Cluster Computing R&D in Australia

ATIP/Australia

ABSTRACT: This report provides overview of Cluster Computing R&D activities in the Asia-Pacific region with particular focus on Australian research activities. We give details of R&D at ten Australian universities as well as highlights of research at several other sites in Asia.

KEYWORDS: Computer Software, Conferences, High-Performance Computing
COUNTRY: Australia
AUTHORS: Mark Baker, Rajkumar Buyya, Ken Hawick, Heath James, and Hai Jin

REPORT CONTENTS

1. INTRODUCTION

The use of clusters of computers as a platform for high-performance and high-availability computing is increasing mainly due to their cost-effective nature. Clusters have been employed as a platform for a number of applications including supercomputing, mission-critical, web-serving, e-commerce, database, and some other commercial applications. Clusters have been used for solving grand challenge applications such as weather modeling, automobile crash simulations, life sciences, computational fluid dynamics, nuclear simulations, image processing, electromagnetics, data mining, aerodynamics and astrophysics. While traditional supercomputers are easier to program, usage of cluster systems is increasing, a trend that is likely to continue.

A number of R&D international projects are investigating the design and implementation of cluster computing technologies and in fact a number of companies are exploiting this technology to develop commercial solutions. Recently many organizations have replaced their traditional or proprietary Supercomputers with clusters as a cost saving measure. Cluster-based systems of supercomputing-class ability have been assembled using commodity-off-the-shelf (COTS) hardware components and freely available software such as Linux operating systems and parallel communication protocols like MPI.

This report provides overview of Cluster Computing R&D activities in the Asia-Pacific region with particular focus on Australian research activities. The report has been organized into three major parts: cluster systems being built, software research projects, and lastly, key cluster computing research projects in the Asia-Pacific region with a focus on Australian academic research. In one sense, activities in Australia mirror similar research activities in other parts of the world. However, cluster systems and software research has been strongly embraced in Australia, perhaps primarily due to the cost performance attractiveness of this technology. Australia has a relatively small population upon whose tax burden research is supported, so the attraction for the clusters is perhaps not surprising. With respect to Australia specifically, some related ATIP Reports include the following.

ATIP98.100 : Some Computational Chemistry and Physics in Australia
ATIP98.099 : Australian Government HPC Initiative - Update
ATIP98.026 : HPC at the Australian National University
ATIP98.025 : Australian HPC Efforts
ATIP96.054 : High Performance Computing Activities in Australia

2. CLUSTER COMPUTING TASK FORCE

Recognizing the emergence and importance of cluster computing technology, the IEEE Computer

Society has set up a Task Force on Cluster Computing[1] (TFCC) in late December 1998. The task force has been acting as an international forum for promoting cluster computing research, promoting education, and collaboration with industry. It participates in helping to set up and promote technical standards in this area. The Task Force is concerned with issues related to the design, analysis, development and implementation of cluster-based systems. Of particular interest are: cluster hardware technologies, distributed environments, application tools and utilities, as well as the development and optimization of cluster-based applications.

The TFCC sponsors professional meetings, publish newsletters and other documents, sets guidelines for educational programs, as well as helps coordinate academic, funding agency, and industry activities in the above areas. The TFCC organizes an annual conference and holds a number of workshops that span the range of activities. In addition, TFCC publishes a biannual newsletter to help IEEE members keep abreast of the events occurring within this field.

The TFCC has been successful in promoting cluster computing through its educational program. One aspect of this program that has been especially popular is an international book donation scheme. Here, authors of cluster-related books are requested to ask their publishers to donate books to the program. In fact, the program has become so popular that the IEEE Computer Magazine editor invited the committee to write an article on this, which appeared in the July 1999 issue of IEEE Computer Magazine under the Technical Activities Forum section.

ATIP's interest in the production of this report stems from a recent technical meeting, the IEEE International Workshop on Cluster Computing (IWCC'99) held in Melbourne, Australia (Dec. 2-3. 1999). The workshop was organised and co-chaired by Rajkumar Buyya (Monash University) and Mark Baker (University of Portsmouth). The workshop attracted participants from academia and industry, as well as users of clusters computing platforms from major 16 countries including US, Japan, France, Hong Kong, Taiwan, Brazil, Germany, China, Singapore, Malaysia, Korea, Sweden, Switzerland, the UK; Mexico, and Australia. Industrial delegates also participated, including Sun, SGI, Hewlett-Packard, Compaq, MPI Software Technology, and Microsoft.

IWCC'99 Technical Program Highlights

The IWCC'99 program consisted of keynote talks, invited talks from industry, regular and poster sessions, as well as a panel on "Cluster Computing R&D in Australia". The program details and presentation slides of some of invited/keynote/industry talks can be found at the meeting's web site[2]. The printed conference proceedings can be obtained from the IEEE Computer Society Press and it as also available online at IEEE Digital Library[3].

Plenary Speaker:
Thomas L. Sterling (California Institute of Technology)
Beowulfs and Beyond: Past Accomplishments and Future Directions

Keynote Speakers:
Kai Hwang, (University of Southern California)
Fault-Tolerant Cluster Architecture for Business and Scientific Applications

Anthony Skjellum (MPI Software Technology Inc.)
Clustering for Research and Production Scale, Parallel and Distributed Computing

[1] http://www.computer.org/tab/tclist/tfcc.htm | http://www.ieeetfcc.org
[2] http://www.csse.monash.edu.au/~rajkumar/tfcc/IWCC99/
[3] http://www.computer.org/proceedings/iwcc/0343/0343toc.htm

David Abramson (Monash University)
From PC Clusters to a Global Computational Grid

Regular Sessions:

- Cluster Setup and Performance Measurement
- Cluster Communications Software and Protocols
- Network Communication Optimization
- Cluster File Systems and Parallel I/O
- Scheduling Programs on Clusters
- Cluster Management and Metacomputing
- Cluster Operating Systems and Monitoring
- Cluster Programming and Analysis Models
- Algorithms and Applications

Industry/Invited Talks:

Sun Cluster Architecture -- Ira Pramanick, Sun, USA.
HP Hyperplex Clustering Technology -- John Cowles, HP, USA/Japan.
Compaq's Directions towards Scalable Supercomputing -- Bruce A. Foster, USA.
Computational Astrophysics on Supercluster -- Matthew Bails, Swinburne University, Australia.

The presentation materials of the above talks are made available online at the IWCC'99 web site. Other details of IWCC'99 has been reported in IEEE Concurrency Magazine (Jan.-March, 2000) issue by Dr. Ahmed in his column on "Cluster Computing: A glance at recent events".

ATIP offers a full range of information services including reports, assessments, briefings, visits, sample procurements, workshops, cultural/business sensitivity training, and liaison activities, all performed by our on-the-ground multilingual experts.

<u>Tokyo Office</u>:
Asian Technology Information Program
Harks Roppongi Building 1F
6-15-21 Roppongi, Minato-ku
Tokyo 106, JAPAN
Tel: +81 3 5411-6670
Fax: +81 3 5411-6671

<u>U.S. Office</u>:
Asian Technology Information Program
PO Box 9678
Albuquerque, New Mexico 87119
Tel: (505) 842-9020
Fax: (505) 766-5166

For further information:
Send us email: info@atip.or.jp
Visit our website: www.atip.or.jp OR www.atip.org

Complete ATIP reports on Asian Science and Technology go to subscribers and collaborating organizations by direct distribution, or via electronic access. These contain text and often, charts, graphs and pictures. Reports for unrestricted distribution often contain summarized *or abstracted information. Sponsors can also obtain specific follow up information - including copies of proceedings, selected papers, exhibition particulars, updates, translations, query searches, etc.*

END OF REPORT ATIP00.015a
[The remaining sections of this report are available to ATIP subscribers]

ATIP00.015 (continued): Cluster Computing R&D in Australia

EXECUTIVE SUMMARY

- Cluster computing is becoming increasingly popular with clusters pushing into mainstream computing. However, cluster computing is primarily and will continue to be a major area of research interest for high-performance and high-end computing community.

- Within Australia, special issues such as wide area clusters and sharing of scarce resources are important and are therefore drivers for software research.

- Application areas including e-commerce and business systems are likely to be major drivers for cluster computing in addition to the more traditional areas such as scientific and engineering applications.

- Challenges to be addressed are related to resource management, scalability, expandability, efficient communication, system administration, and single system image. Some of the research work described here contributes towards addressing these challenges, and a number of them have commercial potential.

- Table 1 lists the major cluster research projects in Australia. The text describes them in detail.

- Of the Australian sites surveyed for this report the strongest, both in terms of current work and also rate of development, are as follows.
  - Adelaide University, Adelaide, for applications on clusters.
  - Monash University, Melbourne for tools.
  - Australian National University, Canberra.

3. MAJOR CLUSTER COMPUTING INITIATIVES IN AUSTRALIA

In this report we highlight work being undertaken in the area of cluster computing at the following institutions, which represent the primary research centers with efforts in this area.

- Australian National University (ANU), Canberra.
- Deakin University, Geelong.
- Monash University, Melbourne.
- Queensland University of Technology (QUT), Brisbane.
- Royal Melbourne Institute of Technology (RMIT), Melbourne.
- Swinburne University of Technology (SUT), Melbourne.
- University of Adelaide, Adelaide.
- University of New South Wales (UNSW), Sydney.
- University of Queensland (UQ), Brisbane.
- University of Western Australia (UWA), Perth.

As in other countries, in Australia cluster-based parallel computing systems for high performance computing have been built by users themselves. The major efforts have been focused on the development of system software, tools, and algorithms that present clusters as a unified computing resource. Applications for clusters have either been explicitly parallelized or automated using tools. Table 1 shows the major cluster computing research projects in Australian universities. The projects can be categorized into high performance networks, operating systems, system management tools, and applications for clusters. (Some of these projects are discussed in more detail in subsequent sections.)

Table 1: Major Cluster Computing Research Projects in Australia

Initiative/Project	Focus and Technologies Developed	URL for Further Information
U. Adelaide: DISCWorld	Service-based metacomputing across LAN and WAN clusters.	dhpc.adelaide.edu.au/projects/DISCWorld/
Monash U: Nimrod/G	A global scheduler for parametric computing over a enterprise wide clusters or computational grid.	www.csse.monash.edu.au/~davida/nimrod.html
Monash U: GUARD	A parallel relative debugger.	www.csse.monash.edu.au/research/guard/
International Effort: Compute Power Grid	A portal and an economic based resource management and scheduling system for global (grid) computing on Internet-wide resources.	www.ComputePower.com/
RMIT: Eddie	A cluster-based robust internet web server.	www.serc.rmit.edu.au/projects.html#Eddie
UWA: Parallel Computing	Scheduling across parallel computing clusters.	www.ee.uwa.edu.au/~paracomp/projects.html
UWA: Fault-tolerant NOW	A fault tolerant switch/network for clusters.	ciips.ee.uwa.edu.au/~morris/Research/FTCilk/
Deakin U: RHODOS	A distributed operating system for clusters.	www3.cm.deakin.edu.au/rhodos/
QUT*:* Gardens	A language and runtime system creating a virtual parallel machine across a network on non-dedicated computers (PCs/workstations).	www.plasrc.qut.edu.au/Gardens/
UQ: DSM on JVM	Distributed Shared Memory on Java Virtual Machine.	www.csee.uq.edu.au/research/report/distsa.jhtml
UNSW*:* Mungi	Mungi is an operating system based on the idea of a single address space, shared by all processes and processors in the system.	www.cse.unsw.EDU.AU/~disy/Mungi/
ANU: Wyrd cluster	A platform for developing parallel computing software and applications.	www.anusf.anu.edu.au/wyrd/
SUT: Astrophysics on Superclusters	Astrophysics, Austronomy, and graphical image rendering applications.	www.swin.edu.au/astronomy/

4. CLUSTER SYSTEMS (HARDWARE) & INFRASTRUCTURE INITIATVES

There is still a lack of common use of terminology for cluster systems. Here, we do not distinguish between cluster systems and Beowulf class systems. Beowulf cluster systems are becoming widespread in both academic and industrial sites around the world. There are clear price-performance benefits over conventional parallel supercomputer systems for many applications. Recent developments suggest that the Beowulf model is becoming viable not only for task farming applications but for large-scale parallel programs with more intensive communications patterns. These issues are of great interest for the future of both cluster computing and parallel computing. Work is in progress to embody the research ideas in parallel computing from the last 20 years into libraries and templates of code to aid the use of Beowulf systems[4]. Many institutions are building cluster systems and we report on some of the primary ones here.

South Australian Centre for Parallel Computing

In Adelaide, in South Australia, there has been a long-standing interest in supercomputing and other high performance computing activities. The South Australian Centre for Parallel Computing has run a number of advanced systems in the past including a Thinking Machines Connection Machine Model 5 and a Silicon Graphics Power Challenge. These systems are still run, but users are migrating onto cluster systems now being built. Two regional infrastructure grants were successfully applied for by the Chemistry department (around A$300k) and the Physics department (around A$670k) at Adelaide University. The Distributed and High Performance Computing (DHPC) Research group[5] has designed and built the system for computational chemistry, consisting of 240 Pentium III processors connected by a Fast Ethernet communications system. This machine is presently running computational chemistry packages such as Gaussian and has produced a number of research results already[6]. The DHPC group is presently designing (and evaluating components for) a second system to satisfy the needs of computational physics at Adelaide and as part of a national consortium for computational quantum physics.

Swinburne

The Swinburne supercluster is a 65-node configuration of Compaq/Digital Alpha workstations, and is involved in several major projects involving large-scale processing. Scientists at the Swinburne Centre for Astrophysics and Supercomputing use this machine for conducting research in computationally demanding problems in pulsar astrophysics[7].

Monash

At Monash, a cluster system[8] called the Monash Parallel Parametric Modeling Engine (PPME) has been assembled with the aim of using it for development of next generation software tools and also for large parametric modeling computations. Parametric computing involves the concurrent execution of one program on multiple machines over a range of parametric simulations; it can achieve performance 62 times faster than would be possible on a single computer. PPME comprises 32 2-way SMP PC nodes (Intel Pentium processors) running the Linux operating system. The machine spreads equally across two Monash University campus, Clayton and Caulfield (16 nodes in each place connected using Fast Ethernet). These two clusters are

[4] http://dhpc.adelaide.edu.au/projects/beowulf

[5] http://dhpc.adelaide.edu.au/

[6] http://dhpc.adelaide.edu.au/projects/beowulf/perseus.html

[7] http://mania.physics.swin.edu.au/

[8] http://hathor.cs.monash.edu.au/

connected using ATM networks, creating a cluster of clusters. A number of research work and student projects are in the areas of factory simulation, fuzzy logic simulation, static channel assignment, N queens problem, modeling OPNET simulations, ad hoc network simulation, disk fracture prediction, biological and financial modeling. These have used the Monash cluster using a tool called Clustor (a commercial version of DSTC/Monash Nimrod project). This machine has also been coupled into the Globus GUSTO Grid Computing test-bed and used for the development of wide-area resource management and scheduling system called Nimrod/G.

Queensland

In Queensland, Queensland University of Technology (QUT) researchers have setup a small Sun workstation cluster interconnected using a Gigabit Myrinet switch. It is being used to develop a programming language and system, called Gardens, for parallel computing on non-dedicated networks of workstations[9].

ANU

The ANU Supercomputer Facility (ANUSF) known for its work on Fujitsu vector processing machines and system software has recently deployed a Linux-Alpha Cluster and plans to use it for solving scientific supercomputing applications in the areas of biology, chemistry, engineering, and environmental modeling. They also plan to conduct research on software for clusters[10].

Australian Partnership for Advanced Computing (APAC[11])

The Australian Partnership for Advanced Computing (APAC), with headquarters at ANU, has been established with a grant of $19.5m from the Federal Government to underpin significant achievements in Australian research, education and technology diffusion by establishing and supporting an effective advanced computing capability ranked in the top 10 countries. One of the roles for APAC is to provide users, particularly in the Higher Education sector, with 'peak' computing systems far beyond the capacity that are currently available. Another important role for APAC is to strengthen the expertise and skills necessary for the effective use and development of these facilities. The broader role for APAC is to form a partnership to lead the development of an Australia-wide computing and communications systems infrastructure supported by Centers of Expertise in advanced computing. APAC is in the process of selecting a peak computing system for a National Facility to be based at ANU, and developing strategies for strengthening complementary infrastructure at other locations around Australia.

Wide Area Clustering

Given its geography, wide area cluster computing activities are of particular importance to Australia. A number of projects and consortia have experimented with wide area systems and technologies. Some of these are as follows.

Research Data Networks Cooperative Research Centre (RDN CRC[12])

The National Government has sponsored a number of collaborative centers in the area of Information technology. The RDN CRC was one of these collaborations and involved Adelaide University, Monash University, Queensland University and the Australian National University.

[9] http://www.plasrc.qut.edu.au/

[10] http://www.anusf.anu.edu.au/wyrd/

[11] http://www.apac.edu.au

[12] http://www.dstc.edu.au/RDNCRC/

These sites were connected using Asynchronous Transfer Mode (ATM) networking technology at 155 Mbps and a number of systems and software research projects were carried out using cluster-computing resources at each site. Some results were: initiation of the DISCWorld wide area cluster computing software project; developments of the Nimrod software project; establishment of collaborations with Tsukuba science city in Japan and establishment of the Australia Pacific ATM network (APAN). The wide area networks set up under the RDN CRC were extensively used to share large data archives and exchange cluster processing cycles between users in a range of scientific and engineering disciplines

The Asia Pacific ATM Network (APAN[13])

APAN is a non-profit international consortium established in June 1997. APAN is intended to be a high-performance network for research and development on advanced applications and services in the Asia-Pacific region. It provides an advanced networking environment for the research community; and also promotes international collaboration. The APAN network connects Australia, Japan, the Philippines, Indonesia, Singapore, Malaysia, Thailand, Hong Kong, China and Korea with Europe and North America.

5. MAJOR CLUSTER SOFTWARE RESEARCH PROJECTS

Beyond hardware and infrastructure, a number of universities in Australia have significant research projects to investigate cluster computing software issues. In many respects these build on distributed or parallel computing activities, from which many of the present ideas in cluster systems are being drawn. Projects range from major systems research and integration projects to narrower projects looking at specific aspects of distributed cluster computing. As in the case of groups building systems, the groups experimenting on software issues are addressing similar topics as their counterparts elsewhere in the world. There is a general trend towards making good use of minimal resources and resource sharing in the Australian projects. There is also a trend towards making good use of wide-area resources.

Clusters offer a very cheap way for gaining access to potentially huge computing power. However, the challenge is to produce software environments that provide an illusion of a single resource, rather than a collection of independent computers. The following subsections discuss some major projects in Australia.

5.1 Distributed Information Systems Control World (DISCWorld[14])

DISCWorld is a prototype-metacomputing model and system being developed at the University of Adelaide. The project was started in 1996 by the Distributed and High Performance Computing (D&HPC) Group. The basic unit of execution in the DISCWorld is a "service." Services are pre-written pieces of Java software that adhere to an API or legacy code that has been provided with a Java wrapper. Users can compose a number of services together to form a complex processing request.

The DISCWorld architecture consists of a number of peer-based computer hosts that participate in DISCWorld by running either a DISCWorld server daemon program or a DISCWorld-compliant client program. DISCWorld client programs can be constructed using Java wrappers to existing

[13] http://www.apan.net/

[14] http://dhpc.adelaide.edu.au/projects/DISCWorld/

legacy programs, or can take the form of a special DISCWorld client environment that runs as a Java applet inside a WWW browser. This client environment can itself contain Java applet programs (Java Beans) which can act as client programs communicating with the network of servers. The peer-based nature of DISCWorld clients and servers means that these servers can be clients of one another for carrying out particular jobs, and can broker or trade services amongst one another. Jobs can be scheduled across the participating nodes. A DISCWorld server instance is potentially capable of providing any of the portable services any other DISCWorld server provides, but will typically be customized by the local administrator to specialize in a chosen subset, suited to local resources and needs. The nature of the DISCWorld architecture means that developers use the terms client and server somewhat loosely, since these terms best refer to a temporary relationship between two running programs rather than a fixed relationship between two host platforms.

DISCWorld is targeted at wide-area systems and to applications that are worthwhile or necessary to run over wide areas. These will typically be applications that require access to large specialized datasets stored by custodians at different geographically separated sites. An example application might be land planning, where a client application requires access to land titles information at one site, digital terrain map data at another, and aerial photography or satellite imagery stored at a third. A human decision-maker may be seated at a low performance compute platform running a WWW browser environment, but can pose queries and data processing transactions of a network of DISCWorld connected servers to extract decisions from these potentially very large datasets without having to download them to his/her own site.

This project has so far produced over 50 technical reports and refereed publications, one PhD thesis, as well as several theses that are in preparation. Researchers in the D&HPC Group are looking into many areas of distributed computing including: scheduling and placement of services and data for near-optimal runtime execution; high performance distributed and parallel file system technologies; the use of object request brokers in metacomputing environments; performance modeling of metacomputing and cluster computing systems; reconfigurability in metacomputing and cluster computing systems; and the use of data tiling in cluster and metacomputing systems.

The D&HPC Group is also actively studying the use of standards-based approaches to manage large geo-spatial data sets. The Group has written implementations of a number of proposed standards and is adapting the standards for use across clusters of workstations and wide-area distributed systems.

5.2 QUT Gardens[15]

The Gardens project aims to create a virtual parallel machine out of a network of workstations interconnected using low latency communication links. It focuses on tightly integrating the programming and system facilities by the use of a single expressive language. The project is composed of a family of sub-projects spanning most areas of interest covered by QUT's Programming Languages and Systems Research Center. Gardens supports abstractions for expressing parallelism through Mianjin language, Tasking Systems for Process Migration and Control, and I/O under migration. The Gardens system programming language Mianjin supports a virtual shared object space and uses type information to enforce safe communication in the presence of abstractions. The interesting features of Gardens are outlined at the project web site.

[15] http://www.plasrc.qut.edu.au/Gardens

5.3 Monash/DSTC Nimrod/G[16]

Nimrod is a tool for parametric computing on clusters and it provides a simple *declarative* parametric modeling language for expressing a parametric experiment. Domain experts can easily create a *plan* for a parametric computing (task farming) and use the Nimrod runtime system to submit, run, and collect the results from multiple computers (cluster nodes). Nimrod has been used to run applications ranging from bio-informatics and operations research, to the simulation of business processes. A reengineered version of Nimrod, called Clustor, has been commercialized by Active Tools[17]. However, research on Nimrod has been continued, to address its use in the global computational grid environment and to overcome shortcomings of the earlier system. For example, the new system can dynamically identify resources.

Nimrod has been used successfully with a static set of computational resources, but is unsuitable as implemented in the large-scale dynamic context of computational grids, where resources are scattered across several administrative domains, each with their own user policies, employing their own queuing system, varying access cost and computational power. These shortcomings are addressed by a new system called Nimrod/G that uses the Globus[18] middleware services for dynamic resource discovery and dispatching jobs over wide-area distributed systems called computational grids.

Nimrod/G allows scientists and engineers to model whole parametric experiments and transparently stage the data and program at remote sites, and run the program on each element of a data set on different machines and finally gather results from remote sites to the user site. The user need not worry about the way in which the complete experiment is set up, data or executable staging, or management. The user can also set the deadline by which the results are needed and the scheduler tries to find the cheapest computational resources available in the global computing grid and use them so that the user deadline is met and cost of computation is kept to a minimum.

The current focus of the Nimrod/G project team is on the use of economic theories in grid resource management and scheduling[19] as part of a new framework called GRACE[20] (Grid Architecture for Computational Economy). The components that make up GRACE include global scheduler (broker), bid-manager, directory server, and bid-server working closely with grid middleware and fabrics. The GRACE infrastructure also offers generic interfaces (APIs) that the grid tools and applications programmers can use to develop software supporting the computational economy.

5.4 UWA Parallel Computing Research Group[21]

In either sequential or parallel systems, the architecture is characterized by functional components, the communication topology & facilities, control structures, and mechanisms. However, there are several issues related to parallelization that do not arise in sequential programming. One of the most important of these is task-allocation, that is the breakdown of the total workload into smaller tasks assigned to different processors, and the proper sequencing of the tasks when some of them are interdependent and cannot be executed simultaneously. To achieve the highest level of performance it is important to ensure that each processor is properly utilized. This process is called load balancing or scheduling, and it is considered to be extremely

[16] http://www.dgs.monash.edu.au/~davida/nimrod.html
[17] http://www.activetools.com
[18] http://www.globus.org
[19] http://www.dgs.monash.edu.au/~rajkumar/talks/WGCC-Japan/
[20] http://www.dgs.monash.edu.au/~rajkumar/papers/WGCCJapan.pdf
[21] http://www.ee.uwa.edu.au/~paracomp/

"formidable" to solve. The scheduling problem belongs to a class of problems known as NP-complete. The UWA (University of Western Australia) parallel computing research group is studying the theory and developing applications (e.g. robotics), scheduling and mapping, heterogeneous and fault-tolerant computing, artificial neural networks, genetic and evolutionary computing, and fuzzy logic[22].

5.5 Deakin RHODOS Distributed OS and Parallelism Management[23]

RHODOS is a micro-kernel based distributed operating system for clusters. The RHODOS group has developed an OS kernel from scratch and many system services have been built at the user level. The services of RHODOS are transparent message passing; process management (includes the support for global scheduling -- both static allocation and dynamic load balancing), and I/O management. On top of these basic services, parallel processing tools such as PVM (Parallel Virtual Machine) and DSM (Distributed Shared Memory) Systems have been built to ease the development of parallel programs. This group is also working on compilers for automatic parallelization of sequential programs to parallel programs. When all these tools developed by the RHODOS group put into production, they may have a significant impact on the cluster computing industry. The group members have produced over 50 publications. Full information the work can be found at the group webpage.

6. ASIA-PACIFIC CLUSTER COMPUTING RESEARCH AT IWCC'99

Researchers in the Asia-Pacific region are attempting to solve a number of issues and problems associated with cluster computing, including: the use of multithreaded DSM runtime systems; reducing network overheads and communication patterns; developing realistic communication models; distributed and parallel file systems.

In this section we discuss selected papers presented at IWCC'99 by researchers from Asia-Pacific region with particular emphasis on Australian research. The workshop's papers spanned a wide range of topics, including:

- Cluster setup and performance measurements
- Communications software and protocols
- Network communication optimization
- File system and Parallel I/O
- Scheduling program on clusters
- Cluster management and metacomputing
- Operating systems and monitoring
- Programming and analysis models, and
- Algorithms and applications.

6.1 Cluster Communications Software and Protocols

Fujitsu Laboratory, Japan

Naoshi Ogawa, Takahiro Kurosawa, Mitsuhiro Kishimoto (Fujitsu Laboratory, Japan), Nobuhiro

[22] http://www.ee.uwa.edu.au/~paracomp/projects.html
[23] http://www3.cm.deakin.edu.au/rhodos/

Tachino, Andreas Savva, and Keisuke Fukui (Fujitsu, Japan) reported on their work in designing a high performance *Virtual Interface Architecture* (VIA) communication system, called Scnet. VIA is a standard defined by Intel, Compaq, Microsoft and others. It is an effort to standardize the network interface and user level communication semantics offered by *Network Interface Card* (NIC). Scnet is the first UNIX VIA system that achieves "full conformance". It is connected with Fujitsu's Synfinity-0 *System Area Network* (SAN). The paper titled "Smart Cluster Network (Scnet): Design of High Performance Communication System for SAN" describes the details of Scnet design and its performance evaluation on a small scale cluster system. The main features of Scnet are:

> A Virtual NIC which allows effective usage of multiple NICs and network support for load balancing as well as partial support for NIC failover.
>
> The protection of a process' resources from other processes by checking the Process ID (PID) every time a process requests an access.
>
> A memory registration mechanism that allows a user application to register any memory region in its space.
>
> A Copy-send-copy communication protocol that is used when transferring small data and when a data alignment error is detected. Also a Send-get protocol is used when the cost of copying exceeds the cost of setup.

The authors evaluated Scnet performance experimentally on two-nodes of a cluster system. When using a single NIC, the ping-pong latency was 42µs measured for a 0 Byte data transfer. The maximum bandwidth was measured at 108 MBytes/s for 128 KByte transfers. By using 2 NICs per node, Scnet achieves up to 210 MBytes/s bandwidth.

6.2 Network Communication Optimization

Australian National University, Canberra, Australia

Jeremy Dawson and Peter Strazdins (Australian National University, Australia) presented techniques and algorithms to improve the performance of various communication patterns on a specific message-passing platform, Fujitsu's AP3000, in their paper "Optimizing User-Level Communication Patterns on the Fujitsu AP3000". On the AP3000, user-level communications must be buffered in special memory on both the send and the receive sides, and the actual transfer of the messages occurs in chunks, for reasons of safety. One method to improve the performance for large messages is the *protocol method*, which involves breaking the message into large chunks and pipelining the chunk transfers.

They extend the protocol method to *communication patterns*, where the pipelining can be performed over several messages involving any one node in the pattern, and/or the copies to special memory can be amortized over different messages. These algorithms can not only minimize message copying but also overlap the copying to/from the special memory with the actual transfer, enabling full bandwidth to be achieved. These patterns include tree broadcast and reduction, (ring-based) multiple broadcasts and reductions, pipelined broadcast, and buffered point-to-point sends. In each case, the messages may have a simple stride and substantial performance improvements were observed. In general, the simpler the pattern, the better the improvement, as it is easier to get full overlapping of message buffering and transmission in this situation. All of these patterns are used in the context of solving dense linear algebraic equations, although they may also be useful for other applications.

Although the authors' algorithms are implemented and evaluated on the AP3000, the algorithms also apply to any platform using a similar mode of user level communications. Worthwhile performance increases are obtained, especially for patterns involving moderately large numbers of processors.

6.3 Cluster File system and Parallel I/O

University of Sydney, Australia

Bruce Janson and Bob Kummerfeld (University of Sydney) reported on their distributed file system design for a loosely coupled group of computers in a paper titled "Soda: A File System for a Multicomputer". Soda presents a consistent file system image to a number of CPU servers. Compared with an existing multicomputer operating system, such as AFS, Coda (from CMU), or NFS (from Sun Microsystems), Soda has the following features:

It provides a single consistent file system image to disks and network files.

It supports limited programmability of the file system’s operations through "active files".

It keeps track of changes made to system files with respect to the original file system sub-hierarchy; thereby simplifying operating system upgrades and problem diagnosis.

It provides some support for process migration by a root-owned, user-mode "wrapper" process on the destination computer.

It provides a privileged user with convenient run-time access to file system status.

It allows dynamic, file system-wide feedback and control via /.Soda.

Currently, the Soda file system borrows the Coda kernel interface. It is comprised of three components: the Coda Linux kernel module, a local cache manager and a remote file server. The cache and file server processes are single-threaded with respect to their clients, and this causes unnecessary blocking. This paper only discusses the early design of the system; there is no performance analysis or benchmark test results. It is thus very hard to predict Soda’s performance. Although there are some problems in the current implementation, Soda still seems well suited for use as one component in a time-sharing, departmental computing system.

Queensland University of Technology (QUT), Brisbane, Australia

Paul Roe and Siu Yuen Chan (Queensland University of Technology, Australia) discussed I/O problems in a shared cluster-computing environment in their paper “I/O in the Gardens Non-Dedicated Cluster Computing Environment”. Gardens is an integrated programming language and system designed to support parallel computing across non-dedicated cluster computers, in particular a network of PCs. To utilize non-dedicated machines, a program must adapt to those currently available. In Gardens, this is realized by over-decomposing a program into more tasks than processors, and migrating tasks to implement adaptation. Communication in Gardens is achieved via a lightweight form of remote method invocation. Furthermore, I/O may be efficiently achieved by the same mechanism. All that is required is to support stable tasks that are not migrated – these are effectively bound to resource such as file systems.

The main contribution of this work is to show how efficient I/O may be achieved in a system utilizing task migration to harness the power of non-dedicated cluster computers. Some preliminary performance comparison of standard Unix file I/O and file I/O using processor bound objects is also presented. The difference in times between the native Unix I/O and bound object I/O may be accounted for by the task context switch overhead. Some comparisons were also made with other related approaches.

The I/O work on Gardens is in its early stages. Several key issues are still to be developed, such as sophisticated cache mechanisms, as well as parallel interactive, and blocking I/O.

National Tsing Hua University, Taiwan

Hau-Yang Cheng and Chung-Ta King (National Tsing Hua University, Taiwan) explored data reliability and availability issues with a parallel I/O system in a NOW environment. Their paper, "File Replication for Enhancing the Availability of Parallel I/O Systems on Clusters" investigates the availability issues in parallel I/O systems with a shared-nothing disk configuration. Normally, there are two basic approaches to achieve a high degree of data availability in a parallel I/O system, data replication with multiple copies of same data stored on different disks and data spreading across an array of disks along with redundant error detection/correction information.

The authors first review several existing data replication techniques, such as mirrored disk, distorted mirroring, interleaved de-clustering, and chained de-clustering. A new file replication method, called *Locality Aware File Replication* (LAFR), is proposed, that takes into account file access patterns and data locality. This is application specific, because it is necessary to consider how the given sets of applications access the file and then attempt to match their locality requirements. LAFR is locality aware because it attempts to achieve the best local access ratios for the multiple access patterns of the given file. This allows users to de-cluster a file in such a way that not only its data availability is enhanced, but also system performance is maintained.

This paper not only gives their detailed algorithm for selecting de-clustering and file allocation for LAFR, but also evaluates the proposed de-clustering method by using three applications, and compares with mirroring and chained de-clustering. From the discussion in this paper, LAFR may lead to better data access locality and introduce less communication, while maintaining the same level of availability.

University of Hong Kong

Roy S. C. Ho, Kai Hwang and Hai Jin (The University of Hong Kong) addressed the importance of a single I/O space in building clusters with I/O centric applications. They proposed a new single I/O space scheme for a Linux cluster. The paper, "Single I/O Space for Scalable Cluster Computing" proposed a novel *Single I/O Space* (SIOS) architecture in the Linux kernel for achieving a *Single System Image* (SSI) at the I/O subsystem level. Traditionally, there are three levels for having SSI for disk I/O operation -- user, file system, and device driver. They use the device driver level to implement SIOS, which has advantages of higher transparency, better performance, lower implementation cost, and higher availability.

The design objectives of SIOS at the device driver level are:

A single address space for all data blocks in the cluster.

High availability feature supported. Besides RAID-1 and RAID-5 schemes, they also propose a new RAID architecture, called RAID-x. It uses orthogonal striping and mirroring for all the data blocks in distributed disks.

Performance and storage size scalability. According to the authors' analysis, their SIOS design offers performance and size scalability. Especially for read operations, SIOS scales quite well even under heavy workload.

High compatibility with current cluster applications.

This paper gives a very detailed description of the implementation of SIOS at the Linux kernel device driver level by using a *Virtual Device Driver* (VDD) for each node. There are also some related issues discussed, such as addressing scheme, potential bottleneck in the buffer cache, data consistency, disk partition and portability. Detailed information of this project can be found at web site[24].

6.4 Cluster Scheduling

Australian National University, Canberra, Australia

B. B. Zhou, P. Mackerras, C. W. Johnson, D. Walsh (Australian National University) and R. P. Brent (Oxford University Computing Laboratory) discussed the problem of gang scheduling, which is currently the most popular scheduling scheme for parallel processing in a time-sharing environment. One major drawback of gang scheduling is fragmentation. The conventional method to alleviate this problem is to allow jobs to run in multiple time slots. The paper, "An Efficient Resource Allocation Scheme for Gang Scheduling", illustrates that the conventional method cannot solve the problem of fragmentation -- on the contrary it may degrade the efficiency of system resource utilization. This is due to the increased system scheduling overhead and unfair treatment of small jobs.

The authors introduce an efficient resource allocation scheme, called *job re-packing*. In this scheme, the order of job execution is rearranged on their originally allocated processors so that small fragments of idle resources from different time slots can be combined together to form a larger and more useful resource in a single time slot. When this scheme is incorporated into the buddy-based system, a *workload tree* can be set up to record the workload conditions of each subset of processors. With this workload tree, the search procedure for resource allocation can be simplified and the workload across the processors can also be balanced. Using job re-packing scheme, jobs can be run in multiple slots to significantly enhance system and job performance.

The authors describe details of the job re-packing scheme and how to build the workload tree. Some experiments were also carried out to illustrate the efficiency of the proposed scheme. According to the authors, there is no process migration involved when job re-packing is applied, therefore, the scheme is particularly suitable for clustered parallel computing systems.

The University of Western Australia, Perth

Kamran Kazemi and Chris McDonald (The University of Western Australia) presented a new method to generate parallel process topology specifications in a message passing system in their paper “Formal Specification of Virtual Process Topologies”. During the development of their Virtual Process Topology Environment, they found lack of adequate and flexible topology support in existing message passing systems such as Parallel Virtual Machine (PVM). This parallel programming environment provides high-level abstraction for inter-process communication, relieving the application developer of the cumbersome task of mapping logical neighbors to their task identifiers within message passing systems. Their approach uses a recurrence relation to define topologies, which separates topological specification from the APIs. This provides flexibility to the developers of applications using regular topologies.

The authors incorporate their formal topology definition language into their Virtual Process Topology Environment. Within such a new environment, the recurrence relationships can be passed to the topology server, which is then used in the generation of the topological specification. The experimental results illustrate that the environment is more efficient than the standard PVM group server. Various process mapping strategies can be examined with extreme ease, allowing application developers to determine the best mapping strategy for each application.

[24] http://andy.usc.edu/trojan/

With the aid of their simple visualizer, the opportunities for developers of visualization tools to assist the parallel application developer in visualization of the topologies as well as verification of the topological definition were also presented. Future plans are to integrate the new approach into their PVM version. This work included conditional tracing, visualization of execution topologies, and programming support.

6.5 Cluster Management and Metacomputing

Real World Computing Partnership, Japan

Yoshio Tanaka, Mitsuhisa Sato (Real World Computing Partnership, Japan), Motonori Hirano (Software Research Associates, Japan), Hidemoto Nakada, and Satoshi Sekiguchi (Electrotechnical Laboratory) present a new type of *Globus Resource Allocation Manager* (GRAM) called RMF (*Resource Manager beyond the Firewall*) for wide-area cluster computing in the paper "Resource Manager for Globus-based Wide-area Cluster Computing". RMF manages computing resources such as cluster systems and enables utilization of them beyond the firewall in global computing environments. It consists of two basic modules, a remote job queuing system (*Q system*) and a resource allocator. The Q system is a remote job queuing system, that schedules jobs submitted from global computing sites. The resource allocator manages resource and allocates them for requested jobs. RMF has the following two features:

> It manages multiple computing resources such as cluster systems, supercomputers, and workstations, and provides them global computing environments.
>
> It provides a mechanism to deploy the Globus gatekeeper and the resource manager on individual systems. By deploying the Globus gatekeeper outside the firewall and resource manager inside the firewall, RMF enables computing resources of clusters inside the firewall to be used for global computing.

For communication beyond the firewall between job processes, they designed the Nexus Proxy, which relays TCP communication links beyond the firewall. RMF and the Nexus Proxy provide a global computing environment in which users can easily utilize such parallel systems as cluster systems and supercomputers beyond the firewall. Since firewalls are obstacles to constructing global computing environments, a RMF-type GRAM is suitable for building Globus-based wide area cluster system.

Besides the issues associated with RMF, the authors also discussed other issues for wide-area cluster computing, such as optimization of a resource allocation algorithm, executable management on remote resource, programming environments.

6.6 Cluster Operating Systems and Monitoring

The University of Hong Kong

Zhengyu Liang, Yundong Sun, and Cho-Li Wang (The University of Hong Kong) reported on the design of an open, flexible and scalable Java-based cluster monitoring tool, called ClusterProbe, in the paper "ClusterProbe: An Open, Flexible and Scalable Cluster Monitoring Tool". ClusterProbe provides an open environment by developing a Multiple-protocol Communication Interface (MCI) that can be connected to various types of external accesses from the clients.

The MCI can support various communication protocols, such as RMI/CORBA, HTTP/HTML, TCP, and UDP. A communication adaptor provides access to a monitoring tool through a particular communication interface. Multiple adaptors are used to provide transparent communication services for external access. ClusterProbe adopts the pre-formatting modules to offer the resource information to client in a user-preferred format, e.g. raw data, compact data, security packets or customized objects.

ClusterProbe is flexible so that the monitoring tool can be easily extended to adapt to resource changes by using the Java-RMI mechanism to gather and transmit data. In addition, the design of ClusterProbe allows it to scale extensively because of its cascading hierarchical architecture.

Several useful services are implemented based on ClusterProbe, including the visualization of cluster resource information in various forms and cluster fault management by using a global event facility. ClusterProbe has been used to assist the execution of a cluster-based scalable WWW search engine (SWSE) and a distributed object-oriented N-body application. The authors also compared ClusterProbe with other related projects and products, including the Java Dynamic Management Kit (JDMK), GARDMON, K-CAP, DOGMA, PARMON[25], Node Status Reporter (NSR), Cluster Administration using Relational Databases (CARD). All experiments demonstrated high efficiency and good performance improvement.

6.7 Cluster Programming and Analysis Models

Queensland University of Technology, Brisbane, Australia

Darren Butler and Paul Roe (Queensland University of Technology) reported on their work developing extensions to Gardens in their paper "Sharing the Garden GATE: Towards an Efficient Uniform Programming Model for CLUMPS". The Gardens project aims to create a virtual parallel machine out of a network of workstations with low latency inter-machine communication links. It integrates a programming language and system to support parallel computation over a network of workstations. The *Gardens Application Tasking Environment* (GATE) kernel’s responsibilities include the maintenance and scheduling of tasks as well as facilitating inter-task communication through global objects.

The extension of GATE allows Gardens to efficiently support clusters of SMP machines under a uniform programming model. These changes allow Gardens to benefit from the high-performance interconnections between local processors while maintaining all the existing benefits of distributed systems. Such support requires the implementation of high-performance shared memory message passing primitives as well as changes to the existing model invariants and the Gardens runtime system itself. There exist two major design issues for high-performance message passing protocols on SMP architectures: the minimization of cache coherent transactions, and the management of concurrent access of the message queues. The first issue has ramification for the performance of the protocol. The second issue involves ensuring integrity of the message queues themselves.

The new communication primitives have demonstrated a minimum latency of 2 microseconds and can maintain a maximum bandwidth of approximately 194 MBytes/s on their test platform. The NAS benchmarks, EP and CG have also demonstrated significant performance increases when parallelized by Gardens. The major contribution of this work is a system that efficiently supports multiprocessor computers in a network of workstations with a uniform programming model. Such a system promises to simplify the task of developing high-performance parallel applications to run on CLUMPS. For detail information of Gardens project, please refer web site at[26].

[25] http://www.dgs.monash.edu.au/~rajkumar/parmon/

[26] http://www.plasrc.qut.edu.au/Gardens/

Deakin University, Melbourne/Geelong, Australia

Michael Hobbs and Andrzej Goscinski (Deakin University, Australia) presented a unique remote and concurrent process duplication method on *Clusters of Workstations* (COWs) in the paper “The Influence of Concurrent Process Duplication on the Performance of Parallel Applications Executing on COWs”. COWs are able to present users and application programmers with a considerable computational resource for relatively low cost. However, software currently used to support application programmers in the development and execution of their parallel programs is often limited and lacks transparency. The initialization stage of a parallel program can prove critical in obtaining good performance and engaging programmers, especially as the number of parallel process increases. The focus of this work was to present a unique process instantiation service that allows a set of parallel child processes to be concurrently duplicated across a COW through the use of group process migration and group communication services.

The concurrent process duplication mechanism is a key component of the process instantiation system, which combines advanced multiple process duplication services with group process migration and group communication to support the execution of SPMD based parallel application on COWs. The concurrent process duplication mechanism was designed and developed as an integral component of the GENESIS operating system specifically to support parallel processing. The concurrent process duplication service was shown to considerably improve the execution performance of loop parallel applications, including Successive over Relaxation (SOR) and Quicksort, when compared to the traditional single process duplication and multiple process duplication services.

The University of Western Australia, Perth

The ability to debug a parallel program with topological knowledge, either explicitly provided or not, is very useful. Simon Huband and Chris McDonald (The University of Western Australia) presented a methodology to identify program topologies using only standard trace facilities. Their paper "Debugging Parallel Programs using Incomplete Information" introduces preliminary work towards developing a complex, automated, topological debugger. Topological information can be exploited in identifying communication errors. Knowing how a program's processes map to their topology allows the graphical display of the program in a manner reflecting the correct topology. The authors demonstrate that under certain circumstance, their methodology is sufficient for debugging purposes. This methodology uses the concept of distance between graphs to deduce the topology of a program given a list of known topologies.

To demonstrate the feasibility of their approach, three related polynomial time generic algorithms were developed (that attempt to determine the minimum distance between a pair of graphs) and implemented on five topologies common to parallel processing: ring, grid, torus, butterfly, and hypercube. Although, in the worst case, calculating the distance is costly, their results demonstrate that when considering topologies that are not considerably corrupted, the distance problem is tractable. Specifically, by using only simple, generic algorithms, the results demonstrate it is possible to identify ring, grid, and hypercube topologies.

In environments such as PVM and MPI, where topologies are not explicitly specified, they have sufficient evidence to suggest that their approach of identifying topologies with no explicit knowledge is plausible. Given the quantity of programs that conform to regular topologies, coupled with usefulness of debugging from topological point of view, their research is an interesting step towards the development of parallel program debugger.

7. CLUSTER COMPUTING BOOK

Rajkumar Buyya from Monash University, with collaboration and contributions from leading experts from all over the world has brought out a two volume book on cluster computing subject published in the US by Prentice Hall.

> High Performance Cluster Computing: Architectures and Systems, R. Buyya (ed.), *Prentice Hall*, ISBN 0-13-013784-7, 1999.
>
> High Performance Cluster Computing: Programming and Applications, R. Buyya (ed.), *Prentice Hall*, ISBN 0-13-013785-5, 1999.

A number of universities have adopted the above books in their graduate courses. They include Technische Universität München (Germany), Syracuse University (USA), Indian Institute of Technology (Delhi, India), Universidad de los Andes, (Colombia - South America), Vrije Universiteit, Amsterdam (The Netherlands), Australian National University, (Canberra, Australia), University of São Paulo (Brazil), State Technical University of Novosibirsk (Russia), Åbo Akademi University (Finland), and others. The book has also received good reviews in IEEE periodicals.

The editor of the book has set up an Internet resources home page called Cluster Computing Info Centre at: http://www.csse.monash.edu.au/~rajkumar/cluster/. This home page provides lecture material and pointers (hot links) to a number of freely downloadable software modules for clusters and related information developed by both academic and commercial researchers.

8. SUMMARY AND CONCLUSION

Cluster computing is becoming increasingly popular and the latest technological developments and research innovations are pushing clusters into mainstream computing. This poses a number of new research challenges that need to be addressed, particularly in the areas of resource management, scalability, expandability, efficient communication, system administration, and single system image. Some of the work described here contributes towards addressing these challenges, and a number of them have commercial potential.

A panel session held at the IEEE International Workshop on Cluster Computing (IWCC'99) in Melbourne highlighted a number of current key issues in cluster computing. They are as follows.

- Cluster computing is primarily and will continue to be a major area of research interest for high-performance and high-end computing community.
- Within Australia, special issues such as wide area clusters and sharing of scarce resources are important and are therefore drivers for software research.
- Application areas including e-commerce and business systems are likely to be major drivers for cluster computing in addition to the more traditional areas such as scientific and engineering applications.

9. CONTACTS

ATIP acknowledges the following contributors in the development of this report (all members of the

IEEE Task Force on Cluster Computing):

Mark Baker (Mark.Baker@port.ac.uk), University of Portsmouth, UK.
*Rajkumar Buyya (rajkumar@csse.monash.edu.au), Monash University, Australia.[coordinator]
Ken Hawick (khawick@cs.adelaide.edu.au), University of Adelaide, Australia.
Heath James (heath@cs.adelaide.edu.au), University of Adelaide, Australia.
Hai Jin (hjin@ceng.usc.edu), Hong Kong University / University of Southern California, USA.

The contents of this report are a result of the authors' understanding of the work of various researchers in the Asia Pacific region, with particular focus on Australia. We have tried to cover as many relevant Australian activities as possible. The original material can either be found at various URL links cited here, or in Proceedings of the International Workshop on Cluster Computing (IWCC'99) published through the IEEE Computer Society Press, USA. All sources are gratefully acknowledged.

For further information about TFCC membership, its planned and future activities, please browse the TFCC home page[27] or contact TFCC co-chairs:

Rajkumar Buyya
School of Computer Science and Software Engineering
Monash University
C5.10, Caulfield Campus
Melbourne, VIC 3145, Australia
Phone: +61-3-9903 1969
Fax: +61-3-9903 2863; eFax: +1-801-720-9272
Email: rajkumar@csse.monash.edu.au | rajkumar@ieee.org
URL: http://www.csse.monash.edu.au/~rajkumar | http://www.buyya.org

Mark Baker
School of Computer Science
University of Portsmouth,
c/o Milton Campus,
Southsea, Hants, UK
Phone: +44 1705 844285
Fax: +44 1705 844006
E-mail: Mark.Baker@port.ac.uk
URL: http://www.dcs.port.ac.uk/~mab/

END OF REPORT ATIP00.015r

atip

[27] http://www.ieeetfcc.org/