interpretation. Exactly how the cache operates also influences the implementation of the emulator's address translation.

Another critical factor was the need to support the host hardware in the field in a manner which our OpenVME customers have come to expect. Industry-standard platforms have, in general, some way to go in this respect, particularly for the remote support of complex systems. Certain features of the host hardware make this easy (or more difficult).

Looking beyond DY, more efficient emulators are required to support the additional features and capabilities of the host hardware. Future emulators will depend on mapping OpenVME's virtual addresses directly on to the host platform's virtual addresses, i.e. using the host hardware to do most of OpenVME's address translation would boost performance. Many commodity processors have two access levels (corresponding to 'system' and 'user') or at most four (compared to OpenVME's 16 Access Levels). The potential to map addresses, without losing OpenVME's security or integrity, depends on exactly how the host hardware implements its own address translation, and how this evolves with later hardware.

During this period of DY development, the Millennium Architecture was confirmed [Procter, 1998] with Windows NT and UnixWare on Intel hardware adopted for future enterprise platforms. Emulation of

byte 0	byte 1	byte 2	byte 3	
01	23	45	67	big endian (openVME)
byte 3	byte 2	byte 1	byte 0	
67	45	23	01	little endian (Intel)

Figure 7: Big-endian and little-endian

OpenVME on Intel hardware completed the picture, with DY becoming the first step.

However, there was a small problem. OpenVME is "big-endian"—byte 0 of a word is the most significant. Intel processors are "little-endian"—byte 0 is the least significant. Figure 7 shows how the integer Hex 01234567 is stored in a 32-bit word.

OpenVME needs to continue to operate in a "big-endian" mode. Everything in OpenVME—VMs, virtual store, Access Levels, PLI, registers, address translation, I/O system, etc.—is "big-endian" and would have to remain so. The emulator, little-endian code on a little-endian Intel host, has to swap the byte order every time it references something inside OpenVME (the Intel order code includes a byte swap instruction for just this purpose). OpenVME's I/O system has the task of passing big-endian OpenVME control structures through little-endian HSC code to the big-endian ICL I/O chips used in the I/O Couplers. OpenVME's big-endian data passes straight through to the DY I/O Coupler.

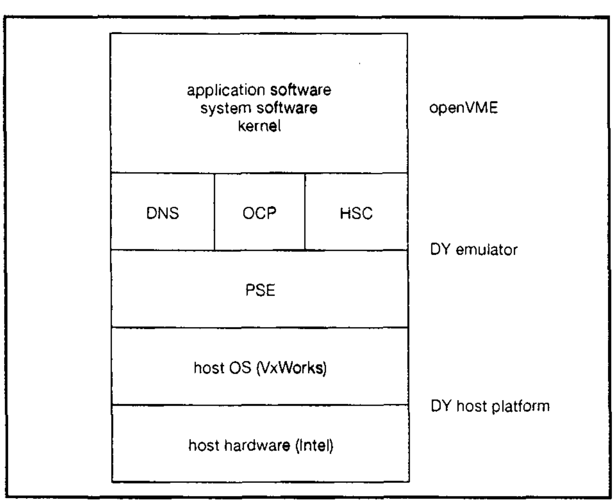

Figure 8: DY Host Platform

The earlier decision to design an abstract platform interface into the emulator made this a much easier task than it would otherwise have been. This point about abstraction providing flexibility for the future was reinforced when the first two Intel platforms considered for DY were subsequently deemed inappropriate for reasons outside the project's control. Finally a suitable Intel processor module, with the relevant features for emulation and remote support of OpenVME, was selected from the Fujitsu range. Together with VxWorks as the host OS this forms the DY host platform shown in Figure 8.

7. DY

Like the rest of the Trimetra range, DY has an OpenVME Subsystem (OVS) and one or more UnixWare/NT Subsystems (UNS).

DY's OVS consists of a Fujitsu M700i Intel Processor module, PCI-based I/O Couplers (PCI is the standard I/O bus on the larger Intel processors), I/O system (disc, tape and LAN controllers) and disc shelves inherited from LY, and an integrated support system. This pro-

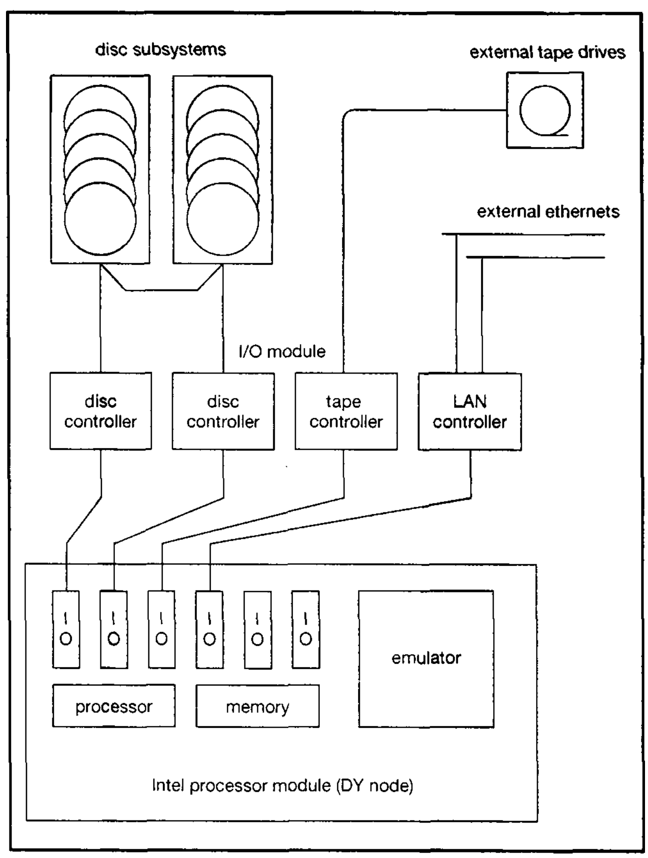

Figure 9: DY OVS

vides a small OpenVME system with the same features and the RAS facilities available to the top of the range customers. Figure 9 shows the arrangement of the DY OVS; for clarity the UNS has been omitted.

A single DY cabinet has enough space to mount the second Fujitsu M700i Intel Processor module and some discs for one UNS. More or larger UNS modules can be mounted in additional cabinets. The whole DY system is under common system management. The customer and user now has the choice of operating systems, services and applications needed by a modern, progressive business.

Choosing what information the emulator had cached and over what period, and how to best exploit the Intel hardware to tune the performance required a lot of investigation as well as experience with building previous OpenVME platforms. In spite of early doubts, the DY interpreter achieved better than its target performance on the Intel Pentium Pro processor.

DY's new I/O Coupler had to interface the Intel PCI bus to the DY I/O controllers, the controllers having been inherited from LY, which turned out to be pushing the limits of the gate array technology used to link ICL's (very fast) I/O chips to PCI. This reinforced the message that to deliver reliable and performing enterprise technology, even when most of it is bought from elsewhere, needs the wide range of skills that ICL is able to bring to a problem. ICL has a strong belief in skills being critical to its own and the customer's future.

A common support system (CSU) was developed for LY. DY then enhanced it to provide the remote support and environmental control needed for the project.

The DY CSU monitors the OVS processor module and its boot sequence, the I/O controllers and the disc hardware. It provides remote diagnostic access to and power control of DNS (DY's Node Support Computer), the I/O controllers and the disc hardware. It also runs the display panel at the front of the cabinet. The combination of the CSU and the DNS component of the emulator bring the support capability of an industry standard Intel platform up to the high standards expected of enterprise systems supporting line of business applications.

DY demonstrates that a robust and secure OpenVME system runs on Intel processing hardware with a performance that is effective for ICL's customers. DY means there is no change to the applications, no change to operations, but more disc space and a faster I/O system. The faster I/O can make some batch jobs run much more quickly, although the actual performance depends on the workload.

DY also means that, for the first time, an OpenVME node runs on the same Intel platform as Windows NT or UnixWare.

8. The Future of OpenVME on Intel

DY is a major milestone in the Millennium programme since it delivers OpenVME running on an Intel processor. Trimetra introduced the concept of a choice of operating systems under a single, integrated systems management image and support route. The Millennium programme continues this evolution by consolidating processing on to common Intel hardware and peripherals on to common standard subsystems; all with the flexibility to re-allocate resources as the business demands. The SY CMOS processor will continue to supply the needs of the high end OpenVME models until the combination of Intel performance growth and emulation technology allows it to be superseded. So what comes next?

Historically Intel processors have doubled in speed roughly every eighteen months, with architectural alterations every few years and more substantial changes less often. For example, the Intel 386 is a 16-bit processor, whereas the Intel 486 is a 32-bit machine. These architectural changes are hidden from OpenVME by the emulator.

However the remote support capabilities of the platforms using industry-standard processors have lagged behind what enterprise systems, including OpenVME, have come to expect. DY had to add significant features to its CSU and DNS to meet customer and ICL's expectations. Emulation using faster platforms is one thing; emulation using faster platforms that cannot be supported properly is not useful. The remote support capabilities of some Intel platforms are now beginning to improve rapidly and de-facto industry standards are beginning to emerge. So future OpenVME systems can expect to take advantage of faster Intel platforms with good remote support in due course.

The emulator's efficiency can be improved by a number of techniques, all of which boost performance. As OpenVME on Intel hardware gets faster, its I/O system must be improved to keep up. More powerful systems demand more resilience and often require disaster tolerance as an option (it is mandatory in some cases).

Future emulators will map OpenVME's virtual addresses on to the underlying Intel hardware's virtual addresses. Like the OpenVME platform, Intel hardware also has its virtual-to-real address translation, but it uses a different architectural model and a different page size to OpenVME. This emulation technique is referred to as Hardware Address Translation (HAT) and will be used instead of the Software Address Translation (SAT) of DY.

With HAT most of the emulator's address translation, including the Access Level checks, is done by the Intel hardware, with the emulator now only handling the set-up and exceptions in its software. To do this OpenVME has to use a page size that is the same as that used by the Intel hardware, i.e. 4Kbytes rather than the 1Kbyte used to date. A side effect of the larger page size is that some checks occur less frequently in the emulator, so speeding up the process still further. The OpenVME architecture ensures that this change to the page size remains invisible to the system software and applications but the store occupancy increases.

Interpretation of PLI can be developed into translation of PLI. Here the emulator compiles the PLI (i.e. OpenVME's binary order code) into the Intel processor's own order code and runs it directly. Effectively a sequence of OpenVME instructions (such as a loop) is converted into an equivalent sequence of Intel instructions that deliver the same result, but much faster. Unusual instructions and instruction sequences that cannot be compiled are still interpreted as before. This allows for a progressive, evolutionary approach to the introduction of PLI translation, rather than a big bang. Again the compilation is invisible to OpenVME and its applications.

Some other manufacturers have already started doing PLI translation for their 'mainframe' architectures. In some cases they have had to restrict or fence its use as their applications can alter their own system or user code. A program altering its own code is a technique that goes back over half a century to the first stored program computer (the University of Manchester's Baby). Altering code dynamically means the translated code is no longer equivalent to the now altered code. It also invalidates state information retained in the emulator, so checks have to be redone. OpenVME's 'pure' architecture avoids these restrictions as the system and applications cannot write to code areas.

A new project is taking on the task of moving OpenVME's external I/O subsystem on to Intel. The I/O subsystem will then run on the same processor module as the emulator, dispensing with the need for separate hardware. More and faster I/O connections will provide the I/O bandwidth for the larger systems reachable with later emulators. Ensuring the I/O throughput scales with the OpenVME power will be critical as many OpenVME applications make very high demands on the I/O power.

Moving the I/O subsystem on to Intel hardware also allows ICL to take advantage of the processor and peripheral clustering subsystems being developed by the industry. Split-site working and disaster standby have been available for many years with the Series 39 and are available with SY. With industry-standard peripheral subsystems using open networks there is the prospect of clustered OpenVME on Intel systems separated by hundreds of kilometres.

This progressive approach means that the OpenVME node and its I/O controllers become 'just software', making the Millennium concept of multiple operating systems on a common platform a reality. ICL's strategy with OpenVME on Intel hardware provides customers with confidence in the future of OpenVME systems and in ICL's ability to keep them benefiting from future technology.

Acknowledgements

I would like to thank all members of the Daisy Project for their help in producing this article.

Java is a trademark of Sun Microsystems, Inc. in the United States and other countries.

Pentium Pro is a trademark of the Intel Corporation.

SCO and UnixWare are trademarks of the Santa Cruz Operation Inc. in the USA and other countries.

UNIX is a registered trademark of The Open Group.

VxWorks is a registered trademark of Wind River Systems Inc.

Windows NT is a registered trademark of Microsoft Corporation in the USA and other countries.

Glossary of Terms

Access Level	The level of access permissions in a Virtual Machine
Address Translation	The process of and checks needed to turn a Virtual Address into the actual (real) address in the physical memory
Big-endian	A processor where Byte 0 of a word is the most significant
CMOS	Complementary Metal Oxide Semiconductor. High density microchip technology used in SY systems
Controller	The I/O unit connecting peripherals (discs, tapes, LANs) to SMARTfibre or Macrolan
CSU	Common Support System
Daisy	The project and internal code name for DY
DM1	Distributed Mainframe. The smallest member of the Series 39 generation of mainframe processors
DNS	DY's NSC
DX	The successor to DM1. A member of the Series 39 generation of mainframe processors
DY	ICL's first OpenVME system running on Intel and the latest addition to the Trimetra range
Emulation	The imitation of one machine by another
Ethernet	A local area network operating at 10 or 100/Mbit per second
HAT	Hardware Address Translation, i.e. address translation done mainly by the host hardware
Host	A machine underlying and supporting the emulator
HSC	High Speed Coupler. The OpenVME coupler responsible for input/output
I/O Coupler	Input/output Coupler
I/O	Input/Output. The transfer of data between a processor and peripherals such as a disc or tape drive or a communications LAN
ICL	International Computers Limited plc
INDB	Internode Coupler
Integral I/O LAN	In internal I/O network used on LY and DY, similar to SMARTfibre
Intel	Intel Corporation
Internode	Connection between the nodes of a system to form a multinode system
Interpreter	An emulator which takes each order-code instruction in turn, decodes and executes the instruction, and then proceeds to the next instruction
IODB	I/O Daughter Board providing the connection between SY and SMARTfibre or Macrolan
IPL	Initial Program Load. The process which loads a node (SY, LY, DY, etc) with the

	firmware and software needed to start running OpenVME
Kernel	The lowest levels of OpenVME
LAN	Local Area Network
Little-endian	A processor where Byte 0 of a word is the least significant
LY	The mid-member of the Trimetra range of mainframe processors
Macrolan	ICL's optical fibre, local area network, connecting the nodes to the peripherals
Mainstore	OpenVME's memory
Memory	The computer's memory
Millennium	ICL's vision of data centres supporting integrated, multiple services on multiple servers of the same or different types
MSU	A unit of mainstore
Multinode	Many nodes connected together to run an OpenVME system to provide more power and better resilience
Node	A processor complete with its memory, I/O Couplers, and environmental control system capable of running OpenVME
NSC	Node Support Computer that monitors and provides general support for the node
NTVM	Nodal Target Virtual Machine. A special VM reserved for use by OpenVME
OCP	Order Code Processor. The processing unit which executes PLI, controls access to mainstore and services interrupts
OpenVME	ICL's operating system
OPER	An OpenVME session specifically used by a systems operator. Usually the first session available externally when OpenVME is started
Order-code	The instruction-set of a particular processor. OpenVME's order-code is more often referred to as PLI
OS	Operating System, eg OpenVME, Windows NT
OVS	OpenVME Subsystem. Part of a Trimetra system
Page	The smallest contiguous element of physical store
PCI	Peripheral Connect Interface. The internal bus supported on Intel processors
Picode	The machine code obeyed by the SY processor
Platform	The combination of operating system and the processor hardware needed to run it
PLI	Primitive Level Interface. The standard interface to the OCP seen by OpenVME and its applications
Primitive Procedure	Special instructions providing access to features inside the underlying OpenVME platform itself, e.g. generate an interrupt or do an I/O transfer
PSE	The part of the emulator providing the common functions and environment for the OCP, HSC, and DNS components on DY
SAT	Software Address Translation, i.e. address translation done by the emulator's software
Segment	A part of Virtual Store containing code or data
Simulation	Emulation used to simulate the design of a new machine
SMARTfibre	ICL's high speed, optical fibre, local area network, connecting the nodes to the peripherals
SPARC	A type of processor supporting a specific machine code
Subject	A machine being emulated
SX	The largest member of the Series 39 generation of mainframe processors
SY	The largest member of the Trimetra range of mainframe processors
Trimetra	OpenVME running alongside, and integrated with, Windows NT and/or UnixWare
UnixWare	SCO's operating system. A particular variant of UNIX for the enterprise market

UNS	UnixWare/NT Subsystem. Part of a Trimetra system
Virtual Store	The virtual memory provided by OpenVME
VM	Virtual Machine. An independent, totally protected virtual computer provided by OpenVME
VxWorks	A real time operating system
Windows NT	Microsoft's operating system

Biography

Andrew Brightwell joined ICT Stevenage in 1967 with a BSc in Physics from Leicester University. He joined a group developing and delivering the first Optical Character Readers. He then moved to developing peripheral and communications controllers for what became the 2900 Range. In 1975 he moved to Kidsgrove.

He was involved in developing the system concepts for what became the Series 39 and was the architect for its I/O system. He moved to West Gorton, Manchester in 1986 and joined Systems Architecture.

He was the systems architect for the Daisy Project with overall responsibility for the design of DY and its relationship to Millennium.

Trimetra UNS

C.J. Martin and C.P. Stewart

ICL High Performance Systems, Manchester, UK

Abstract

This paper describes the architecture of the Y-series Trimetra range of systems. The Y-series systems represent a specific instantiation of the HPS Millennium architecture, packaged to address the requirements of the existing OpenVME customer base. The system provides support for OpenVME, UnixWare and NT operating system environments within a single system framework with properties similar to those of traditional mainframe systems. The approach is aimed at allowing existing OpenVME users to protect and exploit their existing investment in OpenVME applications and data while providing a flexible forward path for developing and running applications on other operating system environments.

1. Introduction

The objective of this paper is to provide an overview of the concepts, structure and facilities provided by the Y-series Trimetra range of data centre systems. The Y-series variant of the Trimetra range is built on common technology used across the Trimetra range, and is packaged specifically for use in the OpenVME data centre market.

1.1 System objectives

Trimetra is aimed at providing a range of systems capable of meeting all of the requirements applying to the datacentre components of a user's IT system.

In the past, the requirement for data centre systems has been met primarily through the use of large central mainframe systems, normally based on the use of a single operating system platform. This requirement has been met extremely successfully over a long period of time by the Series 39 range of server systems.

More recently, however, various processing systems have been developed, including systems based on different versions of Unix and more recently Microsoft's Windows NT. This has resulted in a situation where users have found a need to use several different systems, often supplied by different vendors, in order to run the set of applications and functions needed within their data centre environment.

While the use of multiple platforms in this way has allowed users to run the required set of applications, this approach has proved to have a number of drawbacks associated with it, for instance:

- The costs associated with duplication of processors and peripherals, management systems, support processes and so on across the various platforms
- The lack of flexibility and future-proofing in the system components: this has been a particular problem owing to the rapid evolution in IT system technology, the attendant risks associated with choice of technology and the ability to protect system investment as the technology changes
- The difficulties in integrating applications and data across sets of systems often supplied and supported by multiple vendors. Cross-platform data integration is of particular importance to many traditional mainframe systems users, where there are frequently pressing business requirements to allow data in their existing systems to be accessed and manipulated by applications running on Unix and Windows NT based platforms.

The Trimetra range has been conceived as a way of addressing these problems by providing a system encompassing multiple separate operating system environments, within a single system framework, with properties similar to those of the traditional mainframe system.

For existing OpenVME systems users, the approach is aimed at allowing them to protect and exploit their existing investment in OpenVME applications and data, while providing a flexible forward path for developing and running applications on other operating system environments, as required.

1.2 System structure

Figure 1 provides a conceptual view of the Y-series Trimetra system structure.

The main features of the system are:

- The inclusion of a set of standard branded operating system platforms providing complete support for OpenVME and industry standard applications. The initial system includes OpenVME with a choice of a UnixWare or Windows NT platform. The system is scalable by use of multiple platform instances. The OpenVME component is referred to as the OpenVME subsystem, or OVS, and the UnixWare and Windows NT component (UNS) is referred to as the UnixWare subsystem or the Windows NT subsystem
- The ability to share peripherals amongst all of the component platforms in the system. The peripherals include existing OpenVME devices such as high speed laser printers and tape devices
- The promise of a 'single system' view to be provided to application users of the system, that is the mapping of applications to the various platforms in the system is invisible to the application users
- Software facilities to allow data to be exchanged between applications on the various platforms, including data within existing OpenVME databases
- A common framework to provide for the management and support (by the user and ICL) of all components of the system.

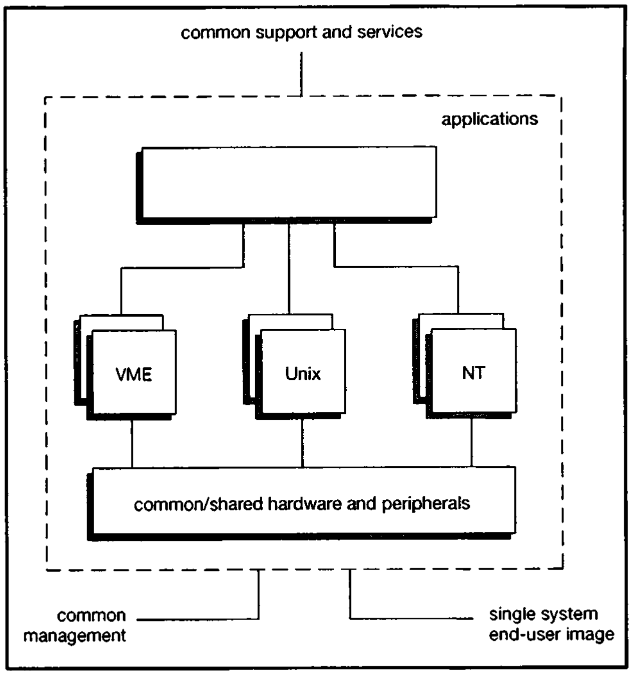

Figure 1: Trimetra structure

The system design is aimed primarily at integrating (in terms of software, hardware and services) a range of selected ICL and third party hardware and software products, with additional product development being targeted at integrating these component products into a single system framework. This represents a very flexible approach to the development of the system which will, in future, allow it to evolve rapidly to incorporate new versions of the component hardware and software products as they become available.

From an end-user point of view, the Trimetra system offers:

- A forward path for existing OpenVME systems
- A (potentially extensible) range of industry-standard application platforms
- Investment protection through continuing support for all aspects of existing Series 39 systems, and the ability to retain data within existing OpenVME databases and applications while simultaneously exploiting it from applications on other Trimetra platforms
- Reduced cost of ownership through the use of common hardware and a common

management and support system for all components of the system

- Flexibility and future proofing, through the decoupling provided within the system design between the choice of operating system environment and the hardware, as well as the management systems used to support these environments.

2. Physical structure and configuration

This section describes the overall physical configuration of the Y-series Trimetra system.

2.1 General structure

Figure 2 illustrates the generic physical structure of the Trimetra system.

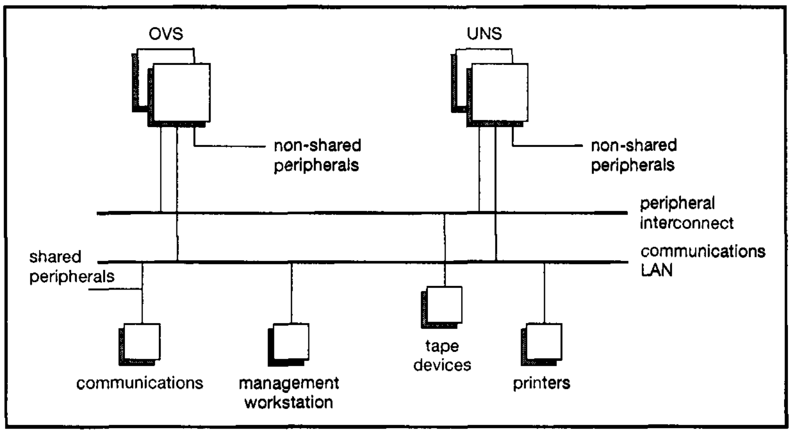

Figure 2: Trimetra Physical Struc-

The main components of the system are as follows:

- A number of OVS subsystem variants, labelled SY, LY and DY, are available. Dependent on the particular system variant, multi-node OVS configurations can be used as normal for resilience and performance purposes. A single Trimetra configuration contains at most one OVS. The system can, depending on the type of OVS, be partitioned if required, in which case the partitions of the system are still all part of the same Trimetra system
- The UNS comprises branded UnixWare and/or Windows NT subsystems running on Intel hardware. A single Trimetra configuration may include multiple processor modules, where a module is a single instance of the operating system running on its own instance of the Intel hardware platform. A single system can contain mixtures of UnixWare and NT systems
- The Systems Management Workstation (SMW) provides a single point of access for the administration of all of the component platforms in a Trimetra configuration. The workstation is based on a Windows PC configured with the software packages needed to administer the various components of the system. If required, multiple workstations can be used within a single Trimetra system and the workstations can coexist with existing dedicated OpenVME management terminals for transition purposes
- Communication between the components of the system, that is the UnixWare, Windows NT and OpenVME subsystems and the system management workstations, is carried over shared communications LAN connections. These connections may currently be either Ethernet or fast Ethernet. Communication between the systems is based on use of OSI and/or TCP/IP based protocols depending on the requirements of the communicating applications
- All Trimetra systems must have at least one common LAN to which all components of the system are connected. Multiple LANs may be configured, if required, for resilience or throughput purposes
- A number of different types of peripheral interconnect are used within the Trimetra system:

  - Connection of disks and tapes to OpenVME is via SMARTfibre or Macrolan or, in the case of the DY and LY server systems, via an integral in-cabinet controller-to-SCSI connection. Disk and tape connections to the UNS processor modules are via PCI-based SCSI connections

- Printer connections to both the OVS and UNS are normally made via Ethernet, although printers can also be connected to individual processor modules via the parallel port on the UNS module
- Peripherals in the system may, depending on the type of peripheral, be dedicated to individual OVS or UNS platforms, or may optionally be shared by several of the components of the system

- Shared external network access to the system is provided via network controllers connected to the shared Ethernet or fast Ethernet LAN (either directly, or indirectly via network bridges or routers). Using this type of connection, remote systems or end users have direct access to applications running on both the OVS and UNS subsystems.

2.2 OpenVME subsystem (OVS)

The OpenVME subsystem (OVS) in current Trimetra configurations comprises a single OpenVME system, which, depending on the system variant, may be capable of being partitioned into multiple OpenVME partitions if required. There are currently three OVS variants, the SY variant, which is designed for larger system users, the LY variant, which is designed for medium to small users and the DY variant which is designed for the smaller end of the system range. The SY and LY variants are based on the use of CMOS chip technology with the DY variant being based on standard Intel technology.

The main features of the OpenVME subsystem are:

- An SY can be a multi-node system, that is the system can include a number of OpenVME cabinets, each forming a single node of a multi-node OVS. LY and DY systems are single node systems each contained within a single cabinet
- Each node of the system runs a single instance of the OpenVME software, running on one or more processors (depending on the system model)
- On SY systems connection to communications LANs or to disk and tape devices etc., is provided via controllers connected to SMARTfibre LANs. On LY and DY systems integral adapters are used for connecting purposes
- The OVS can have access to non-shared peripherals, including peripherals and network controllers carried forward from Series 39 OpenVME systems.

Each of the SY, LY and DY cabinets described above requires a separate communications connection for use in certain diagnostic situations. The asynchronous connection is made by linking an optical modem to a communications port within the cabinet. The optical modem in turn is connected via a serial fibre link leading from the cabinet to an external communications link.

2.3 UnixWare and Windows NT Subsystem (UNS)

The UnixWare and Windows NT subsystem (UNS) of a Trimetra system can include a number of physical modules, each of which runs its own instance of the appropriate operating system.

Free standing UNS cabinets are racked constructs which can hold a number of components. In summary, a single cabinet may contain:

- One or two processor modules each containing up to 4 x 200 MHz Intel Pentium Pro processors, up to 2 Gbytes of memory, 2 x 4 Gbyte internal disks (used primarily to hold system filestore and dump space)
- Each processor module in a separate rack-mountable unit running its own separate instance of the chosen operating system
- In addition, within each processor module, a CD ROM for software distribution/installation, a diskette drive and a DAT tape drive plus industry standard I/O capability (i.e. PCI slots, EISA slots, Parallel printer port, Serial ports)
- Housing for locally connected disk subsystems.

The cabinet provides the cabling access necessary for the interconnection of the components within the cabinet, and for the external connectivity needed by the components in the cabinet.

On DY systems there is an option to include the first UNS module within the same cabinet that houses the DY OVS system. In this configuration two disk shelves can be configured as one for use with the OpenVME subsystem and one for use with the Windows NT or UnixWare subsystem.

An uninterruptible power supply (UPS) can be used to protect the equipment within the cabinet. In cases where a UPS is used, it is housed as a free-standing unit outside the cabinet.

Each processor module requires a local console keyboard directly connected to it for initial bootstrapping and for use in certain diagnostic situations. The console is housed outside the cabinet:

- Each processor module in the cabinet can have its own console if required
- Alternatively, a single console can be connected to a switch unit housed outside the cabinet, which allows the single console to be used by both modules within the cabinet.

Each processor module requires a separate asynchronous connection for use in certain diagnostic situations. The asynchronous connection is made by connecting an optical modem to one of the serial ports on the module within the cabinet. The optical modem, in turn, is connected via a serial fibre link leading from the cabinet to an external communications link.

3. Peripherals

Peripherals provided on Trimetra systems fall into two categories. These are:

- Common or shared peripherals—peripherals of this type can be used by any of the operating systems within the Trimetra system, either concurrently or by reallocating the peripherals for use between the components of the system over a period of time
- System specific peripherals—peripherals of this type include retained peripherals from pre-Trimetra OpenVME systems, Windows NT and UnixWare system-specific peripherals. Peripherals of this type cannot generally be used across the components of the system.

System-specific peripherals are supported in the case of the OpenVME system to allow customers to carry forward their often substantial investment in existing OpenVME peripherals. In the case of the UnixWare and NT systems, peripherals of this type are needed to support particular types of applications provided on these systems, and any conformant UnixWare or NT peripheral can be used with the corresponding Trimetra platform types.

This use of common peripherals represents the target approach on Trimetra, since it allows a user to purchase sets of peripherals as a common resource, with the use of the peripherals being potentially changed over a period of time as the workload mix on the system changes. Peripherals of this type are based on the use of best-of breed third party products, and a recommended core set of products of this type are made available via ICL channels as part of the standard Trimetra product range.

Examples of common peripheral systems within the current Trimetra systems are described in the following sections.

3.1 Disk systems

Commonality of disk systems is provided at two levels in the current systems. In the simpler disk configurations, common Storageworks disks are used across all the Trimetra platforms, including OpenVME, allowing individual disks to be physically moved between platform types and instances as required. A more advanced form of disk sharing is the use of disk arrays, such as those provided in Symmetrix/EMC disk systems, which support multiple simultaneous host connection to the disk controllers. In systems of this type disks can be reallocated between the components of the system dynamically, without the need for physical relocation of system components.

3.2 Tape systems

As with disks, commonality of tape systems is provided at two levels. A common set of tape devices and controllers is available for use on all components of the system and these are based on the use of DLT tape products. In addition, a range of multi-host attach tape libraries/silos is provided, based on systems sourced from StorageTek.

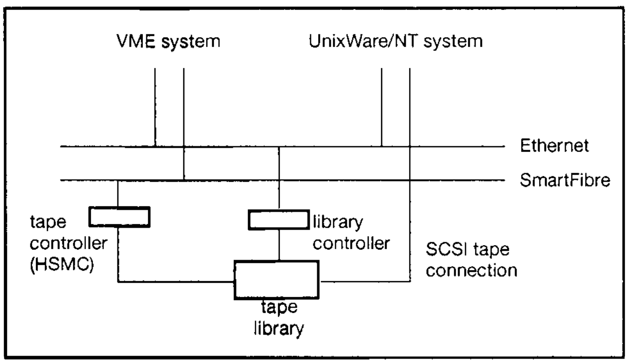

Figure 3: Tape Library Access

As illustrated in Figure 3, the tape libraries provide shared LAN access to the library robotics controller, and multiple tape drives, allowing several systems to access the common tape pool simultaneously. Tape resources from the common pool can be reallocated between the components in the system under software control only, without the need for physical relocation or rewiring of system components.

3.3 Printers

A wide range of standard printers are capable of being used on Trimetra systems, ranging in sophistication from simple LAN based printers to very expensive high speed laser printers. Printers in the Trimetra system are shared through the use of print server software, which allows the use of the printers to be shared between the components of the system, as illustrated in Figure 4.

A number of physical and software configurations are possible in the system depending on the nature of the printer connec-

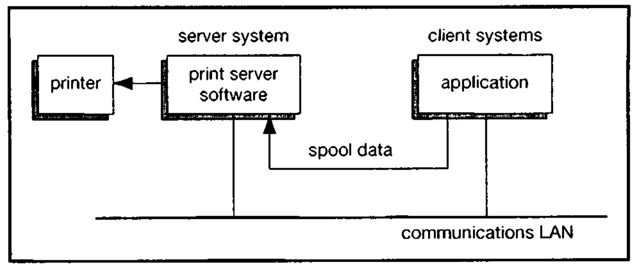

Figure 4: Shared Printer Access

tions and the applications involved. Thus, for instance, in the case of sophisticated high powered laser printers connected to the OpenVME system the print management software can reside on the OpenVME system, allowing NT and UnixWare system components access to the printers themselves, and to the more advance print management features provided on the OpenVME part of the system. Alternatively, simpler printers can be connected to the shared Ethernet LAN via off-the-shelf printer adapters/servers, permitting direct multi-host access to the printers across the LAN.

Advanced print management software is made available on the Trimetra system via collaboration with Gandlake, a specialist supplier of software of this type. Facilities provided in this way include, for instance, secure printing, print routing, advance scheduling facilities, page design, print reformatting etc..

4. Networking facilities

As with the rest of the Trimetra system, the objective of the Trimetra networking system is to provide facilities based on the use of standard networking products which are capable of being shared across the multiple platforms in the system. The networking features can be broken down into two parts, i.e. those relating to the physical network connections and those relating to the communications protocols and standards used over these connections.

In physical terms:

- Communication between the various components of a single Trimetra system is achieved by connection of the components to the common system LANs, which are currently Ethernet and/or fast Ethernet based

- Communication between the components of the system and other external systems is achieved through the use of standard bridges and routers such as those provided by 3-COM and Cisco systems connected to the shared system LAN. Use of this type of arrangement allows all the components of the system to have accessed via shared network adapters and physical connections to external networks.

In addition to this target configuration, support is also provided in the system for the direct connection of unshared network adapters to individual system components. Connections of this type are used, as in the peripheral case, to allow existing OpenVME networking equipment to be carried forward, and to permit direct connection to the UnixWare and NT systems of network adapters needed to support particular types of specialized applications where indirect LAN access to network adapters would not be feasible.

In terms of the protocols carried across the physical interconnections:

- A basic common set of networking facilities, based on the use of TCP/IP and related standards, is supported across all the components of the system. These networking facilities are used both for intercommunication across the shared system LAN between the components of a single Trimetra system, and for communication between the components of the Trimetra system and external systems.
- In addition to the core protocol set, support is also provided, in particular components of the system, for selective OSI networking to support existing OpenVME based interworking requirements, as well as for Novell and NT based networking to meet specific application requirements of the UnixWare and NT systems.

The basic networking facilities provide the capability for interactive and bulk access transfer of data between the components of the system and external systems, as well as the basis for the more complex forms of interworking which are possible between the various systems involved.

5. System Management

The objective of the Trimetra management system is to provide a single consistent framework within which all the components parts of the system can be managed. There are two main components of the management system as follows:

- A common administration and operation system, allowing all the platforms, applications and databases within the system to be controlled from a single management workstation, or, if desired, from a small number of such workstations, each controlling a subset of the components and management functions
- A common set of facilities for the support of the components of the system, covering problem notification, diagnosis and correction.

The following sections provide an overview of these two parts of the management system.

5.1 Administration and operation

Within the overall Trimetra system the features which need to be managed can be separated into two main categories as follows:

- Some aspects of the system should ideally be managed via a single management application operating across all the components of the system. An example of this would be high level scheduling of system components, where there is a need to be able to co-ordinate activities across multiple platforms
- Some aspects of the management system are entirely local to individual platforms,

for instance, the process of cataloguing and controlling local I/O couplers on a particular platform is a function which is completely local to that platform.

The management functions on the Trimetra system and the framework used to access the management functions are aligned with the above view of the distribution of such functions.

The main features of the administration system on Trimetra can be summarized as follows:

- All aspects of the administration and operation of the features of the system can be carried out from a single system management workstation (SMW) or, if desired, from a small number of such workstations
- The management system includes provision for single management applications which can control various aspects of the system across all the component platforms in the system
- Management of local aspects of the individual platforms is carried out using the 'native' management interfaces provided by the basic platforms, for instance, standard SCL interfaces on OpenVME, or the SCO management desktop, which provides the standard management interface for UnixWare systems.

The system management workstation is supplied by ICL as a combined hardware/software package. The package includes the following:

- The SMW platform itself—the platform is a Pentium based Windows PC system and includes CD ROM and tape backup facilities
- A basic set of networking and client products suitable for managing and operating all of the components of the Trimetra system

A single SMW is supplied as a standard component of all Trimetra systems. However, additional workstations can be added, if needed, to allow various management functions to be partitioned between the workstations, or to provide duplicate workstations for resilience purposes.

The SMWs are connected physically to one of the shared Ethernet LANs on the Trimetra system—different workstations can be connected to different LANs, if desired. Communication between the workstations themselves and the OpenVME, Windows NT and UnixWare subsystems makes use of OSI and TCP/IP based connections carried over the common LANs.

The various management facilities provided on the SMW are accessed via the Trimetra System Management Desktop. The management desktop provides a simple framework into which the various management facilities, accessible from the workstation, are hooked and made visible. The framework provided is based on the use of standard Windows facilities and involves grouping the various management applications into sets of nested folders. These contain groups of related applications which are accessible as icons within the folders.

The management desktop may contain three types of groupings of applications as follows:

- Applications which manage on a system wide basis. These applications are started from the top level management desktop
- Applications which are specific to one platform instance. These applications are grouped together into system-specific folders, one for each separate component of the Trimetra system
- Applications which are specific to a single function, but for which separate application instances exist on each component platform. Applications of this type are grouped into function-specific folders, one per separate function.

System-specific and function-specific folders are implemented as nested folders, accessible from icons on the top-level management desktop. Figure 4 illustrates the folder/icon structure of a typical desktop system.

The applications which can be run from the desktop fall into two basic categories:

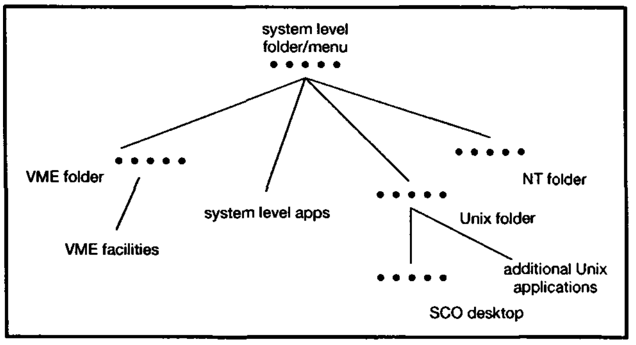

Figure 4: Management Desktop Folders

- A standard set of management products is supplied as part of the overall workstation package
- If required, additional management applications can be added to the workstation and configured within the overall framework as dictated by the user's particular requirements. Applications of this type may be system wide or platform specific, and may be supplied either by ICL or by other third party suppliers.

System wide applications which are supplied as part of the workstation package include:

- Cross-platform problem management
- Cross-platform performance and activity monitoring
- Cross-platform disk administration.

Additional cross-platform applications can be added to the desktop if required, accessed from icons held on the top-level desktop. The additional applications which are available are all based on the use of selected third party products, and include facilities for cross-platform archiving, scheduling, system monitoring and capacity management.

Platform-specific applications provided as part of the standard workstation package include the following:

- In the case of OpenVME, standard 7561 and VT320 emulators, which are used to establish standard OpenVME terminal sessions for administration and operation of the OpenVME system
- In the case of UnixWare, VT320 emulators, X-Windows emulators for access to the standard UnixWare management desktop, and server management client software, which is used in the management of the UnixWare Intel hardware
- In the case of NT, standard NT administrations software together with the Remotely Possible remote access product, which can be used to allow multiple NT domains to be managed from a single NT SMW.

As with the cross-platform applications, additional system-specific applications can be added to the workstation if required. In particular, a number of Windows based optional products are available from ICL which allow OpenVME to be managed through a Windows style user interface rather than via the traditional terminal emulation user interface.

Installation and configuration of the standard SMW package is carried out by ICL as part of the initial installation of a Trimetra system. Installation of optional management products can be carried out by users themselves, using on-line help facilities provided as part of the workstation package.

5.2 Problem management and support

A single consistent set of support procedures is provided on Trimetra systems covering:

- Error detection and reporting
- Problem diagnosis
- Error correction.

The error detection and reporting system provided on Trimetra systems is as follows:

- Hardware and software errors detected within the OpenVME subsystem are detected and reported to the OpenVME

based Support and Maintenance (SAM) system, which also provides the common route for all error handling on the Trimetra system as a whole
- Faults detected on the UnixWare and NT components of the system, which include faults in the server and attached peripherals, are detected by UnixWare and NT-based error detection software supplied as part of the Trimetra system. This software reports the errors detected on the UnixWare system to the SAM software on the OpenVME system, thereby providing the single error handling route
- In cases where errors occur which cannot be reported automatically, for instance system dead conditions, facilities are provided for manual reporting of the errors into the SAM system
- Errors logged via the SAM system are reported electronically to ICL via the SAM problem management system irrespective of the source of the error within the Trimetra system.

The overall process is illustrated in Figure 6.

Once logged into the SAM system, the status of the errors can be checked on the database in the same way for all components of the Trimetra system.

In most cases, problems notified to ICL will be capable of being diagnosed and solved on the basis of the information transmitted to ICL by the automatic error reporting system described above. In some cases, however, it may be necessary for ICL diagnosticians to access the user system on-line and this is achieved using a telesupport communications link to the ICL diagnostic centre.

Using the telesupport link the ICL diagnosticians can obtain direct access to information within the customer system, including details of OpenVME, UnixWare and NT dumps, which would not be transmitted as part of the automatic error notification procedures.

The following telesupport links are provided on the Trimetra system:

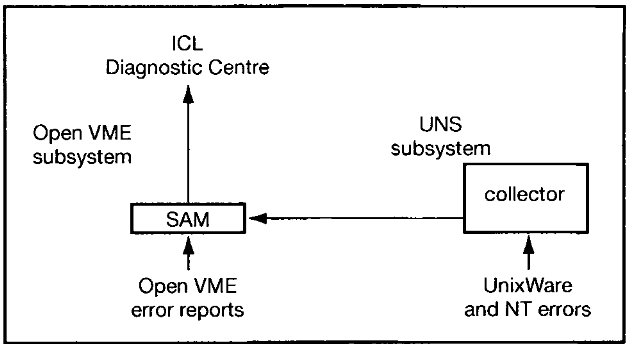

Figure 6: Support & Maintenance System

- In cases where the component of the system to be investigated is fully operational, access is provided via a LAN based SAM support link. The same access link is used for access to all components of a Trimetra system, and the ICL diagnostician can, if required, have simultaneous access to multiple components of the system (OpenVME, NT and UnixWare) via this single support link
- In cases where access via the ISDN link is not possible (for instance where the LAN problems are involved), access to the components of the system is possible via direct asynchronous connections which are available separately to all components of the system. Access to all the system components is again possible from the single support centre.

Establishment of SAM support sessions to the ICL support centre is initiated from the Trimetra system using software based on OpenVME , NT or UnixWare systems, depending on which component of the system is to be accessed.

6. Services on Trimetra

A single integrated set of services is provided by ICL covering all the components of the Trimetra system. Thus for example:

- A single service is provided covering the installation and configuration of the complete Trimetra system
- All the support and upgrade aspects of the system are carried out via a single user

interface provided by a single organization
- A complete set of professional services covering, for instance, filestore design, archiving etc. are provided via a single ICL interface covering all the components of the system.

In operational terms the various services provided on the system are, in fact, provided by a number of separate organizations both within ICL and, in the case of specialized components of the system, by non-ICL organizations. The provision of a single user view of this complete set of services is an extremely powerful factor in providing the single system image of the Trimetra system as a whole.

7. Software environments and system exploitation

The ability to run multiple operating system environments on a single Trimetra system provide a number of exploitation opportunities which would not be possible in systems supporting only a single operating system instance. Specific exploitation aspects of this type include:

- The ability to consolidate multiple discrete applications, which might otherwise have been hosted on separate systems, on to a single shared Trimetra configuration. This ability has become more significant in recent times with the increasing tendency towards re-centralization and consolidation of previously distributed servers
- The use of multiple servers to provide the opportunity to develop high concurrency services by replication of a single application service, for instance, a single Web query service, across a set of servers, thereby avoiding the limits imposed by running such software in a single server instance
- As previously mentioned, the availability of facilities for use with Trimetra to allow users to link applications, hosted on several platforms in the Trimetra system, to form a related set of integrated services.

The latter feature is of particular importance in the case of the OpenVME systems where users can develop new services on Unix and NT and link these to their existing operational OpenVME systems. Examples of such exploitations would include:

- The ability to carry out periodic extracts of data contained within OpenVME systems and to transfer the data into relational database systems contained on the Unix or NT systems, where it can be accessed, for instance, for management support purposes, using standard relational database tools
- Facilities to allow the VME services to be accessed through standard user interface environments, for instance, Web front ends, by development of interfacing tools based on the NT and Unix systems
- The ability to develop new services on the Unix and NT systems, for instance, new transactional applications, electronic commerce services or office automation services, linked in real time, or via batch procedures, to existing OpenVME services.

The ability to develop systems of this type is of considerable importance to the users of these systems since it allows them to retain their often considerable investment in OpenVME based applications, and to exploit these through the development of linked Unix and NT services, all contained within the single managed Trimetra system.

8. Engineering aspects of Trimetra

A final observation on the Trimetra system relates to the nature of the engineering processes involved in the production and delivery of the system. The development processes involved in the production of the system largely comprise the following:

- The design of the system

- The evaluation and selection of the component products, mostly externally sourced, to be used in the construction of the overall system
- The integration and validation of the system constructed from the starting components
- The packaging of the resulting system for delivery as a composite product.

The process involved in developing the Trimetra system is therefore increasingly one of systems design and integration, replacing the process of detailed proprietary hardware and software design and implementation which traditionally characterized the development of large mainframe systems.

In a similar way, the processes involved in the delivery of the system, and in the provision of the services associated with the system, represent increasingly an exercise in logistics and control, providing the customer with a product having the attributes of a single system, while operationally delivering the product through sets of related but coordinated organizations and activities.

The production process involved in the development of the Trimetra system therefore provides an interesting example of the increasing move away from detailed component design in the development of platform systems towards a situation where increasingly powerful and flexible systems can be constructed through the design and integration of larger, but often still relatively uncoordinated, component products.

Acknowledgements

Thanks are due to the various HPS personnel who contributed to the development of the Trimetra system and, in particular, to Derek Ashcroft who conceived and managed the implementation of the orginal system design.

Biographies

Christopher Martin

Chris Martin joined ICL in 1973 after graduating from the University of Leicester with a BSc in Physics. He spent his first 13 years working as a Systems Support Engineer on the medium to large 2900 Series Order Code Processors before joining Mainframe Systems at West Gorton in 1986. After joining Mainframe Systems Chris worked as an Integration team leader on Series 39 Level 80 Systems with particular emphasis on the multinode and partitioning capability of Series 39 Systems before taking over as Manager of Peripheral Integration. With the development of more Unix based systems from ICL HPS, Chris moved into the Unix parallel processor validation group and from there on to the UnixWare and NT subsystem project within Trimetra Systems where he is currently Project Manager for the UNS component of the Trimetra range of systems.

Charlie Stewart

Charlie Stewart currently works as a member of the Systems Architecture group within ICL High Performance Systems. He joined ICL in 1973 following completion of a doctorate degree course in Chemistry, working initially on compilers and subsequently on network system design. Over the last ten years he has been responsible for the development of strategies and architectures in a number of areas, primarily in the field of interworking, and the integration of disparate PC and server systems within large scale networks. His current areas of interest cover the ongoing development and exploitation of the Millennium system architecture, and its integration with Microsoft enterprise technologies.

Trimetra *Xtra*server

David Messham

ICL High Performance Systems, Manchester, UK

Abstract

The ICL Trimetra *Xtra*server is a high-powered server supporting both Windows NT and UnixWare. It has the required scalability, availability and manageability to make it suitable for enterprise-critical services. It is part of ICL's Millennium programme, which will deliver a common Intel-based platform which runs OpenVME as well as NT and UnixWare.

This paper describes how the required qualities were built into the system. *Xtra*server was constructed largely from bought-in components, but these have been configured and validated together to deliver an integrated system. Some development work was necessary, for a management tool for the hardware and a workload scheduler, and is described. The system is delivered with many services available to help the customer exploit it.

1. Introduction

A crucial objective of the Millennium programme is to provide a support environment for the widest possible choice of applications. This means that Millennium platforms offer a choice of operating systems—Windows NT and SCO UnixWare are available as well as VME. Each installation is likely to need several systems, possibly with mixed operating systems depending on the applications.

The initial systems that have been developed as part of the Millennium programme are an important step towards the overall objectives. The Trimetra *Xtra*server is one of these.

The *Xtra*server runs UnixWare and NT. It is directed at the customer who wants an "Enterprise quality" system and either doesn't need VME, or wants the highest performance and scalability from UNIX or NT.

The Millennium programme gives the customer good value for money by building the systems from standard components, thereby taking advantage of the continual improvements generated by the commodity market.

At the same time, the systems are aimed at the customer who is running the most critical or the largest workloads. The customer needs systems which give confidence that the required performance can be achieved, that the system will achieve the required levels of availability, and that the system is manageable. In particular, the system must be manageable in an environment where there are likely to be many Millennium systems and a variety of other systems. These are "Enterprise Qualities". The systems delivered have availability and manageability features already integrated into them and validated to work together.

The challenge for *Xtra*server was to deliver a system with these qualities and to integrate it from standard components. In so doing we had to pioneer this type of integration activity within ICL High Performance Systems (ICL HPS). The components integrated were larger than had previously been dealt with—e.g. disk systems rather than disks. This was a deliberate choice, in order to move the business to a position where ICL HPS could add greater value.

ICL HPS is the division of ICL that focuses on the corporate IT market, and has extensive success in delivering OpenVME systems with excellent scalability, security, manageability, availability and potential for upgrade and change. *Xtra*server had to establish a position in the market for ICL HPS as a vendor of NT and UNIX systems with similar Enterprise qualities. In particular, it had to enable ICL HPS to expand from the current OpenVME base into areas where there is no OpenVME presence.

This paper describes how the objectives of the system were met. It then describes the system in more detail and, finally, it discusses the ways in which value is added in a product that is constructed from large-scale bought-in components.

2. System objectives

2.1 Choice of application

The customer is likely to have chosen a particular application, having identified it as the best solution to a business problem. Offering the widest choice of applications on the platform is therefore essential.

Windows NT almost certainly has the widest choice of applications of any operating system. *Xtra*server offers NT 4 Enterprise Edition, which is the latest version with the most features.

For customers who want applications running on UNIX, SCO UnixWare offers the widest choice of applications of any UNIX platform running on non-proprietary hardware. The latest version, UnixWare 7, is available, as well as the previous version, UnixWare 2.1, for any applications not yet supported by their vendors on UnixWare 7.

By using standard operating systems and building the system out of standard components we can ensure that the system has the potential for change and upgrade. A new application is very likely to be available on the platform, and the current applications will run in a binary-compatible way on the next generation of the platform.

2.2 Performance and Scalability

The requirement is to deal with large workloads and to give the customer confidence that the system can grow to deal with larger ones.

Disk storage is available both in *Xtra*server Disk Arrays and in EMC Disk Systems as described in the section below on Hardware. In either case the maximum storage capacity is approximately 10 Tb (10,000 Gb), or 5 Tb using dual, resilient connections.

Processing power extends from 4 to 12 Pentium Pros, in steps of 2. More than 8 processors cannot be licensed for the standard version of NT, but ICL provides an OEM version licensed for 10 or 12 processors. The *Xtra*server is currently the largest Intel multiprocessor system available from any supplier for NT and UnixWare.

The system performance is particularly enhanced by disk "fast write" cache capabilities. The *Xtra*server Disk Array can be configured with two independent controllers, known as Storage Processors, each with its own cache. Disk writes can be completed as soon as the data has been written to the cache in both Storage Processors, since they are independently stored and powered and no single failure can cause data to be lost from both. This is much faster than waiting for the disks to rotate. It is particularly beneficial when writing database transaction logs, which is not improved by most forms of disk caching. EMC disk systems offer similar powerful cache options.

Not only is the current system expandable to significant levels, but it has an element of future-proofing because it is built from standard components. It is clear that the capacity and power of these components will continue to rise, so future systems running binary-compatible applications will be available with greater power and decreasing relative cost.

2.3 Manageability

A single system must be easy to manage. A more important and general objective is that the customer finds it easy to manage a Data Centre where there are several *Xtra*servers as well as several other systems. The product must not be just a platform that runs the customer's chosen application, but it must be a complete, manageable solution, that fits into an existing complex working environment.

We offer the customer a single point of management (or as many points as required) for several NT, UNIX or OpenVME platforms. This may be the System Management Workstation or a PC supplied with the system, or elsewhere.

Since the system is a standard platform, any management product available on that platform will run. However, an important part of the *Xtra*server system is the offer of management products that have been integrated in and tested with the system.

The system management products chosen are market-leading best-of-breed products that:

- Are at least available on UnixWare and NT, and on a variety of the other systems that are likely to be in the Data Centre
- Can be used from a management station of the customer's choice
- Are suitable for use from within the popular enterprise management frameworks such as CA Unicenter.

The functional areas for which products are supplied are hardware management, backup, performance monitoring, application management, print management, batch management, workload scheduling and support service. For details of the products selected and developed see the section below on Management software.

2.4 High Availability

*Xtra*server is intended for use in enterprise-critical applications, so high availability of the system is a key requirement. The system must prevent the main causes of unavailability—hardware failures, software failures, as well as planned non-productive time due to repair, backup and maintenance.

The effect of hardware failures is minimized by the provision of redundant components, so that, if one fails, the system is not prevented from providing the services. This applies to processors, SCSI adapters and cables, disk system components, disks (with hardware RAID), Ethernet adapters, fans and power supplies. For all failures except those occurring in the processors, the system continues to run without a break.

Disks and other components within the disk systems are "hot replace"—broken ones can be replaced while the system is running and the new one is automatically brought into use. The replacement of other components requires the system to be closed down.

The ultimate in resilience to failures is provided by the High Availability Cluster option, which has two processing systems with shared disk storage (see Figure 3). This provides redundancy for the whole processing system. Failures in either node are automatically detected, and a critical workload is "failed" over to the other node, according to rules pre-defined by the System Administrator. No operator intervention is needed at the time of failure. The operator can also instruct the system to "fail-over" services so that maintenance can be carried out on a node with the services still running.

The frequency of software failures is much reduced by a thorough validation programme. For the components we use, we work with our suppliers to ensure that they are fully stressed for the size and power of an *Xtra*server. Then we run validation on the combinations of products that constitute an *Xtra*server system.

The time needed for backup is another key parameter determining overall service availability. Legato Networker is integrated with the Oracle on-line backup tool on NT to minimize the time for which the service is unavailable. An important option with EMC disk systems is the building of an extra plex of the data, which can then be dropped to be backed up while the service continues to run. This is generally applicable to any software product. Copying the extra plex to tape can be managed by Legato.

EMC disk systems also offer the option of holding a plex of the data remotely on another EMC system. Such a system preserves a copy of the data even if the site on which the original processing is done is incapacitated, for example, by fire. With *Xtra*server processing capability available on the remote site, this provides a disaster-tolerant system.

3. System description

3.1 Hardware

The *Xtra*server Processing Module contains from 4 to 12 200 MHz Pentium Pro processors. Any even number is available. The minimum memory size is 512 Mb and the system is capable of supporting up to 8 Gb in 256 Mb or 512 Mb steps.

The *Xtra*server Disk Array contains between five and thirty disks. 4 Gb, 9 Gb and 18 Gb disks are available. The Disk Array supports hardware RAID 0, 0/1, 1 and 5, individual disks and hot spares.

In addition to offering disk storage in *Xtra*server Disk Arrays, *Xtra*servers are available connected to EMC disk systems (models 3300, 3430 and 3700). This makes all the enterprise features of the EMC system, such as remote mirrors for disaster tolerance, and the on-line creation of backup copies of data available to *Xtra*server customers.

The systems are delivered in two metre high cabinets. Each cabinet contains either one Processing module and (optionally) one Disk Array, or up to three Disk Arrays. The cabinets contain N+1 redundant power supplies and fans, and an optional Uninterruptible Power Supply (UPS).

Figure 1: Xtraserver with System Management Workstation

Figure 1 shows two *Xtra*server cabinets with a System Management Workstation standing on a desk to the left of them.

The left hand cabinet is a system cabinet with a processing system and space for a disk array above (not installed). At the bottom of the cabinet are the Uninterruptible Power Supplies.

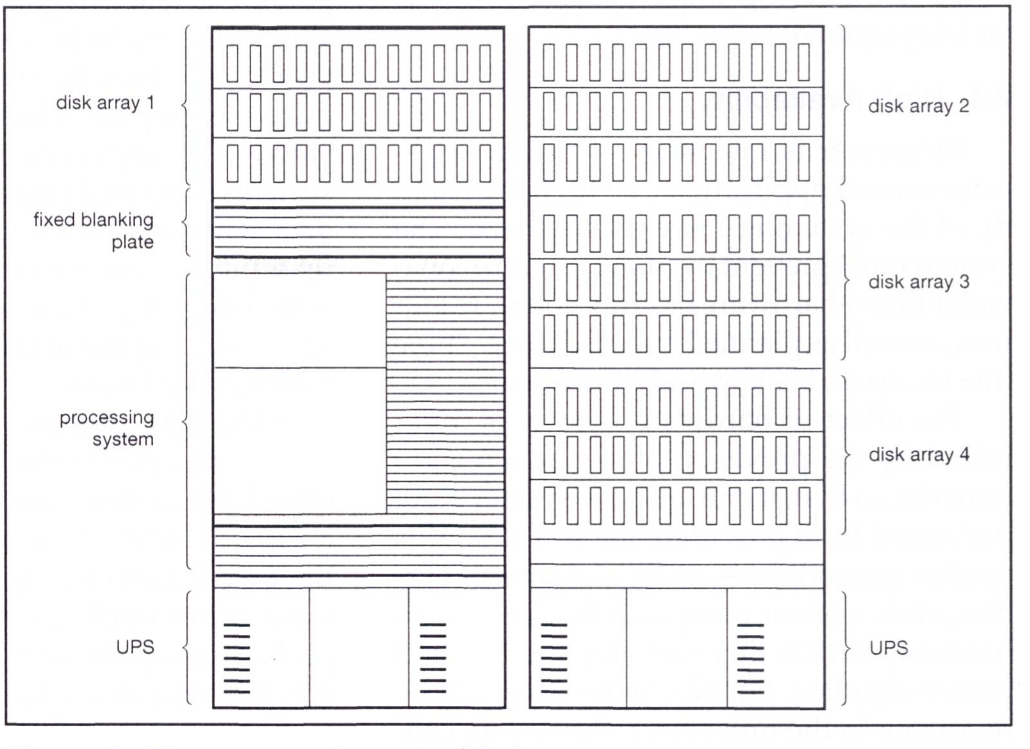

Figure 2: Xtraserver—Layout of Sub-

The right hand cabinet is a disk cabinet with space for up to three disk arrays, each with up to thirty disks. Only the bottom one is fitted. The same space is used at the bottom of the cabinet for the UPS.

The cabinets have similar doors at the rear, which are used for engineering access.

Figure 2 shows in diagrammatic form what a fully populated system would look like.

Fast/Wide Differential SCSI is used to connect the Processing Systems to the disk systems. Dual SCSI connections to the disk systems offer both increased performance and resilience to failure of any interconnect component.

Tape drives, for backup or data interchange, are also SCSI-connected. ICL's strategic alliance with StorageTek ensures that a variety of drive types and silo systems are available with the *Xtra*server system.

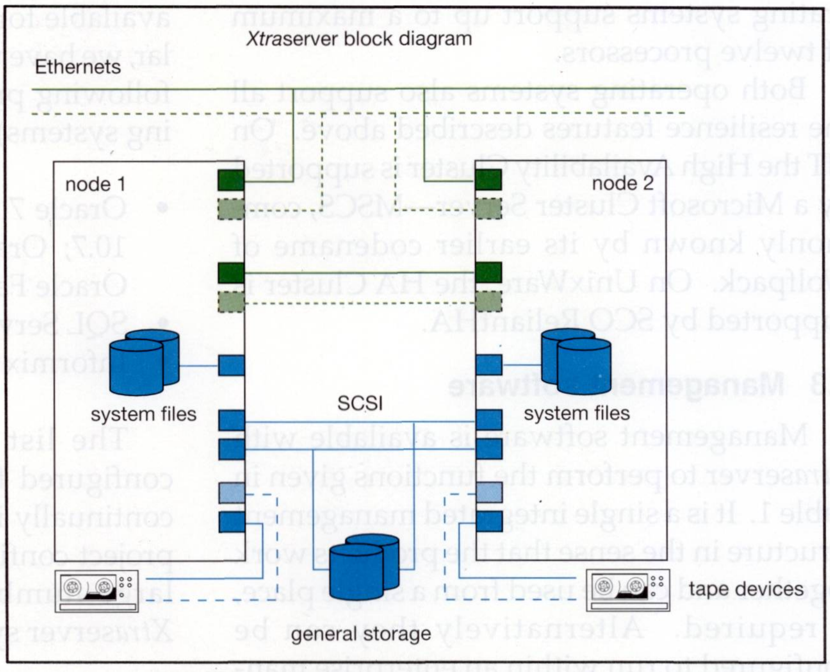

Figure 3: High-Availability Cluster

10 Mbit or 100 Mbit Ethernet is used to connect the *Xtra*server to the customer's network. More than one connection can be configured for increased performance and increased resilience to failures both on the *Xtra*server and in the network.

There is also an option to connect two Processor Modules and a set of Disk Arrays to form a High-Availability (HA) cluster. Figure 3 shows the configuration of a 2-node HA cluster. Ethernets are shown in green and SCSI in blue. Optional components are dashed.

The systems boot from the internal disk holding the system files. These files are mirrored so that the system is resilient to internal disk failure.

The application data is held on the external disk arrays marked general storage on the diagram. Each disk array can be visited by two SCSIs, each of which visits both nodes in a cluster system. This gives resilience to failures of the Host Bus Adapter, the SCSI or the disk controller in the disk array. More than two SCSIs may be used, to give up to 5 Tb of storage with resilient connections or 10 Tb without. EMC disk systems are supported as an option for external storage and they also offer resilient connections.

A variety of tape devices is offered, including attachment to large tape silos.

The pair of internal Ethernets is only used in a cluster for the "heartbeat" to ensure that a node can react to the failure of the other. The external Ethernets, at the top of the diagram, are for connection to the customer's LAN, which may be 10 Mbit or 100 Mbit. Two external Ethernets give resilience to network card failures or to a failure of part of the network.

Both Processor Modules are connected to the SCSIs that connect the Disk Arrays, so that either of them can take over the critical services if the other fails. The SCSIs are externally terminated so that they do not rely on the processing systems for termination. There are additional Ethernet connections between the nodes to carry heartbeat and other intra-cluster messages.

The final hardware component is the System Management Workstation, which is not shown in the diagram. This is a PC connected to the *Xtra*server by Ethernet and is used for the management of the hardware and other system management functions as the customer wishes.

3.2 System software

The operating system software is Windows NT4 Enterprise Edition or SCO UnixWare version 2.1 or version 7. Both operating systems support up to a maximum of twelve processors.

Both operating systems also support all the resilience features described above. On NT the High Availability Cluster is supported by a Microsoft Cluster Server—MSCS, commonly known by its earlier codename of Wolfpack. On UnixWare, the HA Cluster is supported by SCO ReliantHA.

3.3 Management software

Management software is available with *Xtra*server to perform the functions given in Table 1. It is a single integrated management structure in the sense that the products work together and can be used from a single place, if required. Alternatively they can be configured to run within an enterprise management framework such as CA Unicenter or Tivoli. They can be used to manage a single *Xtra*server, several *Xtra*servers, or several *Xtra*servers combined with several other systems where these are supported. The management products can be used to manage the *Xtra*server High Availability cluster systems even when applications are running on a node other than the one they started on.

System Management functional areas and products integrated with *Xtra*server	
Management of disks, processors, and UPS systems	ICL *Xtra*View
Backup	Legato Networker
Performance monitoring	Datametrics Viewpoint
Application Management	BMC Patrol
Print Management	SNI Xprint
Batch Management	Tivoli Maestro
Workload scheduling	ICL Workload Scheduler
Support service	ICL Teleservice and CA Remotely Possible

Table 1: Management Software

Customers may also use other management products where these are available for NT or UnixWare systems.

3.4 Applications

*Xtra*server offers industry-standard platforms, and therefore will run the large number of applications offered on these platforms. Many thousands of applications are available for NT and UnixWare. In particular, we have validated the system running the following products (not all on both operating systems):

- Oracle 7.3.3 and 7.3.4, with Oracle Apps 10.7; Oracle 8.0.4.2 with Oracle Apps 11; Oracle FailSafe for clusters.
- SQL Server 6.5.
- Informix ODS 7.23UC1.

The list of applications particularly configured for and tested on *Xtra*server is continually increasing. There is currently a project configuring MS Exchange to run for large numbers of users across multiple *Xtra*server systems.

4. Value added by ICL

As the *Xtra*server project was different in nature from the development projects previously run by ICL High Performance Systems, it is of interest to examine the contributions made in this way of working.

Value has been added to the delivered product in three broad areas: component selection and integration, product development, and service development.

4.1 Component selection

The selection and integration of the Disk Array will be used as an example.

The product was selected for both high data integrity and performance. The duplicated "fast write" cache described above is a particular feature that contributes to both these requirements.

Many other factors were considered in product selection, including price and discounting, support offered, Year 2000 conformance, and commonality of interests with other collaborators. A process was developed and adopted by the division to ensure that future projects take advantage of the experience gained.

4.2 Integration

The product was tested in the laboratory and, at this stage, we were able to identify the appropriate SCSI adapters to use. The same adapter is not used in all systems, depending on different functional and performance requirements. In one case, the performance of the adapter we chose was 25% greater than its nearest rival.

The management software for the Disk Array did not generate messages in a way that BMC Patrol could pick them up. BMC Patrol is a management product that monitors and controls applications and system components. It is extensible to new applications and components by writing a new "Knowledge Module". We persuaded the suppliers of the Disk Array to generate messages in such a way that they could be picked up by BMC Patrol and we wrote a special "Knowledge Module" to monitor the whole of the *Xtra*server hardware including the Disk Array.

The management software for the Disk Array was not designed to run in a High Availability cluster, where the disk systems are owned by more than one processing system. We configured cluster systems so that they run in only one node, and "fail-over" automatically if that node fails.

Not least, the whole system was validated. The arrangements of the Disk Arrays in the *Xtra*server cabinets were validated for power supply and cooling, and safety features were assured. System startup was validated, both under normal conditions and with hardware failures present in the system. Response to failures while the system was running under stress was checked.

System performance was measured. The results were excellent and justified the effort spent on choice and configuration of components. An unaudited AIM 7 run with UnixWare 2.1, 10 processors and 2 Gb memory gave a peak result of 4,509 jobs/minute. This was the highest performing Intel system by a significant margin.

4.3 Development

The ICL Workload Scheduler manages a system in which several applications can be run each with its own response-time and throughput requirements. This is important in any system running more than one application. It reduces Total Cost of Ownership and improves manageability.

This was one of the areas where it proved impossible to acquire a product with the required capability, since it uses the priority-based scheduling used on OpenVME. Based on this experience with an enterprise mainframe system, HPS then developed a new product.

The initial development was done on UnixWare in collaboration with SCO. An NT version of the product is currently running as a prototype.

The Workload Scheduler was integrated into the management system by developing a Patrol Knowledge Module that enables Scheduler activity to be monitored from the Patrol Management station.

The ICL *Xtra*View Management software provides a framework for the management products supplied for the various hardware components. It ensures that the customer has a coherent view of the hardware platform and is able to manage it as a whole.

At the top level it presents a GUI describing the status of the whole system. If there are problems, it shows in which part of the system the problem has occurred. The icon representing this part of the system can then be expanded to investigate at increasing levels of detail. At the lower levels it uses tools provided by the hardware supplier, but at the top level it presents an integrated set of information to the user.

4.4 Service Development

A crucial part of the total *Xtra*server product is the set of services available to assist customers with the *Xtra*server and to enhance the value of the system for them. Some forty such services have been identified, including establishment, migration to *Xtra*server, backup, database design, performance and capacity management.

Services are particularly important in cluster systems, where it is necessary to configure the clustering software to manage the applications under a variety of failure conditions.

Much of the expertise developed by ICL as part of the project is available to customers through services, e.g. there is a service showing how best to configure Legato Networker to produce timely backups, together with general consultancy on backup management. Again the issue is more complex in cluster systems, where the workload is not always running on the same node when it is backed up.

5. Conclusions

Trimetra *Xtra*server is an important system in the context of the Millennium programme. It offers maximum scalability and performance on NT and UnixWare systems, while providing the Manageability and High Availability necessary to run enterprise-critical applications.

*Xtra*server has allowed ICL High Performance Systems to gain experience of delivering systems by integrating large scale components and adding value to the integrated product and services.

This is a major step in demonstrating ICL's ability to deliver the Millennium programme. It gives us confidence that we shall be able to offer future systems of greater power and to integrate OpenVME onto these systems as well as NT and UnixWare.

Glossary

GUI	Graphical User Interface
HA	High Availability
ICL HPS	ICL High Performance Systems
MS	Microsoft (as in MS Exchange)
MSCS	Microsoft Cluster Server
OpenVME	ICL's proprietary Mainframe operating environment.
RAID	Redundant Array of Inexpensive (now Independent) Disks
RAID 0, RAID 1	Different methods of storing data - striping, mirroring, n+1 redundancy and combinations of these
SCSI	Small Computer System Interface - standard interconnect for disks
UPS	Uninterruptible Power Supply

Acknowledgments

Thanks to Steve Clarke, Mike Curran, John Hayley, Chris Hughes, Moh Khan, Victor Maller, Brian Procter and Chris Williamson who helped with suggestions for this paper. Most of all though, thanks to all who worked on the *Xtra*server project in whatever capacity, without whom there would be no product.

Trademarks

Biography

David Messham graduated from Cambridge University in 1966 with a BA in Engineering. After working for English Electric on computer control system research, he joined ICL in 1970. He worked on the S3 programming language used in VME and was Design Manager of the S3 compiler. He has subsequently worked on a variety of development projects related to the 2900 series and its successors. He had technical responsibility for the integration of the CAFS (Content Addressable FileStore) into VME, for which ICL were presented with the Queen's Award to Industry for technical achievement. He led the VME Kernel design team during the transition to the Series 39 architecture and subsequently for the introduction of Open TP protocols. He was Design Manager for the Goldrush project for commercial exploitation of parallel processing.

David's main interest in the *Xtra*server project has been the specification and integration of the cluster systems. He is a Member of the IEE and a Chartered Engineer.

Millennium Data Access

Richard Day

ICL High Performance Systems, Manchester, UK

Abstract

"Data access" has been identified as a key requirement in ICL's Millennium programme.

Enterprises wish to "access all data from anywhere"—specifically, to be able to use data associated with all the Millennium operating system personalities from any other personality and from other platforms such as PCs.

Enterprises have expressed a set of requirements in the arena of data access. Data access is categorized in terms of five technology areas, in each of which are analysed the relevance, trends, Millennium strategy, products and developments. These analyses are then related back to the requirements, covering the whole set.

The Millennium programme therefore addresses the needs of enterprises for data access in a very wide-ranging way.

Enterprises' existing applications and data are usually extremely valuable assets, and it is normally sensible to try and re-use and encapsulate them rather than let them be dissipated.

In order to reduce costs and improve time to market, enterprises often wish to purchase standard packages. The issue is then how to get these packages to interwork with existing applications and data.

The best way to update data is often through the code that was originally written to access it. This applies both to data associated with standard packages and to legacy data associated with legacy applications. Attempts to access such databases directly without using the associated code may lead to data inconsistency or corruption due to the failure to allow for the complete set of effects and side-effects which the original application code catered for. Therefore application interworking is a key enabler for achieving access to data.

1. Introduction

1.1 Scope of this paper

This paper describes the technologies used within the Millennium programme that enable enterprises to gain access to data, wherever it is held within a Millennium system, from applications running in many different environments. These environments include those running outside the Millennium system as well as inside.

1.2 Structure of this paper

Section 1.3 defines data access in customer terms and states its importance.

Section 2 gives a summary of business needs and trends in the data access area.

Section 3 categorizes data access in terms of technology areas. These areas are then analysed in turn in more depth in subsequent sections in terms of relevance, trends, our strategy, products and developments.

Section 4 describes application integration.

Section 5 describes data movement.

Section 6 describes direct data access.

Section 7 describes user access.

Section 8 describes application development.

Section 9 contains a summary.

This is followed by a glossary of abbreviations and terms used.

1.3 What is data access, and why is it important?

"Data access" means that set of technologies which address an enterprise's requirements to:

- "Access all data from anywhere"
- Re-use existing applications and data.

New business requirements

Enterprises want to provide new functionality in order to meet new business requirements: web access, electronic commerce, data warehousing, and so on. But these new applications need to access the existing mission-critical data, which contains information about the business—such as customers' names and addresses.

The legacy is increasing

The amount of existing mission-critical code and data in the world is increasing rather than decreasing (Figure 1): therefore the opportunity associated with allowing new code to access it is also increasing.

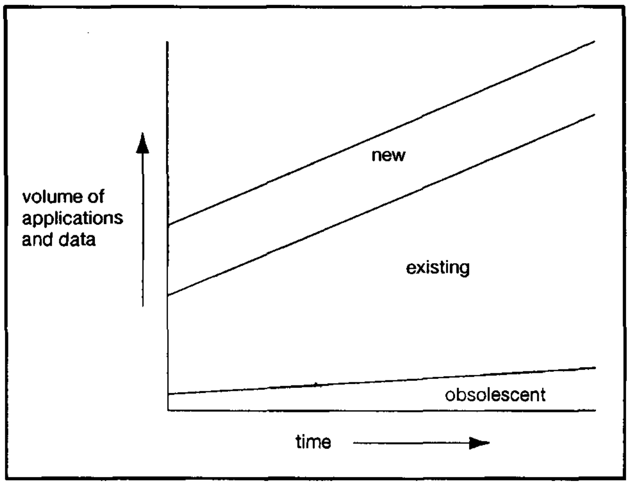

Figure 1: The legacy is increasing

Quantitative data relating to this graph includes the following:

- There are around 400 billion lines of code in use on mainframes
- Contrary to popular belief, total processing mainframe power is increasing. For example, it increased by 20% in 1997 [Gartner (McNee et al.), 1997]
- Mainframes are not the only repositories of legacy code. Considering just the high end of the market, tracked by S/390, Unix and "other" (SNI BS2000; Unisys A series, 2200 and Clearpath; ICL OpenVME and Groupe Bull G-COS 8), research shows that the 1996 market grew by 38% in terms of units [IDC, 1997]
- Today's new code and data is tomorrow's legacy! Total spend on software and services in Europe in 1996 was 61 billion dollars, and this is predicted to increase at a compound annual growth rate of 9.3 per cent over the years 1996 to 2000 [IDC, 1997].

2. Enterprise needs and trends

Enterprise needs

Enterprises wish to "access all data from anywhere".

In UK market research undertaken by ICL HPS for the Millennium strategy, they have expressed this desire in terms of being able to:

1. Have data sharable between line of business applications
2. Give access to data from web browsers
3. Have input and amendment once only
4. Facilitate integration between standard packages and existing applications
5. Move data from operational databases to data warehouses
6. Collect data from many sources
7. Support mobile users
8. Make data available for management information systems
9. Make data available within PC applications
10. Produce data access applications with improved development productivity
11. Reduce the volume of bulk movement of data

This list is a set of headings under which various statements from ICL's customers have been placed. It is not an orthogonal set of technical categories. These headings have been numbered so that they can be referred back to in later sections, showing how each customer requirement is addressed by a technical solution.

Enterprise trends

The trends surrounding data access are summarized below, and expanded in later sections:

- There will be increased use of application packages where competitive advantage is not an issue. (This trend is slowing as users realize that packages can be diffi-

cult to implement and may not meet their needs)
- Wholesale rewriting of applications will not occur. (Re-engineering for its own sake has been shown to be costly and not to produce business benefit, which will only occur where the application functionality no longer meets the business need)
- Businesses are demanding customer orientated systems
- Application interworking will increase as a result
- Transaction Processing will remain a key technology in enterprise scale computing
- The Internet will change the way users access applications, and is increasing demand for access to data from outside the organization
- Information/knowledge is required from the data held by organizations.

3. Categories of data access

Before categorizing data access, some terms need to be recalled relating to splitting an application. The Gartner Group, [Brenner, 1994] and [Day, 1994] has been followed in classifying the content of most applications under three technical headings: data management, application logic and presentation. Often the data management and presentation components are provided by using generic software (such as a database management system supplied by a vendor such as Oracle). The application logic is less generic—it may be bespoke software or it may be supplied as part of a packaged solution. This is illustrated in Figure 2.

If the application is to be split into two parts (one part on a client platform and the other on a server platform), the split can be made at either of the two boundaries between functions, or inside one of the three functions. Consequently there are five main ways of splitting a centralized or personal application into two parts between which there is a client-server relationship. This is the basis of the popular classification into five client-

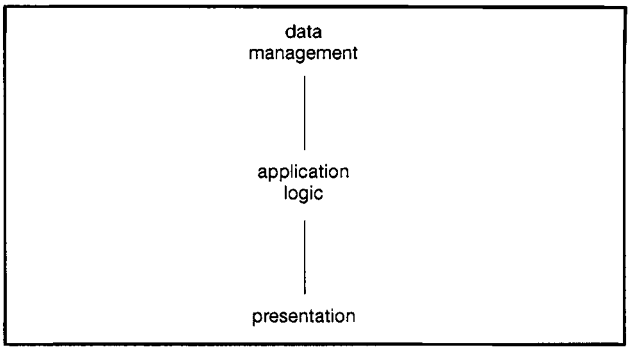

Figure 2: Application software modularity

server styles, which is promoted by the Gartner Group. It is illustrated in Figure 3.

Enterprises typically express their requirements in terms of getting access to data. However, the most appropriate technologies for them to use are not necessarily those that support "direct" access to existing data from new applications. In other words, remote and distributed data management styles should not necessarily be used.

Four categories are described associated with data access. These categories are named as application integration, data movement, direct data access and user access.

3.1 Application integration

Application integration means the technologies associated with enabling a number of on-line applications to interwork synchronously. In terms of Figure 3, this is the use of distributed function mechanisms. This is discussed further in section 4.

3.2 Data movement

Data movement is categorized in terms of *batch data movement, message queuing and data replication.*

Batch data movement refers to the applications which operate in batch mode in order to copy data from one database to another, and the technologies associated with constructing such applications.

Message queuing is the provision of a facility which allows applications to communicate asynchronously with other applications over a network by sending and receiving messages. Messages can contain data in any

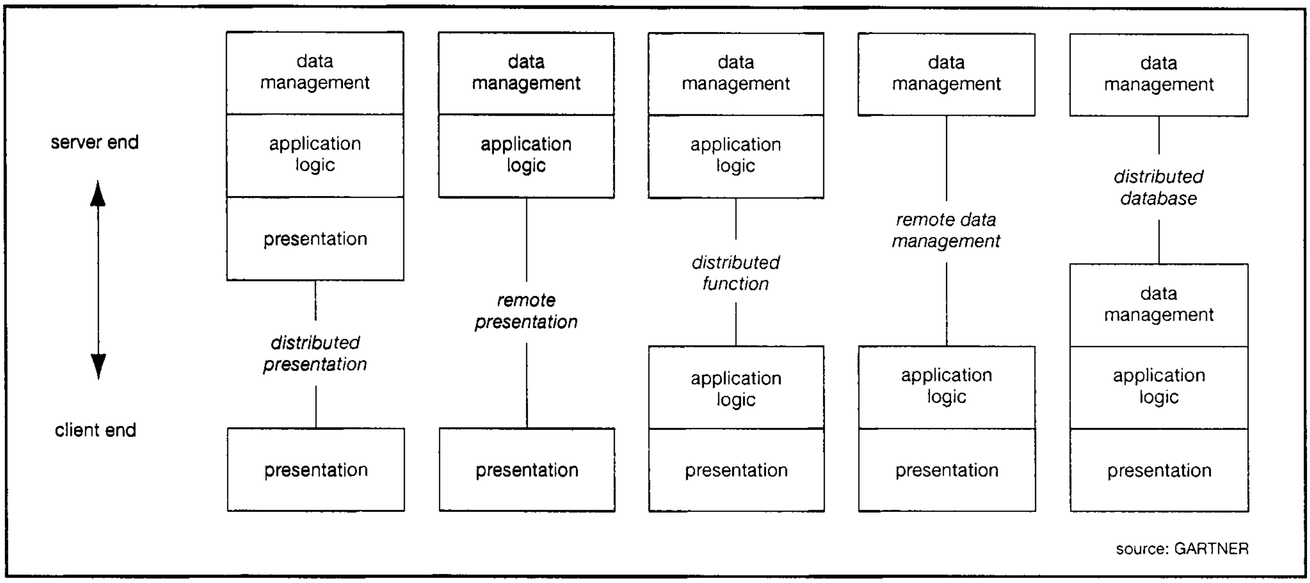

Figure 3: Five generic styles of basic client-server structure

format that makes sense to both the sender and the receiver. When an application receives a request message, it processes the request by reading the contents of the message and acting accordingly. If required, the receiving application can send a response message back to the original sender.

While in transit between senders and receivers, the message queuing system keeps messages in holding areas called *queues*—hence the name *message queuing*. The message queuing system protects messages from being lost in transit and provides a place for receivers to look for new messages when they are ready. Most importantly, applications can use the message queuing system to send messages and continue processing, regardless of whether the receiving application is running or reachable over the network. The receiver may be unreachable because of a problem, or naturally disconnected, as is the case with the applications used by remote or mobile users. Whenever the network becomes available or the receiving application is ready to process requests, the message queuing system will deliver any waiting messages—with the *reliability* required by mission-critical applications.

Data replication is a process that automatically copies information from one database to one or more additional databases.

Data replication often uses message queuing technology. The use of a data replication product typically significantly reduces the amount of application code that needs to be written compared with the direct use of message queuing by an application. However, with data replication there needs to be a structural similarity between the source and the target data sets. See section 5.

3.3 Direct data access

Direct data access means the ability of an application on one system to access data on another system without the intervention of additional application-specific code. See Section 6.

3.4 User access

User access refers to the set of technologies applicable to allowing a variety of types of client systems to gain access to services which access data. See section 7.

3.5 Application development

Application development is the process of constructing and maintaining business solutions. This process may include:

- Acquiring packaged solutions
- Developing bespoke software.

Most organizations have a combination of packaged solutions and bespoke software, and there is a spectrum of organizations between those who have wholly packaged solutions and those who have wholly bespoke software. See section 8.

4. Application integration

4.1 Why application interworking?

The best way to update data is usually through the code that was originally written to access it. This applies both to data associated with standard packages and to legacy data associated with legacy applications. Attempts to access such databases directly without using the associated code may lead to data inconsistency or corruption due to the failure to allow for the complete set of effects and side-effects which the original application code catered for.

In addition, it may be inefficient in performance terms for an application on one system to access a database management system on another. This is particularly the case where either of the following applies:

- Access is over a Wide Area Network
- Database access is at a low level - in other words it is specified in terms of large numbers of accesses each of which returns small amounts of data, such as is the case with IDMSX.

Therefore application interworking is a key enabler for achieving access to data.

Benefits of synchronous working

These are as follows:

- The representation of business model entities (for example, the customer) is consistent throughout the system
- The results of a business process are immediately available throughout the enterprise
- The results of a query are available immediately
- A connected user can conduct a richer dialogue with the enterprise databases and correct a greater quantity of input information immediately.

Many systems require to process data in real time, with input from a number of end users. To take one example, a customer can phone British Gas, and talk to someone who can immediately alter the customer's account details by interacting with a TP system via a PC GUI.

With the advent of mobile systems with users who are not always connected to enterprise systems, architectures need to be put in place that support such usage.

However, in most circumstances, there would be a severe loss of benefits if all users were downgraded to the level of service available to disconnected users.

4.2 Technology trends

Right distribution: two-tier and multi-tier architectures.

Two-tier solutions come in two forms:

- Either the application resides on a server system and has to drive the GUI across to the PCs or terminals
- Or the application resides on PCs and drives SQL across the network.

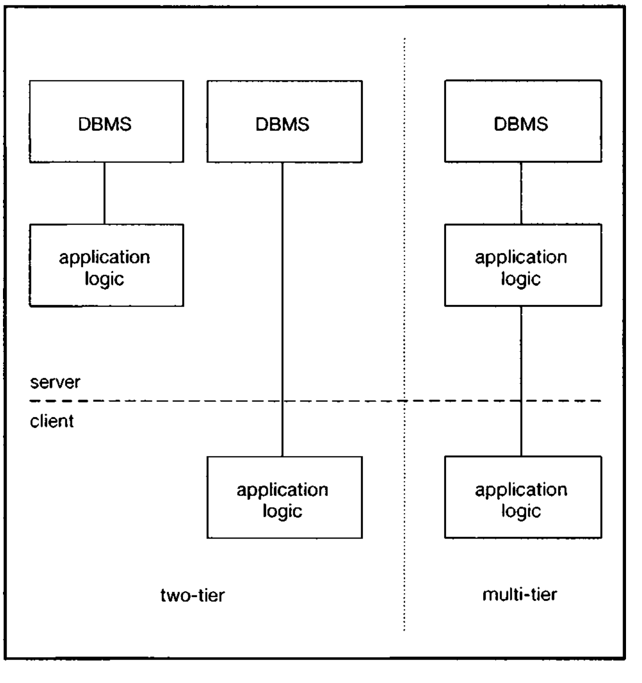

Figure 4: Two-tier and multi-tier architectures

These solutions suffer both from lack of scalability and from difficulty in integration with existing systems.

Where application complexity is low, and the application is expected to have a short life, then a two-tier solution is applicable. The cost of developing and maintaining the solution will be low. Bought-in two-tier packages may be used, as well as application development toolsets that generate two-tier solutions.

But where application complexity is higher, and application life is longer, then three-tier solutions will win on the cost of development and maintenance.

Multi-tier architectures refer to those architectures which place application logic both on the server platform and on client platforms (Figure 4). There is now a massive move to multi-tier (three or more tiers) architectures, object technology and transaction management.

What has caused major companies, including even Microsoft, to make the shift to multi-tier architectures? There are five major influences:

- Desire to reduce the costs of managing client platforms
- Pervasiveness of Web technologies
- Multiple client types
- Spread of object technologies
- Scalability considerations (see Figure 5).

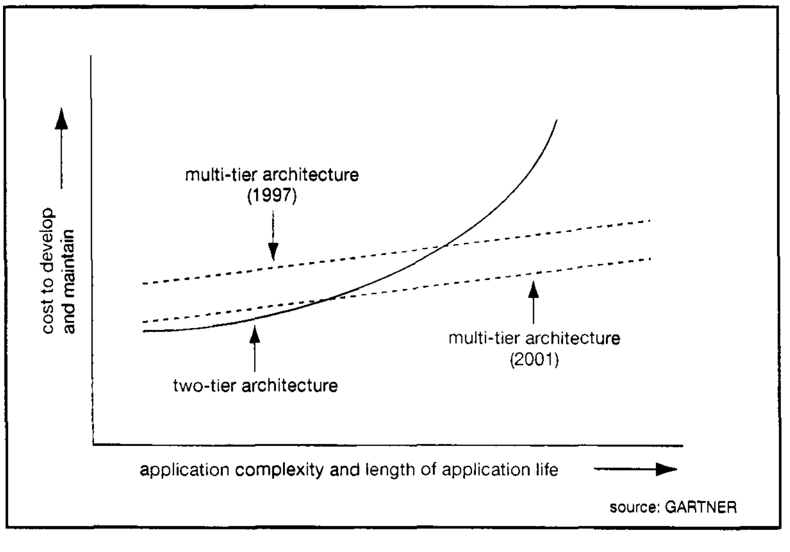

Figure 5: Two-tier versus multi-tier trade-offs

Desire to reduce the costs of managing client platforms

The average cost to an organization of maintaining each PC has been estimated as $10,000 per year. This may even be an underestimate, since it is hard to quantify the hidden costs where staff attempt to manage their own PCs. In order to address the cost issue, PCs need to be managed from a central server. This management includes the ability to distribute client applications to PCs from the server.

This is achieved through the use of web technologies, and these are multi-tier (see below).

Pervasiveness of Web technologies

Web browser technology is the most effective way of:

- Reducing management costs by standardizing the desktop and making it easy to distribute application functionality to the desktop
- Allowing access to existing systems from a wider class of users; for example access by the general public to banking or holiday reservation systems.

Web applications are almost entirely multi-tier. Even with the arrival of client "applets", most of the data access code and business logic will remain on the server. To the extent that browser applications supplant traditional client-server GUIs, the number of two-tier applications will drop correspondingly.

Multiple client types

As well as web browsers there are a number of other front ends that cannot run traditional two-tier client-server applications. These front ends include inter-enterprise messaging, hand-held appliances and some point-of-sale devices and voice response units.

Whereas before, applications had only one type of client access, now there is often a requirement for many. This includes the ability to access applications over Intranets and the Internet, from mobile computing platforms and from "traditional" GUIs. It is far more effective to support multiple client types by placing the common business logic on the server, rather than having different versions of it on the various types of client platforms. In other words, a multi-tier architecture is required.

Spread of object technologies

Object technology provides better productivity in application development and improved application quality and maintainability. It achieves this through natural modelling and reusability. It allows applications to be componentized, and is therefore a key enabler in implementing multi-tier architectures.

Scalability considerations

According to Schulte [Gartner, (Schulte) 1997], a multi-tier architecture should be used if any of the following conditions apply:

- Systems with more than twenty application programs
- Two or more heterogeneous data sources
- Expected application life greater than three years
- Many application modifications or additions anticipated
- More than ten thousand transactions per day
- Significant inter-application communication, such as inter-enterprise communication or access to a mainframe application
- The application may grow over time so that one of the previous conditions will apply.

For most enterprises, at least one of these conditions will apply!

Two-tier computing pushed into a niche

The implication of all this is that two-tier architectures will be pushed into a niche. By 2001, two-tier architectures will only be appropriate for applications which run entirely within a LAN, use only one DBMS, have no need for a browser user interface, and always and only use a desktop PC front-end.

As evidence for adoption, consider the following observations:

- *Industry analysts are stating that a large move to multi-tier applications is happening now:*
  - In 2001, 90 percent of all new client-server applications will be multi-tier [Gartner, (Altman and Reents) 1997]
- *Package vendors are supporting multi-tier solutions:*
  - Peoplesoft to use Tuxedo
  - SCT Banner to rewrite to integrate with X/Open TP
  - Baan and SAP moved to multi-tier architectures years ago
  - Geac and Hyperion are moving to multi-tier
- *Environment suppliers are supporting multi-tier solutions:*
  - Microsoft with Microsoft Transaction Server
  - Oracle with Network Computing Architecture
  - ORB vendors, IBM, BEA with Tuxedo
- Application development tools vendors are supporting multi-tier solutions:
  - Microsoft, Oracle, Nat Systems, etc.
- Customers (large and small) are putting multi-tier solutions in place:
  - British Gas, European Gas Turbines, Companies House, Hyder, Britannia Building Society, etc..

Ninety per cent of all new client-server applications will be multi-tier in 2001.

There is a major architecture and product battle taking place.

The battle: DCOM/MTS versus Java/CORBA

Microsoft promotes DCOM and MTS, and the "anti-Microsoft alliance" (Sun, IBM, Netscape, Oracle, etc.) promotes Java and CORBA. The best prediction is that neither will win outright. The Java/CORBA scene is currently fragmented and one vendor would be expected to win here—perhaps Oracle with NCA.

The common view

However, both sides accept the following propositions:

- Applications should be designed in terms of distributed objects [Day, 1998b], in order to achieve better productivity and maintainability
- There is an architectural shift in new on-line processing architectures. This is the shift to three-tiered architectures, as outlined above. In 90% of organizations the requirements are such that a three-tier architecture is mandated
- This architecture is enabled today by transaction management systems (and in the future by object transaction managers). This is a forced choice where high levels of reliability, availability and throughput are required and, in any event, provides the simplest way to develop server applications.

Both transaction managers and object environments are evolving to become object transaction managers. See Day for a definition of this term [Day, 1998c].

Which products will win?

Which middleware products will win in the arena of enterprise on-line processing environments?

According to Gartner the following environments will cover 90 per cent of all new enterprise client-server development after 2001—and more than half of new midsize and large OLTP applications will use MTS.

- Tuxedo from BEA Systems
- Microsoft's Transaction Server (MTS)
- IBM's Component Broker
- Oracle's Network Computing Architecture products (NCA) which supports CORBA.

All these environments are evolving into object transaction managers.

4.3 Millennium Strategy

Enterprises will increasingly add new functionality by means of applications that will run on NT or UNIX platforms. They will want these to interwork with existing VME applications so as to utilise their large investment in VME.

ICL High Performance Systems has defined middleware architectures that will meet these requirements, such that enterprises will be able to implement the architectures using leading enterprise environments provided on the three platforms. The product sets will be validated to work together on Millennium and will be supported by documentation and services.

This will make it attractive for enterprises to implement their new applications using Millennium components; that is, enterprises that use Millennium will obtain benefits such as a validated product set and single supplier interfaces.

The strategy is to focus on a small number of leading enterprise environments. The chosen environments are MTS and Tuxedo, with access to CORBA environments also provided through a software gateway.

This supports enterprise needs numbers 1 to 4 inclusive (section 2), see Figure 6.

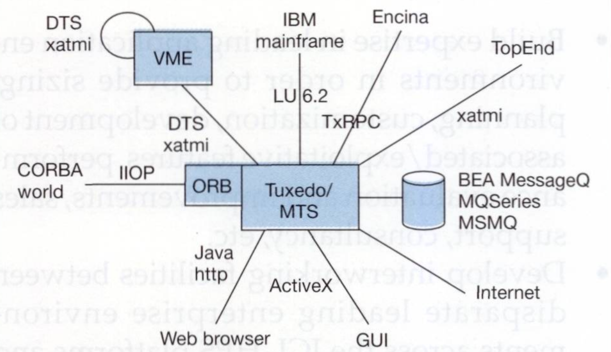

Figure 6: Integrating applications

The combination of MTS and Tuxedo provides the interfaces needed in order that applications on Millennium component platforms (VME, NT and UNIX) will interwork with applications running on other vendors' environments (including IBM mainframe environments). Tuxedo and MTS run on

UNIX and NT, and the number of interfaces needed from VME is small and to a large extent available already.

Through expertise in architectures and product solutions, ICL HPS is able to deliver generic solutions. The added value in these generic solutions is derived from the customers' trust in ICL HPS's ability to deliver pre-validated interworking solutions, associated consultancy and services and better scalability through correct solution choice.

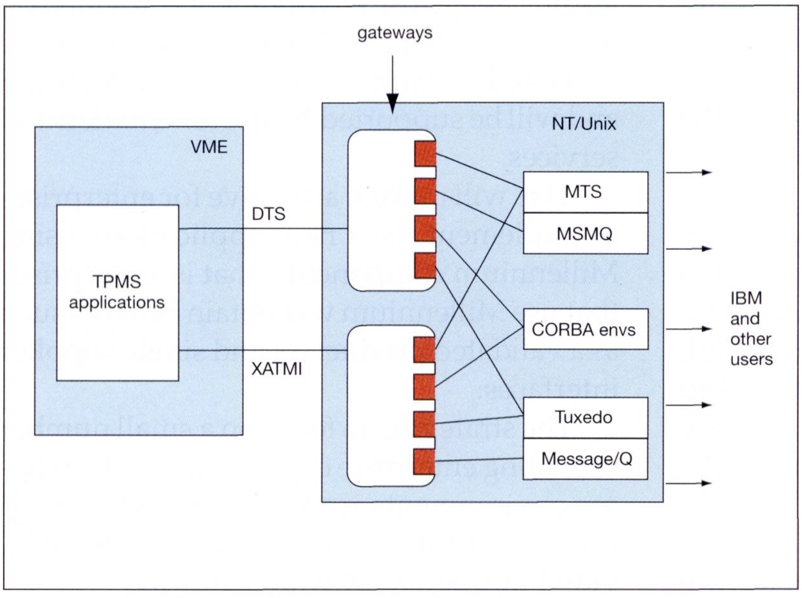

Figure 7: Fundamental application interworking

ICL HPS will develop the capability to:

- Build expertise in leading application environments in order to provide sizing, planning, customization, development of associated/exploitative features, performance evaluation and improvements, sales support, consultancy, etc.
- Develop interworking facilities between disparate leading enterprise environments across the ICL HPS platforms and specifically to ensure VME applications can interwork with applications on other platforms
- Support Internet access to applications (see section 7).

4.4 Products

These are focussed around facilitating interworking between the leading enterprise environments and include VME integration:

- MTS/TPMS interworking (DTS, XATMI)
- Messaging/TPMS interworking (MSMQ, MessageQ)
- TPMS/CORBA interworking.

It is a fundamental principle that VME interworking will be built using gateways on Unix and NT, rather than by porting code to or writing new code for VME (see Figure 7). This has the following benefits:

- Reduction in "treadmill" costs of re-porting and re-writing to support new standards
- Minimization of the number of VME interworking components for which performance (DTS, XATMI) has to be optimized
- Minimization of additional CPU cycles on VME, leading to minimization of performance degradation of existing customer systems on VME.

5. Data movement

5.1 Why data movement?

It is not always possible or desirable to have tightly coupled systems.

Loosely coupled and strongly coupled systems

In a strongly coupled system all the databases can be regarded as one logical database and these databases must be kept up-to-date against each other in real-time. Many of the benefits of strongly coupled systems are obvious:

- The results of an update can be immediately available across the whole of the enterprise

- The results of a query are available to the user immediately
- Information in the system about customer details, for example, can be consistent across the whole system at any point in time
- A connected user can have a rich dialogue with the enterprise databases and, as a result, correct input information immediately.

There are circumstances where loosely coupled systems are preferable to strongly coupled systems. These include:

- Where the performance overheads of keeping all the databases in the system consistent are too high in relation to the benefits that would be obtained
- Where some parts of the system need to continue even if other parts have failed. In a strongly coupled system, if any of the platforms, communications and applications in the system are not operating then no part of the total system is allowed to proceed
- Where some components of the system are not always connected, for example, mobile users [Day, 1998d].

Both strongly and loosely coupled systems are used by ICL HPS's customers, and many customers use both types of system in combination.

5.2 Trends

Data movement is therefore relevant for moving data from operational systems to data warehouses; supporting mobile working and moving data between corporate and branch office databases.

Data warehousing

Data warehousing growth is guaranteed. Organizations are now willing to spend significant sums of money to get competitive advantage from making more data available for analysis by more users. They are expecting to increase their average number of warehouses and marts to nearly six by 1999 and expand the typical size from 132 GB to 259 GB [Forrester, 1997].

Seventy-five per cent of the task of implementing a data warehouse is in data extraction from operational systems, according to the Gartner Group.

Mobile working

Mobile sales executives need to connect to the data centre to confirm transactions.

There are various examples of mobile working, for example, hand held meters for reading gas and electricity meters. A large motoring organization currently downloads and uploads information about jobs, but there is no automatic link to membership—so if you join at a motorway service station you are still not covered for the remainder of your journey. Their vision is of intelligent engine management systems diagnosing the fault using the onboard satellite navigation system to locate the car and the mobile phone to contact the motoring organization.

Building Societies have tried connecting laptops by digital phone. This will become practical in the not too distant future.

The growth in middleware support, and improvements and price reductions in portable systems, communications adapters, modems and cellular services are leading to an increasing use of mobile and remote working.

5.3 Strategy

There is therefore a need to support data movement between:

- Operational databases and data warehouses [Day, 1998f]
- Mobile systems and corporate databases
- Corporate databases and branch databases.

This supports enterprise needs numbers 5 to 7 inclusive (see section 2).

Additionally, needs numbers 8 to 10 must be supported. The architecture for supporting both these sets is illustrated in Figure 8.

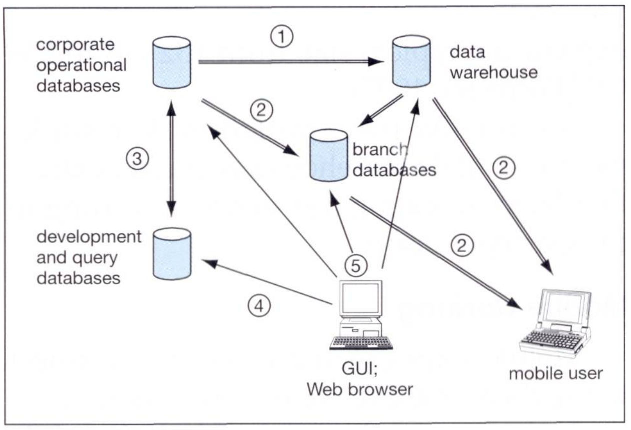

Figure 8: Data movement and access

Key to Figure 8:

(1) Data movement into a data warehouse (enterprise needs numbers 5 and 6)

(2) Using replication or message queuing for moving data between corporate databases, branch databases and mobile systems (enterprise need number 7)

(3), (4) Improved development productivity (enterprise need number 10; see section 8)

(5) Data made available within PC applications for MIS purposes (enterprise needs numbers 8 and 9; see section 6).

There is no single technology that supports all the attributes; therefore a number of technologies are required [Day, 1998e]. These include:

- Batch data movement and associated tools
- Message queuing
- Data replication.

Batch data movement

Applications can be produced which run in batch mode and extract data from source data-stores and load it into the target database.

There are tools that generate such applications. These tools differ in the extent to which they can cater for transformations, conversions, cleansing, scrubbing and so on. ETI-EXTRACT from Evolutionary Technologies Inc. is a best-of-breed tool in this category. It supports data collection, conversion and migration from virtually any platform, operating system and database management system to any other environment.

ETI-Extract is available for VME for which it generates COBOL and SCL and can already be used to extract data from IDMSX. Because it is not always possible to identify which records in the database have been updated since the last extract, bulk re-copying of possibly the whole of the database may need to take place. Sometimes the amount of time required is too long even to fit into a weekend.

The key development is enabling the extraction of updates from IDMSX logs. This reduces the amount of time needed for the extract and, therefore, allows more frequent extraction, thereby keeping the target database more up-to-date. In addition, because the extraction process works off the logs rather than the database, the operational system may be running at the same time as the extraction process.

Message Queuing

This technology is used where there is a requirement to keep the target database more up-to-date against the source databases than would be feasible using batch processing and where the data transformations involved are too extensive for replication technology to cater for it.

The key message queuing products are:

- Microsoft Message Queue (MSMQ)
- BEA's MessageQ.

The key development is enabling messages to be sent from VME (TPMS) applications into such systems, using common technology.

Reducing the volume of bulk movement of data

Enterprise need number 11 is addressed here.

The strategy is to provide a *software pipe* facility for Millennium systems, which will support the interchange of data between co-operating applications on different operating system personalities within Millennium. It allows applications on one personality to access serial character-encoded files on another.

An example of the effectiveness of this solution follows. Typically, when data is to be moved from an operational database to a data warehouse the steps are:

1. Extract the data from the source database into a serial file on the same platform
2. Transfer the file into a serial file on the target platform
3. Update the target database from the serial file.

The software pipe solution improves the performance and manageability of this process by:

- Eliminating the writing and reading of one of the files, and the associated processing
- Removing the explicit file transfer process
- Improving the performance of the transfer process.

The associated products are:

- *Millennium IntraSystem Communications*, which provides the management of the communications service and the API for VME, NT and UNIX
- *Remote File Access to NT and UNIX files* from applications running on VME
- *Remote File Access to VME files* from applications running on NT and UNIX
- *File transfer utility*: A new file transfer command and a File Transfer Remote Application for VME, NT and UNIX is provided.

Data replication

Many situations occur in the daily operation of an organization that involve the need to have the same information in more than one location. This situation naturally occurs in large organizations with many geographic locations, but it is also common in smaller organizations with branch offices. A growing use of data replication, as a component for building a data warehouse, is appropriate even at a single location. Data is replicated from an on-line transaction processing system to a data warehouse for analysis.

The key products are Sybase Replicator, Praxis's OmniReplicator and Sybase's SQL Anywhere. The latter is particularly useful in the context of mobile computing.

The key development is to enable replication of updates from IDMSX logs, in conjunction with one of these products, selected following a feasibility study.

6. Direct data access

6.1 Why direct data access?

A common enterprise requirement is for ad hoc access from PC tools into corporate databases. This access is typically required for management information (MIS) purposes.

Examples of access types include:

- Obtaining reports
- Using spreadsheets
- Ad hoc queries.

6.2 Technology trends

The trend is towards using SQL to mediate between the PC tools and the data sets. Most PC tools that are relevant in this context support SQL (e.g. spreadsheets, report writers, word processors, query tools). Clearly, relational databases support SQL, and the technology is now available which supports SQL access to many other types of database.

This technology is appropriate for low volume ad hoc read access. For high throughput or update access or pre-defined access, application integration technologies are usually more appropriate.

6.3 Strategy

The strategy is to support SQL access to all data sets on Millennium, including read access to IDMSX.

6.4 Products

Information Builders' EDA/SQL product supports SQL between PC tools and most databases. ICL has provided the necessary components to allow EDA/SQL to also access IDMSX.

7. User access

7.1 Definition

This refers to the way in which people access systems. There are a number of classes of user access including:

- On-line users
- Mobile users, who are intermittently connected
- Usage through other enterprises' systems.

The end users' access may be through:

- Graphical user interfaces from PCs
- Web browser interfaces from PCs and network computers
- Character based user interfaces or block mode terminals.

7.2 Technology trends

The most significant trend is that towards allowing user access through web browser interfaces; for the reasons outlined in section 4.2.

There are continuing trends to:

- Improve the productivity of users by replacing character based user interfaces and block mode terminals by graphical user interfaces
- Putting new and improved user interfaces on to legacy applications
- Putting in place systems such that input and amendment are required once only to a set of applications
- Enable services, including services provided by legacy applications, to be accessed by new classes of user and client types.

7.3 Strategy

There are a number of existing mechanisms that address some of the above trends. These include Cochise, Dialogue Manager and SCO's Tarantella.

The existing Cochise product [Beale, 1997] enables web access to a VME application. Its ability to integrate access to a number of applications is limited.

The existing Dialogue Manager product [Thompson and Robertson, 1994] enables a variety of applications on various server platforms to be integrated and accessed through new user interfaces. However, it has limited capabilities in the area of provision of web browser access.

Tarantella allows access to existing applications from web browsers, but it provides neither the ability to integrate the applications, nor to modify the user interface.

The chosen architectural solution addresses these issues. It provides for integrated access to numbers of applications on disparate servers, through a variety of client types, and includes access from web browsers. New functionality may be integrated. The architecture is multi-tier, for the reasons given in section 4.

Solutions conforming to the architecture may be put in place using any of the key application environments (CORBA systems, Tuxedo or MTS). The diagram below shows an MTS based solution with the ICL HPS-provided product IPR.

Figure 9 shows a scalable solution, which supports (through the integration of "new Cochise/DM" with MTS):

- Improved user interfaces to legacy applications
- Input and amendment once only for a number of applications, including both legacy and new applications
- Different classes of user
- A variety of client types, including web browser interfaces.

This diagram does not show how the needs of intermittently connected users are met—for which see section 5, above and Day [Day, 1998d].

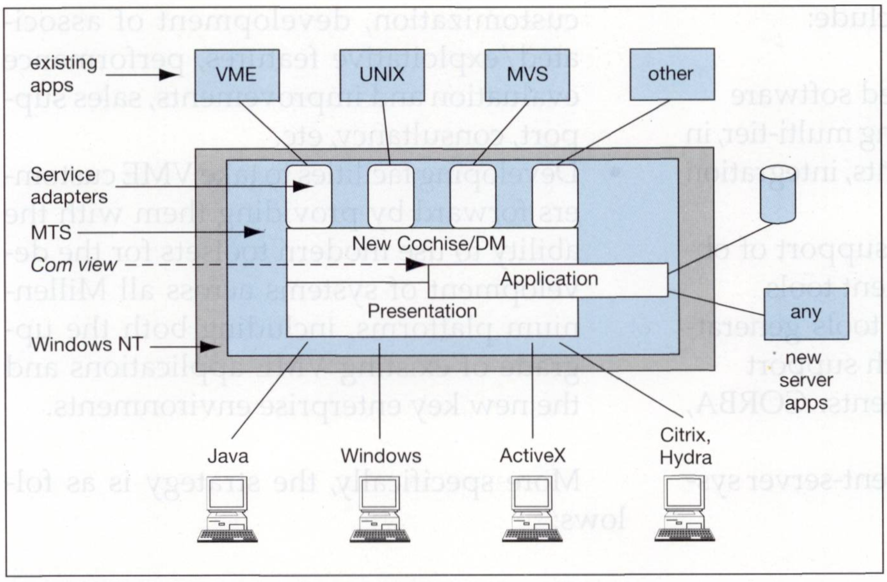

Figure 9: User access strategy

The strategy is to develop solutions using Microsoft technology, specifically NT and MTS. Enterprises will be able to construct similar solutions using Tuxedo, but fewer generic components will be available—there will therefore be more bespoke middleware in such solutions.

Web and electronic commerce

Access to systems from web browsers, including access for the purpose of electronic commerce, uses the above architecture with the addition of web servers. In the case of the Microsoft environment, their web servers are well integrated into their architecture (with MTS, etc.). In the other cases (Tuxedo and CORBA-based systems) this integration in not yet so well advanced.

8. Application development

8.1 Significance

When enterprises require new or changed applications, their concerns include:

- Containment of risk
- The desire to improve time-to-market, reduce development costs and improve development productivity
- The need to integrate new functionality (packaged or bespoke software) with existing applications and data.

Application development is important to ICL HPS because customers will tend to choose systems and solutions where they are persuaded that their concerns have been addressed. The ability of organizations to use modern development toolsets to construct VME applications helps to extend the life of VME systems. Toolsets that generate code that can run in either VME or NT/Unix help customers to construct applications that can be deployed across multiple Millennium operating systems.

There are many different aspects of applications development. For example, there are a number of different categories of application development tools, and different customers will choose different tools depending on the nature of their organization. These aspects are discussed by Day [Day, 1998g].

8.2 Trends

Perhaps the most significant trend is that both the suppliers of packaged applications and the vendors of development tools are moving to supplying components. A component is a piece of reusable functionality with well-defined interfaces. Thus packaged applications are becoming much less monolithic, resulting in an increased ability to use packaged components in conjunction with legacy applications and other new applications. The vendors of packaged applications are now supplying development tools in order to help customers tailor these packages. Correspondingly, vendors of development tools are moving towards providing generic business-oriented functionality in the form of components, and to providing increased ability to integrate packaged components from many and various suppliers.

Other important trends include:

- Continuing use of packaged software
- Packaged software becoming multi-tier, in order to support components, integration and web access
- Object languages and the support of object methods by development tools
- Application development tools generating for environments which support
- Object oriented environments: CORBA, DCOM/ActiveX
- Transactions, scalability, client-server systems
- Web integration

8.3 Strategy

Enterprises want to use best-of-breed toolsets. In addition, our VME customers want improved development tools to maintain and modify VME applications, in order to reduce development spending and improve time-to-market.

VME applications are being modified to support Year 2000 and EMU, and to allow them to interwork more easily with new applications. Providing improved development and maintenance tools for VME applications extends the life of VME systems.

These enterprises want to use the same toolsets for developing both VME and non-VME applications, because this leads to reduced development staff costs. Customers would also like to be able to define their applications in a way that is independent of the target environment.

The strategy is, therefore, to ensure that a coherent set of best-of-breed enterprise toolsets in each application development category (CASE, 4GL, 3GL, etc.) are available to ICL HPS customers. These toolsets will support all the platform environments (NT, Unix and VME) and generate code that will run in the key environments described in application environments (MTS, CORBA/NCA and X/Open TP (TPMS and Tuxedo)).

This is achieved by:

- Building expertise in these key toolsets, in order to provide sizing, planning, customization, development of associated/exploitative features, performance evaluation and improvements, sales support, consultancy, etc.
- Developing facilities to take VME customers forward by providing them with the ability to use modern toolsets for the development of systems across all Millennium platforms, including both the upgrade of existing VME applications and the new key enterprise environments.

More specifically, the strategy is as follows:

- Targeting the key environments with best-of-breed toolsets for NT, Unix and VME, developments are focussed in the first instance around Nat Systems' Natstar (in the CASE and 4GL arena) and Micro Focus COBOL (in the 3GL arena)
- A forward route to object technology is provided through Natstar
- The application development process is allowed to be carried out on the PC, initially by
  - Natstar and Micro Focus COBOL tools, which target VME, Unix and NT
  - XITEC technology in order to improve the maintenance and development process for Application Master and COBOL for VME
- A forward route is provided for existing design data into the new world, through mechanisms which allow design data to be exchanged between DDS and tools' repositories
- Expertise is built in the above best-of-breed tools and in the best-of-breed tools for Unix and NT from Oracle and Microsoft.

9. Summary

In UK market research, many enterprises rated data access as a key requirement, and expressed a set of needs in this arena. Consequently, data access has been identified as a key requirement for ICL's Millennium programme. The overall requirement can be stated as the ability to "access all data from everywhere" and to reuse existing applications and data.

In this paper data access has been defined in customer terms and its importance stated. Data access was then categorized in terms of five technology areas: application integration, data movement, direct data access, user access and application development. In each of these areas the relevance, trends, Millennium strategy, products and developments were analysed and these analyses were related back to the requirements.

Data access solutions need to satisfy various quality attributes (such as appropriate levels of scalability). This requires fundamental decisions to be taken regarding which architecture, technologies and products should be used to satisfy these requirements.

Appropriate middleware architectures have been defined which will meet enterprises' requirements, such that enterprises will be able to implement the architectures using leading enterprise environments provided on the three platforms (NT, UNIX and VME), together with "adapter" middleware products that join the environments together. These joins allow legacy applications to be reused and data to be accessed wherever it resides. The product sets will be validated to work together on Millennium and will be supported by documentation and services.

That the Millennium programme does satisfy enterprises' data access requirements has been shown. It does this through the provision of ICL and third party products, generic building blocks that form the infrastructure of an enterprise's total solution, and architectural guidance.

Glossary of terms used

3GL	Third Generation programming Language, for example COBOL or C
4GL	Fourth Generation Language. These are very high-level languages which are specialized for particular kinds of problems
ActiveX	A set of technologies, produced by Microsoft, which enables software components to interact with one another in a networked environment. ActiveX(tm) is built on COM
API	An Application Programming Interface allowing applications to access services
Application Master	A 4GL which runs on VME systems
Availability	A measure of whether the information system is available to its users when required
CASE	Computer Aided Software Engineering—supporting a design method, and computerising the application development life cycle from design diagrams through to code generation
COM	An architecture, defined by Microsoft, for cross-platform development of client/server applications based on object-oriented technology
CORBA	Common Object Request Broker Architecture: an architecture for run-time support of object-oriented distributed computing
DBMS	Database Management System
DCOM	An object protocol, specified by Microsoft, that enables ActiveX(tm) components to communicate directly with each other across a network
DDS	Data Dictionary System: a repository for holding and managing development-time information, which runs on VME
DTS	Distributed Transaction (Processing) System: a TP communications protocol supported by TPMS
GUI	Graphical User Interface
HPS	High Performance Systems Division
Java	Java is a programming language and a run-time environment that allows code to be "written once and deployed everywhere"; i.e. it is operating system independent. Java applications reside on centralized servers. The network delivers the applications to the client on request
LAN	Local Area Network

Middleware	Client-server middleware is the infrastructure that provides support to client-server application software, and insulates it from platforms
MIS usage	A type of usage which gives managers, and others, access to information in ways that are not pre-determined
MSMQ	Microsoft Message Queue Server is Microsoft's product offering supporting message queuing technology
MTS	Microsoft Transaction Server: a product that combines the features of a TP monitor and an object request broker
NCA	Network Computing Architecture is a cross-platform environment, defined by Oracle, for developing and deploying networked applications
ORB	Object Request Broker
Platform	A collection of hardware and software components with the ability to process and store information. The hardware includes processors, store, peripherals and network controllers; the software includes operating systems
TCP/IP	Transmission Control Protocol/Internet Protocol; part of the Internet Protocol Suite
TP	Transaction Processing. A transaction is a unit of work which is atomic, starts and finishes with the system in a consistent state, is isolated from other transactions and has a durable effect once it has been committed
Tuxedo	A leading TP monitor available on many platforms
XATMI	X/Open Application Transaction Manager Interface: an API defined by X/Open for transaction based applications
X/Open TP	A set of interface standards which many transaction monitors (including Tuxedo and VME OpenTP) support.

Bibliography

BEALE, A., "Cochise: a World Wide Web interface to TPMS applications," ICL Systems Journal, Volume 11, Issue 2, January 1997

BRENNER, J.B., "Client-server architecture," Ingenuity, The ICL Technical Journal, Volume 9 Issue 1, May 1994

DAY, R.P., "How ICL Corporate Systems support Client-server: an Architectural Overview," Ingenuity, The ICL Technical Journal, Volume 9 Issue 1, May 1994

DAY (a), R.P., "Data access," ICL HPS, April 1998p, http://www.hps.icl.co.uk/sa/techstrat/rpd-aevo.htm

DAY (b), R.P., "What is object technology?" ICL HPS, April 1998, http://www.hps.icl.co.uk/sa/pu/wiao.htm

DAY (c), R.P., "Types of middleware," ICL HPS, April 1998, http://www.hps.icl.co.uk/sa/techstrat/rpd-middef.htm

DAY (d), R.P., "Mobile and remote working," ICL HPS, May 1998, http://www.hps.icl.co.uk/sa/techstrat/rpd-mob.htm

DAY (e), R.P., "Data Warehousing topics," ICL HPS, April 1998, http://www.hps.icl.co.uk/sa/dw/dwmove.htm

DAY (f), R.P., "Data Warehousing," ICL HPS, April 1998, http://www.hps.icl.co.uk/sa/techstrat/rpd-dw.htm

DAY (g), R.P., "Application Development Overview" ICL HPS, April 1998, http://www.hps.icl.co.uk/sa/adover.htm

FORRESTER Research Inc. (Schadler, T., Dolberg, S., Boehm, E. W., Massey, C., and Buss, T.), "Data Warehouse Strategies", The Forrester Report, Software Strategies, Volume Eight, Number Six, September 1997, http://www.analysis.icl.co.uk/forrester/software/1997/reports/sep97swr.htm

GARTNER Group Inc., (Schulte, R.), "Architecture and Planning For Modern Application Styles," System Software Architectures, Strategic Analysis Report, R-ARCH-104, April 1997, http://www.analysis.icl.co.uk/gartner/ssa/00037280.htm

GARTNER Group Inc., (McNee, B., Keller, E., Stewart, B., Tunick, D., and Andren E.), "The Year 2000 Challenge: Opportunity or Liability?" Industry Trends & Directions, Strategic Analysis Report, R-Y2K-115, 29 July 1997, http://www.analysis.icl.co.uk/gartner/itd/00042078.htm

GARTNER Group Inc., (Altman, R. and Reents, S.), "System Software Architectures Scenario," System Software Architectures, Strategic Analysis Report, R-SSA-106, October 1997, http://www.analysis.icl.co.uk

/gartner/ssa/00067620.htm

IDC (International Data Corporation), (Doyle, A., and Cranna, M.), "Western European Information Technology Spending Patterns 1995-2001," IDC Report #A01D, October 1997, http://www.analysis.icl.co.uk/idc/A01D-03.htm

THOMPSON, R. and ROBERTSON, I., "Dialogue Manager: Integrating disparate services in client-server environments," Ingenuity, The ICL Technical Journal, Volume 9, Issue 1, May 1994.

Biography

Richard Day joined ICL in 1969 with an honours degree in Mathematics from Warwick University. He is a Chartered Engineer.

After early work in design, implementation and testing of System 4 COBOL components, he went on to lead the design and implementation of a PL/1 code generator. Involvement in 2900 COBOL compilation systems resulted in his becoming chief designer and then project manager for that development, which was released as VME (C2) COBOL.

He then had management and design responsibility running a strategy and design unit responsible for VME application development and databases, including language compilers, testing tools, a programmers' workbench and VME INGRES.

Subsequently he was appointed a Systems Architect, with overall responsibility for Corporate Systems' Superstructure (application development and database) and client-server technical strategy and architecture.

He is currently a Systems Architect in ICL High Performance Systems Division, responsible for defining the architecture and specifying the product developments and collaborations for above-platform software within ICL's Trimetra and Millennium range. Additionally he delivers architectural advice to customers in these areas and in the areas of migration and interworking of applications and data.

Previous Issues

To order back issues

Contact

Josie Abbey
Group Technical Directorate
ICL, Westfields, West Avenue, Kidsgrove, Stoke-on-Trent, ST7 1TL
Telephone +44 (0)1782 794815
Email: Josie.Abbey@tscexch.x400.icl.co.uk

or

The Editor, V.A.J. Maller
Telephone +44 (0)1438 833514
Email: V.A.J.Maller@lboro.ac.uk

ICL Systems Journal

Guidance for Authors

1. Content

The ICL Systems Journal has an international circulation. It publishes papers of a high standard that are related to ICL's business and is aimed at the general technical community and in particular at ICL's users, customers and staff. The Journal is intended for readers who have an interest in computing and its applications in general but who may not be informed on the topic covered by a particular paper. To be acceptable, papers on more specialised aspects of design or application must include some suitable introductory material or reference.

The Journal will not usually reprint papers already published but this does not necessarily exclude papers presented at conferences. It is not necessary for the material to be entirely new or original. Papers will not reveal material relating to unannounced products of any of the ICL Group of companies.

Letters to the Editor and book reviews may also be published.

2. Authors

Within the framework defined in paragraph 1, the Editor will be happy to consider a paper by any author or group of authors, whether or not employed by a company in the ICL Group. All papers will be judged on their merit, irrespective of origin.

3. Length

There is no fixed upper or lower limit, but a useful working range is 4,000-6,000 words; it may be difficult to accommodate a long paper in a particular issue. Authors should always keep brevity in mind but should not sacrifice necessary fullness of explanation.

4. Abstract

All papers should have an Abstract of approximately 200 words, suitable for the various abstracting journals to use without alteration.

5. Presentation

5.1 Printed (typed) copy

A typed copy of the manuscript, single sided on A4 paper with the pages numbered in sequence, should be sent to the Editor. Particular care should be taken to ensure that mathematical symbols and expressions, and any special characters such as Greek letters, are clear. Any detailed mathematical treatment should be put in an Appendix so that only essential results need be referred to in the text.

5.2 Electronic version

Authors are requested to submit either a magnetic disk version of their copy in addition to the manuscript or an e-mail attached file or both. The format of the file should conform to the standards of any of the widely used word processing packages or be a simple text file.

5.3 Diagrams

Line diagrams will, if necessary, be redrawn and professionally lettered for publication, so it is essential that they are clear. Axes of graphs should be labelled with the relevant variables and, where this is desirable, marked off with their values. All diagrams should be numbered for reference in the text and the text marked with the reference and an appropriate caption to show where each should be placed. Authors should check that all diagrams are actually referred to in the text and that copies of all diagrams referred to are supplied. If authors wish to submit drawings in an electronic form, then they should be separated from the main text and be in the form of EPS files. If an author wishes to use colour, then it is very helpful that a professional drawing package be used, such as Adobe Illustrator.

5.4 Tables

As with diagrams, these should all have captions and reference numbers. If they are to be provided in electronic form, then either a standard spreadsheet (Excel) should be used or the data supplied as a file of comma/tab separated variables. A printed version should also be supplied, showing all row and column headings, as well as the relevant units for all the quantities tabulated.

5.5 References

Authors are asked to use the Author/Date system, in which the author(s) and the date of the publication are given in the text, and all the references are listed in alphabetical order of author at the end. e.g. in the text: "...further details are given in [Henderson, 1986]" with the corresponding entry in the reference list:

HENDERSON, P., "Functional Programming Formal Specification and Rapid Prototyping," IEEE Trans. on Software Engineering SE 12, 2, 241-250, 1986.

Where there are more than two authors it is usual to give the text reference as "[X et al ...]".

Authors should check that all text references are listed; references to works not quoted in the text should be listed under a heading such as Bibliography or Further reading.

5.6 Style

A note is available from the Editor summarising the main points of style—punctuation, spelling, use of initials and acronyms etc. preferred for Journal papers.

6. Referees

The Editor may refer papers to independent referees for comment. If the referee recommends revisions to the draft, the author will be asked to make those revisions. Referees are anonymous. Minor editorial corrections, to conform to the Journal's general style for spelling, punctuation or notation, will be made by the Editor.

7. Proofs, Offprints

Printed proofs are sent to authors for correction before publication. The Editor will, however, always be prepared to send electronic versions to authors, either in PDF or as output files of the production system used for the JournalÊPageMaker, Illustrator and Photoshop.

8. Copyright

Copyright of papers published in the ICL Systems Journal rests with ICL unless specifically agreed otherwise before publication. Publications may be reproduced with the Editor's permission, which will normally be granted, and with due acknowledgement.

ICL Systems Journal Autumn 1998

ICL Group Technical Directorate
Westfields
West Avenue
Kidsgrove
Stoke-on-Trent ST7 1TL