A Survey of Distributed Computing, Computational Grid, Meta-computing and Network Information Tools

R.J. Allan and M. Ashworth

Computational Science and Engineering Department,
CCLRC Daresbury Laboratory,
Daresbury, Warrington WA4 4AD, UK

Abstract

A software environment of unprecedented quality and functionality is emerging in which coupled computing resources are accessed via client-server and Web-based tools. This development is being driven by a combination of the computer industry, which is rapidly developing software for e-commerce and leisure use, and the loose collection of world-wide "freeware" programmers. Geoffrey Fox has referred to it as the "Distributed Commodity Computing and Information System".

In this survey we examine a number of tools and projects for science and engineering applications on wide-area network based systems. This includes distributed computing, computational steering and meta-computing techniques.

We have also included a few "collaborative working" and "distance education" projects which share a number of the same goals and difficulties.

Keywords

distributed computing, computational steering, meta-computing, network solvers, collaborative working, distance education, programming tools, networks of workstations, cluster computing, e-Services.

This is a Technical Report of the UKHEC Collaboration.

Report available from http://www.ukhec.ac.uk/publications/reports/survey.pdf

Table of Contents

1 Introduction

Computational scientists are always hungry for computer resources. Use of single very large computers, even if they are of Distributed Memory Multi-computer (DMM) architecture, is limited — more often by cost than by technology. There are however many more modest systems with a total power far outweighing the few supercomputers available. It is this fact which has sparked an interest in harnessing distributed resources to create “computational power grids”. Such grids can be used for throughput (accessing free resources), high performance or data access.

At the departmental level grids are built from workstations or PC commodity clusters that would otherwise be infrequently used but can be harnessed to obviate the need to purchase mid-range servers. At the national level they may be built by collaborating computer centres or university research groups. Activities at the international level involve government laboratories and large national centres.

To realise the performance offered by a grid-based computing environment, a program must be *ubiquitous*, adaptive, robust and scalable. Ubiquity means that there must be binary images of the program able to run on whatever resource is available. An agent-based system may be used to evoke these images or other service components. This is required because the grid is a federation — the owners of the individual resources maintain ultimate authority over their use. As such, the resource pool may change without notice as resources are added, removed, replaced or upgraded. If a component is not compatible with all potentially available software infrastructures, operating systems and hardware architectures it will not be able to draw some of the “power” that the grid can provide. Adaptivity is required to ensure performance. If the resource pool changes or the performance of the resources fluctuates due to contention (e.g. from other users), the component must be able to choose the most profitable combination of resources that are available at a given time. Similarly, if resources become unavailable due to reclamation, excessive load or failure (network or server), a component serving the users' requirements must still be able to make progress. Scalability, in a grid setting, allows a distributed task to use resources efficiently. The greater the degree to which this can be dispersed, the more flexible the grid system has to be in order to meet its performance needs.

Distributed software tools, and especially those which facilitate very complex coupled applications to be constructed and used are likely to be of growing interest over the coming few years. They are however difficult to implement in an efficient federal manner, and it is more likely that data management or throughput services will prevail in the short term. We have carried out a survey of the current research and tools that are being constructed in these areas. We do not however attempt to provide an introduction to all the underlying distributed-computing techniques that are both complex and diverse. There are numerous discussions in the computer-science literature which should be consulted for background information (see e.g. Orfali and Harkey (1997), Hwang and Xu (1998)).

Distributed computing systems offer more than just a large CPU resource. A software environment of unprecedented quality and functionality is emerging along with the use of the Internet for E-commerce and leisure purposes. This development is being driven by a combination of the computer industry and the loose collection of world-wide "freeware" programmers. Fox has referred to this as the "Distributed Commodity Computing and Information System".

In the USA and Japan there are several alliances of computing centres separated by large distances. In Europe, Germany has taken a lead because of the regional computing centres. In the UK the JREI-funded centres may be (but are not yet) a source of similar resources.

Some of the technology gaps that have become focus areas for meta-computing and related research are:

? Execution environments that are portable and scalable;

- resource management;
- data storage and movement;
- security and authorisation.

? Tools that enable the use of the execution environment:

- automated tools for porting legacy code;
- collaborative problem solving environments for complex scientific and engineering tasks to extend the capacity of teams;
- formal, portable programming paradigms, languages and tools that express parallelism and support software synthesis and re-use.

? Develop execution environments to support applications of the future:

- structures of application software that can make use of up to 10,000 processors;
- methods for coupling multiple disciplines for analysis and optimisation and coupling disciplines to optimisation;

? Design and architecture to integrate execution environment, user environment and applications.

Software implementations for metacomputing are often described in software layers. Typical of this is the Integrated Grid Architecture proposed by the Grid Forum. Its four components are:

1. Grid Fabric — the lowest level with primitive mechanisms to provide support for high-speed network I/O, differential services, instrumentation, etc.
2. Grid Services — the typical middleware level with a suite of grid-aware services implementing authentication, authorisation, resource location, event services, etc.
3. Application Toolkit — provides more specialised services for diverse application classes, e.g. data-intensive, visualisation, distributed computing, collaborations, problem solving environments (PSE);
4. Grid-aware Applications — implemented in terms of grid services and application toolkit components.

The packages which are relevant to meta-computing applications include: Legion, GLOBUS, AppLeS, CORBA, Nimrod, NetSolve, Synthetix, Chorus, InfoSpheres, Amoeba, MILAN, Arjuna, Apertos, GrassHopper, WAVE, Locust, HPVM, HPC++, CC++, MIST, GA, Fortran-M, HPF, Java, Raja/RMI, Jini, ANDF, DQS, NQS, LSF, Condor, NQE, LoadLeveler and Cumulvs. We survey only a sample of them in this report.

We do not describe in detail the software that contributes to the layers of middleware. Examples of group communication systems, used for security purposes include the Transis Research Project at the Hebrew University, Jerusalem and the Rampart project at AT&T Research, USA. In addition to these, components must be provided to monitor availability and performance of services on the network.

Examples include NetPerf (Jones, 1999), GlobPerf (Lee *et al*, 1999) and the Network Weather Service (NWS) (Wolski *et al*, 1998).

The simplest tools provide an “object broker” mechanism to access computational resources, for instance a numerical library on a remote platform. More sophisticated systems provide fault-tolerance and checkpointing for meta-computing experiments (Fox and Furmanski, 1997). An example of software of this type is NetSolve. We do not discuss the broker software here, but the computer industry has emerging standards such as CORBA, Java JWORB, COM etc. and many references can be found via the Web. Two “middle layer” tools which are briefly described are GLOBUS (NCSA Alliance in the US) and Legion (US NPACI Collaboration). These form the basis of many other tools and grid demonstrations such as GUSTO.

Building systems that alter program behaviour during execution, based on meta-computing techniques and user-specified criteria (computational steering), has also recently become a research topic of particular interest. It may be thought that a discussion of computational steering does not fit well into a survey of distributed computing tools. Nevertheless the techniques are related in requiring methods to alter running processes, either to move them for the purpose of load balancing or to alter their execution path.

Steering and meta-computing tools require powerful visualisation facilities to run in a distributed computing environment with a distributed infrastructure (run-time system). Building such an infrastructure requires devising strategies for co-ordinating execution across machines (concurrency control mechanisms), mechanisms for fast data transfer between machines and mechanisms for user manipulation of remote execution.

A project at the Georgia Institute of Technology has developed the Falcon run-time monitoring system and the Progress and Magellan computation steering tools which use it to develop and control large-scale applications.

SCIRun (pronounced “ski-run”) is a simulation-steering tool designed for shared-memory multiprocessors and now ported to a distributed environment at the University of Utah, Salt Lake City with funding from NCSA.

Other projects, such as NetSolve, are part of the NetLib activity of Dongarra *et al*. http://www.netlib.org/.

Finally, meta-computing infrastructures may be, and are being, applied for data management enabling large data sets stored and indexed at remote sites to be analysed and re-used in inter-disciplinary projects.

Foremost in data grid developments are the particle physics, astrophysics and climate modelling communities. The first of these are stimulated by the imminent appearance of very large quantities of data from the CERN Large Hadron Collider (LHC). A large number of countries will participate in analysis of the data, with interacting grids organised as follows:

Tier 0 :	CERN, Geneva where the ATLAS, CMS and other experiments will be run on LHC;
Tier 1 :	independent national centres in the USA and Europe;
Tier 2 :	a number of regional centres in each country, probably deployed at universities or national laboratories;
Tier 3 :	computing resources of an individual university group;
Tier 4 :	an individual workstation.

This structure is typical of any grid organisation.

User requirements in data management are discussed in a separate report (Kleese, 1999).

1.1 Criteria for Inclusion

Packages covered here are those which can be run on either true parallel machines, workstation and PC clusters or on shared-memory systems. Very often machines of several types will be connected across a wide-area network (perhaps via ATM). We have in the main restricted our attention to packages which are intended for use with Fortran 77, Fortran 90, C and C++ applications, since these are the most widely used languages in scientific research. However, Java is playing an increasingly large role in this area, especially for the construction of network-enabled tools.

The main criterion for inclusion is that a package should be of use in a scientific or engineering application. Some of the entries cover packages which are already in existence and available. However, many packages are under construction or proposed software projects, and are included if they are thought to be of sufficient interest.

1.2 Individual Entries

The different fields in the entry for each package should be self-explanatory. Names and addresses given are simply somebody who can be contacted about the package; they are not meant to represent the entire cast responsible for the software. For full lists of the organisations and people involved the actual documentation (or Web page) should be consulted.

1.3 Other Sources of Information

As well as the individual Web sites and references which are listed in the individual entries there are collections of grid- and meta-computing links maintained by Rajkumar Buyya, Monash University, Australia and Mark Baker, University of Portsmouth, UK. These lists are available on the Web at URLs http://www.gridcomputing.com/, http://www.dcs.port.ac.uk/~mab/Computing-FrameWork/list.html and http://computer.org/channels/ds/gc.

Many projects are listed, including: ARCADE; BAYANIHAN; BOND; COVISE, PACX-MPI and G7; Charlotte; CONDOR; DISCWorld; DOCT; EROPPA; FAFNER; GLOBUS; HARNESS; HPVM; Hector; I-WAY; IceT; InfoSpheres; JET; JavaDSM; JavaNOW; JAVELIN; LEGION; MOL; MPL for meta-computing; NEOS; NETSOLVE;
Nile; NIMROD; NINF; NSCP; NWS; NYU; PARDIS; SNIPE; Symera; WAMM; WebFlow; WebSubmit and UNICORE.

For information on the GLOBUS project and its components there is a very comprehensive book by Ian Foster and Carl Kesselman (1998). This provides the most complete discussion of grid-based computing in general.

1.4 Intended Audience and Feedback

This survey is particularly geared towards users of the UK national academic computing facilities and for this reason also has a slight "UK slant", although most of the work has been done in the USA. However, the information contained here should also be useful to a wider audience.

It is our intention to keep this report as up-to-date as possible. To this end, we would be very keen to hear about any packages that are of interest in parallel scientific computing and are not currently included. Corrections and comments are also welcomed.

The authors can be contacted by email at r.j.allan@dl.ac.uk and m.ashworth@dl.ac.uk.

2 Low-level Protocols

2.1 TCP/IP

For the last 20 years, TCP/IP has been accepted as the most common low-level communication method between processes on the Internet. It underlies most of the tools used today, whether browsing Web pages, downloading files or connecting to other hosts. There is a well-established procedure for using TCP/IP to establish socket connections for bi-directional communication in server-client applications (e.g. Stevens, 1998). The Internet protocol provides a reliable way to communicate, but has no security or authentication and no higher-level process control.

2.2 RPC

Remote Procedure Calls (RPC) is a powerful technique for the development of client-server distributed applications. It is based on extending the notion of conventional, or local procedure calling, so that the called procedure need not exist in the same address space as the calling procedure. The two processes may be on the same system, or they may be on different systems with a network connecting them. By using RPC, programmers of distributed applications avoid the details of the interface with the network. The transport independence of RPC isolates the application from the physical and logical elements of the data communications mechanism and allows the application to use a variety of transports.

RPC has an ISO standard specification and is part of the Distributed Computing Environment (DCE). RPC implementations are commonly restricted to C or C++ programs running on UNIX systems. As an exception to this, the Jakarta project (http://jakarta.apache.org/) offers a Java implementation of XML-RPC – a popular protocol that uses XML over HTTP to implement remote procedure calls.

2.3 CORBA

The Common Object Request Broker Architecture (CORBA) (Mowbray and Zahari, 1995, Siegel, 2000) is a low-level architecture established in 1989 by the Object Management Group (OMG) (see http://www.omg.org/). CORBA is an open, vendor-independent architecture and infrastructure that applications can use to work together over networks. Using the standard Internet Inter-ORB Protocol (IIOP), built on top of TCP/IP, a CORBA-based program from any vendor, on almost any computer, operating system, programming language, and network, can inter-operate with any other CORBA-based program.

CORBA provides a specification for the Interface Definition Language (IDL). IDL lets developers define interfaces to their programs and objects in a standardized fashion. With the IDL are mappings that map the IDL definitions and types to programming languages such as C, C++ and Java. CORBA offers developers complete language transparency.

Developer and vendor objects interact with one another through an Object Request Broker (ORB). Using the language mappings, developers can create client-side "stubs" and server-side "skeletons" that their ORBs will understand.

CORBA applications are composed of objects. For each object type you define an interface in IDL. The IDL interface defines the syntax for the contract that the server object offers to the clients that invoke it. Any client that wants to invoke an operation on the object *must* use this IDL interface to specify the operation it wants to perform, and to marshal the arguments that it sends. When the invocation reaches the target object, the *same* interface definition is used there to unmarshal the arguments so that the object can perform the requested operation with them. The interface definition is then used to marshal the results for their trip back, and to unmarshal them when they reach their destination.

The IDL interface definition is independent of programming language, but maps to all of the popular programming languages via OMG standards. OMG has defined standard mappings from IDL to C, C++, Java (Brose et al, 2001), COBOL, Smalltalk, Ada, Lisp, Python, and IDLscript.

The OMG provide a specification – not an implementation. It is up to other individuals, groups and companies to provide implementations.

2.3.1 CORBA implementations

There are many implementations of CORBA currently available. They vary in the degree of compliancy, portability and language mapping features. We present briefly the principal implementations:

- VisiBroker from VISIGENIC and BORLAND is one of the leading commercial ORBs available. It supports 14 platforms including Linux, HP-Unix, NT, RTOS, VxWorks, etc. It complies with the latest CORBA standards: Real-Time, Minimum CORBA, Naming, Event services, load balancing, caching, security,

and persistence. It is also a high performance bus, which can communicate over shared memory, a backplane or TCP/IP. Gateways are also available for COM/DCOM. It also supports SNMP and CMIP for network management.

- ORBix from IONA is also a very solid, fully compliant commercial implementation implemented in Java.
- Orbacus from IONA is standard-compliant with language mappings only for C++ and Java. It supports Solaris, HP-Unix, NT/2000, Linux, Compaq Tru64, AIX and SGI IRIX. It also complies with latest CORBA standards: Real-Time, Minimum CORBA, Naming, Event services, and load balancing. However it is a free solution for non-commercial use. Source code is also available.
- TAO is a freely available implementation of a CORBA Object Request Broker (ORB) developed at Washington University. TAO is a C++ ORB that is compliant with most of the features defined in the CORBA specification. Importantly, it complies with the real-time implementation of CORBA, which focuses on an efficient, predictable and scalable quality of service. It can be downloaded from the Web and used and redistributed without licensing cost. The commercial support, the documentation and training is available on the OCI site (http://www.oci.com/), which also maintains the FAQ. TAO has been used by many commercial applications.
- JacORB (http://www.jacorb.org/) is a 100% pure Java, JDK 1.2 compatible, CORBA implementation. It offers a high-performance, fully multithreaded ORB, and an IDL compiler which supports the OMG IDL/Java language mapping rev. 2.3. A GUI (POAmonitor) allows you to inspect your object adapters. JacORB is freely available with downloads from the aforementioned URL. Examples and full source code are included.

2.4 Java RMI

Java Remote Method Invocation (RMI) enables the Java programmer to create distributed Java-to-Java applications, in which the methods of remote Java objects can be invoked from other Java virtual machines, possibly on different hosts.

A Java program can make a call on a remote object once it obtains a reference to the remote object, either by looking up the remote object in the bootstrap naming service provided by RMI or by receiving the reference as an argument or a return value. A client can call a remote object in a server, and that server can also be a client of other remote objects. RMI uses object serialization to marshal and unmarshal parameters and does not truncate types, supporting true object-oriented polymorphism.

RMI applications often use the client-server model. A typical server application creates a number of remote objects, makes references to those remote objects accessible, and waits for clients to invoke methods on those remote objects. A typical client application gets a remote reference to one or more remote objects in the server and then invokes methods on them. RMI provides the mechanism by which the server and

the client communicate and pass information back and forth. Such an application is sometimes referred to as a distributed object application.

For more information see the Sun Java RMI Web pages at http://java.sun.com/products/jdk/rmi/index.html

3 Network-based Solvers and Information Systems

Computational "power grids" have recently become a hot topic. They do not necessarily give higher computing power for a given application than the largest parallel supercomputers, but they do enable maximal exploitation of existing resources. This is particularly important in the middle range of computing where many computers and clusters are linked over a wide-area network providing a very flexible environment and making use of systems which otherwise would only be lightly loaded. This however raises serious problems of security and authentication, and demands a level of collaboration rarely seen in facility management. Indeed this may become a major success of the technology.

A book on computational grids (Foster and Kesselman, 1998), based on experiences with the GLOBUS project, appeared in July 1998 emphasising its growing importance. It contains a large amount of background material and descriptions of past and current projects by 30 expert authors.

There is also an "Information Power Grid Hotlist" from the NASA Web site, which includes information on distributed computing, meta-computing and Java http://www.nas.nasa.gov/NAS/Tools/.

Power grids provide more than just computational resources. They can also provide access to distributed information sources and instruments (e.g. on telescopes or synchrotron sources). They are fundamental in funding solutions to challenging problems such as smart instruments, collaborative engineering or data mining.

The first computational power grid was GUSTO, a project connecting 27 computer centres across the USA led by Ian Foster (ANL) and Carl Kesselman (University of Southern California). This is described in the GLOBUS entry and in Foster and Kesselman (1998).

NetSolve is a large research project driven from the University of Tennessee, Knoxville and Oak Ridge National Laboratory by Jack Dongarra *et al*. Other network software includes NEOS and NetLink. The NetLink project has similar goals fo NetSolve but via the NetLink access agent. The NEOS network package is specifically designed for optimisation problems. Several other projects are in the pipeline, but being collaborations of major computing centres are on the national scale and mainly originating in the USA.

3.1 AppLeS

Application Level Scheduler. See EveryWare below.

3.2 CAD Services

Name: CAD Services

Description: The Object Management Group (OMG), in addition to producing the CORBA standard, have a number of other application-based working groups. The Manufacturing Domain Task Force (MfgDTF) mission is to foster the emergence of cost effective, timely, commercially available and interoperable manufacturing domain software components through CORBA technology. It has produced the CAD Services specification in order to provide interoperability between Computer Aided Design (CAD), Computer Aided Manufacturing (CAM) and Computer Aided Engineering (CAE) systems. The aim is to provide users of design and engineering systems the ability seamlessly to integrate software across a wide range of CAD/CAM and CAE applications through the use of CORBA interfaces. The specification focuses on establishing CAD system interfaces that provide geometry and topology data to analysis applications and tools.

Two alpha implementations of CAD services have been produced: one implemented by EDS over its CAD system, and the other written by TranscenData over Parametric Technology Corporation's ProE CAD system.

Systems: As CAD Services is built on CORBA technology, it relies on CORBA availability for interfaces with programming languages and hardware portability.

Contacts: Russ Claus,
NASA John H. Glenn Research Center
Lewis Field, 21000 Brookpark Road
Cleveland, Ohio 44135, USA

Email: claus@lerc.nasa.gov

URL: http://www.omg.org/homepages/mfg/mfgcadv1ftf.htm

Comments:

References:

3.3 EveryWare

Name: EveryWare

Description: A user-level software toolkit to write applications to use a computational grid. It consists of a portable set of processes and libraries which can be incorporated in the application so that a wide variety of changing distributed resources can be used together to achieve supercomputer performance.

EveryWare uses three main software components:

lingua franca — a portable communication system;
performance forecasting service — assess availability and loading of resources;
distributed state exchange — synchronise and manage distributed program state.

EveryWare has been used as a tool to interface to a number of environments: Unix; Globus; Legion; Condor; NT and Win32 (using the CygWin set of UNIX emulation tools); Java; and NetSolve.

An EveryWare application implementing a Ramsey Number search algorithm (Wolski *et al*, 1999) claims to be the first true meta-computing application which is pervasive, dependable, consistent and inexpensive. It was a high-performance computing challenge entry at SuperComputing '98.

Systems:

Contacts: Rich Wolski
Department of Computer Science and Engineering, 0114
University of California San Diego
La Jolla, CA 92093 USA

Email: rich@cs.ucsd.edu

URL: http://nws.npaci.edu/EveryWare/

Comments: Developed with NPACI funding and based on previous work on Application Level Scheduler AppLeS (Su *et al*, 1999) http://apples.ucsd.edu/

References: Wolski *et al*, 1999

3.4 GIOD

Name: Globally Interconnected Object Databases

Description: Joint project between Caltech, CERN and Hewlett-Packard is addressing the data storage and access problems posed by the next generation of particle collider experiments which will start at CERN in 2005.

The data rates from the experiments' online systems will be of order 100 MBytes/sec (each event's data is around 1 MByte), giving rise to a yearly accumulation of several PetaBytes. Large processor farms based on commodity hardware will reconstruct the raw data from the online systems to particle tracks, energy clusters, etc. in near-real time. We expect farms of 10^7 MIPS will be required. The reconstructed data (around 100 kBytes per event) will be stored (perhaps with the raw data) in ODBMS.

Object data from around 10^9 particle collisions will need to be made available each year to collaborating physicists. This will require replication of significant fractions of the ODBMS amongst "regional centres" (which serve outlying collaborating institutes), which are scattered across the globe.

The project has been investigating the scalability of commercial ODBMS, and working on models of organising the data to optimise access and analysis for the end-user physicist. There are some serious problems with devising a system architecture that allows sufficient flexibility while at the same time prevents inadvertent abuse.

Systems: Partners are using several existing leading-edge hardware and software systems, namely the Caltech HP Exemplar (a 256-PA8000 CPU SMP machine of some 10^5 MIPS) the High Performance Software System (HPSS) from IBM, the Objectivity/DB Object Database Management System, the Java 3D API from Sun Microsystems, the Versant ODBMS, and various high speed Local Area and Wide Area networks.

Contacts: Julian Bunn,
158-79 CACR,
California Institute of Technology,
1200 E. California Blvd., Pasadena, CA 91125

Harvey Newman,
256-48 HEP,
California Institute of Technology,
1200 E. California Blvd., Pasadena, CA 91125

Email: julian@cacr.caltech.edu, newman@hep.caltech.edu

URL: http://pcbunn.cithep.caltech.edu/

Comments:

References:

3.5 IPG

Name: IPG — Information Power Grid

Description: IPG is designed to implement seamless access to resources between NASA sites and a few NSF and PACI sites. This followed from a number of workshops and reviews in autumn 1997. It grew from the Advanced Computing Networks and Storage (ACNS) and Computation Aerospaces (CSA) programmes at NASA. Goals of the project are to provide access to all resources for a single large simulation and to include virtual reality and access to large-scale data stores. A number of middleware implementations and demonstrator applications are being developed in phase II of the project starting in 3Q99 and continuing until 3Q04. The full project was planned to develop over a seven-year time scale.

The goal is to develop the information technology that enables a geographically distributed national computing and information infrastructures and demonstrate a system prototype.

Applications are likely to include aeronautics and other areas of interest to NASA such as space sciences and earth sciences. For the aircraft development cycle the following requirements were identified:

- seamless networked access to distributed legacy applications;
- cross-platform, interactive visualisation of large 3D data sets;
- intelligent, distributed data mining across unspecified heterogeneous data sources, with privacy and security and using agent technology;
- tools for the development of multi-disciplinary systems integrating co-operating, distributed applications, with process/knowledge capture and exploitation and using automated software engineering tools;
- grid interfaces for process invocation and status monitoring with scheduling support for jobs.

In addition there were a number of real-time requirements for aircraft operations systems.

Systems:

Contacts: Bill Johnston,
Project Manager,
Lawrence Berkeley Laboratory,
USA

Email:

URL: http://science.nas.nasa.gov/Groups/Tools/IPG

Comments:

References:

3.6 NEOS

Name: NEOS — Network-Enabled Optimization System

Description: The Optimization Technology Center (OTC) (see http://www.ece.nwu.edu/OTC/). A joint project between the US Argonne National Laboratory and NorthWestern University which started in 1994. The mission of the centre is to widen the community awareness of optimisation techniques and to promote the use of such techniques. NEOS is a WWW interface to the software and computational resources at the OTC and is provided as three levels: a server; a guide (decision tree); and a set of tools. NEOS currently solves normal and stochastic problems and linear network optimisation problems. The NEOS guide also contains information about software packages and case studies.

An extension to NEOS is the MetaNEOS project that provides an interface to a computational grid running NEOS software via Condor flocking and glide-in mechanisms. This was used to solve the nug30 quadratic assignment benchmark (see below). It is supported by the MW communication harness in Condor, which can be accessed via the Web pages.

Systems: Intel, SGI, Sun, HP etc.

Contacts:

Email: metaneos@mcs.anl.gov

URL: http://www-neos.mcs.anl.gov/neos/
http://www-unix.mcs.anl.gov/metaneos

Comments:

References: Czyzyk *et al*, 1996

Researchers at the University of Iowa and Argonne National Laboratory announced in June 2000, that they had solved the nug28 quadratic assignment problem using the Condor high-throughput computing system. The system was developed at the University of Wisconsin, one of the National Computational Science Alliance's Partners for Advanced Computational Services (PACS) and a member of the Enabling Technologies team.

The quadratic assignment problem (QAP) is a standard model in the area of applied mathematics known as location theory. In such a problem, there are a set of n

locations and a set of n facilities, and each facility must be assigned a location. To measure the cost of each possible assignment, the flow between each pair of facilities is multiplied by the distance between the pair's assigned locations, and then a sum is taken over all of the pairs.

The goal is to find the assignment that minimizes total cost. Despite the simplicity of the problem statement, QAPs are incredibly difficult to solve to optimality.

"The QAP is, for its size, among the hardest of all combinatorial optimization problems," says Kurt Anstreicher, a professor of management sciences at the University of Iowa. Anstreicher and his student Nathan Brixius designed the algorithm that solved the nug28 problem. This problem is derived from a particularly notorious QAP now known as nug30, which was first suggested as a test problem in 1968. Despite enormous advances in computational power and discrete optimization theory, the nug30 problem remained unsolved until July 2000.

"We designed a state-of-the-art algorithm," says Anstreicher, "but without a state-of-the-art computational platform we would never be able to attack a QAP like nug28." Anstreicher and Brixius worked in collaboration with Jeff Linderoth and Jean-Pierre Goux of Argonne National Laboratory, an Alliance partner, to implement their QAP algorithm using the Master-Worker runtime support library.

The Master-Worker library, developed by researchers at Argonne, Northwestern University, and Wisconsin as a part of the MetaNEOS project, uses Wisconsin's Condor system to send work to a potentially large number of processors working in parallel. The system is well suited to algorithms that can exploit a high degree of paralellism with relatively low bandwidth requirements.

Condor harnesses the power of desktop computers and commodity clusters by monitoring their status and running jobs on them when they are available. Machines at Wisconsin and the Albuquerque High Performance Computing Center, both Alliance PACS site, ran the computation, as well as computers at the Italian Instituto Nazionale di Fisica Nucleare.

The solution consumed over 18,000 CPU-hours in just over four days. About 200 workstations were being used to solve the problem at any given time. The computation would have taken over 400 days to complete on a single workstation.

"Optimization is one example of the many scientific disciplines that have been served by Condor," says Miron Livny, a computer science professor at Wisconsin who heads the Condor project. "By harnessing the huge computing power of desktop and commodity hardware and making it accessible to the scientific community, Condor enables computationally intensive science and the development of a new generation of computing technology."

The National Computational Science Alliance is a partnership to prototype an advanced computational infrastructure for the 21st century and includes more than 50 academic, government and industry research partners from across the United States. The Alliance is one of two partnerships funded by the National Science Foundation's Partnerships for Advanced Computational Infrastructure (PACI) program, and receives cost-sharing at partner institutions.

Nug30 was solved using a system of over 2500 processors consuming over 60,000 HP-C3000 equivalent hours of computation in 7 days.

3.7 NetLink

Name: NetLink

Description: Has the objective to find a data distribution architecture that, in a distributed manner, can help to centralise library maintenance and tuning. Uses an “access agent” component for a variety of software and target hardware and to provide authentication via secure domains. Uses a sophisticated cache mechanism for object searches.

Systems:

Contacts: I. Holmqvist and E. Lindström,
Umea University,
S-90187 Umea,
Sweden

Email: dpiht@cs.umu.se

URL: http://www.hpc2n.umu.se/

Comments: Prototype only. Uses resources at HPC2N.

References: Holmqvist and Lindström, 1998

3.8 NetSolve

Name: NetSolve

Description: Uses a client-server-agent software architecture to harness loosely coupled systems on a network. NetSolve is intended to provide transparent access to a whole variety of software libraries, highly tuned for the target architecture. This improves maintainability of software and avoids the end user having to download and compile it.

NetSolve is implemented as a three-tiered system:

1. a client — specifies the problem. May be a C or Fortran program linked to the NetSolve library, Mathematica sessions, or Java applets calling the NetSolve library. A Java GUI is also provided;
2. agents — C programs running as daemons act as resource brokers;
3. servers — registered with agents and can perform certain services (e.g. have particular applications installed). Provide optimal computation environment for their particular architectures.

At the API level NetSolve looks like a high-level library with a single function call `netsolve`. Character strings are introduced to specify the required action. A non-blocking version is also available, but the user has to take care of resource usage and determinism.

NetSolve currently has an interface to ScaLAPACK and related components.

Systems: uses Condor (Litzkow and Livny, 1990) for its distributed computing management

Contacts: J. Dongarra and H. Casanova,
University of Tennessee,
Knoxville,
TN 37996,
USA

Email: dongarra@cs.utk.edu or casanova@cs.utk.edu

URL: Available from NetLib http://www.netlib.org/

Comments: Currently a prototype

References: Casanova and Dongarra, 1997

3.9 Nile

Name: Nile

Description: Nile is developing a distributed computing solution for the CLEO High Energy Physics experiment. The goal is to provide a self-managing, fault-tolerant, heterogeneous system of hundreds of commodity workstations, with access to a distributed database in excess of 100 TB. These resources are spread across the United States and Canada at 24 collaborating institutions. Nile will allow any resource to be accessed and used transparently by any member of the collaboration, from anywhere within the collaboration. The Nile system must out live its development phase, adapt to and scale with changes in CLEO's computing needs, be easily maintained, and be able to

incorporate new software components as they become available. To address the longevity, maintainability, and adaptability concerns Nile uses the CORBA standard, whilst the scalability and fault tolerance concerns led to using a widely distributed architecture.

Systems:

Contacts: Cornell

Email:

URL: http://www.nile.cornell.edu/

Comments:

References:

3.10 Ninf

Name: Ninf — Network-based Information Library for Global World-wide Computing Infrastructure

Description: Ninf is an ongoing global network-wide computing infrastructure project that allows users to access computational resources including hardware, software and scientific data distributed across a wide area network with an easy-to-use interface. Ninf is intended not only to exploit high performance in network parallel computing, but also to provide high-quality numerical computation services and access to scientific databases published by other researchers. Computational resources are shared as Ninf remote libraries executable at a remote Ninf server. Users can build an application by calling the libraries with the Ninf Remote Procedure Call, which is designed to provide a programming interface similar to conventional function calls in existing languages, and is tailored for scientific computation. In order to facilitate location transparency and network-wide parallelism, the Ninf meta-server maintains global resource information regarding computational server and databases, allocating and scheduling coarse-grained computation to achieve good global load balancing. Ninf also interfaces with existing network services, such as the Web, for easy accessibility.

Ninf IDLs have so far been defined for LAPACK, LibSci and other numerical libraries and databases. There is a major project to develop a CFD modelling network.

Systems: Cray J90, SUN UltraSparc and SparcStation 20, NOW clusters

Contacts: Ninf Administration Group,
Electrotechnical Laboratory,
Umezono, Tsukuba 305, Japan

Email: ninf@apgrid.org

URL: http://ninf.apgrid.org/

Comments: Ninf is an on-going research project. ETL also maintains a useful mirror site for numerical algorithms and high-performance computing at http://phase.etl.go.jp/

References: Sekiguchi *et al*,1996

3.11 PPDG

Name: PPDG — the Particle Physics Data Grid

Description: A number of US University and DoE collaborators proposed the PPDG for next-generation Internet funding in 1999. The objectives are:

1. delivery of an infrastructure for widely distributed analysis of particle physics data at multi-petabyte scales by thousands of physicists;
2. acceleration of the development of network and middleware infrastructure aimed broadly at data-intensive collaborative science.

The proposed research is intended to test a number of hypotheses in designing, developing and deploying a network and middleware infrastructure capable of supporting data analysis and data flow patterns common to many particle physics experiments:

- an infrastructure built on emerging network and middleware technologies can meet the functional and performance requirements of wide area PPD analysis;
- specific data flow patterns, including sustained bulk data transfer and distributed data access by large numbers of clients, can be supported;
- the infrastructure can be compatible with commercial middleware technologies such as object databases, ORBs and common object services.

The project builds on a number of related projects of the collaborators: Globus; Objectively Open File System (OOFS); Globally Interconnected Object Databases (GIOD); Sequential Access Method (SAM); Storage Access Co-ordination System (STACS); Scalable Object Storage and Access; Condor; Storage Request Broker (SRB) and the China Clipper project.

Systems:

Contacts: Harvey B. Newman

California Institute of Technology
1200 East California Blvd.,
Pasadena, CA 91125,
USA

Richard P. Mount,
Stanford Linear Accelerator Centre,
Mail Stop 97, PO Box 4349,
Stanford, CA 94309,
USA

Email: newman@hep.caltech.edu, richard.mount@stanford.edu

URL: http://www.ppdg.net/

Comments:

References:

3.12 PSUE

Name: PSUE — Parallel Simulation User Environment

Description: The Parallel Simulation User Environment (PSUE) is a suite of modules that are closely linked and enable the user to carry out computational engineering problems easily and quickly. The main objectives of the PSUE is to decrease problem set-up time for computationally intensive tasks and to allow inexperienced users quickly to learn about such problems and go on to use the environment for more complicated tasks.

The PSUE includes the following capabilities:

- geometry builder;
- geometry repair;
- unstructured grid generation;
- grid quality analysis;
- remote/parallel platforms;
- post-processing and data analysis;
- help facilities;
- application integration.

Systems: The PSUE has been developed using X, OSF Motif and OpenGL library routines, which are available across most UNIX platforms. These routines give a consistent, modular feel to the graphical interface.

Contacts: Prof. N. Weatherill,
Dept. Civil Engineering,
University of Wales, Swansea,
Singleton Park, Swansea SA2 8PP, UK

I.C. Risk,
British Aerospace (Operations) Ltd.,
Sowerby Research Centre,
Filton, Bristol BS12 7QW, UK

Email: n.p.weatherill@swansea.ac.uk, ian.risk@src.bae.co.uk

URL: http://www-simulations.swan.ac.uk/PSUE/

Comments: Part of the EU projects Caesar (FP3 number 8328) and Julius (FP4 number 25050)

References:

The geometry builder is capable of importing CAD data that can be modified and simple geometrical entities created using point, line and surface creation. A geometry repair facility overcomes topological inconsistencies correcting surface overlaps and gaps. Each of these tools prepares the data for the unstructured grid generation module that uses a Delaunay triangulation algorithm to efficiently generate 2D planar/3D surface triangles and 3D volume tetrahedra. Grid point density is controlled using boundary point distribution and point, line and triangular sources either imported or created using the geometry builder. Grid cosmetics are incorporated to improve grid quality that can be examined using histogram and visual techniques. Numerical libraries are available to create element-, side- and face-based data. The ability to utilise remote and parallel platforms allows the use of the most suitable machine size/type for specific operations, particularly, grand challenge problems. The incorporation of links to visualisation packages AVS and ENSIGHT allow easy access to the wealth of post-processing facilities available in commercial software.

Application integration allows users the flexibility to incorporate their own, commercial or public domain software into the environment. This can be performed at many levels of operation through user defined script files. Recompilation of the PSUE is not required, providing a fast and efficient method of consolidating all the user's software together into a single package. Once integrated data may be sent to and from applications via a data transfer interface that may use file, pipe or socket transfer.

3.13 SAM

Name: SAM — Sequential Access Method

Description: A data access framework for analysis of particle physics experiment data. SAM is implemented as a set of distributed servers, with well-defined access services provided include writing, cataloguing and reading data. Data discovery services resolve logical definitions of data and queries on data into physical files or in some cases specific events within files.

Systems:

Contacts: Fermilab, USA

Email: sam-users@fnal.gov

URL: http://d0db.fnal.gov/sam/
http://projects.fnal.gov/act/sam_cluster/WhitePaper.htm#_edn1

Comments:

References:

3.14 WebHLA

Name: WebHLA — Web-based High-Level Architecture

Description: WebHLA is a US Department of Defense project to develop distributed computing services via commodity systems. An interactive programming and training environment for high-end computing. Uses a three-tier approach implemented in JWORB middleware over the GLOBUS meta-computing or NT cluster back end:

front end — WebFLOW enables visual collaborations and visual authoring tools connect meta-computing application;
distributed objects — WebFLOW servers acting as proxies to computer systems;
back end — WebFLOW clients.

This follows the philosophy of the Pragmatic Object Web of Fox *et al*.

Systems:

Contacts: G.C. Fox and W. Furmanski,
Syracuse University,
Syracuse, NY, USA

Email: gcf@npac.syr.edu, furm@npac.syr.edu

URL: http://www.npac.syr.edu/

Comments: Currently a prototype, but an early application was demonstrated at SuperComputing'98.

References: Fox and Furmanski, 1997, Orfali and Harkey, 1997

4 Distributed Application Management Tools

This section describes some of the "middle-ware" tools that can be used to develop applications using distributed resources. They are used by many of the other tools in this report, except the ones that have their own client-server components.

There is a great deal of overlap between tools in this section and the ones which we have designated “meta-computing tools”. Particular issues for middleware software are: security; authentication; system failure recover; checkpointing; system availability; load balancing; job management and process migration. An overview is provided by Hwang and Xu, 1998.

Typically users want to be able to create a request to run a job containing a list of restrictions regarding its operation, e.g. number of nodes, latest computational time, etc. The local resource management service (RMS) will try to match the requirements against resources available via its own scheduler. It is cannot satisfy the request it will pass on a request to another RMS on the wide area network.

Agents, Software Agents, Intelligent Mobile Agents (IMAs) and Softbots are terms used to describe the concept of mobile computing or mobile code (Bradshaw, 1997). An example of agent technology is D'Agents from Dartmouth College, USA.

Meta-computing environments introduce the following RMS problems (Czajkowski *et al*, 1998):

? Site authority — resources are typically owned by different organisations in different administrative domains. Policies of scheduling, security etc. will be site-specific in nature and make inter-working difficult;

? Heterogeneous substrate — different sites use different local RMS, some of which are mentioned below. Even if the same system is used between two sites, configuration differences and local modifications can change apparent functionality;

? Policy extensibility — meta-computing applications span a large range of disciplines. The RMS must support domain-specific and diverse requirements to allow new development and accommodate existing users;

? Co-allocation — true meta-computing requires simultaneous access to multiple resources and also monitoring and managing computations;

? Online control — negotiation may be required to adapt resource availability to application requirements in a dynamic manner. Tele-immersive applications are typical of this class. A RMS must support such negotiation via a priority system, perhaps via a firrerential service.

Whilst theoretical solutions of these problems are challenging, practical implementations are even more so and the need to communicate over a wide-area network (WAN) may eliminate some otherwise ideal solutions.

If the various resource-management and security issues can be overcome giving a user access to a well defined set of computational resources with hierarchical interconnect, then there remains the problem of programming. At the time of writing there are few algorithms or applications capable of using heterogeneous systems,

although they are likely to become more widespread with the availability of NUMA architectures stimulated b the ASCI programme. Projects addressing programming issues are not discussed here, but include MPICH-G, PACX-MPI (Gabriel *et al*, 1997), STAMPI (Imamura et al, 2001) and MagPIe (Kielmann, 1999).

4.1 ASCI Distributed Systems

Name: ASCI Distributed Systems project

Description: The Distributed Systems portion of the ASCI Problem Solving Environment consists of the networking, Distributed Computing Environment (DCE), and Distributed Resource Management (DRM) activities. These three research and development activities are very interdependent and are fundamental to the creation of the basic infrastructure for the ASCI environment.

For the past two years, the DCE and networking projects in ASCI have been tightly coupled. For example, the DCE goal of providing a common cross-cell authentication capability that will enable a user's single authentication to be honoured on all ASCI computing platforms and servers is key to removing major networking barriers and achieving the necessary network throughput performance.

The DRM activities are closely tied to the DCE project in that:

- DRM depends on DCE for authentication;
- DRM will offer access to DFS from batch jobs;
- work is under way to manage the lifetimes of DCE credentials for use by batch jobs;
- strong consideration is being given to replacing the current socket-based communication in DPCS at LLNL with DCE RPCs.

In the future, networking and visualisation resources will need to be managed and scheduled. Therefore, the DRM project will require an even greater awareness of networking and visualisation issues in order to begin to include these elements in the resource management strategy.

In terms of correspondence to other ASCI programs, the Distributed Systems activities are closely aligned with most of those PSE research projects that directly relate to the Distance Computing portion of DisCom2, and to a lesser extent to the Distributed Computing portion. Similarly, the Distributed Systems activities relate well to NEWS visualisation corridor efforts. By administratively grouping networking, Distributed Computing Environment and Distributed Resource Management. Co-ordination between these three closely related and interdependent activities is being increased.

Systems: ASCI tri-lab platforms

Contacts: Bob Tomlinson, LANL
Barry Howard, LLNL
Doug Brown, SNL

Email: bob@lanl.gov, bhoward@llnl.gov, cdbrown@sandia.gov

URL: http://www.llnl.gov/asci/pse/ds/dist_sys.html

Comments:

References:

The Tri-Lab distributed computing team (a consortium of computer scientists from LLNL, LANL and SNL) is collaborating to co-ordinate:

- implementations of the DCE and DFS technologies from the Open Group;
- security plans and MOUs for cross-cell trust between sites;
- with other PSE components and platforms for a well integrated solution;
- with other collaboratory efforts, including the ESnet DCE Working Group;
- early access with users to establish testbeds and evaluate technologies.

4.2 Codine/GRD

Name: Codine/GRD — COmputing in DIstributed Networked Environments

Description: Codine is a Resource Management System targeted to optimise utilisation of all software and hardware resources in a heterogeneous networked environment. The easy-to-use graphical user interface provides a single-system image of all enterprise-wide resources for the user and also simplifies administration and configuration tasks. The tools originated as public-domain projects in the USA c.1992 but are now marketed by Gridware Inc. that was formed in a merger in late 1999 of Genias Software GmbH in Germany and Chord Systems Inc. in the USA. It was then acquired by Sun in summer 2000.

Codine contains components for:

- resource license management;
- heterogeneous distributed computing;
- SMP computing support;
- job queue management;
- fault tolerance;
- checkpointing and migration.

X11 and Motif GUI interfaces are provided for programming and system management.

An additional product based on Codine is the Global Resource Director (GRD). GRD provides optimal utilization of computing resources, allows policy-oriented modification of priorities during execution and guarantees optimum turnaround times for batch users. It is a suite of tools that provide enterprises with powerful and cost-effective workload management. Features include:

- centralised workload management;
- global workload management strategies including share-based, priority and deadline scheduling;
- advanced job-queuing and load-balancing;
- automated policy enforcement;
- global resource management.

GRD was a joint development of Genias Software and Raytheon Systems Company.

Systems: DEC Unix, SUN SunOS and Solaris, Parsytec, HP, SGI, IBM AIX

Email:

URL: http://wwws.sun.com/software/gridware/

Comments: Codine/GRD was marketed by Gridware Inc. until it was taken over by Sun Microsystems in 2000.

References:

4.3 Condor

Name: Condor v6.1.5

Description: The goal of the Condor project is to develop, implement, deploy, and evaluate mechanisms and policies that support high-throughput computing on large collections of computing resources with distributed ownership. Guided by both the technological and sociological challenges of such a computing environment, the Condor Team has been building software tools that enable scientists and engineers to increase their computing throughput.

Condor manages processes in a pool of workstations. It provides transparent checkpointing and restart facilities so that computations can be moved from over-loaded or failed machines onto lightly loaded ones. Current limitations (not unique to Condor) include:

- only migrates processes between machines of the same architectures;
- only migrates processes within its own server;

- only works with serial (single-process) programs;
- system calls are always executed on “host” machine.

Several run-time mechanisms are provided in the Condor model to facilitate different load-sharing strategies. These include “flocking” and “glide-in”. These were recently used in the MetaNEOS project to link over 2500 processors.

A communication harness, MW, has been introduced into Condor to support master-worker applications on a grid. This includes a top-level API and a bottom-level grid interface.

Systems:

Contacts: M. Livny,
University of Wisconcin, USA

Email:

URL: http://www.cs.wisc.edu/condor/

Comments: Used as a middle layer in NetSolve.

References: Litzkow and Livny, 1990, Epema *et al*, 1996, Buyya, 1999

4.4 D'Agents

Name: D'Agents — Dartmouth Agents

Description: A mobile agent is a program that can migrate from machine to machine in a heterogeneous network. The program chooses when and where to migrate. It can suspend its execution at an arbitrary point, transport to another machine and resume execution on the new machine. In the picture below, an agent carrying a mail message migrates first to a router and then to the recipient's mailbox. The agent can perform arbitrarily complex processing at each machine in order to ensure that the message reaches the intended recipient.

Mobile agents have several advantages over the traditional client/server model:

- efficiency: mobile agents consume fewer network resources since they move the computation to the data rather than the data to the computation;
- fault tolerance: mobile agents do not require a continuous connection between machines;
- convenient paradigm: mobile agents hide the communication channels but not the location of the computation;

- customisation: mobile agents allow clients and servers to extend each other's functionality by programming each other.

There are alternative techniques that have many of these same advantages such as queued RPC, proxy servers, etc. The problem with these alternative techniques is that each one is only suitable for certain applications. A mobile-agent system on the other hand is a single, unified framework in which a wide range of distributed applications can be implemented easily and efficiently.

D'Agents is a mobile-agent system under development at Dartmouth College. The ultimate goal of D'Agents is to support applications that require the retrieval, organisation and presentation of distributed information in arbitrary networks. Some of the research areas are:

- Security mechanisms;
- Support for mobile and partially connected computers;
- Navigation, network sensing and resource discovery tools;
- Automatic indexing, retrieval and clustering techniques for text and other documents (D'Agents is used in several information-retrieval and workflow applications).

Other notable mobile agent systems include:

- Telescript and Odyssey from General Magic;
- ARA (Agents for Remote Access);
- TACOMA;
- IBM Aglets.

Systems:

Contacts: Prof. G. Cybenko,
Thayer School of Engineering,
Dartmouth College, USA

Email: rgray@cs.dartmouth.edu.
There is also an un-moderated majordomo mailing list for users.

URL: http://www.cs.dartmouth.edu/~agent/

Comments: Source code and documentation is available from the Web page.

References: Brewington *et al*, 1999

4.5 DOME

Name: DOME — Distant Object Migration Environment

Description: The goal of the Dome project is to build sets of distributed objects that can be used to program heterogeneous networks of computers as a single computing resource. Dome addresses the problems of load balancing in a heterogeneous multi-user environment, ease of programming, and fault tolerance.

Project components include:

- architecture independent checkpointing;
- dynamic load balancing at runtime

Example applications are a distributed dot product and a simple molecular dynamics code.

Systems:

Contacts: Adam Beguelin,
Carnegie Mellon University,
USA

Email: adamb@cs.cmu.edu

URL: http://www.cs.cmu.edu/~Dome/

Comments:

References:

4.6 GLOBUS

Name: GLOBUS

Description: The Globus project is developing basic software infrastructure for computations that integrate geographically distributed computational and information resources.

GLOBUS is a joint project of Argonne National Laboratory and the University of Southern California's Information Sciences Institute. Project team includes groups at Argonne, USC/ISI, and the Aerospace Corporation, with significant contributions also being made by other partners.

Core components include:

- GRAM — Globus Resource Allocation and process Manager provides uniform resource allocation, object creation, computation management and co-allocation mechanisms for diverse resource types;

- Nexus — heterogeneous communication infrastructure, supports unicast and multicast;
- MDS — Meta-computing Directory information Services, structure and state information;
- GSI — Grid Security Infrastructure, authentication and related security services, provides public key based single sign-on, run-anywhere capabilities for multi site environments. GSI supports proxy credentials, inter-operability with local security mechanisms, local control over access, and delegation. A wide range of GSI-based applications has been developed ranging from `ssh` and `ftp` to MPI, Condor and the SDSC Storage Resource Broker;
- GASS and GEM — Global Access to Secondary Storage and Global Executable Manager. GASS provides a uniform name space (via URLs) and access mechanisms for files accessed via different protocols and stored in divers storage system types (HTTP, FTP, HPSS, DPSS etc.).

Participants in the GUSTO Consortium of NPACI sites are testing GLOBUS concepts on a global scale.

GLOBUS is also the first software on the TransPAC network which links the Asia Pacific Advanced Network (APAN) and vNBS academic research networks, see URL http://www.transpac.org/. Scientific applications will be ported as the system is build throughout 1999.

Systems: Runs on many UNIX-style operating systems such as AIX, FreeBSD HPUX, Digital HPUX, IRIX, Linux SPP-UX, Solaris and UNICOS/mk.

Contacts: Ian Foster,
Argonne National Laboratory, USA

Carl Kesselman,
University of Southern California, USA

Email: foster@mcs.anl.gov, carl@isi.edu

URL: http://www.globus.org/

Comments: Tutorials are available at http://www.globus.org/training/. Foster and Kesselman (1998) feature Globus in “The Grid” book.

References: Foster and Kesselman, 1997, Foster and Kesselman, 1998, Buyya, 1999

Several steps are required in writing an application to use GLOBUS. Tools are, for instance, provided in EveryWare to do this. Each component of the GLOBUS toolkit may be used independently or in connection with the other services. GRAM can be used for process creation and control. It acts as a “gatekeeper” that first creates certificates of authenticity for each user enabling access to remote compute

resources. It interfaces to local RMS software such as NQE or LSF as shown in Figure 1. Once processes are executing GRAM provides the user with a means to check job status, kill jobs or read output. GASS can be used to access common persistent storage. Servers essentially allow remote processes, which use the GASS client utilities or library functions, to access local file systems. GASS servers thus act as simple file servers binding to a port and transferring files to and from the local file system driven by requests from remote processes. The BDS is based on a Lightweight Directory Access Prototol (LDAP) (Yeong *et al*, 1995). This acts as a general-purpose repository for information about resources in the GLOBUS testbed. Among other data it stores information about where each gatekeeper is running, how to contact it (i.e. TCP/IP port number) and how many nodes are free on the resource it manages.

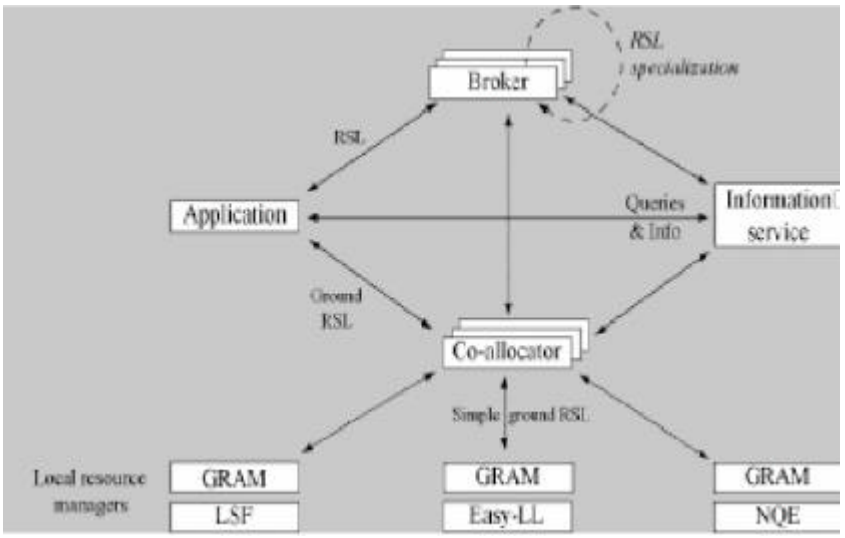

Figure 1: GLOBUS GRAM structure.

4.7 Hector

Name: Hector

Description: Supplies systematic support for run-time checkpointing and process migration, providing information for dynamic data-parallel load balancing. Contains a complete MPI implementation based on MPICH with interfaces to a self-migration facility, command and control structure and instrumentation facility. Linking MPI programs to a library

provides access to these services. Runs in a distributed/centralised manner.

Run-time information about CPU loading and memory usage of every candidate platform is collected to determine machine's status. Support is provided for data-parallel load balancing techniques such as "factoring" and "fractiling" which enable data exchange between peers or "Pirhana" which uses a master process. Data migration off a machine does not imply that the running process is terminated, so data can be moved back at a later date.

Systems:

Contacts: S.H. Russ,
Mississippi State University, USA

Email: russ@erc.msstate.edu

URL:

Comments:

References: Russ *et al*, 1996

4.8 LSF

Name: LSF — Load Sharing Facility

Description: Widely used for corporate computational resource management, especially in the engineering industry. Platform aim to provide the best application resource management solutions for enterprise, allowing them to intelligently harness and leverage the maximum power from their existing computing systems by using idle cycles in a flexible and dynamic manner.

Some users who are working in this way include AMD (design of the 1 GHz Athlon chip), Sanger Centre, Cambridge (decoding human chromosome 22), NCSA Alliance (running world's largest NT cluster), Daimler-Chrysler (engine and car development and crash simulation), Ford (advanced vehicle design), Bombardier (aerospace), Lockheed-Martin (aerospace), Mill Film (computer graphics), Cine Studio (computer graphics), Ericsson (Electronic Design Analysis), Alcatel (EDA), Infineon (EDA), Shell (seismic data processing and oil reservoir modelling) and RABA (defence analysis and research).

The current product is LSF Suite 4.0 that includes a number of modules (base, batch, analyzer, parallel, multicluster, jobscheduler and make). The base module maintains a load information manager which, for each host on the LSF network, collects data on indices

such as CPU queue length and utilisation, available user memory, paging and disk i/o rate, number of users, host idle time, available swap space, available /tmp space, host status. Additional external indices may be configured such as machine type, software license availability, network load etc. The batch schedule engine uses this information and a description of the requested job and priority, to decide when and where to start jobs. The analyzer keeps records of this and is able to produce accounting information and usage reports as required. The parallel module provides integrated MPI libraries with appropriate accounting procedures. The jobscheduler is used for data processing, and the make module uses GNU make for remote compilations.

In addition to scheduling jobs according to the system indices and a priority, limits may be set on run time, number of jobs per host, job type or user. Chaining of jobs is possible for pre- and post-execution processes. Configuration is fully dynamic to account for changes in workload type, e.g. interactive to batch at night and weekends.

Checkpointing and restart facilities are built in and can be activated by LSF at given periods or by user-provided hooks. Job suspension and migration can be carried out in this way.

Systems: Many including: Sun, HP, Compaq, SGI, IBM, Fujitsu, Hitachi, NEC, Sony and NT systems.

Contacts: Roland Richardson,
Platform Computing Ltd.,
Units 18/19, Intec 2, Basingstoke, Hampshire RG24 8NE

Email: info-europe@platform.com

URL: http://www.platform.com/

Comments: A number of applications are integrated with LSF including: NQS, SNMP, Fluent, Maya, ClearCase, Unicenter TNG, Condor (for checkpointing).

References:

4.9 Nexus

Although often referenced separately, this is now a heterogeneous communication layer of GLOBUS.

4.10 SPEEDES

Name: SPEEDES — Synchronous Parallel Environment for Emulation and Discrete Event Simulation

Description: Promotes the development of complex, inter-operable simulations on geographically dispersed high-performance parallel computers.

SPEEDES correctly co-ordinates event processing in a run-time environment to link disjoint simulations in logical time through the use of optimistic rollback techniques that are fully stable even when processor work-loads are poorly balanced. The Breathing Time Warp algorithm accomplishes this by only sending messages between processors that are unlikely to be rolled back.

High Level Architecture (HLA) services provided include:

- Object management;
- declaration management;
- data distribution management;
- ownership management;
- time management;
- federation management.

Simulation objects must have state information so that events can be time-ordered and “lazy” roll-back can occur for straggling events. This is managed by defining “publication regions” and “update regions” in the simulation objects.

Uses the IMPORT simulation language and compiler to target its process model capabilities.

Systems: MPI and shared-memory under UNIX and NT are supported.

Contacts: J.S. Steinman,
Metron Incorporated, 512 Via de la Valle, Site 301,
Solana Beach, CA 92075, USA

Email: steinman@ca.metsci.com

URL:

Comments:

References: Steinman *et al*, 1999

4.11 UNICORE

Name: UNICORE — UNIform access to COmputing Resources

Description: Funded by the German ministry for science and education, starting in August 1997 for two years, as a prototype for sharing access to facilities at German supercomputing centres. The aim is to provide seamless, secure and intuitive batch access for diverse computing resources. Genias and Pallas implemented prototype software. Development is to continue in a follow-on project by Pallas.

Centres that are currently linked include:

- Rechenzentrum der Universität Stuttgart (RUS);
- Rechenzentrum der Universität Karlsruhe;
- Leibnitz Rechenzentrum München (LRZ);
- Konrad-Zuse-Zentrum Berlin (ZIB);
- Paderborn Centre for Parallel Computing (PC2).

Other project partners include:

- Forshungszentrum Jülich;
- Deutsche Wetterdienst (DWD);
- ECMWF.

Early users included Debis and INPRO who carry out modelling work for the German automobile industry.

UNICORE includes a Web-base Java GUI for batch submission with the same look and feel independent of target system and facilitates distribution of work to the most suitable platform and site. Information about resources is provided. Use is made of existing technology with access to distributed data. The three-layer approach comprises a browser running on the user's workstation that communicates with a UNICORE gateway running at any of the collaborating sites. This contains a UNICORE authentication procedure (using X.509 certificates) and site-specific authentication and login authorisation. Finally a resource-management layer will submit the job to the local system or initiate further authentication for submission to a remote site.

Systems: Fujitsu, Hitachi, HP, IBM, NEC, SGI/Cray and SUN are currently included.

Contacts: D. Erwin,
Forshungszentrum Jülich,
Germany

Karl Solchenbach,
Pallas GmbH,
Bonn, Germany

Email: d.erwin@fz-juelich.de, info@pallas.com

URL: http://www.unicore.de/
http://www.fz-juelich.de/unicore/

Comments: A tutorial CD is available.

References:

The UNICORE project has developed a software infrastructure that provides users with seamless and secure access to distributed computer systems. UNICORE focuses on batch processing.

The typical workflow for a simulation running on a supercomputer is assumed to consist of the following steps:

- pre-processing (e.g. grid generation or data acquisition);
- simulation (one or several batch jobs);
- postprocessing (e.g. visualization or data archiving).

In UNICORE, the pipeline can be set up in a single step. This includes data transfer and submission of batch jobs to multiple sites. In fact, the most general situation that can be realised as a single UNICORE job is a structure that can be represented as a directed acyclic graph. UNICORE is designed to incorporate features like load balancing and meta-computing support in the future.

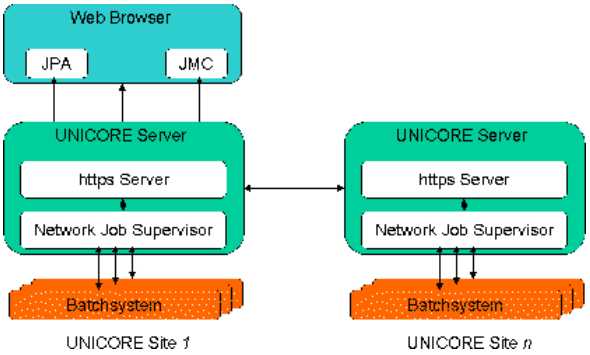

Figure 2: UNICORE software architecture

The main UNICORE components, shown in Figure 2 are the Job Preparation Agent (JPA), the Job Monitor Controller (JMC), the UNICORE https server, also called the Gateway, and the Network Job Supervisor (NJS).

Users interact with the JPA to create UNICORE jobs and to submit them to a UNICORE site. Job descriptions can be saved to disk on the local system and can easily be modified and re-used. Descriptions are system-independent, all platform and site-specific parameters and procedures are filled in by the NJS at submission time.

The JMC provides a seamless interface that lets you monitor the progress and control the execution of UNICORE jobs, including access to the job's output and to result files.

JPA and JMC feature an intuitive graphical user interface with online help and assistant functionality. They are implemented as Java applets — requiring only a Web browser for execution.

The UNICORE Gateway performs authentication based on X.509 certificates, plus optional site-specific authorisation. UNICORE uses the secure http protocol (https), thus guaranteeing secure communication between JPA/JMC on the local workstation and NJS on the remote UNICORE site.

The Network Job Supervisor (NJS) produces site-specific UNICORE jobs by filling in parameters and procedures relating to site and platform, and interfaces with the local queuing system(s). For jobs spanning multiple UNICORE sites, the NJS at the different sites co-operate to co-ordinate job scheduling and execution.

4.12 WebOS

Name: WebOS

Description: A software framework WebOS is being constructed which will provide incrementally scaleable geographically aware Web service. This is part of the Berkeley NOW II project.

Systems:

Contacts: Amin M. Vahdat,
Department of Computer Science,
Levine Science Research Center (D308),
Box 90129, Duke University,
Durham NC 27708-0129

Email: vahdat@cs.duke.edu

URL: http://www.cs.duke.edu/ari/issg/webos/

Comments:

References: Vahdat *et al*, 1998

4.13 WOS

Name: WOS — Web Operating System

Description: WOS is a component-based approach to demand-driven services provision on heterogeneous and dynamic resources. Communication replaces the notion of servers. Different versions of services and WOS may be running on a given network. Services are provided by "eduction engines" and "warehouses" connected by a discovery/ location protocol (WOSRP). There is also a generic services protocol (WOSP). This could be implemented as an interface to CML, CBL or CORBA.

WOS considers communication to be the central issue, rather than maintaining a list of central servers. This promotes a "net-centric" approach. TCP/IP is assumed for the base protocol.

The underlying structure of WOS assumes that subsets of nodes define a particular environment and context. These can be dynamically and autonomously created and a node may belong to more than one context. The concept of "versioning" is introduced to avoid conflicts and a linked to the discovery/ location and service communication protocols.

Systems:

Contacts: Prof. P.G. Kropf,
Département d'informatique et recherche opérationnelle (DIRO)/
Department of Computer Science and Operations Research
C.P. 6128, Succursale Centre-Ville
Montréal, Qc
Canada H3C 3J7

Email: kropf@iro.umontreal.ca

URL: http://citeseer.nj.nec.com/unger98simulation.html

Comments: Funded under EU Framework IV. A follow-on project WOSSYSTEM has been proposed. A related but not identical approach is Jini (Sun Microsystems, 1998).

References: Kropf *et al*, 1997

5 Computational Steering

Program steering has been defined as the capacity to control the execution of long-running, resource-intensive programs. This may include modifying the program state, managing data output, starting and stalling program execution, altering resource allocations etc. Dynamic steering requires the user to monitor the program or system state and have the ability to make changes. This could be through subroutine calls added as "instrumentation" (perhaps by an automatic tool such as SvPablo

http://www-pablo.cs.uiuc.edu/Project/SVPablo/SvPabloOverview.htm) or by interacting with the data structures in the code. An extensive survey of research in this area was carried out in November 1994 by Gu *et al*, 1994, however not many of the projects led to practical tools. In the intervening four years more progress has been made, and we describe just a few of the current projects here.

5.1 COVISE

Name: COVISE

Description: A tool for visualisation and interactive steering built as an AVS system that can be attached to executing jobs. A "creator" module allows attributes to be attached to data objects. These attributes can be accessed via menu entries and changed through "interactors". Displays in 3D are included with variable parameters. Upstream feedback is implemented to change the executable behaviour.

Systems:

Contacts: Dr. Ulrich Lang,
Computing Centre University of Stuttgart,
Visualisation Department,
Allmandring 30, Stuttgart, Germany

Email: lang@hlrs.de

URL: http://www.hlrs.de/organization/vis/covise/

Comments:

References:

5.2 CUMULVS

Name: CUMULVS

Description: Allows the scientist to modify a fixed set of parameters while using AVS to visualise the computational model. Implemented as a user-programmed AVS module. CUMULVS has been used to distribute CFD applications. Supports collaborative visualisation and simulation allowing several viewers to "plug in" and steer. A check-pointing facility allows cross-platform migration and heterogeneous restarts, so it may be of interest in meta-computing.

Systems: A middle layer between PVM and AVS.

Contacts: Distributed Computing Group,
Computer Science and Mathematics Division,
Oak Ridge National Laboratory,
P.O. Box 2008, Bldg 6012, MS 6367
Oak Ridge, TN 37831-6367

Email: cumulvs@msr.csm.ornl.gov

URL: http://www.epm.ornl.gov/cs/cumulvs.html
Available from NetLib http://www.netlib.org

Comments:

References: Geist *et al*, 1996

5.3 FALCON

Name: FALCON

Description: This is a monitoring system with low monitoring latency and perturbation. Steerable applications are developed through source-code modifications using Progress or Magellan (see below) and steering is assisted by the run-time system. FALCON captures many of the same things that a debugger would, and modifying variables enables you to steer your program while it is executing. Provides hooks to visualisation systems such as Iris Explorer. Can also monitor applications at the thread level, but a Cthreads package is required.

Uses DataExchange and PBIO for event transport in a heterogeneous environment and event filtering and processing.

Falcon, Progress and Magellan form part of a larger project to create "distributed laboratories" which is described on the Web pages.

Systems:

Contacts: Karsten Schwan,
College of Computing, Georgia Institute of Technology,
Atlanta, GA 30332-0280

Email: schwan@cc.gatech.edu

URL: http://www.cc.gatech.edu/systems/projects/FALCON

Comments: Not yet available for public release.

References: Gu *et al*, 1995

5.4 Progress

Name: Progress

Description: Steerable applications are developed through source-code modifications and steering is assisted by a run-time system. Progress assists scientists to develop steerable applications by instrumenting their code with library calls and using "steerable objects" that can be altered at run time. The latter include sensors, actuators, probes,

function hooks, complex actions and synchronisation points — in fact many of the same concepts found in VR applications. Has been used for MD simulations.

Systems:

Contacts: Jeffrey Vetter, Karsten Schwan,
College of Computing, Georgia Institute of Technology,
Atlanta, GA 30332-0280

Email: schwan@cc.gatech.edu, vetter@cc.gatech.edu

URL: http://www.cc.gatech.edu/systems/projects/Steering/

Comments: Uses the FALCON run-time system for on-line monitoring.

References: Vetter and Schwan, 1995

5.5 Magellan

Name: Magellan

Description: Derived from the Progress system. Extends the client and server steering models. Uses a specification language ASCL to provide commands for monitoring and steering using the same objects as Progress. The application code must still be instrumented. Has been used for MD simulations.

Systems:

Contacts: Jeffrey Vetter, Karsten Schwan,
College of Computing, Georgia Institute of Technology,
Atlanta, GA 30332-0280

Email: schwan@cc.gatech.edu, vetter@cc.gatech.edu

URL: http://www.cc.gatech.edu/systems/projects/Steering/

Comments: Uses the FALCON run-time system for on-line monitoring.

References: Vetter and Schwan (1997)

5.6 SciRun

Name: SciRun — Scientific Computing and Imaging

Description: A scientific problem-solving environment (PSE) which provides the ability interactively to guide or steer a running computation. The entire process of computation modelling, simulation and visualisation is built and executed within the PSE. SciRun was designed initially for multi-threaded shared-memory multiprocessors using C++ classes. A distributed-memory version is being produced and threading is now used to hide latency and perform other tasks.

Applications may be composed from new and existing components which are C++ classes describing geometries, meshes, fields, surfaces etc. Current applications are based on 3D tetrahedral unstructured meshes and SciRun defines the format of the data structure with which it interacts. A hexahedral mesh interface is being developed. The Diffpack, SparseLib++ and PETSc libraries are currently included. Components are linked in a dataflow network familiar to AVS Express users. All components support steering which is implemented in three distinct ways in the system:

- direct lightweight parameter changes — affect a running module;
- cancellation — when a parameter change cancels and re-starts a module;
- feedback loops — changes to parameters require other modules to be re-run.

Meta-computing is supported. SciRun aims to address the problems of interaction and integration of scientific simulation and visualisation in a distributed computing environment.

Systems: Uses AVS Express on an SGI for control and visualisation.

Contacts: S.G. Parker, D.M. Weinstein and C.R. Johnson
University of Utah, Salt Lake City, UT 84112, USA

Email: crj@cs.utah.edu

URL: http://www.sci.utah.edu/research/pse_fields.html

Comments: Funded from the NCSA PACI Alliance and venture capital. May become a commercial product in time.

References: Miller *et al*, 1998, Arge *et al*, 1999
See also http://www.sci.utah.edu/pubs/scirun_pubs.html

5.7 VASE

Name: VASE — Visualization and Application Steering Environment

Description: VASE presents an abstraction for a steerable program and offers tools that create and manage collections of steerable codes. VASE annotates existing Fortran code to create a high-level model of the application. Users therefore do not have to work at the source code level. Software developers must however annotate the existing code. Once this has been done CASE co-ordinates the execution of codes in a distributed environment under and SPMD execution model. A powerful C-like scripting language provides flexible support for data selection and steering during execution.

Systems: Sun SPARCstation, SGI Iris workstation, Cray Y-MP

Contacts:

Email:

URL:

Comments: Presented at SuperComputing '93. No known work on this project since 1994.

References: Jablonowski *et al*, 1993

6 Meta-computing Environments

A number of the tools and collections of software described in the previous sections are of use in a full meta-computing environment (Catlett and Smarr, 1992). In this section we identify projects which provide the full functionality of a meta-computing infrastructure.

6.1 GLOBUS

Name: GLOBUS

Description: In addition to the GLOBUS distributed middleware, part of the GLOBUS project also focuses on meta-computing using facilities at the GUSTO Consortium sites.

The Globus grid programming toolkit is designed to help application developers and tool builders overcome these obstacles to the construction of "grid-enabled" scientific and engineering applications. It does so by providing a set of standard services for authentication, resource location, resource allocation, configuration, communication, file access, fault detection, and executable management. These services can be incorporated into applications and/or programming tools in a mix-and-match fashion to provide access to needed capabilities.

High-level services include:

- MPICH-G and PAWS — grid-enabled MPI communications libraries. MPICH-G uses Nexus for heterogeneous communication (see previous entry);
- CC++ and HPC++ — parallel languages;
- grid-enabled libraries to provide uniform programming environment;
- remote-access and visualisation;
- DUROC and Nimrod — resource brokers;
- graphical status monitors.

The meta-computing directory service (MDS) maintains lists of resource objects in a distributed directory. Information can be updated from the Globus system, other information providers and tools and from applications. Information is provided dynamically to tools and applications. A lightweight directory access protocol has been developed. MDS tools include object class browser, explorer, various APIs and search tools and translators from GLOBUS object definition language.

Contacts: Ian Foster, Argonne National Laboratory, USA
Carl Kesselman, University of Southern California, USA

Email: foster@mcs.anl.gov, carl@isi.edu

URL: http://www.globus.org/

Comments: Kesselman provides a useful tutorial at URL http://www.npaci.edu/Training/NPACI_Institute98/Presentations/kesselman.
Also featured in the “Grids” Foster and Kesselman (1998).
A tutorial on GLOBUS was presented at SuperComputing '99.

References: Foster and Kesselman, 1997, Foster and Kesselman, 1998, Buyya, 1999

6.2 Legion

Name: Legion

Description: An integrated meta-system or "grid" system that has been deployed at a number of sites. It arose from an object-based software project at the University of Virginia beginning in 1993. The goal has been a highly useable efficient and scaleable system founded on solid principles. It is guided by work in object-oriented parallel processing, distributed computing and security. The group has wide experience of distributed computing systems.

Legion supports existing codes written in MPI and PVM, as well as legacy binaries. Key capabilities include:

- eliminating the need to move and install binaries manually on multiple platforms;
- providing a shared, secure virtual file system that spans all the machines in a Legion system;
- providing strong PKI-based authentication and flexible access control for user objects;
- supporting remote execution of legacy codes, and their use in parameter space studies.

Legion thus addresses the issues of scalability, programming ease, fault tolerance, security, site autonomy etc. It is designed to support large degrees of parallelism in application codes and manage the complexities of the physical system for the user.

Components include:

- method invocation service;
- file system;
- security system;
- context space directory services;
- resource management service;
- core-object management service.

These components link the computer operating system to the application codes.

A variety of diverse applications have been ported to Legion, e.g. CHARMM, ocean models, CCM, particle-in-cell codes, and several parameter-space studies. Web pages contain historical information about the project, documentation, discussion of the key features and a download facility.

Systems: Linux86 and alpha, Sparc, RS/6000, SGI, Alpha, Cray T90 and C90, HP. Legion has been run on Centurion, NPACI, DoD and NASA testbeds.

Contacts: University of Virginia

Email: legion@virginia.edu

URL: http://www.legion.virginia.edu

Comments: Based on earlier software called MENTAT. Work is in progress to create an “open” system that allows and actively encourages third-party development of applications, run-time library implementations and core system components. Legion can be downloaded from the Web page. A tutorial on Legion was given at SuperComputing '99.

References: Grimshaw *et al*, 1994a, Buyya, 1999 and additional publications at http://legion.virginia.edu/papers

6.3 LSF

Name: LSF — Load Sharing Facility

Description: The standard application resource management components of LSF have been described in Section 4. Of relevance to grid technology, the concepts are extended in the multi-cluster module which is able to

transfer work between locally managed or remote systems, e.g. to access particular software licenses. This can work over separately managed sites, e.g. multiple departments or divisions of large companies, computer centres supporting many sites, multiple co-operating organisations. It also has support for loosely connected sites with long distances or slow links and WAN with possible time differences. This maintains autonomy with each cluster having its own LSF administrators and policies, but negotiation to set up inter-cluster resource sharing.

Systems: Many including: Sun, HP, Compaq, SGI, IBM, Fujitsu, Hitachi, NEC, Sony and NT systems.

Contacts: Roland Richardson,
Platform Computing Ltd.,
Units 18/19, Intec 2, Basingstoke, Hampshire RG24 8NE

Email: info-europe@platform.com

URL: http://www.platform.com

Comments:

References:

6.4 MILAN

Name: MILAN — Meta-computing In Large Asynchronous Networks

Description: An ongoing research project aimed at making efficient use of distributed systems. The ultimate goal of the project is to build a software environment emulating a collection of virtual machines on a non-dedicated, unpredictable, distributed platform. The scope of the project is large and it encompasses a gamut of approaches from fundamental research through prototypes, up to working usable systems. It is the fundamental tenet of the research approach that only commodity components will be used. MILAN will provide software running in user space assuming standard hardware, operating systems, and compilers. Use is made of existing Web tools, many based on Java, for access to wide area resources. This means that authentication and security are not addressed specifically. Emphasis is placed on components for heterogeneous programming which are directed as separate research projects:

- Calypso/Aguda — tools to utilise heterogeneous networks for high performance parallel computing. The distributed machine is viewed as pseudo-SMP device. A very course decomposition of work (threads) is distributed to these nodes. Core algorithms for load

balancing are eager scheduling and a two-phase idempotent execution strategy;

- Chime — a compiler/ language based approach to heterogeneous computing (Windows NT) which is an extension of C++. This is similar to the OpenMP paradigm for programming SMP systems (Allan and Müller, 1999);
- Malaxis — another distributed shared memory package for Windows NT;
- Charlotte — provides a similar interface to Calypso and Chime but uses the Web as a computing platform. Java is used to run the remote threads;
- KnittingFactory — a Web middle layer in Java which provides the necessary meta-computing directory service and signals advertisements for machines to run threads of a job. It also includes point-to-point communication that is not possible via the Java applets.

KittingFactory aims to solve several problems with Web-based computing:

- how can Java applets find other members of the collaboration session;
- how to deal with the restrictions imposed by the Java security model;
- how to overcome the inability of applets to communicate directly.

Previous solutions have been either to rely on untrusted native code or to use a single forwarding agent. Untrusted native code can be used either as a Java native-method, or as an extension to a Web browser in the form of a plug-in. However, this has two drawbacks. First, it creates an insecure environment to execute foreign code. Second, it would require administrative effort that otherwise would not be needed.

The use of a single agent to forward message among applets also has several drawbacks. First, requires an HTTP server to run on the same machine as the initiating application, which limits the machines that can be used for this purpose. Second, every message of every application must be channelled through a single Java application. This clearly limits the scalability. Finally, locating a single routing agent for each collaborative application can be a daunting task.

Systems: Linux, Solaris and Windows NT.

Contacts: Z.M. Kedem, New York University, USA
P. Dasgupta, Arizona State University, USA

Email: ziv.kedem@nyu.edu, partha_dasgupta@asu.edu

URL: http://cs.nyu.edu/milan
http://milan.eas.asu.edu under Research Areas

Comments: Currently only small-scale research prototypes are available.

References:

6.5 NWIRE

Name: NWIRE

Description: Aims to provide a meta-computing infrastructure similar to GLOBUS. Each machine runs its own RMS and collections of machines are overseen by a meta-manager that is responsible for scheduling jobs within a domain. There is no central server, but meta-managers can communicate via a known list and poll for specific resources. Jobs may be run across domains via a “contract” mechanism. A differential service is maintained with minimal impact on local services. The use of CORBA for communication also means that local security mechanisms are maintained.

Systems:

Contacts:

Email:

URL:

Comments: Prototype only in 1999

References: Brooke *et al*, 2000

6.6 STA

Name: STA — Seamless Thinking Aid

Description: From the Centre for Promotion of Computational Science and Engineering (CSSE), part of JAERI, Japan. It is a Web-enabled Java-based environment that includes a number of tools for assisting parallel programming and using computers connected by networks.

The goal is to allow larger calculations and to couple applications with different memory or architectural requirements.

Editors, parallelising compilers, debuggers and performance tuning tools are able to exchange data in a seamless way. The kpx performance monitor is described in Allan *et al*, 1999.

In order to use a heterogeneous computing resource a modified version of MPI called STAMPI is provided. This is able to dynamically allocate child processes to computers, provides a varying number of message routers to optimise communications uses VSCM or CCMI for internal or external communication respectively and uses the MPI-2 standard programming interface.

STAMPI has been used to link together e.g. vector and RISC-based systems (NEC and IBM) in coupled 3D CFD and structural mechanics calculations for aircraft modelling.

Systems: STAMPI works on systems such as Fujitsu VPP, NEC SX, Cray, IBM SP, Hitachi SR2201, Fujitsu AP3000 and Intel Paragon.

Contacts: Hiroshi Koide,
CCSE, JAERI, 2-2-54,
Nakemeguro, Meguro, 153-0061, Japan

Email: koide@koma.jaeri.go.jp

URL: http://www.globus.org/mpi/related.html

Comments:

References: Imamura et al, 2001

7 Activities World-wide

A number of consortia of research groups world-wide are developing large-scale distributed computing systems and meta-computing systems. These are also in many cases the focus of the software development described in the rest of this report. Some of the projects are first described.

We do not discuss the underlying network testbeds, which include MREN, CalREN-2, SarTap, EMERGE, ESNet, NTON, Berlin and South German Gigabit Network etc.

Grid Fora have been established in the USA, Europe and the Asia-Pacific region. These fora maintain discussion lists and working groups for software developers and users. They are collaborating to develop software components and standards. In July 2000, during the iNET conference in Yokohama, Japan there was a meeting of GridForum, European Grid and Asia-Pacific Grid Forum. The meeting was devoted to the Global Grid Forum idea, which was also discussed during GridForum-4 meeting at MicroSoft's Headquarters in Redmond, WA. The idea is, briefly, to merge the three Fora into one body with three "chapters": N. American, European and Asia-Pacific.

7.1 The US Grid Forum

The Grid Forum is an informal consortium of institutions and individuals working on wide area computing and computational grids: the technologies that underlie such activities as the NCSA Alliance's National Technology Grid, NPACI's Metasystems efforts, NASA's Information Power Grid, DoE ASCI's DISCOM program, and other activities world-wide.

The Grid Forum is modelled, in many respects, on the Internet Engineering Task Force IETF, see http://www.ietf.org/ and focuses on the promotion of Grid computing via the documentation of "best practices" and "standards", with an emphasis on rough consensus and running code. The processes, under which the Grid Forum operates, are still being established.

So far the Grid Forum has selected a number of working groups and organised several workshops, including a “Birds of a Feather” meeting at SuperComputing ‘99. Working groups currently include:

- Scheduling Working Group (Sched-WG);
- Grid Information Service Working Group (GIS-WG);
- Security Working Group (Security-WG);
- Remote Data Access Working Group (Data-WG);
- Application and Tools Requirements Working Group (Apps-WG);
- Grid Performance Working Group (Perf-WG);
- Advanced Programming Models Working Group (Models-WG);
- Account Management Working Group (Accounts-WG);
- User Services Working Group (Users-WG).

International participation is encouraged.

Grid Forum Web pages are at URL http://www.gridforum.org/.

7.2 European Grid Forum

The European Grid Forum, Egrid, aims at fostering the co-operative use of distributed computing resources that are accessible via wide area networks. EGRID was formed in late 1999 and an organisational structure and a charter are currently under discussion. EGRID is an open forum, the community includes individuals from European research institutes, universities and companies working in the field of wide area computing and computational grids.

Egrid members come from both worlds: the application-oriented end-users and the system software developers.

In its early phase Egrid is meant as a discussion platform for interested parties in Europe. Multiple workshops are planned in the near future throughout Europe. It is planned to form working groups on the different subjects.

Egrid has now established a number of working groups that are collaborating with the US Grid Forum as follows:

- Data Storage and Management http://www.zib.de/merzky/EGrid/Data/
- Resource Management http://www.egrid.org/rm-wg/
- Programming Models
- Testbeds
- Performance Analysis

Egrid Web pages are maintained at URL http://www.egrid.org/.

7.3 Asia-Pacific Grid Forum

APGrid Forum Web pages are maintained at URL http://www.apgrid.org/.

7.4 NSF PACI

PACI is the Partnerships for Advanced Computational Infrastructure programme funded by the USA National Science Foundation Advanced Scientific Computing Division. PACI has established two national centres at the Universities of California at San Diego (NPACI) and Illinois (NCSA) which involve over a hundred academic institutions in collaborative HPC projects. This costs around $64M per year.

The core idea of the Alliance-PACI project is to establish a leading-edge computational grid, termed the “National Technology Grid”, linking partner sites including NCSA and SDSC. Another project is termed the “National-Scale Machine Room”.

7.4.1 NPACI

The National Partnership for Advanced Computational Infrastructure is based at the SDSC, University of California San Diego. It includes CalTech, U Texas, U Michigan, UC Berkeley, UC Davis, UCLA, UC Santa Barbara, U Houston/Keck, U Maryland and U Washington.

NPACI are developing the Legion software.

7.4.2 NCSA

The National Computational Science Alliance (the Alliance) is based at the University of Illinois at Urbana-Champagne. It includes OSC, UIUC, U Kentucky, UI Chicago, U Boston, Rice, Stanford, Princeton, MIT, U Wisconsin, U Minneapolis plus Allstate Insurance, American Airlines, AT&T, Caterpillar, Dow Chemical, Eastman Kodak, Eli

Lilly, FMC, F.P. Morgan, McDonnell Douglas, Motorola, Phillips Petroleum, Schlumberger Ltd., Sears, Shell Oil, Tribune Co., and United Technologies.

At SuperComputing '99 (Portland, Oregon, USA), Alliance research teams showed how the Alliance is developing a prototype virtual workspace, called the NCSA Access Grid that can be used for collaborative scientific research, distance education, and remote meetings and seminars. Some demonstrations utilised the Access Grid, connecting to remote locations either on the SC exhibit floor or in other cities. The Alliance also showed how its work in developing a national-scale technology Grid is enabling science and will exhibit new computational tools and infrastructure that are being integrated into the Grid. Projects of the Alliance Partners for Advanced Computational Services were also demonstrated.

NCSA are developing the GLOBUS software.

7.4.3 GUSTO Consortium

GLOBUS distributed and meta-computing concepts are being tested on a global scale by participants of the Globus Ubiquitous Supercomputing Testbed Organization (GUSTO). This is an agreement between PACI sites to develop a meta-computing testbed.

GUSTO is based at Argonne National Laboratory and the University of Southern California and started in 1997. It currently spans over twenty institutions and includes some of the largest computers in the world. Both dedicated and commodity Internet services are used.

GUSTO is further described in the GLOBUS Web pages at URLs http://www.globus.org/ and http://www.globus.org/research/testbeds.html.

7.4.4 US Data Analysis Grid

This project is planned to run during 2001-2005 to enable collaborative analysis of experimental data coming from the CERN LHC (see GriPhyN below). It is a joint proposal of the CMS/US ATLAS/LIGO experimental groups, involving individuals from Florida, FNAL, North-eastern, and Caltech. The project aims to build an ensemble of Tier 2 Centres, well co-ordinated with Tier 1 Centres (see Introduction). It includes three projects on a shared network infrastructure.

7.5 GriPhyN

US physicists from the CMS and Atlas experiments at the Large Hadron Collider (LHC) at CERN, the LIGO experiment and the Sloan Digital Sky Survey received $11.9M funding in October 2000 from the NSF for the **Gri**d **Phy**sics **N**etwork. The proposal

was motivated by the fact that, whilst all four experiments have been approved for running and have received substantial construction funds, the computing needs of the US based physicists who will be analysing the data have barely been addressed.

The scale of the required computing resources is enormous: each of the four experiments requires enormous computing capacity and has a massive dataset (up to Petabyte size) that must be accessed by a widely distributed (international) user base served by networks having bandwidths that vary by orders of magnitude. Clearly, a computing solution for the four experiments requires dedicated, large-scale funding.

All four experiments receive considerable federal support, approximately $1.4 billion by the time the experiments come online (SDSS begins data taking in 2000, LIGO starts in 2002 while Atlas and CMS commence around 2005).

Whilst the LHC, LIGO and SDSS experiments plan to generate massive datasets, they are not unique. It turns out that many scientific and financial endeavours involve the rapid generation and analysis of large datasets. These include:

- The Earth Observing System Data Information System (EOSDIS) (3 PB by 2001);
- The Human Brain Project. Time series of 1 Terabyte scans of the human brain, generating of the order of a Petabyte of data in a short period of time;
- The Human Genome Project;
- Automated astronomical scans Geophysical data;
- Satellite weather image analysis, where chaotic processes are studied;
- Point of sale receipts, in which patterns of consumer spending are tracked;
- Banking records, which are analysed for spending cycles or unusual transactions that may relate to illegal activities.

For LIGO, CMS and Atlas the one thing that stands out is the massive dataset that must be managed and accessed. While huge CPU resources must be used to analyse this data, the overwhelming problem is posed by the data itself. Accordingly, we prefer to think of the resources as comprising a Data Grid.

It is assumed that each experiment will have one (or more) so-called Tier 1 computing centres within the US, e.g. Fermilab for CMS, Caltech for LIGO. For LHC these Tier 1 centres might have roughly 20% of the CPU and storage capacity available at CERN. The LHC Tier 1 centres are expected to have about 10^5 SpecInt95s in compute capacity and several PB in storage, along with perhaps several hundred TB in disk cache. The corresponding Tier 1 compute site for LIGO will be about an order magnitude smaller.

Further information on the GriPhyN project, related work and a white paper can be found at URL http://www.griphyn.org/.

7.6 NASA IPG

The NAS "Information Power Grid" project is designed to implement seamless access to resources between NASA sites and a few NPACI sites. This followed from a number of workshops and reviews in autumn 1997. It grew from the Advanced Computing Networks and Storage (ACNS) and Computation Aerospace (CSA) programmes at NASA. Goals of the project are to provide access to all resources for a single large simulation and to include virtual reality and access to large-scale data stores. A number of middleware implementations and demonstrator applications are being developed in phase II of the project starting in 3Q99 and continuing until 3Q04. The full project was planned to develop over a seven-year time scale and cost around $63M per year.

Further information available the Web: http://www.nas.nasa.gov/Groups/Tools/IPG.

There is also an "Information Power Grid Hotlist" from the NASA Web site, which includes information on distributed computing, meta-computing and Java http://www.nas.nasa.gov/NAS/Tools.

7.7 ASCI PSE

A major activity driven from Los Alamos National Laboratory is the Problem Solving Environment (PSE) for the Accelerated Strategic Computing Initiative (ASCI). It also includes Lawrence Livermore National Laboratory and Sandia National Laboratory. Information on the ASCI projects is available from URL http://www.lanl.gov/asci. Components of PSE relevant to network-based computing are:

? High Performance Computing Support:

High Performance Computing Support's (HPCS) role is to provide a supporting infrastructure between platforms and applications for effective high-end application execution and tera-scale data management:

- Archival storage;
- Scientific data management;
- High speed interconnect;
- Scalable I/O;
- Distributed resource management;
- Platform and service integration.

? Tri-Lab Networking:

Designing and implementing this wide/local area network architecture that enables uniform, transparent, and efficient distributed classified and unclassified computing among the three defence programs laboratories continues to be a formidable technical and administrative task that involves every aspect of networking:

- Tri-lab connectivity;
- New secure service and encryption upgrades;
- Network modelling.

? Distributed Computing Environment:

The purpose of the Distributed Computing Environment team is to provide a common set of key core services throughout the ASCI community, common both inter-organisationally (within a single lab) and between the ASCI computing environments at each of the three laboratories:

- Production DCE core services;
- Tri-lab distributed services/support;
- DCE secure web pilot;
- Tri-lab DFS deployment;
- DFS/HPSS integration testing;
- Expanded desktop deployment;
- Distributed objects;
- Assessment study of PKI and DCE;
- ASCI application support.

The debugging and visualisation activities will be described in a separate report (Allan *et al*, 1999).

7.8 iGrid and StarTap

iGrid is a collaboration between University of Illinois at Chicago, Indiana University, Tokyo University and Keio University with the aim of “empowering global research community networking”.

iGrid is part of the StarTap initiative. StarTap (Science, Technology, And Research Transit Access Point) is a persistent infrastructure, funded by the National Science Foundation Advanced Networking Infrastructure and Research division, which is part of the Computer and Information Sciences and Engineering (CISE) directorate, to facilitate the long-term interconnection and interoperability of advanced international networking in support of applications, performance measuring, and technology evaluations. The StarTap anchors the international vBNS connections program.

Physically, StarTap connects with the Ameritech Network Access Point (NAP) in Chicago, as does the vBNS and other high-speed research networks. It enables traffic to flow to international collaborators from over 100 U.S. leading-edge research universities and supercomputer centres that are now, or will be, attached to the vBNS or other high-performance U.S. research networks.

StarTap is documenting the international collaborations it helps foster. These applications are among the most computation-demanding and/or data-intensive today, and serve as test cases for the various network features StarTap deploys. Not only do these science applications help promote the exciting research being carried out worldwide, but they serve as a reference for others interested in computational science and engineering problems, or in the computer and communication technologies used to help solve them.

Major demonstrations have been held at SuperComputing '97, Alliance '98, iGrid '98 and one is planned at iGrid 2000. In the past these have included meta-computing using the Globus software and collaborative virtual reality using the CavernSoft software. In an example of the latter engineers at Caterpiller Inc. at NCSA were able to demonstrate a new tractor design to customers in Germany using an Immersadesk facility at GMD, Bonn.

Very informative Web pages are maintained at http://www.startap.net/.

7.9 JAERI/STA

The Center for Computational Science and Engineering (CCSE) was established within the Japanese Atomic Energy Research Institute (JAERI) in 1995. It is playing a leading role in the research and development of computational science and engineering in Japan. This is continuing the work started in the Science and Technology Agency (STA) and will continue to satisfy their requirements.

Principal strands of the research and development at CCSE are:

? development of parallel basic software;

? development of parallel algorithms;

? development of parallel processing tools;

? studies of numerical simulations on complex phenomena by particle and continuum methods;

? new computer architectures.

These feed into applications of special interest to the STA laboratories and Japanese Universities and software is available on JAERI and STA computers. Fortran 90 and MPI is used and the software is portable across many platforms, including Intel Paragon, Fujitsu VPP, Hitachi SR2201, Fujitsu AP3000, IBM SP, NEC SX4, Cray T90.

A specific deliverable, relevant to this report, is the Metacomputing environment STAMPI and its associated tools (see 6.6).

7.10 METODIS

The MEtacomputing TOols for DIstributed Systems project is an EU-funded project involving a collaboration between HLRS (Stuttgart, Germany), CRIHAN (France), Pallas (Bonn, Germany), DASA (Germany) and Aerospatiale (France). The aim is to build an ATM-based meta-computing system for aerospace applications. Tools include COVISE, PACX-MPI and VAMPIR.

This project runs alongside the UNICORE project to provide a seamless interface for submitting jobs to German regional supercomputers.

A large-scale implementation of these tools was demonstrated at SuperComputing '99 and involved Stuttgart, Manchester, Pittsburg, San Diego and Tskuba.

7.11 Berkeley NOW and NOW II Projects

Network of Workstations (NOW) is a research system that started around 1995 and was designed as a dedicated high-performance platform. It consists of Sun UltraSparc systems linked by Myrinet. Key software includes GLUnix (middleware for queuing, gang scheduling and other resource management), xFS (scaleable parallel file system) and Active Messages (low-latency communications layer based on Thinking Machines CM-5 software and achieving 10 µs latency and 40 MB/s bandwidth).

NOW II extends the original project to provide multi-user access. A software framework WebOS is being constructed which will provide incrementally scaleable geographically aware Web service.

In the NOW and NOW II projects active messages are used for fast communication. An asynchronous communication mechanism is provided which uses a user-defined message handler invoked via an interrupt mechanism. This is very similar to the mechanism provided by Intel in their NX/2 operating system. This is implemented as GAM (Generic Active Messages).

For further information see Web URL http://now.cs.berkeley.edu/. See also Hwang and Xu (1998) for an introduction to the project and comparison with other systems.

7.12 Illinois HPVM Project

The goal of this project, which started around 1997, is to develop shared controllable high-performance components for distributed systems. This includes predictable communication, management of heterogeneity, stable performance models and adaptive resource management. Virtual reality is supported using high-speed networking and an ability to manipulate large data sets. A variety of compute and networking components are being evaluated.

Software includes Illinois Fast Messages with APIs to MPI, SHMEM and Global Arrays, Dynamic co-scheduling resource management, FM-QoS heterogeneous communication layer and front-end administration tools using Java.

Further information is available from Web URL http://www-csag.ucsd.edu/projects/hpvm.html.

7.13 Real-World Computing Partnership

A project is under way to build distributed systems with shared resources. For further information contact:

K. Kubota,
Real-World Computing Partnership,
Tsukuba, Ibaraki 305-0032, Japan

Web URL http://www.rwcp.or.jp/.

7.14 DoE2000 Programmes

7.14.1 NC

The DoE2000 National Collaboratories are developing a set of tools and capabilities which will permit scientists and engineers working at different US Department of Energy and other facilities to collaborate on solving problems as easily as if they were in the same building. The programme supports research into the tools required by a virtual laboratory: collaborative tools; information surety (authentication plus security); and high-performance networking and one pilot implementation of these tools in partnership with other DoE programmes (e.g. ASCI).

7.14.2 ACTS

The Advanced Computational Testing and Simulation programme is developing an integrated set of algorithms, software tools and infrastructure that will enable computer simulation to be used in place of experiments when real experiments are too dangerous, expensive, inaccessible or politically unfeasible.

7.15 DARPA QUORUM

The QUORUM programme is developing the technologies which will allow end users to achieve predictable and controllable end-to-end quality services for critical defence computing needs in a global heterogeneous distributed computing environment. These technologies will be integrated into a global operating system so that this resource pool appears to end users as a single computing platform and demonstrated on key defence problems in shipboard computing and command and control.

7.16 US Defense Modelling and Simulation Office

The Defense Modelling and Simulation Office (DMSO) High-Level Architecture (HLA) project offers standards for an inter-operability framework enforcing re-usability and share-ability of objects and components based on new technology standards. The aim is to couple disjoint simulations on HPC systems and commodity clusters.

There are a number of parallel and distributed simulation tasks at Metron Inc. that are sponsored by the High Performance Computing Modernization Office (HPCMO) through its Common HPC Software Support Initiative (CHSSI) project. These include: (i) modifying NSS to execute with high-performance on massively parallel machines; (ii) extending the IMPORT simulation language and SPEEDES modelling framework to provide easier-to-use programming interfaces; and (iii) developing a High-Level Architecture (HLA) Run-Time Infrastructure (RTI) for supporting interoperability between different simulations on high-performance computers. These efforts are being co-ordinated by the Naval Research Laboratory (NRL), Space and Naval Warfare Systems Command (SPAWAR), and the DMSO.

Another project sponsored by DMSO at Syracuse University is WebHLA.

The DMSO Web site is at http://www.dmso.mil/public/.
The Metron Web site is at http://www.ca.metsci.com/.

7.17 US National Scalable Cluster Project

NCSP is developing a prototype meta-computing system including three university clusters in Illinois at Chicago, Maryland at College Park and Pennsylvania. Goals are to develop software and demonstrate scalable clustered computing enabling data transfer between geographically remote sites.

vBNS (very Broad Network System) is used to construct a fast network. It uses ATM (Asynchronous Transfer Mode) protocols to achieve transfer speeds, sufficient to link nodes in local and wide area computing clusters, with the power to transfer Terabytes of data within minutes.

Other activities include data mining, data warehousing and medical supercomputing.

Project DataSpace is a five-year project that started in 1999 with the goal of establishing protocols and standards for high performance and distributed data mining. Protocols for mining distributed data were demonstrated at SuperComputing '99 and it was established that these protocols are effective for distributed workstation clusters connected with high performance networks (super-clusters) and with commodity networks (meta-clusters).

Contact: Prof. R. Hollebeek,
University of Pennsylvania,
Department of Physics and Astronomy,
209 So. 33rd St., Philadelphia, PA, 19104, USA
hollebeek@nscp.upenn.edu. Web URL is http://nscp.upenn.edu/.

7.18 Waseda University Parallel and Distributed Computing Environment

The Parallel and Distributed Computing Environment Project is a project supported by the Japanese government through the Japan Society for the Promotion of Science. Its objective is to develop a parallelising restructuring compiler and related tools for parallel and heterogeneous distributed computing environment. The project puts equal emphasis on both practical and theoretical sides. To pursue the project, they built a network of high performance computers as a research infrastructure. The final goal of the project is to build a "super Grid", which can be interconnected to an international Grid.

7.19 Particle Physics Data Grid (PPDG)

Several data grids are being constructed for analysis of experimental results, which will come from the CERN LHC (see GriPhyN above).

PPDG is a DoE/NGI funded initiative involving US High Energy Nuclear Physics US laboratories FNAL, BNL, ANL, LBNL, SLAC and JLAB, together with Caltech, CACR, SDSC, and University of Wisconsin. PPDG aims to exploit expertise and existing tools for distributed data management; Globus, SRB, Condor matchmaking etc. PPDG will develop, acquire and deliver vitally needed Grid-enabled tools for data-intensive requirements of particle and nuclear physics, in general and the High Energy and

Nuclear Physics (HENP) community, in particular. The PPDG Web site is at http://www.ppdg.net/.

There will be a DoE IT2/SSI project building on the PPDG work.

7.20 China Clipper Project

This is a US joint project between Argonne National Laboratory (ANL), Lawrence Berkeley National Laboratory (LBNL) and Stanford Linear Accelerator Center (SLAC). It focuses on developing technologies for widely distributed data-intensive applications, mostly for particle physics experiment analysis. Software used includes a distributed parallel storage system and GLOBUS on the accessible networks. As well as demonstrating the feasibility of high data transfer rates to participating sites the project has developed network instrumentation, optimisation and debugging tools.

Work in the Clipper project is now being extended in the US Particle Physics Data Grid (PPDG).

There is further information on the China Clipper project at URL http://george.lbl.gov/Clipper/.

8 Collaborative Working and Distance Education

Visualisation systems nowadays provide a rich set of functions to read, filter, map and render data. The dataflow systems such as AVS/Express (AVS UK Ltd) or Iris Explorer (Nag Ltd) not only do that but also exhibit a flexible framework for composing visualisation applications out of a network of component parts. These systems are in use in industry, academic institutions and government organisations.

In certain organisations, mainly industrial ones, dataflow visualisation systems are used for seriously large and complex 3D problems resulting from the increased size of problems that can be simulated on today's high performance computers. Understanding the 3D scenes that result from this process can present a problem. The viewing operations usually supported are data-centred and involve manipulating the data to find features of interest within the data. Exploration of complex datasets using such mechanisms can become a tedious task and in consequence the user may never find that unique perspective on the data which affords the crucial insight being sought.

The work described here aims to harness the user-centred navigation and interaction capabilities of virtual environment (VE) systems to exploit the exploratory skills that the user already possesses. This will provide a more interactive and responsive environment for data visualisation, allowing the user to observe subtle but important effects and new relationships within the data.

8.1 Distributed, Collaboratory Experiment Environments (DCEE) Programme

The DCEE programme was funded by US Dept. of Energy, Energy Research Division, Mathematical, Information, and Computational Sciences office under contract DE-AC03-76SF00098 with the University of California, and ended in February 1997. Its scope was extended in other directions using grid technology within the DoE2000 Programmes.

As a program, DCEE has had several major successes:

1. Scientific collaboratories represent a new way of doing science, and many aspects of the traditional characteristics of "real" laboratory environments are changed or missing. DCEE has successfully developed tools and techniques supporting widely distributed collaboration, and has introduced them into all of its testbed environments. The testbeds represent a wide spectrum of scientific environments: medium-scale environments (tens of scientists and graduate students) in the PNL Environmental and Molecular Sciences Laboratory, the ANL Advanced Analytical Electron Microscope, and the Berkeley Lab Advanced Light Source; and a large-scale science experiment (hundreds of scientists and graduate students) of in the General Atomics D-IIID Tokamak Fusion facility.

2. DCEE clearly identified, and in some areas started to develop, integrate, and/or examine alternative approaches for, the technologies needed to provide the basic remote human and remote instrument capabilities to support remote scientific collaboration: distributed, cross-platform electronic notebooks are combinations of distributed publishing and data management systems, and interfaces to laboratory data, that allow remote scientists to collect, catalogue, and share results and insights new multi-party data communications protocols provide the means to disseminate data and control in widely distributed systems so that scientists can effectively engage in personal and group communication and share data from instruments open, secure, and easily administered global file systems are needed to provide transparent data sharing, and integration with local computing facilities flexible and capable control arbiters are needed to manage the variety and complexity of distributed resources in scientific collaboratories decentralised, public-key security infrastructure provides protection for all of the collaboratory components while at the same time allowing the parties directly responsible for controlling access to specific data or instruments to easily specify and enforce their own use-conditions

3. Distributed collaboratories represent a new paradigm for human-human and human-instrument interaction. The program has started to examine the (sometimes

considerable) sociological issues of distributed collaboration and instrumentation, and the potential use and utility of various "shared space" / VR approaches to distributed scientific collaboration that attempt to more closely mimic direct contact / real presence interactions.

Some individual testbed projects within DCCE include:

- Argonne National Laboratory: Electron microscopy, physics and virtual reality;
- Lawrence Livermore National Laboratory, Princeton Plasma Physics Lab, Oak Ridge National Laboratory, and General Atomics: Remote control for fusion;
- Pacific Northwest National Laboratory: Remote collaboration for environmental molecular sciences;
- University of Wisconsin-Milwaukee and Lawrence Berkeley National Laboratory: Remote SpectroMicroscopy at the Advanced Light Source;
- UC Santa Barbara: Multicast data communication;
- Lawrence Berkeley National Laboratory: Electronic notebooks;
- Lawrence Berkeley National Laboratory: Security architectures for open distributed laboratory systems;
- General Atomics: Sociological aspects of remote collaboration.

These projects and related links are explained on the Web page http://www-itg.lbl.gov/DCEEpage/DCEE_Overview.html

8.2 NCSA Access Grid

The Access Grid is a collaborative virtual workspace that brings people together in real time. Collaborative sessions on the Grid could include scientific research, distance education, or remote training. Grid entry points can be as simple as a desktop computer or as sophisticated as a large format multimedia display system.

8.3 Emory University CCF

CCF (Collaborative Computing Frameworks) is a suite of software systems and tools, communications protocols, and methodologies that enable collaborative, computer-based, co-operative work. CCF constructs a virtual work environment on multiple computer systems connected over the Internet to form a collaboratory. Participants may interact, simultaneously access/operate computer applications, access data repositories or archives, collectively create and manipulate documents and spreadsheets, perform computational transforms, and conduct a number of other activities via tele-presence. CCF is an integrated framework that consists of multiple co-ordinated infrastructural elements, each of which provides a component of the virtual collaborative environment. A prototype of a complete collaboration system was exhibited at SUPERCOMPUTING'99 with live demonstrations, and includes an underlying reliable multicast protocol suite, application sharing and X-multiplexer systems, shared dataspace and filesystem, the computing harness, and the clearboard, audiotool, multiway chat and video-conferencing tools.

8.4 BioCoRe ¾ Biological Collaborative Research Environment

A number of new collaboratory and interactive/steered molecular dynamics projects are being developed at the Theoretical Biophysics Group at NIH Resource for Macromolecular Modeling and Bioinformatics at the Beckman Institute for Advanced Science and Technology University of Illinois at Urbana-Champaign.

BioCoRe is being designed as a tool-based solution to all of the collaborative issues in structural biology.
See http://www.ks.uiuc.edu/Research/biocore/localServer/announce.shtml.

This network-centred meta-application is intended to:

- improve collaboration between biomedical researchers located at the same or geographically distant sites; facilitate the transparent use of and communication between Resource and third-party programs, tools, and databases;
- allow researchers to share information, computational and data-storage resources; enable scientists to interact in both synchronous and asynchronous fashion with each other or the modelling tools through a common infrastructure;
- enable scientists to initiate new collaborations through its communication interface and to reduce the need for travel between collaborating research groups.

It is particularly aimed at the four main areas of: a “workbench” with analysis tools, data sharing, resource allocation, simulation control and interactive MD; a “notebook” with record keeping, mentoring and monitoring; a “conference” platform for communication, visualisation, control, audio/video features and training; and “documents” support for reports, publications and programs.

8.5 Environmental Molecular Science Collaboratories

The EMS Laboratory at Batelle Pacific Northwest National Laboratory (PNNL) are developing a number of tools for building collaboratories. These currently include Eccé, CORE 4.0 and Notebook 4.11.

Eccé is the Extensible Computational Chemistry Environment Project is developing a suite of software tools built around an object-oriented chemistry data model and an object-oriented database. The ecce data model will facilitate the integration of multiple applications beneath a unifying graphical user interface that provides a common look-and-feel.

CORE is a Collaborative Research Environment - see BioCoRe above.

For further information on these and related projects see http://www.emsl.pnl.gov:2080/docs/collab/.

8.6 MANICORAL ¾ Multimedia and Network in Co-operative Research and Learning

MANICORAL aims to address the problems of networked scientific meetings. It was an EU FPIV Telematics project RE1006.

8.6.1 Summary

Funded under the Telematics for Research initiative of the Telematics Application programme, the general objectives of the project were to explore, develop, implement and evaluate a Computer Supported Co-operative Work (CSCW) system among the members of a geographically distributed European research group. The project finished at the end of 1997. The consortium formed to undertake the project was multidisciplinary and consisted of

- AFRICAR (Altimetry for Research in Climate And Resources), an end-user research consortium of geoscientists investigating the expoitation of radar altimetry data;
- a group studying Human Communications, Co-operation and Cognition (HCCC) whose members have backrounds in humanistic and social sciences and human computer interaction;
- CSCW technology providers.

Extensive experience indicates that provision of technology alone does not however lead to effective solutions. A prime purpose of MANICORAL was to enable a long term study of how a community evolves; how technology may enable new ways of collaboration and new ways of dealing with scientific data; how scientific work may be enhanced and how technology may change the roles among community members.

8.6.2 Key technical developments

? Use, effects, and requirements of CSCW technology:
  - generate improved understanding of how groups collaborate and co-operate in a distributed context;
  - develop a conceptual framework and methods which will serve as a basis for developing user-driven requirements and functional specifications.

? Specific CSCW technology development and augmentation:
  - develop distributed co-operative visualisation system;

- develop improved and uniform access to distributed domain data and processing software.

To find out more about the project, consult the web page http://www.bitd.clrc.ac.uk/Activity/Manicoral or http://www-geomatics.tu-graz.ac.at/mgqi/manicoral/home_textonly.html or contact Julian Gallop j.gallop@rl.ac.uk.

8.7 CAVERNSoft

CAVERN, the CAVE Research Network, is an alliance of research and industrial institutions equipped with CAVEs, ImmersaDesks, and high performance computing resources, interconnected by high-speed networks to support collaboration in design, training, education, scientific visualisation, and computational steering — using virtual reality. Supported by advanced networking on both the national and international level, CAVERN focuses on tele-immersion — the union of networked virtual reality and video in the context of significant computing and data mining.

CAVERNsoft is the systematic software architecture for CAVERN. Being developed at the Electronic Visualization Laboratory (EVL) at the University of Illinois at Chicago, CAVERNsoft is designed to enable the rapid construction of tele-immersive applications; to equip previously single-user applications with tele-immersive capabilities; and to provide a testbed for research in tele-immersion. The outcome of this research includes new techniques for network quality of service; database access for the recording and intelligent querying of tele-immersive sessions; collaborative information visualisation; and mediating time and distance in tele-immersion.
The home page of CAVERN is at URL http://www.evl.uic.edu/cavern/ and examples of its use can also be found via http://www.startap.net/.

8.8 OSC Gateway

OSC Gateway is a collaboration between Ohio Supercomputer Center and the US Department of Defense MSRCs and NPAC. Called “Gateway” this project uses the Web to create a problem-sharing environment for DoD scientists and engineers that enables secure and seamless access to high performance resources. The project was demonstrated at SuperComputing ‘99 with other partners on the show floor and at OSC in Columbus and ASC MSRC in Dayton.

8.9 Pittsburgh Supercomputing Center

This Collaboratory project facilitates distributed software development and advances scientific research by bringing modern computing tools to bear on the productive

activities of collaborating scientists. We will show live demonstrations of tools used by our collaborators in the area of code/ document management, remote debugging, person-to-person conferences, and application sharing.

8.10 Diesel Combustion Collaboratory

DCC is a pilot project to develop and deploy collaborative technologies for combustion researchers through the US DoE National Laboratories, academia and industry. This is part of the DoE2000 framework. The result is to produce a problem-solving environment for combustion research.

The requirements of the collaborators are:

1. share graphical data easily using desk top workstations;
2. discuss modelling strategies and quickly exchange model descriptions between groups;
3. archive collaborative information in a Web-accessible electronic notebook;
4. utilise a distributed execution management system to run combustion models at widely separated locations;
5. quickly analyse experimental data and modelling results in a Web-accessible format;
6. video conference for one-on-one collaborations and group meetings using desktop workstations.

Internet-based DCC tools therefore include: a distributed execution management system for distributed workstations and supercomputers; web-accessible data archiving capabilities for sharing graphical data; electronic notebooks and shared workspaces for facilitating collaboration; visualisation of combustion data using an image library; and video-conferencing and data-conferencing at remote sites. Security and efficiency are key aspects of the collaborative tools.

To implement these tools a framework must be built to provide a seamless distributed computing service. The Product Realization Environment (PRE) developed at Sandia National Laboratory is used (Whiteside *et al*, 1998). PRE is built on top of CORBA (Yang and Duddy, 1996), which acts as an integration framework. The architecture of PRE consists of seven major pieces including uniform data objects and transport, a trading service, security, a conversion broker, integrated applications and user interfaces. Applications to be used in PRE must be “wrapped” and made into re-usable components.

The project is led by C.M. Pancerella, L.A. Rahn and C.L. Yang at Sandia National Laboratory. A paper giving further information was presented at SuperComputing '99 (Pancerella *et al*, 1999).

9 References

R.J. Allan and C.J. Müller
Shared-Memory Programming Paradigms: edition 1, Parallel Application Software on High-Performance Computers
CLRC Daresbury Laboratory, Daresbury, UK (1999)

R.J. Allan, R.R. Ward and M.C. Goodman
Survey of Parallel Performance Tools and Debuggers: edition 1, Parallel Application Software on High-Performance Computers,
CLRC Daresbury Laboratory, Daresbury, UK (1999)

P. Arbenz, W. Gander and M. Oettli
The Remote Computational System
Lecture Notes in Computational Science 1067 pp. 662-667 (Springer, 1996)

E. Arge, A.M. Bruarset, H.P. Langtangen (eds.)
Modern Software Tools for Scientific Computing (Birkhäuser, 1997) ISBN 0-8176-3974-8 and 3-7643-3974-8

I. Banicescu and H. Unger
Running Scientific Computations in a Web Operating System Environment in "2nd Int. Workshop on Embedded HPC and Applications" at 11th IEEE Int. Parallel Processing Symposium (1997)

I. Banicescu and H. Unger
Running Scientific Computations in a Web Operating System Environment
in "High Performance Computing 1999" A. Tentner (ed.) (Society for Computer Simulation International, 1999) ISBN 1-56555-166-4 pp. 356-362

D. Bernholdt, G.C. Fox, W. Furmanski, B. Natarajan, H.T. Ozdemir, Z.O. Ozdemir and T. Pulikal
WebHPC ¾ an interactive Programming and Training Environment for High Performance Modelling and Simulation in Proc. DoD HPC'98 Users' Group Conf. (Rice University, Houston, TX, June 1998) available from URL http://www.npac.syr.edu/iwt98/pm/documents

W.J. Bolosky
Operating System Directions for the next Millenium (Microsoft Research, 1999) http://research.microsoft.com/sn/Millenium/mgoals.html

J.M. Bradshaw
Software Agents (AAAI Press, the MIT Press, Menlo Park, CA, 1997)

T. Brecht, H. Sandhu, M. Shan and J. Talbot
Towards world-wide Supercomputing 7th European Workshop on System Support for World-wide Applications (1996)

B. Brewington, R. Gray, K. Moizumi, D. Kotz, G. Cybenko and D. Rus
Mobile agents in distributed information retrieval
in Intelligent Information Agents, chapter 12. M. Klusch (ed.) (Springer-Verlag, 1999) To appear as ISBN 3-540-65112-8.

J. Brooke, S. Pickles, F. Costen and S. Ord
Using metacomputing to process scientific data
in Proc. of The 2nd Intl. Workshop on Next Generation Climate Models for Advanced High Performance Computing Facilities, Toulouse, France, Feb 23 -25, 2000
Available from http://www.csar.cfs.ac.uk/staff/brooke/

G. Brose, A. Vogel, K. Duddy
Java Programming with CORBA: Advanced Techniques for Building Distributed Applications, 3rd Edition
2001, Wiley, ISBN 0-471-37681-7

R. Buyya (ed.)
High Performance Cluster Computing: Architectures and Systems, Volumes 1 and 2
(Prentice Hall, NJ, USA, 1999)
http://www.phptr.com/ptrbooks/ptr_0130137847.html
http://www.phptr.com/ptrbooks/ptr_0130137855.html
For more information, please see:
http://www.dgs.monash.edu.au/~rajkumar/cluster

W. Caripe, G. Cybenko, K. Moizumi and R. Gray
Network awareness and mobile agent systems
IEEE Communications Magazine 36 (1998) pp. 44-49

N. Carriero and D. Gelernter
Coordination Languages and their Significance
Communications of the ACM 35 (1992)

L. Cardelli and R. Davies
Service Combinators for Web Computing
System Research Center Technical Report (Digital Equipment Corp. 1997)

K. Czajkowski I. Foster, N. Karonis, C. Kesselman, S. Martin, W. Smith and S. Tuecke
A resource management architecture for meta-computing systems
Proc. IPPS/SPDP '98 Workshop on Job Scheduling Strategies for Parallel Processing, 1998.
Available at http://www.isi.edu/~karlcz/

G. Fox, W. Furmanski, M. Chen, C. Rebbi and J. Cowie
WebWork: Integrated Programming Environments Tools for National and Grand Challenges NPAC Technical Report SCCS-0715 (Syracuse University, USA, 1995)

H. Casanova and J. Dongarra
NetSolve: A Network-enabled Server for solving Computational Science Problems Int. J. Supercomputer Applications and High-performance Computing 11 (1997) pp. 212-223

H. Casanova and J. Dongarra
NetSolve's Network-enabled Server: Examples and Applications IEEE Computational Science and Engineering 5 (1998) pp. 57-67

C. Catlett and L. Smarr
MetaComputing Comm. ACM 35 (1992) pp. 44-52

J. Czyzyk, M. Mesnier and J. Moré
NEOS: The Network Enabled Optimization System Technical Report MCS-P615-1096 (Argonne National Laboratory, 1996)

T. Eickermann
Meta-computing in Gigabit Environments: Networks, Tools and Applications
Parallel Computing (1998) pp. 1847-1872

D.H.J. Epema, M. Livny, R. van Dantzig, X. Evers and J. Pruyne
A Worldwide Flock of Condors: Load Sharing among Workstation Clusters
in Future Generations of Computer Systems, P. Sloot (ed.), Vol 12 (1996)

I. Foster and K. Kesselman
GLOBUS: a Metacomputing Infrastructure Toolkit
Int. J. Supercomputing Applications (1997) pp. 115-128

I. Foster and K. Kesselman
The Globus Project: a status report
IPPS/SPDP'98 Heterogeneous Computing Workshop pp. 4-18 (1998)
http://www-fp.globus.org/documentation/papers.html

I. Foster and C. Kesselman (eds.)
The Grid: Blueprint for a new Computing Infrastructure
(Morgan Kaufmann, 1998) ISBN 1-55860-475-8. Abstracts of chapters and ordering information from URL http://www.mkp.com/grids

G.C. Fox and W. Furmanski
Petaops and Exaops: Supercomputing on the Web
IEEE Internet Computing 1 (1997) pp. 38-46
http://www.npac.syr.edu/users/gcfpetastuff/petaweb

G.C. Fox, W. Furmanski, T. Haupt, E. Akarsu and H. Ozdemir
HPcc as High-Performance Commodity Computing on top of Integrated Java, CORBA, COM and Web Standards
in “Euro-Par'98: Parallel Processing” D. Pritchard and J. Reeve (eds.) LCNS 1470 pp. 55-74 (Springer 1998) ISBN 3-540-64952-2

G. Fox and W. Furmanski
Java for Parallel Computing and as a general Language for Scientific and Engineering Simulation and Modelling Concurrency: Practice and Experience 9 (1997) 415-26

G.C. Fox, W. Fremanski, Z. Ozdemir, T. Pulikal
Using WebHLA to integrate HPC FMS Modules with Web/ Commodity based Distributed Objects Technologies of CORBA, Java, COM and XML
in “High Performance Computing 1999” A. Tentner (ed.) (Society for Computer Simulation International, 1999) pp. 273-278 ISBN 1-56555-166-4

E. Gabriel, M. Resch, R. Rühle
Implementing MPI with Optimised Algorithms for Meta-computing
in “Proceedings of the Third MPI Developer's and User's Conference”, A.J. Skjellum, P.V. Bangalore, Y.S. Dandass, eds. MPI Software Technology Press, Starkville Mississippi, 1999, pp. 31-42.
http://www.hlrs.de/people/gabriel/PAPER/atlanta99.ps

G.A. Geist, J.A. Kohl and P.M. Papadopoulos
CUMULVS: Providing Fault Tolerance, Visualisation and Steering of Parallel Applications
SIAM (August 1996)

D. Gillmor
Move over Supercomputers
San Jose Mercury News (1999)
http://www.sjmercury.com/business/computelprimell22.htm

A.S. Grimshaw, W.A. Wulf, J.C. French, A.C. Weaver and P.F. Reynolds Jr.
A Synopsis of the Legion Project
Technical Report CS-94-20 (University of Virginia, 1994)

A.S. Grimshaw, W.A. Wulf, J.C. French, A.C. Weaver and P.F. Reynolds
Legion: the next Logical Step toward a Nationalwide Virtual Computer
Technical Report CS-94-21 (University of Virginia, 1994)

A.S. Grimshaw and W.A. Wulf
Legion ¾ a view from 20,000 feet

in “Proc. 5th IEEE International Symposium on HPDC” IEEE Computer Society Press (1996)

A.S. Grimshaw, W.A. Wulf, J.C. French, A.C. Weaver and P.F. Reynolds Jr.
The Legion Vision of a Worldwide Virtual Computer
CACM (1997) Vol. 40 No. 1

W. Gu, J. Vetter and K. Schwann
An annotated Bibliography of Interactive Program Steering
SIGPLAN Notices 29 (1994) 140-8 and Technical Report GIT-CC-94-15 (Georgia Institute of Technology, 1994) available from Web URL
ftp://ftp.cc.gatech.edu/pub/tech_reports/1994/GIT-CC-94-15.ps.Z

W. Gu, G.S. Eisenhauer, E. Kramer, K. Schwan, J. Stasko and J. Vetter
FALCON: On-line Monitoring and Steering of Large-scale Parallel Programs
in Proc. 5th Symposium of the Frontiers of Massively Parallel Computing (February 1995) 422-29 and Technical Report GIT-CC-94-21 (Georgia Tech 1994)
ftp://ftp.cc.gatech.edu/pub/tech_reports/1994/GIT-CC-94-21.ps.Z

W. Gu, G.S. Eisenhauer and K.Schwan
FALCON: On-line Monitoring and Steering of Parallel Programs
Concurrency: Practice and Experience (1998) 10 No. 9, pp. 699-736.

E. Holmqvist and E. Lindström,
NetLink: A Modern Data Distribution Approach Applied to
Transparent Access of High Performance Software Libraries
in “Applied Parallel Computing” Proc. 4th International Workshop PARA'98.” B. Kagström, J. Dongarra, E. Elmroth and J. Wasniewski (eds.) Lecture Notes in Computer Science 1541 pp. 248-254 (Springer, 1998) ISBN 3-540-65414-3

S.A. Hopper, A.R. Mikler, P. Tarau, F. Chen and H. Unger
Mobile Agent-based File System for the WOS: an overview
in “High Performance Computing 1999” A. Tentner (ed.) (Society for Computer Simulation International, 1999) ISBN 1-56555-166-4 pp. 363-368

K. Hwang and Z. Xu
Scalable Parallel Computing: Technology, Architecture, Programming
(WBC/McGraw Hill, 1998) ISBN 0-07-031798-4

Toshiyuki Imamura, Yuichi Tsujita, Hiroshi Koide, and Hiroshi Takemiya
An Architecture of Stampi: MPI Library on a Cluster of Parallel Computers
In “Recent Advances in Parallel Virtual Machine and Message Passing Interface”, Proc. 7th European PVM/MPI Users' Group Meeting, Balatonfüred, Hungary, September 2000, J. Dongarra, P. Kacsuk, N. Podhorszki (Eds.), Lecture Notes in Computer Science 1908, p. 200 ff., Springer, 2001.
http://www.link.springer.de/link/service/series/0558/bibs/1908/19080200.htm

D.J. Jablonowski, J.D. Bruner, B. Bliss and R.B. Haber
VASE: The Visualisation and Application Steering Environment
in Proc. Supercomputing '93 pp. 560-569 (IEEE Computer Society Press, 1993)

R. Jones
NetPerf
Technical Report (1999)
Available from http://www.netperf.org/netperf/NetperfPage.html

B. Kagström, J. Dongarra, E. Elmroth and J. Wasniewski (eds.)
Applied Parallel Computing Prod. 4th International Workshop PARA'98. Lecture Notes in Computer Science 1541 (Springer 1998) ISBN 3-540-65414-3

T. Kielmann
MagPIe: MPI collective communication operations for clustered wide area systems
in "Proc. ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming" PPoPP'99 (Atlanta, Georgia, USA, 1999)
See http://www.cs.vu.nl/~kielmann/pubs.html

K. Kleese
Requirements for a Data Management Infrastructure to support UK High-End Computing
Technical Report DL-TR-99-04 (Daresbury Laboratory, 1999)

D. Kotz and R.S. Gray
Mobile agents and the future of the Internet ACM Operating Systems Review 33 (1999) pp. 7-13

P.G. Kropf
Overview of the WOS Project
in "High Performance Computing 1999" A. Tentner (ed.) (Society for Computer Simulation International, 1999) ISBN 1-56555-166-4 pp. 350-355.

P.G. Kropf, J. Plaice and H. Unger
Towards a Web Operatims System
in "Proc. WebNet" (Toronto, 1997)

M. Litzkow and M. Livny
Experiences with the Condor Distributed Batch System
in Proc. IEEE Workshop on Experimental Distributed Systems. (University of Wisconsin, Madison, 1990)

M. Miller, C.D. Hansen and C.R. Johnson
Simulated Steering with SCIRun in a Distributed Environment

in “Applied Parallel Computing” Proc. 4th International Workshop PARA'98. B. Kagström, J. Dongarra, E. Elmroth and J. Wasniewski (eds.) LNCS 1541 pp. 366-376 (Springer, 1998) ISBN 3-540-65414-3

D. Milojicic
Operating Systems ¾ now and in the future
IEEE Concurrency 7 (1999) 12-21

Thomas J. Mowbray, Ron Zahavi
The Essential CORBA: Systems Integration Using Distributed Objects
1995, Wiley, ISBN 0-471-10611-9

R. Orfali and D. Harkey
Client/Server Programming with Java and CORBA
(Wiley, 1997) ISBN 0-471-16351-1

C.M. Pancerella, L.A. Rahn and C.L. Yang
The Diesel Combustion Collaboratory: Combustion Researchers Collaborating over the Internet
Technical Paper presented at SuperComputing'99 (Portland, Oregon, USA, November 1999)

S.G. Parker, D.M. Weinstein and C.R. Johnson
The SCIRun Computation Steering Software System
in “Modern Software Tools in Scientific Computing” E. Arge, A.M. Bruaset and H.P. Langtangen (eds.) pp. 1-44 (Birkhäuser, 1997) ISBN 0-8176-3974-8 or ISBN 3-7643-3974-8

J.S. Plank and H. Casanova and M. Beck and J. Dongarra
Deploying Fault-tolerant and Task Migration with NetSolve in “Applied Parallel Computing” Proc. 4th International Workshop PARA'98. B. Kagström, J. Dongarra, E. Elmroth and J. Wasniewski (eds.) LNCS 1541 pp. 418-432 (Springer, 1998) ISBN 3-540-65414-3

D. Pritchard and J. Reeve (eds.)
Euro-Par'98: Parallel Processing
Lecture Notes in Computer Science 1470 (Springer 1998)
ISBN 3-540-64952-2

J. Pruyne and M. Livny
Parallel Processing on Dynamic Resources with CARMI
in “Job Scheduling Strategies for Parallel Processing” D.G. Feitelson and L. Rudolph (eds.) Lecture Notes in Computer Science 949 (Springer-Verlag, 1995)

J. Pruyne and M. Livny,
Interfacing Condor and PVM to harness the cycles of workstation clusters

to appear in “Future Generations of Computer Systems” P. Sloot (ed.)

F. Reynolds
Evolving an operating system for the Web
IEEE Computer 29 (1996) 90-92

J. Robinson, S.H. Russ, B. Flachs and B. Heckel
A Task Migration Implementation of the Message-Passing Interface
in Proc. Fifth High Performance Distributed Computing Conference (Syracuse, NY, Aug. 1996)

S.H. Russ
Using Hector in an Architecture for Rapid Distributed Fault Tolerance
Technical Report MSSU-EIRS-ERC-97-17 (Mississippi State University, Dec. 1997)

S.H. Russ, B.K. Flachs, J.A. Robinson and B. Heckel
Hector: Automated Task Allocation for MPI
Technical Report MSSU-EIRS-ERC-95-6 (Mississippi State University, Sept. 1995)

S.H. Russ, B. Flachs, J. Robinson and B. Heckel
Hector: Automated Task Allocation for MPI
in “10th International Parallel Processing Symposium” (IEEE Press, Honolulu, HI, Apr. 1996)

S.H. Russ, B. Meyers, M. gleeson, J. Robinson, L. Rajogopalan, C. Tan and B. Heckel
User Transparent Run-time Performance Optimisation in “2nd Int. Workshop on Embedded HPC and Applications” at 11th IEEE Int. Parallel Processing Symposium (1997)

S.H. Russ, J. Robinson, and M. Gleeson
Dynamic Communication Mechanism Switching in Hector
Technical Report MSSU-EIRS-ERC-97-8 (Mississippi State University, Dec. 1997)

S. Sekiguchi, M. Sato, H. Nakada, S. Matsuoka and U. Nagashima
Ninf: Network based Information Library for Globally High-performance Computing
in Proc. POOMA (Santa Fe, 1996)

J. Siegel
CORBA 3 Fundamentals and Programming, Second Edition
2000, Wiley, ISBN 0-471-29518-3

M. van Steen, P. Homburg and A.S. Tanenbaum
The Architectural Design of GLOBE: a wide-area distributed system
Technical Report IR-422 (Vrije Universiteit, Amsterdam, 1997)

J.S. Steinman, G. Berliner, G.E. Blank, J.S. Brutocao, J. Burckhardt, M. Peckham, S. Shuper, K. Stadsklev, T. Tran, R.V. Iwaarden, L. Yu
The SPEEDES-based run-time Infrastructure for higl-level Architecture on high-performance Computers in "High Performance Computing 1999" A. Tentner (ed.) (Society for Computer Simulation International, 1999) pp. 255-266 ISBN 1-56555-166-4

W.R. Stevens
UNIX Network Programming: Volume 1: second edition
1998, Prentice Hall, ISBN 0-13-490012-X

A. Su, F. Berman, R. Wolski and M.M. Strout
Using AppLeS to Schedule Simple SARA on the Computational Grid
International Journal of High Performance Computing Applications, 1999, Vol. 13, No. 3, pp. 253-262
http://apples.ucsd.edu/hetpubs.html

Sun Microsystems Inc. (1997)
Java Remote Method Invocation Specification

Sun Microsystems Inc. (1998)
Jini Specification rev. 1.0 beta
http://www.javasoft.com/products/jini/specs

T. Tannenbaum and M. Litzkow
the Condor Distributed Processing System
Dr. Dobbs' Journal (February, 1995)

D. Tavangarian
in "Special issue on Cluster Computing"
Journal of Systems Architecture 44 (Elsevier Science, 1997)

L.H. Turcotte
A Survey of Software Environments for Exploiting Networked Computing
Technical Report MSU-EIRS-ERC-93-2 (Mississippi State University, 1993)

H. Unger
A new Security Mechanism for the use in large distributed systems
in "High Performance Computing 1999" A. Tentner (ed.) (Society for Computer Simulation International, 1999) ISBN 1-56555-166-4 pp. 369-376

A. Vahdat, T. Anderson, M. Dahlin, D. CUller, E. Belani, P. Eastham and C. Yoshikawa
WebOS: Operating System Services for Wide Area Applications
in "Proc. 7th Symposium on High Performance Distributed Computing" (1998)
http://now.cs.berkeley/edu/WebOS/publications.shtml

J. Vetter and K. Schwan

PROGRESS: A Toolkit for Interactive Program Steering
in Proc. 24th Ann. Conf. on Parallel Processing (1995)

J. Vetter and K. Schwan
High-performance Computational Steering of Physical Simulations
in Proc. 11th Int. Parallel Processing Symposium (Geneva, Switzerland, April 1997)

J. Wallace, L. Petersen, G. Leonard, J. Celano, J. Brutocao and J. Caldwell
Import v2.0 beta 1: a Tool for large-scale, complex System Simulation in "High Performance Computing 1999" A. Tentner (ed.) (Society for Computer Simulation International, 1999) pp. 267-272 ISBN 1-56555-166-4

R.A. Whiteside, E.J. Friedman-Hill and R.J. Detry
PRE: a framework for enterprise integration
in "Proc. Design of Information Infrastructure Systems for Manufacturing - DIISM'98" (Fort Worth, Texas, May 1998)

R. Wolski, J. Brevik, C. Krintz, G. Obertelli, N. Spring and A. Su
Running EveryWare on the Computational Grid
Technical Paper presented at SuperComputing'99 (Portland, Oregon, USA, November 1999)

R. Wolski, N. Spring and H. Hayes
The Network Weather Service: a distributed resource performance forecasting service for meta-computing in "Future Generation Computing Systems" (1998). Available from http://www.cs.ucsd.edu/users/rich/papers/nws-arch.ps

Z. Yang and K. Duddy
CORBA: a platform for distributed object computing
OSR 30 (1996)

W. Yeong, T. Howes and S. Kille
Lightweight Directory Access Protocol
RFC 1777 (March 1995)

C.A. Lee, J. Stepanek, R. Wolski, C. Kesselman and I. Foster
A Network Performance Tool for Grid Environments Technical Paper presented at SuperComputing'99 (Portland, Oregon, USA, November 1999)