High Flexibility Multi-Platform Hybrid Computer Cluster

Shuo Liang

School of Computing and Engineering

University of Huddersfield

A thesis submitted to the University of Huddersfield
in partial fulfillment of the requirements for

MSc by Research

2012

To my Parents
for their unending support and encouragement

献给我亲爱的父母
为了他们无尽的支持和鼓励

Acknowledgements

I thank my supervisor Dr Violeta Holmes, for her supervision and support on my undergraduate and Master. It would be very tough to complete such a challenge, without her endless support and encouragement.

Very deep appreciation must be expressed to my colleague and friend, Ibad Kureshi. He is the one backing up me in the office all the times during these two years. The author especially thank to David Gubb and Yvonne James for their solicitude and encouragement.

Special gratitude to my friend Yu Gao, for her solicitousness, inspiration and accompany in the past.

The gratefulness to my parents is far beyond words.

Copyright Statement

Abstract

Along with the development of computer hardware and open source software, the Beowulf Clusters have become an economic and practical choice of small-and-medium-sized institutions to shorten their research cycle. In many growing Universities, departmental clusters are being set up on the departmental demand for running simulations, rendering and other calculations. Hence, in the University, individual departments may have their own HPC resources.

However, the staff and students of these Higher or Further Education Institutes are familiar with various software, which run on different operating systems, e.g. Windows and Linux. Furthermore, the platform-crossing software generally gives Windows users easier and friendlier interface. To support these software, their small-scale computer clusters have to be divided in two or more computer clusters. Although the virtualisation technology could be an effective choice, many of the Beowulf Clusters, which are built by legacy hardware, do not support the virtualisation.

As a result of our research work, a cluster middleware based on OSCAR 5.2 beta 2 is being developed to achieve the function of implementing a Linux-Windows Hybrid HPC Cluster, which holds the characters of each operating system and accepts and schedules jobs in both batch job schedulers. By using Linux CentOS 5.5 with the improved OSCAR middleware with Windows Server 2008 and Windows HPC 2008 R2, a bi-stable hybrid system has been deployed at the University of Huddersfield. This hybrid cluster is known as the Queensgate Cluster.

Contents

List of Figures

List of Tables

List of Abbreviation

API	Application Program Interface
BSD	Berkeley Software Distribution
COTS	Commercial Off-The-Shelf
DHCP	Dynamic Host Configuration Protocol
DLL	Dynamic-Link Library
EXT	Extended File System
FAT	File Allocation Table
GRUB	GNU GRand Unified Bootloader
GUI	Graphical User Interface
HPC	High-Performance Computing
HPC-RC	HPC Resource Centre
IEEE	Institute of Electrical and Electronics Engineers
LAN	Local Area Network
LDAP	Light Weight Directory Access Protocol
MAC address	Media Access Control address
MBR	Master Boot Record
MPI	Message Passing Interface
NAS	Network-Attached Storage
NFS	Network File System
NGS	National Grid Service
NIC	Network Interface Controller
NTFS	New Technology File System
NW-Grid	North West Grid
OSCAR	Open Source Cluster Application Resources
PBS	Portable Batch System
POSIX	Portable Operating System Interface for Unix
PXE	Preboot eXecution Environment
QGG	QueensGate Grid
RHEL	Red Hat Enterprise Linux
ROM	Read-Only Memory
SDK	Software Development Kit
SMP	Symmetric MultiProcessing
SSH	Secure SHell
SSI	Single System Image
SVN	SubVersioN
TCP/IP	Transmission Control Protocol and Internet Protocol
TFTP	Trivial File Transfer Protocol
TORQUE	Terascale Open-Source Resource and QUEue Manager
WAIK	Windows Automated Installation Kit
XML	Extensible Markup Language

Chapter 1

Introduction

Due to the movement of open-source software and the development of general-purpose computer hardware mainly multi-core computer processors, building a low cost supercomputer from general-purpose computers and open-source software has become an economic and practical choice of small-and-medium-sized institutions to shorten their research cycle. In many growing Universities, departmental High-Performance Computing (HPC) resources are being set up on demand for running simulations, rendering and other calculations.

As a medium-sized Higher Education institution, the University of Huddersfield is still finding its place within the research community. The High-Performance Computing Resources Centre (HPC-RC) was built to aid new research efforts. This project has contributed to the design and implementation of these resources.

1.1 High-performance computing and Computer Cluster

Today, more and more researchers or institutions have established or found their suitable open-source projects or proprietary software for modelling, simulation or other large-scale calculation tasks based on High-Performance Computing resources.

The High-performance computing centralizes computing resources for solving vast problems that a general-purpose computer could not solve or find very difficult to solve. These kind of resources was a luxury for

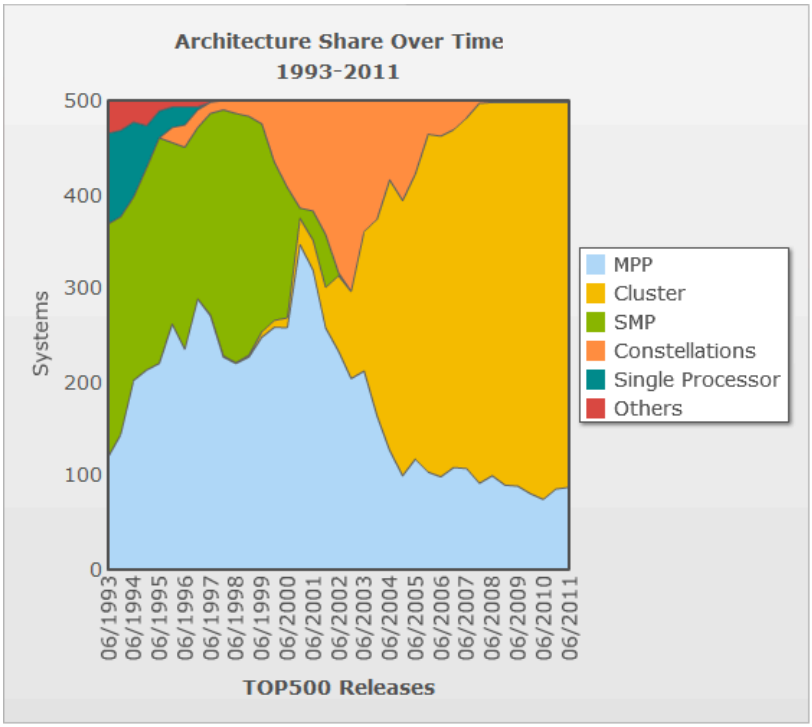

Figure 1.1: Architecture share of TOP500 in performance over time(TOP500.Org, 2011)

a small/middle-sized institution or company before the computer cluster had become an economical solution.

As Figure 1.1 shows, before 1999 in the TOP500, which is a website promoted by German company 'Prometeus GmbH' lists 500 fastest computer systems twice a year since 1993, the architecture of supercomputers were mainly Symmetric Multi-Processing (SMP) and Massively Parallel Processing (MPP) systems, whose system needed to be specially designed for working together. Since 1999, the computer cluster system has become the main stream of HPC.

Rather than SMP or MPP, a computer cluster is a group of computers, each of which is independent and complete computer system, linked by network, that work together to represent functions of a single computer. The Beowulf cluster is even built-up by commodity computers.

Many universities in the UK started to establish their departmental and campus clusters in the last decade. For example, a computer cluster at the University of Nottingham was built in 2004/5, and it had hit the June 2005 TOP500. (TOP500.Org, 2011)

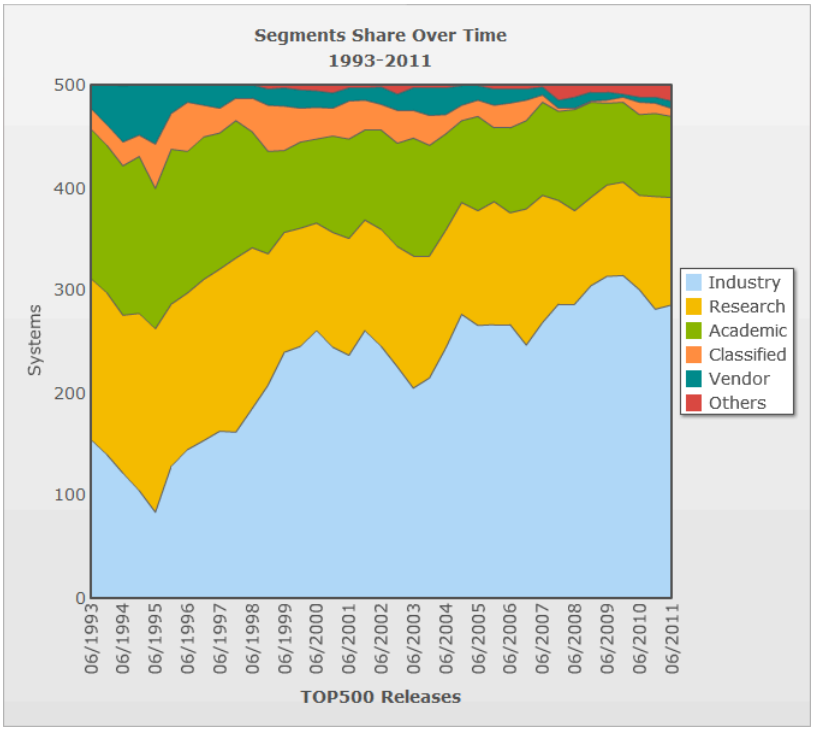

Figure 1.2: Swift growth of research activity(TOP500.Org, 2011)

Figure 1.2 shows these HPC resources are not confined to the laboratories of research institutions and universities. The industrial companies are also using HPC to model and simulate their processes, deploy a variety of hardware and software architectures, open source and commercial products, such as Windows and Unix-like systems.

For example, Audi: "Scheuchenpflug never considered using Linux as the operating system for the high-performance computing needed for lighting simulation, even though Audi uses Linux computing clusters almost exclusively for classical simulations, such as crash and flow analyses. "It's simple—our CAD system, CATIA V5, is a Windows application," he says. "And it was very important to us to be able to work in a seamless environment."" (Microsoft Corporation)

1.2 The University of Huddersfield HPC Resources

In recent years, a few research groups in University of Huddersfield had started building department level HPC resources on their demand. By the

Software Name	Description
Abaqus	Finite Element Analysis
Amber	Assisted Model Building with Energy Refinement aimed at biological systems
Backburner	Rendering software for 3ds Max
Blender	Open Source 3D Modeller and Renderer
CASTEP	CAmbridge Sequential Total Energy Package
COMSOL	Multiphysics Modeling, Finite Element Analysis, Engineering Simulation Software
DL_POLY	General purpose classical molecular dynamics (MD) simulation software
ANSYS FLUENT	Computational Fluid Dynamics (CFD)
GAMESS-UK	Molecular QM code
GULP	General Utility Lattice Program
LAMMPS	Large-scale Atomic/Molecular Massively Parallel Simulator
MATLAB	Numerical Computing Environment
METADISE	Minimum Energy Techniques Applied to Defects, Interfaces and Surface Energies
NWChem	Multi-purpose QM and MM code
OpenFoam	
Opera	Finite Element Analysis for Electromagnetics

Table 1.1: Applications on QGG

aim of stimulating the demand for HPC at the University, using existing hardware and free software, a project to establish such resources had been launched and successfully completed over the past two years. This project also reallocated the computing resources distributed across the university departments, into the University of Huddersfield HPC Resource Centre.

However, School of Applied Science (SAS), which loaned its rack-mount computer to HPC-RC, are using software based on Unix-like system and Command Line Interface (CLI); the research members in School of Computing and Engineering (SCE), which contributed its tower-case computer, are more familiar with Windows applications and Graphical User Interface (GUI).(Kureshi, 2010)

The software currently used for research and development at the University of Huddersfield are as follows:

To support the above software modelling and simulation, QueensGate Grid (QGG) had been established, which comprises three computer clus-

ters and runs different operating systems:

Normally, for supporting software running on different operating systems, a computer clusters have to be divided into two or even more computer clusters. Although the latest virtualisation technology could be an effective choice, Queensgate Beowulf Clusters, which are almost built by legacy hardware, do not support this technology.The platform-crossing software generally gives Windows users easier and friendlier interface. The aim of this research work was to devise a system that will enable better utilisation of our small HPC resources and support our research community. Our initial objective was to devise a simple dualboot system to support both Windows and Linux platforms for the software in Table 1.1. A number of case studies were conducted to identify the efficiency and ease of use of these software from the scientific and administrative users' point of view.

1.3 Project Objectives

The distinctive objectives of this project are:

- To undertake an in-depth literature review of the existing HPC systems
- To evaluate current solution in HPC hybrid resources management
- To investigate commercial and open-source software for multi-platform system
- To investigate dualboot and multi-boot systems
- Design and develop a dualboot OSCAR and Windows HPC system

Our ultimate goal was to design and implement multi-boot solution that will be both easy to install and maintain by administrators of the clusters, as well as researchers using the HPC resources.

1.4 Dissertation Outline

This dissertation contains 5 chapters.
Chapter 1 (Introductions) discusses the motivation and the background of the project. It identifies the main aim and objectives of our work.
Chapter 2 (Background Research) presents the research into the state of High-Performance Computing.
Chapter 3 (Literature Review) analyses the current solution of multi OS system and multi-boot cluster.
Chapter 4 (Establishing QGG) presents the results of the HPC resources development at the University of Huddersfield.
Chapter 5 (Dualboot QSCAR) presents the design and development of the dualboot OSCAR software.
Chapter 6 (Future work and Conclusion) outlines recommendation for future work and conclusion.
The appendices include information that supports the research work, but is not included in the main body of the dissertation.

The majority of source code is attached in appendixes. In addition, this project is hosted on Google Code open source hosting service: `http://code.google.com/p/dualboot-oscar/`

Chapter 2

Literature Review

2.1 Computer clusters software infrastructure

Computer cluster is a collection of computers that works together and represents an illusion which is created by software or hardware as a single computer to the users. There are several types of computer cluster, which are High-Arability (HA) cluster, Load-Balancing (LB) cluster, High-Performance Computing (HPC) cluster etc.

2.1.1 HPC cluster architecture

A cluster is a collection of computers that appear as a single large computer. As shown in Figure 2.1, the computer cluster architecture consists of three layers: hardware layer, middleware layer, and applications layer. Hardware layer includes all the physical equipments, such as computers (head nodes and compute nodes), network switches and cables, etc. Cluster middleware is a combination of software that helps applications or users utilising the cluster's computational or other functional hardware easily or efficiently. It could have a parallel filesystem that lets files be accessed on any node of the system, or a job scheduler that helps user's tasks running fairly or highly-utilised, etc. Users' HPC tasks and general administration are in the Application layer. The Applications could be parallel or sequential programs. Sequential applications get advantages of the higher CPU speed, larger data storage and running remotely.

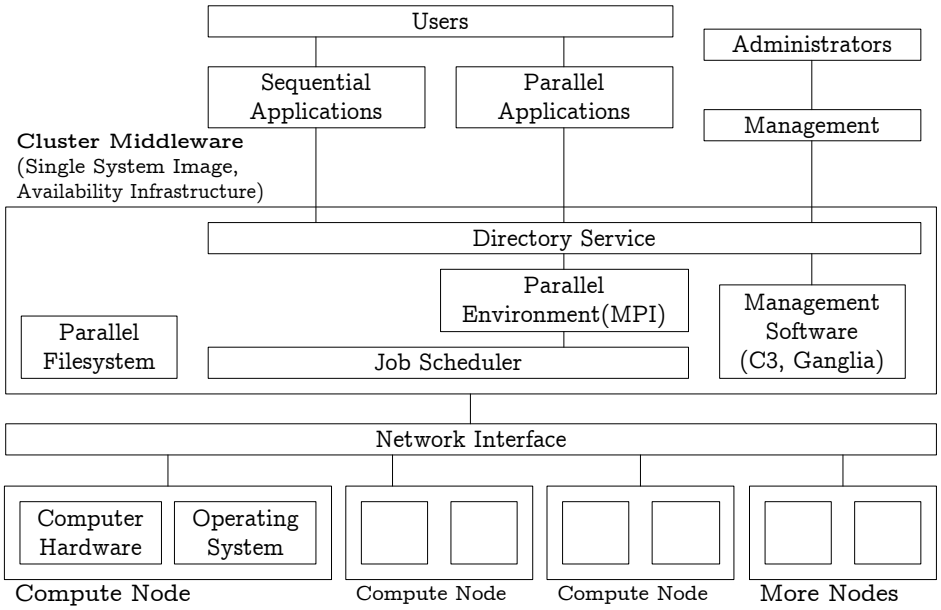

Figure 2.1: Computer Cluster Architecture

Single System Image (SSI) is a concept in distributed computing. SSI presents a computer cluster as one single computer, which has multi-core processors. It is generally implemented in several levels: implementation level, programming level and management level.

Figure 2.2 shows a example of computer cluster structure. It includes one single head node, as a single entry point into the cluster resources, linked to a number of worker nodes with a high-speed network.

Generally, to submit a HPC job, users need to log into cluster system via SSH (Secure Shell) connection, and authenticated by an LDAP (Light Weight Directory Access Protocol), which is used to store their user identities. Users also can submit jobs through a web-based interface, which many groups are developing.

2.1.2 High-performance Networks

The interconnection of computer cluster is an important factor of performance. In parallel computing, due to the Amdahl's law (Amdahl & Sunnyvale, 1967), which points out that a parallel programme speed-up is

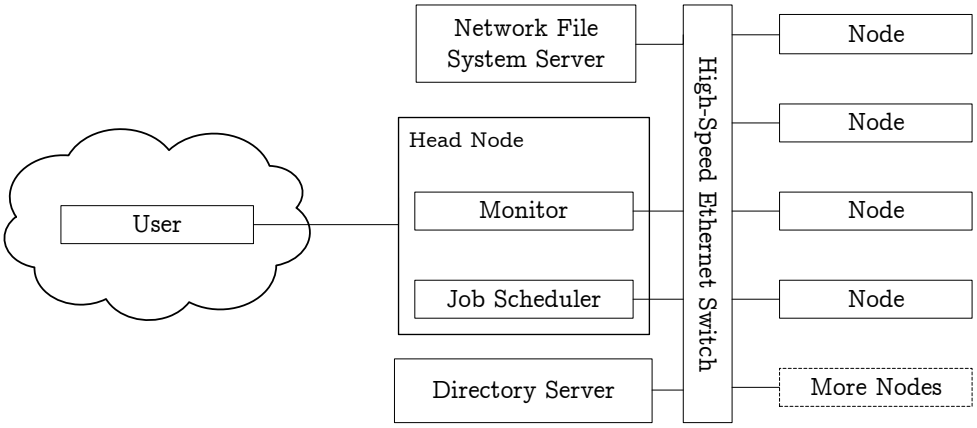

Figure 2.2: Example of a Computer Cluster Structure

limited by its sequential time spending, communication latency may significantly affect the system performance. Since computer cluster entered HPC field, there were four main methods for connecting computer cluster: Myrinet, Ethernet, Fat tree and Infiniband. Ethernet is the current general solution of Local Area Network (LAN), because its characteristic meets the ordinary usage of commodity PC. Myrinet, Fat tree and Infiniband, are designed for High-performance requirement.

2.1.3 Operating system for computer cluster

Operating system is a set of programmes, which manages hardware and software resources, provides Application Program Interface (API) for other applications between hardware and software.

2.1.3.1 Unix family and GNU/Linux

Unix is an operating system initially developed in AT&T Bell Laboratories. It was published with the source code and AT&T could not commercialise it in the first ten years, because AT&T was asked to authorise its non-telephone technology to anyone who requested it. Therefore, Unix became very popular all over the world. After the restriction had been raised, Unix then had become proprietary and closed-source software. To maintain open source version of Unix, The Berkeley campus of the University of California was distributing their variant of Unix, which is called

Berkeley Software Distribution (BSD), free and under BSD license, which has loose regulations for redistribution.

While Berkeley was involved in the lawsuit settled by Unix Systems Laboratories (USL), Berkeley could not publish their BSD for Intel x86-architecture. During this time, GNU/Linux, which is an open-source implementation for Unix interface POSIX (Portable Operating System Interface for Unix), had become the mainstream on x86 platform. Currently, the Linux family's share in TOP500 June 2011 is 91.2% (TOP500.Org, 2011).

General Linux distributions for clusters are in Table 2.1:

Distribution	Derivatives	Package management system
Red Hat Enterprise Linux (RHEL)	CentOS, Oracle Linux and Scientific Linux	RPM-based
Debian	Ubuntu	Debian-based
Slackware	SUSE, OpenSUSE	Various
Gentoo		Portage

Table 2.1: General Linux distributions for clusters

There are a number of proprietary software like IBM AIX, HP-UX, and Solaris. Their share in TOP500 is getting less.

2.1.3.2 Windows family

Unlike the Linux family share in the TOP500, Windows family has dominated the desktop market. A statistics from StatCounter shows the market share of Windows family is over 91%.(StatCounter, 2011) A large group of users from academia and industry and even computing field are not familiar with Unix-like system and Command Line Interface.

Microsoft entered the HPC field in 2006. The Windows Compute Cluster Server (WCCS) 2003 is their first HPC product, which provides the function of deployment, management, and other utilities. (Microsoft Corp., 2006)Microsoft had formally released "HPC Pack 2008 R2" in 2010. However, the HPC Pack is not the only way to construct a Windows cluster. Some third party software can also provide the function of deployment or management.

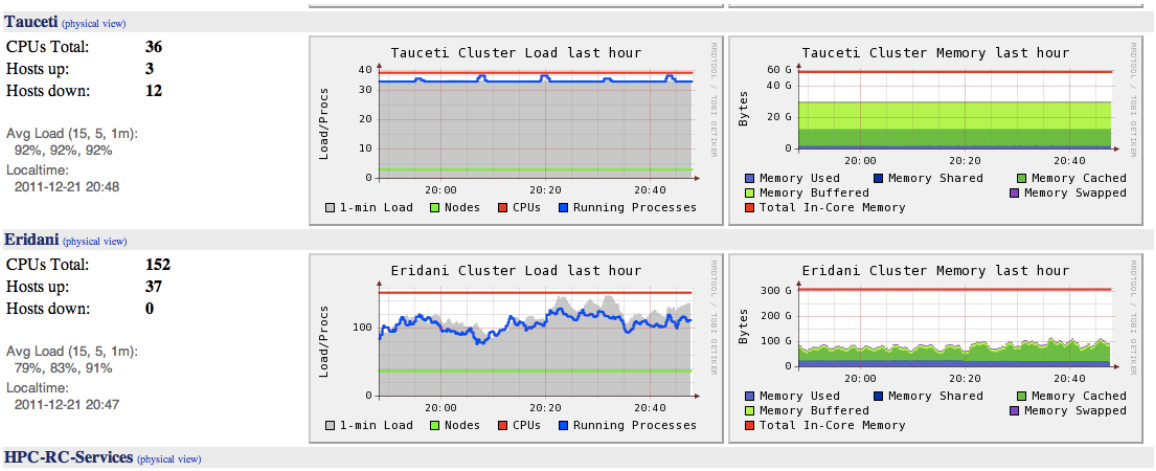

Figure 2.3: Ganglia page of "Eridani"

A render software BlackBurner which is from Autodesk 3Ds MAX, only runs on Windows and is not officially compatible with other job scheduler e.g. Windows HPC scheduler.Therefore, HPC resources need to have a provision for both Linux and Windows Operating Systems support.

2.1.4 Cluster Middleware

Cluster middleware refers to many things. It is the software for enabling the parallel computing such as deployment, job scheduling, hardware monitoring, Message Passing Interface (MPI) for parallel communication or other tasks. It could come separately or as single package/toolkit like OSCAR, ROCKS and HPC Pack 2008 R2 or as an operating system distribution MOSIX.

2.1.4.1 Individual programmes

Monitoring (Ganglia) "Ganglia" is an open-source monitoring programme for HPC system under BSD-license. It is widely used on cluster systems and grids for monitoring CPU, Memory, Network activities , etc.

Job scheduling in HPC cluster There is a number of job schedulers for HPC management. Due to their licences have different restrictions, different scheduler are used for commercial or non-commercial purpose.

TORQUE Resource Manager TORQUE (Terascale Open-Source Resource and QUEue Manager) is developed from OpenPBS by Adaptive Computing Enterprises Inc. TORQUE could use its build-in scheduler pbs_sched or others like "Maui" which is another job scheduler developed by Adaptive Computing Enterprises Inc. Whatever the scheduler is, it does not affect the using of the commands of PBS as the front-end. OSCAR is using TORQUE as the cluster resource manager.

C3 C3 (Cluster Command & Control) is a command line interface tool for cluster management. It has a set of commands that can execute shutdown nodes, delete files, copy file from nodes, deploy image to nodes and so on, on a range of particular nodes by a single command. C3 has been included in OSCAR.

2.1.4.2 Collection of programmes

OSCAR OSCAR (Open Source Cluster Application Resources) is a collection of open-source tools that enables users to install and administer their Beowulf HPC Linux cluster with a friendly user interface. It contains a group of essential programmes to manage and program the HPC cluster. Its 5.x version mainly supports RHEL, CentOS, Fedora, openSUSE and YellowDogLinux (for IBM PowerPC CPU e.g. Sony PlayStation 3). The 6.x version supports RHEL 5/CentOS 5, Debian 5 and Ubuntu 10.04 (LTS). "In practice, it might be more fitting to say that OSCAR delays the need for expertise and allows you to create a fully functional cluster before mastering all the skills you will eventually need." (Sloan, 2004) It is a proper software for the learner of computer cluster. However, OSCAR project is now less active than ROCKS.

ROCKS ROCKS is a complete "cluster on a CD" solution for Linux clusters. It does not require much knowledge in building cluster, and have most essential open-source tools for computer cluster. Unlike OSCAR which is supporting multiple Linux distributions as an independent application, ROCKS come as a distribution based on CentOS (Red Hat family). Both OSCAR and ROCKS provide guided graphic interface to install and manage clusters. "It is for this reason the OSCAR has been

chosen as the proposed systems middleware as in case the e-Science community moves to another operating system that is not similar to CentOS or Scientific Linux, the same middleware can be used to reduce a layer of complexity as a new system is deployed."(Kureshi, 2010)

Microsoft HPC Server 2008 R2 "Microsoft HPC Server 2008 R2 provides a robust, scalable, cost-effective, and easy-to use HPC solution. Windows HPC Server 2008 R2 can integrate with an existing Active Directory® directory service infrastructure for security and account management; and can use Microsoft System Center Operations Manager for data center monitoring." (Microsoft Corp., 2010) Microsoft HPC server, as a commercial software targets Windows users, provides the User Interface and functions more friendly. Most operations could be done in the graphic interface. It also offers complete resources for Developers.

SSI system cluster The SSI cluster e.g. OpenSSI, MOSIX (Multicomputer Operating System for Unix) and z/VM are developed for achieving the goal of Single System Image. The advantage is they can run normal programmes without modifying source code. However, they do not use PBS-like method to manage computing resources. Instead, they are load-balancing cluster. They allocate their resources dynamically.

2.1.5 Portable Programming Environment

IEEE POSIX The IEEE (Institute of Electrical and Electronics Engineers) POSIX (Portable Operating System Interface for Unix) is a standard, whose aim is to support the portability of programmes among different UNIX System environments. Developer can easily migrate UNIX programme with few change of source code. As they are UNIX-like system, The Linux distributions, support various version of IEEE POSIX. The current Windows kernel supports POSIX in a subsystem, however it is an early POSIX standard from 1990.

MPI As the platforms are various, portability is also important in parallel programming. MPI (Message Passing Interface) is an interface standard for parallel computing in C/C++ and Fortran. Similar to POSIX, it formulates the standard rather than implementation. Therefore, users can run specialised MPI compiler from vendors of hardware e.g. Intel MPI compiler, or use free available version e.g. OpenMPI, MPICH, etc.

Java Language Except making general standard, there are other solution for platform crossing. Java was designed by the purpose "write once, run anywhere." A Java programme can be compiled to a Java bytecode, then be executed in a Java Virtual Machine.

2.1.6 Why do we need more than one operating system? Is there an ultimate choice?

The two leagues of operating system represent the two different design principle, application-driven and user-driven.

A principle of UNIX programme designing is KISS (Keep it simple, Stupid!), which put stability at the top of their priority list. Instead, Windows and Mac, the consumer level software makes the user interactive experience as a priority, although Mac OS X is using an Unix-like system as kernel.

Currently Windows family and Unix family are dominating different market fields. The big share of Windows in consumers field, leads a number of programmer concentrating on learning Windows programming. On the other side, Unix-like system has got a larger share because of the stability, performance and free of use of open-source system. (*Usage Statistics and Market Share of Operating Systems for Websites, November 2011*, n.d.)

The two different philosophies target different markets. In the near future, the gap between the two operating systems will still exist.

2.2 Ability of cross-platform

Different programmes, depending on their developing method and programming language, have more or less difficulty to support multi-platform.

2.2.1 Closed-source software

A lot of the closed-source non-Java software, especially some large-scale Windows software which uses a large number of system exclusive API, have weakness in system platform migration. Since their source code are not published, user can only run the programme on the platform which developer has offered. Although some of the company offers multi-platform supports, others' platform policy would not be changed for a few of users' request.

2.2.2 Open-source applications

Although not every open-source project gives multi-platform support due to developer's finite focus or less demand, an open-source project is capable to be forked from [1] by another new project in order to support new platforms or functions as other developers' need.

However, an open-source project cannot just be simply recompiled to be running on non-official supported. Due to different developing ability, to fork a new distribution is not suitable for every research groups or commercial organisations. It is also not achievable for a small-scale HPC resource centre.

2.2.3 Java applications

As mentioned above, Java programmes can be run on any platforms which has Java Runtime Environment (JRE) support.

2.3 Migration difficulty of large organisations

Many of the large organisations use Microsoft AD (Active Directory) as the directory service solution for account login, data storage. Although Windows AD provides UNIX support, it is not fully compatible with

[1] Fork from: Software Engineering term, means *developed an independent project based on another open-source project's source code.*

LDAP. The computer services of University of Huddersfield, runs two directory service, one is for the Windows PC, another is for Macintosh.

2.4 Platform crossing solution

2.4.1 Subsystem for UNIX-based Applications (SUA)

Windows NT kernel offers a subsystem for a very old version of POSIX (IEEE standard 1003-1:1990.), and the latest version is IEEE standard 1003.1-2008. Since Windows is not paying attention on supporting and developing this system, programmers are forced to use other Windows POSIX emulator, e.g. Cygwin.

2.4.2 System Emulator

Wine Wine is an open-source project for Unix-like system to run Windows applications. The recent stable version 1.2.3 is licensed by GNU LGPL v2.1. It emulates Windows application's running environment by reverse engineering. Therefore, it is not offering perfect capability.

Although Wine could make some Windows applications running in Unix-like environment, the reverse engineering can hardly bring perfect compatibility for different Windows applications and good performance.

Cygwin Cygwin is a Linux emulator for Windows. It provides "a DLL (cygwin1.dll) which acts as a Linux API layer providing substantial Linux API functionality" and "a collection of tools which provide a Linux look and feel environment for Windows"(*Cygwin*, n.d.). Its latest DLL version 1.7.9-1 is licensed by GNU GPLv3.

2.4.3 Virtualisation technology

Virtualisation technology is not a new term. It had been developed and used in the mainframe computers, which are not built by x86 CPUs. Virtualisation has become popular since Intel (VT-x) and AMD (AMD-V)

had started to support hardware-assisted virtualisation for x86 architecture in recent years. With hardware enhancement, the guest OS can achieve better performance and more compatibility.

However, the hardware support wasn't coming with their entire products. "Eridani", A Beowulf cluster of University of Huddersfield built from eliminated laboratory computers, mainly use Intel Core™ 2 Quad-core Q8200 processor that has no virtualisation support. We have to look for other ways to enable the multi-platform cluster.

2.4.4 Multi-boot cluster computer

Multi-boot approach, in in our opinion, suitable for the legacy hardware that has no hardware virtualisation support. A multi-boot computer could be implemented by various approaches.

1. Changing active partition
2. Multi-system bootloader, e.g. GRUB (for Linux as main system), GRUB4DOS (for Windows as main system)
3. PXE also can enable multi-boot function

Pros: Multi-boot solution has wide range compatibility of hardware. It does not require new technology, and there is no performance reduction.

Cons: Reboot takes time, normally about 5mins.

Because of native incompatibility between Windows system and UNIX-like system, and the above-mentioned solutions' limitations, the multi-boot computer cluster system could be a viable solution for legacy hardware.

Dualboot or Multi-boot? The main problem we discovered is the incompatibility between Windows family and Unix-like family, a dualboot system is suitable to solve this major problem.

Chapter 3

Methods of Multi-boot Cluster

3.1 Active partitions

Active partition is a concept of Master Boot Record (MBR), which is used in IBM PC. MBR is stored at the first 512 bytes of a hard disk. It contains the partition table of its primary partitions, flags, etc. Different operating system might have their version of MBR. A flag in MBR marks which is the default boot active partition. By changing the flag, the computer can be loaded to different partitions system. Active partition is an easy method for multi-boot system switching. However, it needs a local disk modification each time it changes OS, which need system-level authority.

3.1.1 Local multi-boot bootloader (GRUB)

GRUB (GNU GRand Unified Bootloader) is a bootloader of mainly Unix-like system. It is the bootloader for most Linux distributions. It could be installed to the MBR section or head of a primary partition. When it is installed on MBR section, it boots any primary partition and logical partition. If GRUB is installed in MBR, it will ignore active partition. Instead, it reads its configuration file and follows its logic.

3.1.2 Remote multi-boot bootloader (PXEGRUB)

PXE (Preboot eXecution Environment) was developed by Intel in 1999. Its aim is to help computer management, e.g. in remote network setup, in

remote network boot (diskless), or in emergency boot up, etc. However, PXE provides environment rather than entire bootloader. It executes other programme (Boot ROM) as bootloader. A general PXE ROM of Linux, PXELINUX, a part of SYSLINUX, has weakness in controlling local partition. It only can turn boot task over to local MBR.

A sub-programme of GRUB PXEGRUB came for supporting PXE. The advantage of PXEGRUB is that GRUB bootloader can be remotely run fully functional, so the configuration could be stored and controlled at server-side rather than client-side.

3.2 Existing solutions

There are several software in developing for platform-crossing cluster solution.

3.2.1 Meta-Scheduler and Moab

"As a meta-scheduler, Moab optimally determines when the OS mix should be modified based upon defined policies and service level agreements as well as current and projected workload. When the specified conditions are met, Moab triggers the OS change using a site's preferred OS-modification technology, such as dual boot, diskfull and/or diskless provisioning or virtualization"(Cluster Resources Inc., 2007)

The concept of meta-scheduler was born for Grid-Computing to manage and unify the resources of different systems. They normally don't replace the local job-scheduling system, but running as a middleware, bridge the meta-scheduler and local-scheduler. The popular grid-middleware, e.g. gLite and Globus, all have meta-scheduling system. Moab is commercial software, which does meta-scheduling between TORQUE and Windows HPC Scheduler.

3.2.2 xCAT

xCAT (Extreme Cloud Administration Toolkit) is an open-source deployment and administration tool for clusters. It supports several operating

systems e.g. RHEL, SUSE, IBM AIX, Windows HPC, and virtual machine platforms.

A paper from IBM (Le, Ghidali, & Thiru, 2010) offered a solution that deploying hybrid cluster of RHEL 5 and Windows Server 2008 R2. However, it needs manually kickstart configure for RHEL 5 and WAIK configuration for Windows Server 2008 R2, and it switches systems manually by changing active partitions.

3.3 A solution for QGG

All these solutions above had been considered not suitable for QGG's hardware model and scale, provision of HPC resources, HPC administrative and support staff at the University of Huddersfield. The hybrid cluster was designed and implemented in the process of establishing the university campus grid, using our unique multi-platform hybrid solution.

Chapter 4

Establishing QGG

4.1 QGG and HPC Resource Centre

The Queensgate Grid (QGG) is established as the University of Huddersfield Queensgate campus' computer grid, which has high throughput and capacity of calculations and data. The development of topology, network file system, virtual machine based server management, HPC development tools, application implementation, hardware installations and configuration have been done while establishing the QGG.

The HPC-RC (HPC Resource Centre) of University of Huddersfield is a One-Stop-Shop for all things HPC on the Campus. It handles computer clusters, HPC license server, HPC directory server and all other HPC resources within QGG. Table 4.1 shows the servers of HPC-RC in various types. "Eridani" and "TauCeti" are the dedicated clusters for HPC usage; the "Mid-night Render Farm" is the utilisation of the desktop computers in the idle time at student laboratory;"Spica" and "Shaula" are Xen server for virtual machines, which holds LDAP server, license server and Condor server, etc.; "Mimosa" is the central network-mounted storage server. The virtual machines are used for enabling the high-availability of critical servers.

Name	Type	Aim of Design
`Eridani`	Tower-case Computer	Hybrid Cluster
`TauCeti`	Rack-mount Computer	Pure Linux Cluster
Mid-night Render Farm	Laboratory Desktop Computer	3Ds Max Render Farm
`Spica`	Rack-mount Computer	Xen Server
`Shaula`	Rack-mount Computer	Xen Server
`ldap`	Virtual Machine	LDAP server
`lrc2`	Virtual Machine	Engineering School License Server
`condor_server_1`	Virtual Machine	Render Farm Condor Server
`Mimosa`	Rack-mount Storage Server	Shared Network Storage

Table 4.1: Servers of QGG

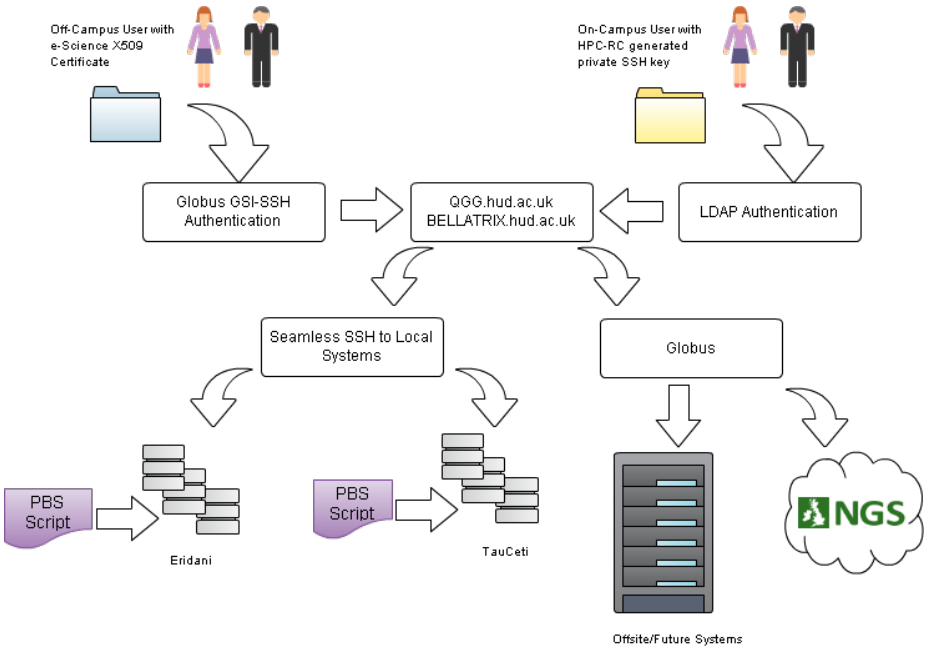

Figure 4.1: The QGG Workflow

4.2 Xen-based High-Availability x86 virtual servers

As the x86 hardware virtualisation have got into most of the server CPUs, managing servers that run different service in a virtualised environment brings a huge convenience for system maintenance. Backing up and restor-

ing of virtual machines are simplified to the operations on the disk image and configure files.

CentOS 5 and Windows Server 2008 R2 are the main systems used in our virtual servers.

4.2.1 LDAP server

Instead of using the existing Active Directory service from university, we configured our own users directory server, a LDAP server, because managing through Active Directory is not flexible and QGG needs independent administration.

LDAP is only a protocol, not an implementation. After several attempts, CentOS Directory Server (also known as Red Hat Directory Server or 389 Directory Server) had been implemented in a dedicated virtual machine. CentOS Directory has a graphic interface for administrators to do add/remove/edit users, import/export database, etc.

By connecting all servers of QGG to the LDAP server, it effectively avoids the user ID conflicts between different machines. Any shutting down of other servers will not affect users logging into a computer of QGG. As one user identity can be used among all these machines, the home folder of each user can be mounted from a dedicated network storage server as well.

4.2.2 SVN server

SVN (Subversion) is a version control system that is mainly used for computer programme projects. It versions all the files you submitted to the server and keeps them for revising.

During the development of QGG, a lot of Linux scripts and documents have been written at a different location of different machine. To manage the development of whole QGG, we have set up a virtual SVN server to manage the programmes we wrote for QGG's front-end and back-end. By bringing all these things into several project in SVN, it was easier to archive what we have created originally.

4.2.3 License server

Most of the commercial applications of HPC requires a local server to fetch the software usage state over the campus network. Another license server `mech1` is a very old computer, which had been working non-stop for many years; the longer it runs, the more chance it might go wrong. License server in virtual machine isolates the software and hardware, gives a more reliable availability and better portability.

4.3 Dedicated NAS (Network-Attached Storage) server for HPC resources

Initially, all the user data were stored at the head node of OSCAR cluster, and shared to the compute nodes by creating a NFS (Network File System) service at head node. As more and more HPC users are joining, and more and more computer cluster or server are joining QGG, having individual storage on each computer brought huge inconvenience for user's data transferring and capacity upgrading.

A dedicated rack-mount shared network storage server, which has been called "mimosa", had been configured to share files to all the servers of QGG, include Linux clusters and Windows clusters.

NFS service for Linux systems NFS is a protocol for remote files accessing via network, which was initially developed by Sun Microsystems. It is widely used in Unix-like systems, and can be easily configured in most Linux distributions. By working with LDAP server, all the QGG user can have same home folder in each machine of QGG.

Samba service for Windows clusters Samba is a open-source implementation of Windows files sharing system support by reverse engineering. "mimosa" enables Samba service for the render-farm machines accessing shared texture files and general using of Windows HPC users.

4.4 HPC applications, developing tools and libraries

4.4.1 Environment Modules

At the beginning of our development, the initial "Eridani" cluster only had the MPI library modules installed by OSCAR, e.g. `openmpi-1.2.4`, `mpich-1.2.7` and `lam-7.1.4`. If any user needs a different version of MPI or any other libraries, they have to compile their own version in their home folder or ask administrator to install for them.

After the NAS server had been set up, all the QGG applications and libraries are transferred to the NAS server. Then, a later version 3.2.8 of the Environment Modules on "Eridani", was provided to make it support "tab" function, which lets users easily find the modules they need by tapping "tab" button.

4.5 Joining Grid Virtual Organisations

QGG have joined the NGS and NW-Grid with mainly the resources of "Eridani". (NGS, 2011) Eridani provides them availability, that running jobs with maximum 32 cores and 48 hours length each .

`http://www.ngs.ac.uk/huddersfield`

4.6 The hybrid cluster in QGG

The cluster in Figure 4.2 is "Eridani", which is the hybrid cluster of QGG. It has 36 COTS (Commercial Off-The-Shelf) tower-only case desktop computers as compute nodes, stacked on a self-made shevlf, as seen in figure 4.2. Each node has an Intel Core Q8200 or Q8300 CPU, 8GB DDR2 RAM, and gigabit ethernet LAN card. The two head nodes run CentOS 5 with OSCAR 5 and Windows HPC 2008 R2.

Figure 4.2: The Hybrid Cluster “Eridani”

Chapter 5

Dualboot OSCAR

OSCAR, the open-source cluster middleware kit, has a friendly and clear interface for setting up a computer cluster. It supports various Linux distributions, which gives us more flexibility, easier maintenance and quicker updates. It was used in University of Huddersfield to deploy a small cluster using COTS components.

5.1 How OSCAR and Windows HPC 2008 R2 Works

5.1.1 General OSCAR deployment

OSCAR provides almost graphical user interface (GUI) in its deployment. After the installation of OSCAR through command-line, users would get an interface as shown in Figure 5.1.

The general OSCAR deployment steps are:

1. Command-line: Head node installation.
2. Graphical: Configure cluster packages and build compute node images.
3. Graphical: Configure deployment method and media. (through network or discs)
4. Graphical: Deployment and complete installation.

Figure 5.1: Installing interface of OSCAR 5.2b1

Unlike ROCKS and RHEL, which use "kickstart", a deployment tool developed by Red Hat, OSCAR includes several small open-source programmes for implementing the deployment: [sic] `systemimager`, `SystemInstaller`, `System Configurator`. These three programmes do the most of deployment. Because of this, OSCAR offers an easier and cleaner way for installing the clusters.

Step 0 to 3, user configures what cluster middleware to be installed on cluster.

Step 4, user specifies what system packages to be installed in compute nodes, and build node disk image.

Step 5 to 6, defining number of computer nodes and method of deployment.

Step 7 to 8, finalizing cluster installation.

In step 4, OSCAR calls "systemimager" to install system packages and cluster middleware into an "image", which is a folder containing most files

in a deployed compute node. The scripts of deployment are generated in the systemimager's folder.

The deployment bash scripts are stored by default at `/var/lib/systemimager/scripts/` as in Figure 5.2. OSCAR's network install mode remotely run these scripts on the compute nodes.

```
hosts      oscarimage.master        oscarnode01.sh  pre-install
hosts.bak  oscarimage.master.orig   post-install
```

Figure 5.2: An example of files list in `scripts` folder before imaging nodes

The original bash script `oscarimage.master` (will be different if we change the default name) formats the whole disk of computer node. Then this script initializes system files, transfers the system image and calls System Configurator to install GRUB, etc.

5.1.2 General Windows HPC 2008 R2 Deployment

Windows HPC 2008 R2 offers a full GUI for system managers as Figure 5.3 shows. Cluster administrators can easily deploy a cluster by using several configurations.

Configure your network Windows HPC 2008 R2 gives 5 different types of network topology. In these five layouts, they all requires at least one enterprise connection to Active Directory, which gives unity for enterprise management but also inconvenience.

1. Compute nodes isolated on a private network.
2. All nodes on both enterprise and private networks.
3. Compute nodes isolated on private and application networks.
4. All nodes on enterprise, private, and application networks.
5. All nodes only on enterprise network.

Provide installation credentials In this step, an authorised AD user can become the administrator of this HPC suite.

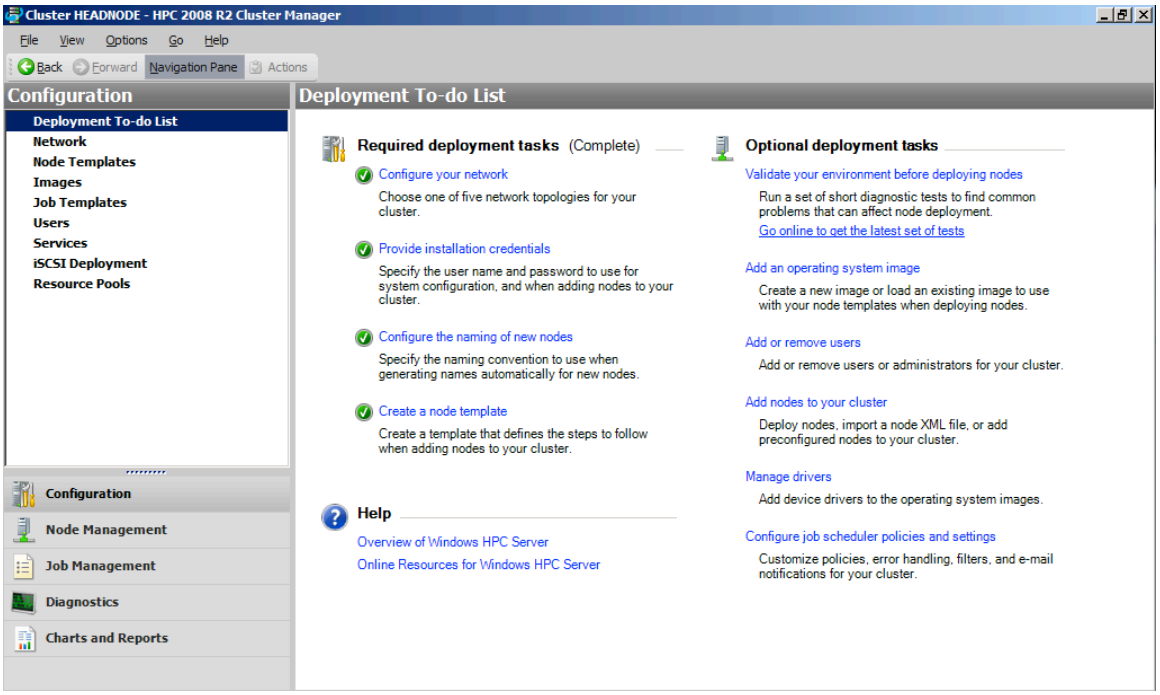

Figure 5.3: Management interface of Windows HPC 2008 R2

Configure the naming of new nodes Defining the hostname of compute nodes in network.

Create a node template There are four types of template: Compute node, Broker, Workstation node, Windows Azure node. Only compute node is required in a basic computer cluster. in the template, Windows Installation Image, Production Key (Volume license) can be specified.

Add new nodes After all above configurations, the head node can deploy new nodes or add pre-configured nodes into nodes list.

In deploying new nodes, it enables PXE to initialise the deployment mode on compute nodes. All the steps will be done automatically. At the beginning of installation, a script for `diskpart`, which is a Windows partitioning programme, in head node `diskpart.txt` is called for disk partition formatting.

5.2 How dualboot OSCAR v1.0 works

Initially the 1.0 version of dualboot-oscar was made for a small-scale cluster (16 machines and 64 cores). The implementation is simple and concise. The figure below illustrates the design of this system. It consisted of two head nodes and a number of worker nodes.

The function of dualboot-oscar can be divided into dualboot controller and dualboot deployment.

5.3 Dualboot controller of v1.0

5.3.1 Local grub loader

The idea to implement dualboot solution came from an article of an IBM engineer (Carter, 2006). It offers a method to switch a single machine between Windows and Linux.

By creating a public partition, which is formatted to FAT (File Allocation Table) file system, a GRUB configuration file `controlmenu.lst` is stored at this FAT partition. The default GRUB's configure file `menu.lst` in the Linux EXT3 partition is linked to the file in FAT partition, by adding the command `configfile` in `menu.lst`. Then both Windows and Linux has been enabled the read/write ability to GRUB configure file. Thus, they both can control default OS. With the disk layout in Table 5.1, and the OS switching script, compute node can be easily switched between two different systems.

For switching multi-system a Perl script `bootcontrol.pl` written by (Carter, 2006) is used to modify GRUB configure file. For our dualboot system, this universal Perl script can be replaced by Windows and Linux batch scripts, which rename filenames among three files, `controlmenu.lst`, `controlmenu_to_linux.lst` and `controlmenu_to_windows.lst`. The two files `controlmenu_to_linux.lst` and `controlmenu_to_windows.lst` are pre-configured and copied into this FAT partition. Figure 5.4 shows the `menu.lst` in it's original place, which is `/boot/grub/menu.lst`.

Partition	/dev/sda1	/dev/sda2	/dev/sda5	/dev/sda6	/dev/sda7
Type	Primary NTFS	Primary EXT3	Logical SWAP	Logical FAT	Logical EXT3
Linux Mount point			/boot	/boot/swap	/
Name in GRUB	(hd0,0)	(hd0,1)	(hd0,4)	(hd0,5)	(hd0,6)
Windows Drive Letter		C:		E:	
Description	Windows	Linux Boot	Swap	dualboot files	Linux
Dualboot files			menu.lst	bootcontrol.pl controlmenu.lst	

Table 5.1: Disk layout of dualboot oscar v1.0 computer node

`menu.lst` is set to redirect to the file `controlmenu.lst` ,which is given in Figure 5.5, in FAT partition for controlling default operating system.

```
default=0
timeout=5
splashimage=(hd0,1)/grub/splash.xpm.gz
hiddenmenu

title changing to control file
        root (hd0,5)
        configfile /controlmenu.lst
```

Figure 5.4: An example of modified `menu.lst`

5.3.2 Batch job submitted to scheduler for system switching

The system switching action is packed as a PBS or Windows HPC job script, which locates a single node, modifies GRUB's configure file, and reboots the machine. The advantage of sending switch orders through job scheduler is that job scheduler can automatically locate free nodes, and all the running jobs can be protected from other accidental operations.

The PBS batch job , which is a BASH script as in Figure 5.6, it books one full node (with 4 cores), changes default boot OS, and reboot. The com-

```
default 0
timeout=10
splashimage=(hd0,1)/grub/splash.xpm.gz

title CentOS-5.4_Oscar-5b2-linux
root (hd0,1)
kernel /vmlinuz-2.6.18-164.el5 ro root=/dev/sda7 enforcing=0
initrd /sc-initrd-2.6.18-164.el5.gz

title Win_Server_2K8_R2-windows
rootnoverify (hd0,0)
chainloader +1
```

Figure 5.5: An example of modified `controlmenu.lst`

mand `sleep 10` is to avoid rebooting action interrupting the OS changing action.

Figure 5.7 is the screenshots of an OS switch job in Windows HPC 2008 R2. This job is similar to the one in PBS, it runs a program to switch default OS, lets system restart, then leaves job idle longer than the time it reboots.

The OS switch job in Windows HPC is stored in a XML format file, it could be submitted by a command line, with submitter's user name and password as clear text. It is more secure than making an individual account for submitting these jobs.

5.3.3 Daemon programmes for queue monitoring

The key to make the dualboot cluster switch idle resources automatically, is the daemon (background) programmes. Two daemon programmes are running at each head node, in order to determine the job queue state then judge the system switching actions by sending the switching batch job.

In the OSCAR head node, PBS does not provide APIs (Application Programming Interface) for other programmes. Several Perl programmes had been written for parsing the output of PBS commands and submitting OS switching job. A C++ programme is written for TCP/IP communication with Windows HPC 2008 R2 head node.

```
#####################################
### Job Submission Script         ###
# Change items in section 1         #
# to suit your job needs            #
#####################################
# Section 1: User Parameters        #
#####################################
#
#!/bin/bash
#PBS -l nodes=1:ppn=4
#PBS -N release_1_node
#PBS -q chemq
#PBS -j oe
#PBS -o _reboot.out
#PBS -r n
#
#####################################
# Section 3: Executing Commands     #
#####################################
echo \$PBS_JOBID >>/home/sliang/reboot_log/rebootjob.log #
   write logs
sudo /boot/swap/bootcontrol.pl /boot/swap/controlmenu.lst
   windows #changes default boot OS
sudo reboot #reboot node
sleep 10 #leave 10 seconds to avoid job be finished before
   reboot
```

Figure 5.6: An example of PBS job An OS switch job in torque

In the Windows HPC 2008 R2 head node, Microsoft provides a SDK (Software Development Kit) for programmes to fetch the data and send tasks, e.g. get the queue state and nodes state. In the dualboot OSCAR v1.0, two programmes are made for fetching queue state and communicating with OSCAR head node. To reduce the difficulty of programming, the communicating programme is compiled in the Cygwin environment.

5.3.3.1 Queue state fetching programmes (detector)

To define the queue state, we define a scheduler is "stuck", when the scheduler has no job running and several jobs queuing. The detector

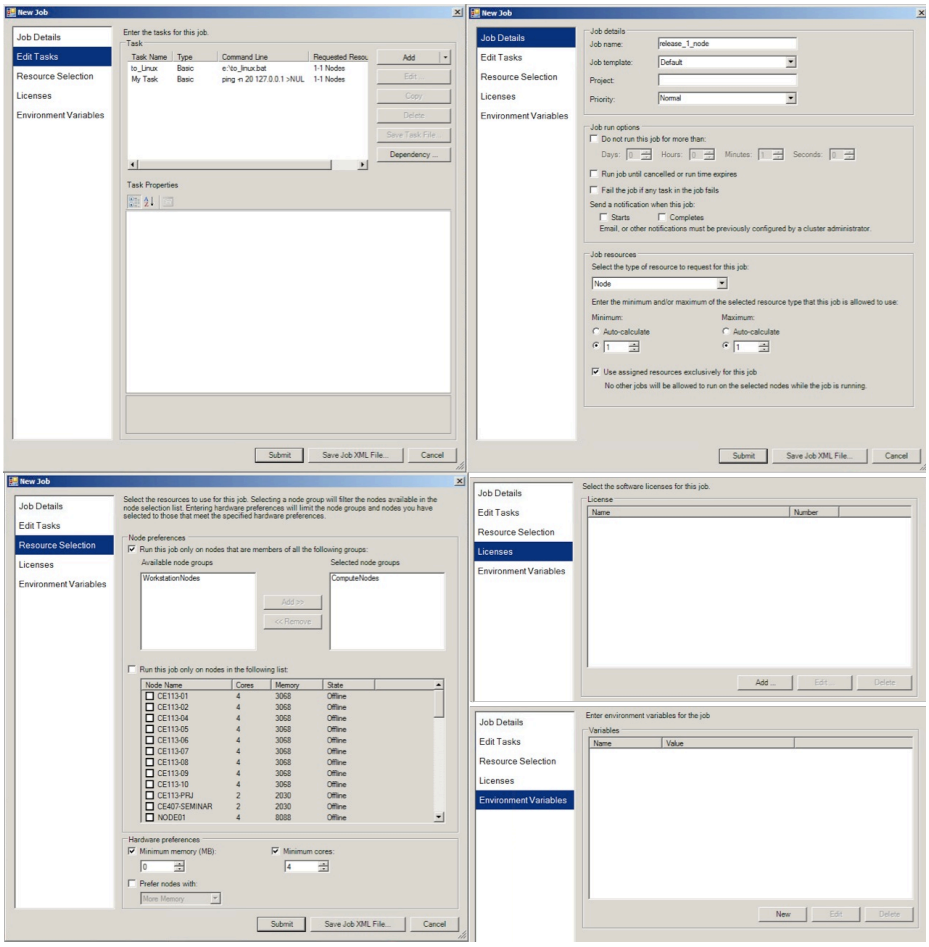

Figure 5.7: An OS switch job in Windows HPC 2008 R2

reads how many compute nodes the first queuing job needs.

The detector gives a text output of scheduler, by parsing the PBS command `pbsnodes` (full detail of nodes in PBS, shown in Figure 5.10) and `qstat -f` (for full detail queue status, shown in Figure 5.11). The first line is the information for the communicator, others are debug information. The format is explained in figure 5.8, which is a character sting sent through network. the detector's outputs in the queue state of running, stuck and others is shown in Figure 5.9.

Char output	0	1-4	5-67	68-
Data	Stuck=1 Other=0	[Needed CPUs] Default=0000	[Stuck job ID] Default="none"	Undefined

Figure 5.8: Output format of detectors

In Windows HPC 2008 R2 head node, the detector fetches data through

the API it provided, and follows the same output format as in figure 5.8

```
[sliang@linhead pbs]$ /dualboot/checkqueue.pl
00000none
Other state
R=0 nR=0

[sliang@linhead pbs]$ /dualboot/checkqueue.pl
00000none
Job running, no queuing.
R=1 nR=0
1186.linhead.queensgate-cls
        Job_Name=sleep
        Job_Ownner=sliang@linhead.Queensgate-CLS
        state=R
        time=2010 04 17 20 11 12

[sliang@linhead pbs]$ /dualboot/checkqueue.pl
100041191.linhead.queensgate-cls
Queue stuck
R=0 nR=1
```

Figure 5.9: Three kinds of outputs of PBS detector (Liang, 2010)

```
enode01.eridani.qgg.hua.ac.uk
        state = free
        np = 4
        properties = all
        ntype = cluster
        status = opsys=linux, uname=Linux enode01.eridani.
           qgg.hua.ac.uk -2.6.18164.el5 #1 SMP Fri Sep 9
           03:28:30 EDT 2011 x86_64,sessions=? 0,nsessions=?
            0, nusers=0, idletime=257163, totmem=15881584kb,
            availmem=15825740kb, physmem=8069096kb, ncpus=4,
            loadave=0.00, netload=154924801596, state=free,
           jobs=? 0,rectime=1271497128
```

Figure 5.10: An example of `pbsnodes` output

```
Job Id: 1185.linhead.queensgate-cls
        Job_Name = release_1_node
        Job_Owner = sliang@linhead.Queensgate-CLS
        job_state = R
        queue = chemq
        server = linhead.queensgate-cls
                 elease_1_node.e1185
        exec_host = node16.Queensgate-CLS/3+node16.
           Queensgate-CLS/2+node16.Queensgate-CLS/1+node16.
           Queensgate-CLS/0
        Priority = 0
        qtime = Fri Apr 16 17:55:40 2010
        Resource_List.nodes = 1:ppn=4
        Variable_List = PBS_O_HOME=/home/sliang,PBS_O_LANG=
           en_US.UTF-8,
                 PBS_O_PATH=/usr/kerberos/bin:/usr/local/bin
                    :/usr/bin:/bin:/usr/X11R6/
```

Figure 5.11: An example of `qstat -f` output (Liang, 2010)

5.4 Deployment of v1.0

The initial dualboot deploying method is achieved by modifying OSCAR's `oscarimage.master` and Windows HPC's `diskpart.txt`.

5.4.1 Modifying OSCAR deployment tool

The deployment of dualboot OSCAR v1.0 requires several manual changes in the deployment script generated by OSCAR. It has to be re-done each time administrator rebuilds node image, which brings a lot of inconvenience in system managing. The changes are adding FAT partition into node image and empty the beginning part of disk for Windows installing.

The standard node image is a pure EXT format located in local disk of head node. To add a FAT partition and a empty partition for Windows, the disk layout file `ide.disk` and the deployment script it generated, `oscarimage.master`, needs to be manually modified.

the main points to be edited are:

```
[sliang@linhead pbs]$ /dualboot/checkqueue.pl
00000none
Other state
R=0 nR=0

[sliang@linhead pbs]$ /dualboot/checkqueue.pl
00000none
Job running, no queuing.
R=1 nR=0
1186.linhead.queensgate-cls
        Job_Name=sleep
        Job_Ownner=sliang@linhead.Queensgate-CLS
        state=R
        time=2010 04 17 20 11 12

[sliang@linhead pbs]$ /dualboot/checkqueue.pl
100041191.linhead.queensgate-cls
Queue stuck
R=0 nR=1
```

Figure 5.12: Three kinds of outputs of Windows HPC 2008 R2 detector (Liang, 2010)

1. Reserved space in `ide.disk` by adding partitions of Windows and dualboot FAT.
2. In `oscarimage.master`, replace `mkpart` by `mkpartfs`, to make FAT works proper.
3. Add `-modify-window=1 -size-only` argument to `rsync` commands, to support syncing FAT format partitions.
4. remove the lines of Windows partition in `fstab` and unmount commands to avoid errors.

5.4.2 Patching Windows HPC deployment tool (simple)

Windows HPC has stored its configure file in a clear-text file, which is `C:\Program Files\Microsoft HPC Pack 2008 R2\Data\InstallShare\Config\diskpart.txt` shown in Figure 5.13.

```
select disk 0
clean
create partition primary
assign letter=c
format FS=NTFS LABEL="Node" QUICK OVERRIDE
active
exit
```

Figure 5.13: original `diskpart.txt`

Because we already know our disk size, the modified version only uses a part of the disk. In our case, it is a 250GB hard disk, so we reserved 150GB for Windows. The modified version is shown in Figure 5.14.

```
select disk 0
clean
create partition primary size=150000
assign letter=c
format FS=NTFS LABEL="Node" QUICK OVERRIDE
active
exit
```

Figure 5.14: modified `diskpart.txt` in dualboot OSCAR 1.0

Because this `diskpart.txt` script wipes out the whole disk, the Windows partition has to be installed first, and each time during reinstallation of Windows, Linux needs to be reinstalled as well.

5.5 How dualboot OSCAR v2.0 Works

Because the local GRUB bootloader has to be correctly pointed by MBR section on a hard disk, the reimaging of Windows partitions always rewrites MBR and damages GRUB which boots Linux. Therefore, Windows has to be installed before Linux. This is a considerable inconvenience during the system maintenance. the dualboot OSCAR v2.0 only places the PXE network bootloader in the head node. Thus, the MBR information in each computer node does not have to be fixed after either system

reimaging. Figure 5.15 is the flow chart of dualboot OSCAR, it is triggered by each queue state fetching and end at rebooting job queued and run.

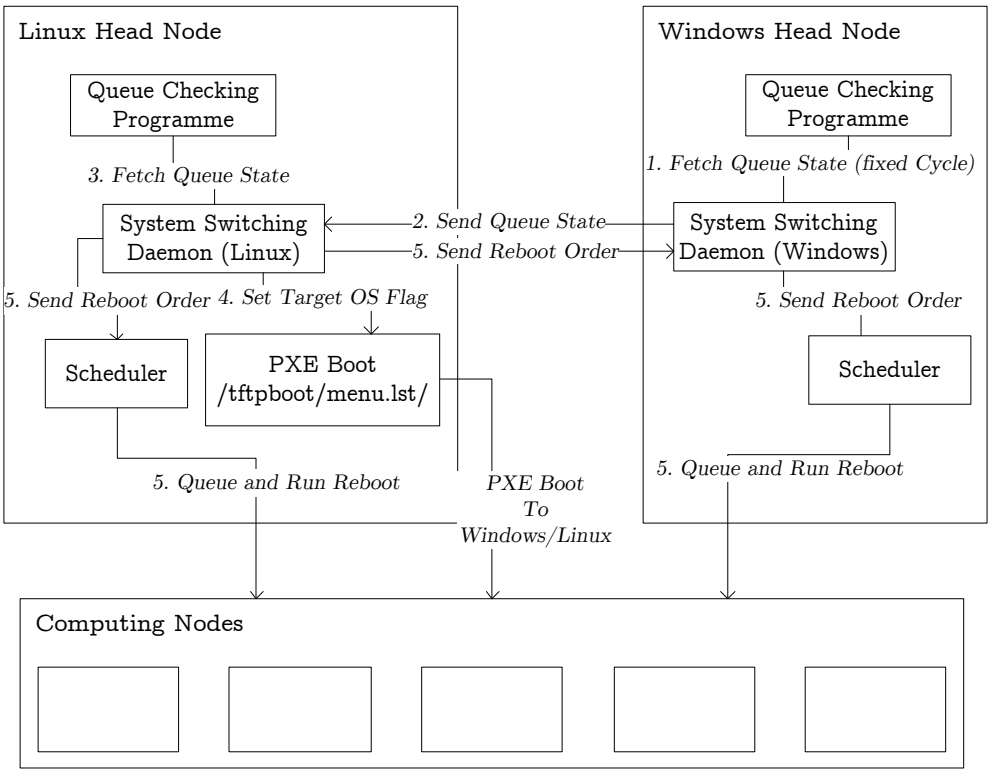

Figure 5.15: Flow chart of dualboot OSCAR 2.0

5.6 Dualboot controller of v2.0

5.6.1 PXE boot ROM

PXELINUX is sub-project of SYSLINUX, whose aim of design is to help Linux booting from different storage formats, e.g. CD-ROM, PXE and FAT. PXELINUX could load our machines into PXE environment, and it is also the method that OSCAR uses to deploy compute nodes. However, PXELINUX has less ability in controlling local partitions booting. It only can quit PXE and lead to normal boot order.

The solution to PXELINUX control nodes' boot order is that firstly load a ROM of PXE, such as PXEGRUB, then let PXEGRUB load remote node.

5.6.1.1 PXEGRUB as a network bootloader

Initially, the ROM of PXEGRUB in GRUB 0.97 was chosen for the network bootloader, and it tested well in virtual machine. Until it had been found that not all new LAN cards can be supported.

PXEGRUB is an optional component of GNU GRUB. It could be obtained by compiling from source code with parameter `--enable-diskless` and `--enable-(suited NIC drivers)`. DHCP and TFTP services could specify individual boot ROM and configure file for each node.

The PXEGRUB also makes system switching simpler. The dualboot OSCAR v1.0 method uses a FAT partition which stores configure file. In improvement, as their configure files are stored at the head node, a compute node could be switched by any reboot action, including soft reboot and physically power reset.

A lot of tests have been done in VirtualBox virtual machine. They worked well in the virtualised environment. However, due to the developing of GRUB 0.97 had been stopped, it could not support new model of LAN cards in the market. So it has to be change in another way.

5.6.1.2 GRUB4DOS as a net boot ROM

GRUB4DOS is an open-source fork of GRUB. GRUB4DOS supports a wider range of disk formats and load methods than GRUB. it also has a very easy-use PXE ROM.

The PXE ROM of GRUB4DOS reads different menu files,which are named from compute nodes' LAN cards MAC address, at the directory `menu.lst/` of PXE directory (normally is `/tftpboot/`).

By modifying the menu files, dualboot OSCAR can control any machine's boot order, as long as they are connected to their head node.

5.6.1.3 How they work together

Initially, the OS loading method is designed to make `menu.lst` for each machine's MAC, then it could boot specific machines to specific operating system. However, the daemon programme in OSCAR head node would be difficult to get information about which machine is scheduled to be rebooted. The flow chart of the initial way is shown in Figure 5.16.

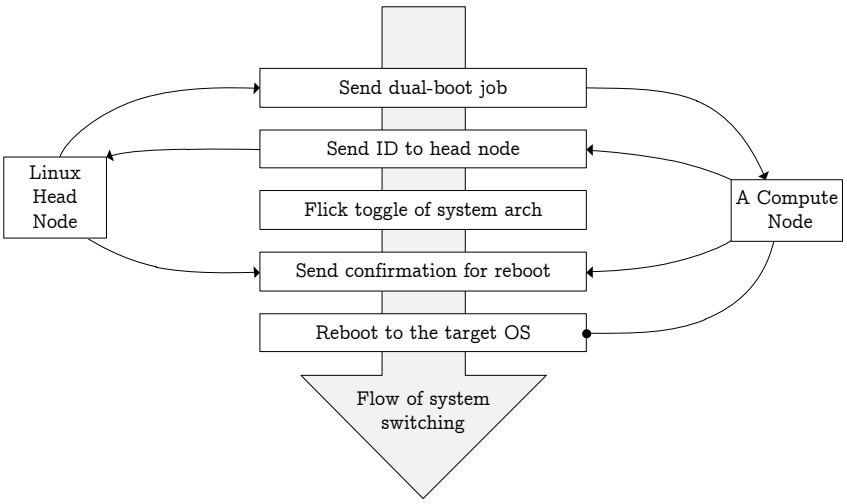

Figure 5.16: Initial way of PXE OS boot control of v2.0

Eventually, the method is developed into a single "flag" control system. All the rebooting nodes will be led to the same operating system, because the whole dualboot cluster will only need one system at one time. As in Figure 5.17, the current way is more concise.

5.6.2 Fetching queue states

The former way to get the queue information can be kept. The output of queue state checking programmes could be collected and used by new communicator programmes.

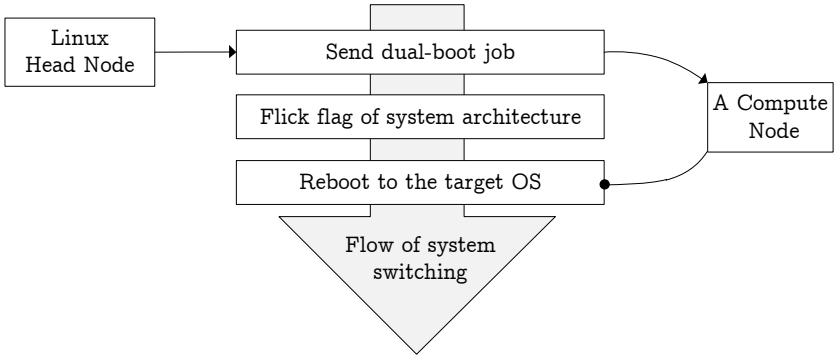

Figure 5.17: Current way of PXE OS boot control of v2.0

5.6.3 Head node communicators between Windows head and OSCAR head

The two communicator are written in Perl. Strawberry Perl or Active Perl is required in Windows Head node. Its flow chart is in Figure 5.15. Windows queue status is submitted to Linux side by TCP/IP socket communication. Multi-boot service send 'switch batch job' (just reboot).

1. Windows communicator fetch queue state by fixed cycle, e.g. 10mins.
2. Windows communicator send queue state to Linux communicator.
3. Linux communicator fetch PBS queue state and decide how many machines are needed to switch OS.
4. Set target OS flag.
5. Send reboot order to Windows HPC scheduler or PBS scheduler.
6. Machines will be rebooted when it is totally free, then be loaded into another OS.

5.7 Deployment of v2.0

A significant improvement is that the deployment of v2.0 has become more convenient than v1.0. After patching several supporting packages, OSCAR's dualboot deploying scripts can be generated automatically each

time. Windows partition and OSCAR partition can be individually reimaged without corrupting each another. But in first time deployment, Windows-then-Linux is more suitable.

5.7.1 Patching OSCAR deployment tool (less manual change)

OSCAR uses `systemimager`, `systeminstaller`, and `systemconfigurator` to build compute node image and configure the node which has just been installed.

By patching `systemimager` and `systeminstaller`, a new disk format label `skip`, is enabled in OSCAR's disk image configure file, e.g. `ide.disk` shown in Figure 5.18. The first partition with label `skip` will be reserved for Windows.

```
/dev/sda1         16000   skip
/dev/sda2         100     ext3   /boot       defaults bootable
/dev/sda5         512     swap
/dev/sda6         *       ext3   /           defaults
/dev/shm          -       tmpfs  /dev/shm    defaults
nfs_oscar:/home   -       nfs    /home       rw
```

Figure 5.18: `ide.disk` in v2.0

Patches are attached in appendixes, it also can be found in dualboot OSCAR's google code project.

5.7.2 Patching Windows HPC deployment tool

Deployment for Windows HPC compute nodes in v2.0 is as easy as in v1.0. By modifying `diskpart.txt`, Windows HPC deploy tool will only make one primary partition with specified size with the script in Figure 5.14. When some compute nodes need to be reimaged, a different `diskpart.txt` in Figure 5.19, which only format the Windows partition, can be applied to take effect.

```
select disk 0
select partition 1
format FS=NTFS LABEL="Node" QUICK OVERRIDE
active
exit
```

Figure 5.19: `ide.disk` in v2.0 for reimaging

Chapter 6

Future Works and Conclusions

Dualboot OSCAR benefits both users and administrators. QGG has got a flexible cluster as a multi-purpose cluster that enables other clusters to concentrate on pure Unix Environments.

The version 1.0 of dual-boot OSCAR had been fully deployed on "Eridani". The principle of version 2.0 is stable and has been tested on "Eridani". Since this project has been hosted on google code as a public open-source project, it will be carried on by anyone who want to contribute to it.

Currently the daemons for queue monitoring are still following the rule "first-come first-serve". This could be improved to adapt the rules from diverse administration requirements. The further version of this dual-boot package is being considered as either an independent software patch or a dual-boot enabled version of OSCAR 6.x, or an integrated module of OSCAR project.

During the developing of dualboot OSCAR, QGG, the university campus grid, and HPC-RC, the complete HPC resource centre, had been established. A group of facilities or servers for QGG has been developed and deployed by HPC-RC. QGG so far have joined two Virtual Organisations, NGS and NW-Grid.

As a result of this research a conference paper on dualboot cluster system with OSCAR and Windows HPC was presented at All Hands Meeting 2010, Cardiff.

In conclusion, this thesis have discussed the current situation of High-Performance Computing, evaluated current solution in HPC hybrid resources management, investigate commercial and open-source software for

multi-platform system, designed and developed a dualboot cluster system with OSCAR and Windows HPC.

Appendix A

Appendixes

A.1 Patches for systemimager and systeminstaller

A.1.1 Server.pm.patch

Path : /usr/lib/systemimager/perl/SystemImager/Server.pm

```
--- Server.pm.orig      2011-12-22
   10:50:23.000000000 +0000
+++ Server.pm   2011-12-12 02:01:46.000000000 +0000
@@ -397,21 +397,21 @@
          print $out qq(logmsg "Old partition table
             for $devfs_dev:"\n);
          print $out "parted -s -- $devfs_dev print\n
             \n";

-         print $out "# Wipe the MBR (Master Boot
   Record) clean.\n";
-         $cmd = "dd if=/dev/zero of=$devfs_dev bs
   =512 count=1 || shellout";
-         print $out qq(logmsg "$cmd"\n);
-         print $out "$cmd\n\n";
-
-         print $out "# Re-read the disk label.\n";
-         $cmd = "blockdev --rereadpt $devfs_dev";
-         print $out qq(logmsg "$cmd"\n);
-         print $out "$cmd\n\n";
-
```

```
-       print $out "# Create disk label.  This
   ensures that all remnants of the old label,
   whatever\n";
-       print $out "# type it was, are removed and
   that we're starting with a clean label.\n";
-       $cmd = "parted -s -- $devfs_dev mklabel
   $label_type || shellout";
-       print $out qq(logmsg "$cmd"\n);
-       print $out "$cmd\n\n";
+###    print $out "# Wipe the MBR (Master Boot
   Record) clean.\n";
+###    $cmd = "dd if=/dev/zero of=$devfs_dev bs
   =512 count=1 || shellout";
+###    print $out qq(logmsg "$cmd"\n);
+###    print $out "$cmd\n\n";
+
+###    print $out "# Re-read the disk label.\n";
+###    $cmd = "blockdev --rereadpt $devfs_dev";
+###    print $out qq(logmsg "$cmd"\n);
+###    print $out "$cmd\n\n";
+
+###    print $out "# Create disk label.  This
   ensures that all remnants of the old label,
   whatever\n";
+###    print $out "# type it was, are removed and
   that we're starting with a clean label.\n";
+###    $cmd = "parted -s -- $devfs_dev mklabel
   $label_type || shellout";
+###    print $out qq(logmsg "$cmd"\n);
+###    print $out "$cmd\n\n";

        print $out "# Get the size of the
           destination disk so that we can make the
            partitions fit properly.\n";
        print $out qq(DISK_SIZE=`parted -s
           $devfs_dev unit MB print ) . q(| grep '
           Disk geometry for' | sed 's/^.*-//g' |
           sed 's/\..*$//' | sed 's/MB//' `) . qq(\
           n);
@@ -674,6 +674,8 @@

            print $out qq(START_MB=$startMB{$m}\n);
            print $out qq(END_MB=$endMB{$m}\n);
+
+
```

```
             my $swap = '';
             if ($flags{$m}) {
@@ -696,6 +698,20 @@
                 $cmd = qq(parted -s -- $devfs_dev
                    mkpart $p_type{$m} $swap) . q(
                    $START_MB $END_MB) . qq( ||
                    shellout);

             }
+
+                     ### Skip the "skip"
   partition and remove the rest partitions. this is
    not considering gtp.
+                     if ($flags{$m}) {
+                 if ($flags{$m} =~ /skip/) {
+                     print $out qq(logmsg "parted -s
    -- $devfs_dev rm 2"\n);
+                     print $out "parted -s --
   $devfs_dev rm 2\n";
+                     print $out qq(logmsg "parted -s
    -- $devfs_dev rm 3"\n);
+                     print $out "parted -s --
   $devfs_dev rm 3\n";
+                     print $out qq(logmsg "parted -s
    -- $devfs_dev rm 4"\n);
+                     print $out "parted -s --
   $devfs_dev rm 4\n";
+                                         $cmd = "echo
    \"Skiped parttion\"";
+                 }
+             }
+
             print $out qq(logmsg "$cmd"\n);
             print $out "$cmd\n";

@@ -753,6 +769,7 @@
                     if (($flag eq "lba") and (
                        $label_type eq "gpt")) {
                        next; }
                     # Ignore custom flag 'swap'. -
                       AR-
                     if ($flag eq "swap") { next; }
+                    if ($flag eq "skip") { next; }
                     $cmd = "parted -s -- $devfs_dev
```

```
                        set $m $flag on || shellout
                       \n";
                    print $out "logmsg $cmd";
                    print $out "$cmd";
@@ -1350,7 +1367,14 @@

                print $out "\n";

-            # ext2
+            # skip
+            } elsif ( $xml_config->{fsinfo}->{$line
   }->{fs} eq "skip" ) {
+
+                # create fs
+                $cmd = "mke2fs -q $real_dev ||
   shellout";
+                print $out qq(logmsg SKIPED "$cmd"\
   n);
+
+           # ext2
             } elsif ( $xml_config->{fsinfo}->{$line
                }->{fs} eq "ext2" ) {

                # create fs
@@ -1633,6 +1657,12 @@
         my $dump      = $xml_config->{fsinfo}->{
           $line}->{dump};
         my $pass      = $xml_config->{fsinfo}->{
           $line}->{pass};

+               ### Skip the "skip" partition
+               if ($fs) {
+                       if ($fs =~ /skip/) {
+                       next;
+                       }
+       }

         # Update the root device. This will be used
            by systemconfigurator
         # (see below).
```

A.1.2 IA.pm.patch

Path : /usr/lib/systeminstaller/SystemInstaller/IA.pm

```
--- IA.pm.orig  2011-12-22 10:50:23.000000000 +0000
+++ IA.pm       2011-12-12 02:01:46.000000000 +0000
@@ -143,6 +143,13 @@
                                        $flags="swap
                                           ";
                                    }
                                }
+                               if ($DISKS{
   PARTITIONS}{$parname}{TYPE} == "777") {
+                                   if ($flags) {
+                                       $flags .= ",
   skip";
+                                   } else {
+                                       $flags="skip
   ";
+                                   }
+                               }
                                if ($flags) {
                                     print
                                        AICONF "
                                        flags=\"
                                        $flags\"
                                         ";
                                }
@@ -178,6 +185,13 @@
                                        $flags="swap
                                           ";
                                    }
                                }
+                               if ($DISKS{
   PARTITIONS}{$parname}{TYPE} == "777") {
+                                   if ($flags) {
+                                       $flags .= ",
   skip";
+                                   } else {
+                                       $flags="skip
   ";
+                                   }
+                               }
                                if ($flags) {
```

```
                                        print
                                           AICONF "
                                           flags=\"
                                           $flags\"
                                            ";
                                    }
```

A.1.3 Partition.pm.patch

Path : /usr/lib/systeminstaller/SystemInstaller/Partition/Partition.pm

```
--- Partition.pm.orig   2011-12-22
   10:50:23.000000000 +0000
+++ Partition.pm        2011-12-12
   02:01:46.000000000 +0000
@@ -406,6 +406,8 @@
          return 83;
      } elsif ($fstype =~ /^(swap)$/) {
          return 82;
+     } elsif ($fstype =~ /^(skip)$/) {
+         return 777;
      } elsif ($fstype =~ /^(extended)$/) {
          return 5;
      } elsif ($fstype =~ /^(fat16|vfat|msdos)$/) {
```

References list

Amdahl, G. M., & Sunnyvale, C. (1967, December). Validity of the single processor approach to achieving large scale computing capabilities. In *Afips spring joint computer conference* (Vol. 52 Suppl 2). Available from http://www-inst.eecs.berkeley.edu/~n252/paper/Amdahl.pdf

Carter, M. (2006, March). *Automate OS switching on a dual-boot Linux system.* Available from http://www.ibm.com/developerworks/linux/library/l-osswitch/

Cluster Resources Inc. (2007). MOAB HYBRID CLUSTER. Available from http://download.microsoft.com/download/c/3/1/c318044c-95e8-4df9-a6af-81cdcb3c53c5/ClusterResources_Windows_Linux_Hybrid_Cluster.pdf

Cygwin. (n.d.). Available from http://www.cygwin.com/

Kureshi, I. (2010). *Establishing a University Grid for HPC Applications.* Master thesis, University of Huddersfield. Available from http://eprints.hud.ac.uk/10169/

Le, Q., Ghidali, D., & Thiru, M. (2010). Using xCAT to Provision Windows HPC Server 2008 and Red Hat Enterprise Linux. (September).

Liang, S. (2010). *A Dynamic OS Switching Solution for Dual-boot Clusters.* Unpublished doctoral dissertation.

Microsoft Corp. (2006, June). *Microsoft Releases Windows Compute Cluster Server 2003, Bringing High-Performance Computing to the Mainstream.* Available from http://www.microsoft.com/presspass/press/2006/jun06/06-09computeclusterserver2003pr.mspx

Microsoft Corp. (2010). *Technical Overview of Windows HPC Server 2008 R2 - White Paper.* Available from http://download.microsoft.com/download/A/2/C/A2C6AB4C-50E4-4AF0-AEF8-55040EB18D1F/WindowsHPCServer2008R2TechnicalOverview_final.docx

NGS. (2011). *New resources now available through the UI-WMS | NGS* (Tech. Rep.). Author. Available from http://www.ngs.ac.uk/news/new-resources-now-available-through-the-ui-wms

Sloan, J. D. (2004). *High Performance Linux Clusters with OSCAR, Rocks, OpenMosix, and MPI* (1st ed.; A. Oram, Ed.).

StatCounter. (2011). *Top 5 Operating Systems from March to Aug 2011 | StatCounter Global Stats.* Available from http://gs.statcounter.com/\#os-ww-monthly-201103-201108-bar

TOP500.Org. (2011). *TOP500 Supercomputing Sites.* Available from http://www.top500.org/

Usage Statistics and Market Share of Operating Systems for Websites, November 2011. (n.d.). Available from http://w3techs.com/technologies/overview/operating_system/all

Bibliography

All the Details of many versions of both MBR and OS Boot Records. (n.d.). Available from http://thestarman.pcministry.com/asm/mbr/index.html\#Win7

Amdahl, G. M., & Sunnyvale, C. (1967, December). Validity of the single processor approach to achieving large scale computing capabilities. In *Afips spring joint computer conference* (Vol. 52 Suppl 2). Available from http://www-inst.eecs.berkeley.edu/~n252/paper/Amdahl.pdf

Barry, P. (2002). *Programming the network with Perl* (1st ed., Vol. 54) (No. 2). John Wiley & Sons Inc.

Brown, J. (2006). *Computer cluster to boost UK brain power.* Available from http://www.computing.co.uk/ctg/news/1842927/computer-cluster-boost-uk-brain-power

Buyya, R. (n.d.). *Single System Image and Cluster Middleware.* Available from http://ww2.cs.mu.oz.au/678/SSI-Clusters.ppt

Calegari, P. B., & Varlet, T. M. (2009). A hybrid OS cluster solution: Dual-Boot and Virtualization with Windows HPC Server 2008 and Linux Bull Advanced Server for Xeon.

Carter, M. (2006, March). *Automate OS switching on a dual-boot Linux system.* Available from http://www.ibm.com/developerworks/linux/library/l-osswitch/

Cluster Resources Inc. (2007). MOAB HYBRID CLUSTER. Available from http://download.microsoft.com/download/c/3/1/c318044c-95e8-4df9-a6af-81cdcb3c53c5/ClusterResources_Windows_Linux_Hybrid_Cluster.pdf

Collins-sussman, B., Fitzpatrick, B. W., & Pilato, C. M. (2011). *Version Control with Subversion For Subversion 1.6 (Compiled from r4096)* (Vol. 6). Available from http://svnbook.red-bean.com/en/1.6/svn-book.pdf

Cygwin. (n.d.). Available from http://www.cygwin.com/

Flanery, R., Geist, A., Luethke, B., Schwidder, J., & Scott, S. (2000). The Scalable System Administrator : via C3 & M3C Tools. In *The second international workshop on cluster-based computing* (pp. 1–5).

Georgie, S. (2009). *Evaluation of Cluster Middleware in a Heterogeneous Computing Environment.* Unpublished doctoral disser-

tation. Available from http://www.risc.jku.at/publications/download/risc_3855/MasterThesis-StefanGeorgiev.pdf

Gropp, W., Lusk, E., & Sterling, T. (2003). *Beowulf Cluster Computing with Linux* (2nd ed.).

grub4dos guide - pxe booting. (n.d.). Available from http://diddy.boot-land.net/grub4dos/files/pxe.htm

IBM DOS 2.00 Master Boot Record. (n.d.). Available from http://thestarman.pcministry.com/asm/mbr/200MBR.htm

Intel Corporation. (1999). *Preboot Execution Environment (PXE) Specification.* Available from http://download.intel.com/design/archives/wfm/downloads/pxespec.pdf

Kureshi, I. (2010). *Establishing a University Grid for HPC Applications.* Master thesis, University of Huddersfield. Available from http://eprints.hud.ac.uk/10169/

Kureshi, I., Holmes, V., & Liang, S. (2008). Hybrid HPC – Establishing a Bi-Stable Dual Boot Cluster for Linux with OSCAR middleware and Windows HPC 2008 R2. In *Ahm2010.* Cardiff. Available from http://www.allhands.org.uk/2010/sites/default/files/2010/TuesW2KureshiHybridPC.pdf

Le, Q., Ghidali, D., & Thiru, M. (2010). Using xCAT to Provision Windows HPC Server 2008 and Red Hat Enterprise Linux. (September).

Li, P., Member, S., Ravindran, B., & Member, S. (2004). A Formally Verified Application-Level Framework for Real-Time Scheduling on POSIX Real-Time Operating Systems. , *30*(9), 613–630.

Liang, S. (2010). *A Dynamic OS Switching Solution for Dual-boot Clusters.* Unpublished doctoral dissertation.

Mellanox Technologies Inc. (n.d.). InfiniBand ™ Frequently Asked Questions.

Microsoft Corp. (2006, June). *Microsoft Releases Windows Compute Cluster Server 2003, Bringing High-Performance Computing to the Mainstream.* Available from http://www.microsoft.com/presspass/press/2006/jun06/06-09computeclusterserver2003pr.mspx

Microsoft Corp. (2010). *Technical Overview of Windows HPC Server 2008 R2 - White Paper.* Available from http://download.microsoft.com/download/A/2/C/A2C6AB4C-50E4-4AF0-AEF8-55040EB18D1F/WindowsHPCServer2008R2TechnicalOverview_final.docx

Modules – Software Environment Management. (n.d.). Available from http://modules.sourceforge.net/

NGS. (2011). *New resources now available through the UI-WMS | NGS* (Tech. Rep.). Author. Available from http://www.ngs.ac.uk/news/new-resources-now-available-through-the-ui-wms

Open Source Cluster Application Resources (OSCAR) Administrator's

Guide. (n.d.). Available from http://oscar.openclustergroup.org/public/docs/oscar5.0/OSCAR5.0_Users_Manual.pdf

OSCAR Homepage. (n.d.). Available from http://www.csm.ornl.gov/oscar/home.html

Raymond, E. S. (2003). *The art of unix programming.* Addison-Wesley. Available from http://portal.acm.org/citation.cfm?id=829549http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.103.2799\&rep=rep1\&type=pdf

Rocks V and Windows 2008 HPC Server - Rocks Clusters. (n.d.). Available from https://wiki.rocksclusters.org/wiki/index.php/Rocks_V_and_Windows_2008_HPC_Server

Severance, C., & Dowd, K. (2010). *High Performance Computing* (1.5 ed.). Connexions Web site. Available from http://cnx.org/content/col11136/1.5/

Sloan, J. D. (2004). *High Performance Linux Clusters with OSCAR, Rocks, OpenMosix, and MPI* (1st ed.; A. Oram, Ed.).

StatCounter. (2011). *Top 5 Operating Systems from March to Aug 2011 | StatCounter Global Stats.* Available from http://gs.statcounter.com/\#os-ww-monthly-201103-201108-bar

syslinux. (n.d.). Available from http://www.syslinux.org/

TOP500.Org. (2011). *TOP500 Supercomputing Sites.* Available from http://www.top500.org/

Usage Statistics and Market Share of Operating Systems for Websites, November 2011. (n.d.). Available from http://w3techs.com/technologies/overview/operating_system/all

Walker, B. J. (n.d.). *Open Single System Image (openSSI) Linux Cluster Project.* Available from http://openssi.org/ssi-intro.pdf

WineHQ. (n.d.). *WineHQ.* Available from http://www.winehq.org/

xCAT - Extreme Cloud Administration Toolkit. (n.d.). Available from http://xcat.sourceforge.net/