Deploy Kerrighed SSI massively using DRBL

Wen-Chieh Kuo, Che-Yuan Tu, Yao-Tsung Wang
National Center for High-Performance Computing, Taiwan
{rock,rider,jazz}@nchc.org.tw

Abstract

Single System Image (SSI) Cluster played an important role on science, engineering and related domain research. Kerrighed is one of the SSI Clustering systems, users can run OpenMP or MPI applications with its features such as distributed shared memory (DSM), process migration, and resource management. In spite of having such practical SSI Clusters, users still have to choose a management system to deploy SSI Clustering systems that provide ease of use and management for scientists or IT technicians. To deploy and manage Clusters is not a user friendly work for most of users, especially non-IT users. That's why a convenient tool is so crucial.

In recent years, there are several solutions that combining SSI and Linux Clusters for high performance computing applications. In order to find a better way to deploy and manage Linux Clusters, we bring up an open source solution called Diskless Remote Boot in Linux (DRBL) [3] to make it ease of use. DRBL makes it easier to managing the deployment of the GNU/Linux operating system across many clients and it also keeps the configuration of all users' client computers on centralized server machine. This paper expounds the deployment of Kerrighed SSI by using DRBL.

1. Introduction

As the growth of the High-performance computing applications, there are more and more researches and science organizations that use GNU/Linux to build their Clusters. From these key elements of Cluster build-up, the distributed architecture of Clusters implies two major issues: Clusters are difficult to manage (Cluster installation, update and upgrade) and to use (application programmers and users have to manage Cluster resources by themselves) [6]. Although there are several solutions, we still try to find a better and easier way to make deployment more friendly and common in different GNU/Linux distributions.

In distributed computing field, a SSI Cluster is a system in which multiple networks, distributed databases or servers appear to the users as a single system. In other words, the operating system environment is shared by all nodes in the system. If users want to use and manage SSI Cluster easily, a user-friendly open source solution may be necessary. That's why we use DRBL to deploy SSI Clusters. Users just need to install SSI Clustering systems (Kerrighed) with DRBL on one server machine, and follow the DRBL installation wizard step by step. After installation, users only need to modify some of the DRBL configurations to fit the deployment environment. Users can also manage and modify those client computers (client nodes) on DRBL server-side, and run HPC-related applications on each node. SSI Cluster deployment and management becomes easier by combining DRBL and Kerrighed, and it makes Linux Clusters easy to use.

2. Background

2.1. Types of Deployment Tools

According to the classification of E. Imamagic and D. Mihajlovic [4], they divide deployment tools in two types: image-based and package-based. The package-based deployment tools are suitable for heterogeneous environment and their advantage is easy to update and modify packages for each nodes.

The disadvantage of package-based deployment tools is lack of common packaging method, package-based usually limit to operating system package systems (Deb or RPM). The advantages of image-based deployment tools are very intuitive and simple than package-based. The disadvantage of image-based is that users have to build specific packages for some requirements in each node. It is suitable for homogeneous environment (Cluster). The following will discuss some deployment tools especially to Kerrighed SSI.

2.2. SystemImager

SystemImager[12][13] is software which automates Linux installs, software distribution, and production deployment. SystemImager brings the automatic installation of Linux to masses of identical machines for quicker system deployment, and it is one of the easiest ways to do automated installs, software distribution, content or data distribution, configuration changes with operating system updates to user's network of Linux machines. It's the most useful tool in a large numbers of similar machines environment. So it really brings benefits to those environments include web servers, high performance Clusters, laboratories, and corporate desktop environments where all workstations have the same basic hardware configuration.

In order to easily deploy Kerrighed nodes, it's important to simplify installation process, like building DHCP/TFTP server or boot loader configuration with PXE environment. In SystemImager, users should run server setup; configuration and image creation to deploy to client machines after the SystemImager and related packages installation are completed. It seems easy to deploy Kerrighed nodes by running through the deployment process, but it is inconvenient to create images for Kerrighed configuration, like authentication and KerFS [9] configuration for most of users those are not familiar with GNU/Linux.

2.3. OSCAR

OSCAR allows users, regardless of their experience level with a *nix environment, to install a Beowulf type high performance computing Cluster [4] [5]. OSCAR is a wizard based Cluster installation, so it provides a friendly user interface for users to easily setup their machines and Clusters.

OSCAR has already contained almost user's need in configuring complex Cluster administration and communication packages. Users can easily download OSCAR related HPC packages by entering the package sources provided by INRIA in OSCAR package selector. It installs and configures all required software for the selected packages according to user’s input. Then it creates customized disk images which are used to provision the computational nodes in the Cluster with all the client software and administrative tools needed for immediate use. OSCAR also includes a robust and extensible testing architecture, ensuring that the Cluster configuration is ready for production [4].

2.4. Kadeploy

Kadepoly [8] is belonging to image-based deployment tool; it can deploy customized images. It is used to massively deploy Cluster and grid nodes. It has a set of deployment, configuration and management tools for Clusters nodes. Currently it deploys Linux, *BSD, Windows, Solaris on x86 and 64 bits computers successfully. Comparing to other deployment tools, Kadeploy has to do complicated configuration and non-automatic packages installation. The major different between Kadeploy and other tools is that it needs database (MySQL) to maintain information about the Cluster composition and current state.

2.5. FAI

Fully Automatic Installer (FAI) [5] is a package-based deployment tool to deploy Debian GNU/Linux and other distributions (Ubuntu, Mandriva, Suse, solaris...) to Cluster. It's more flexible than other tools like kickstart for Red Hat, autoyast and alice for SuSE or Jumpstart for SUN Solaris. It uses class concept to support heterogeneous configuration, and flexible update running system without installation.

3. Architecture

3.1. DRBL

Diskless Remote Boot in Linux (DRBL) is an open source solution to managing the deployment of the GNU/Linux operating system across many clients [3]. DRBL supports lots of popular GNU/Linux distributions, and it is developed based on diskless and systemless environment for client machines. DRBL uses PXE/Etherboot, DHCP, TFTP, NFS and NIS to provide services to client machines so that it is not necessary to install GNU/Linux on the client hard drives individually [3] (see Figure 1). Users just prepare a server machine for DRBL to be installed as a DRBL server, and follow the DRBL installation wizard to configure and push the environment for client machines step by step. That’s should be an easy way for users to deploy a DRBL environment even a GNU/ Linux beginner.

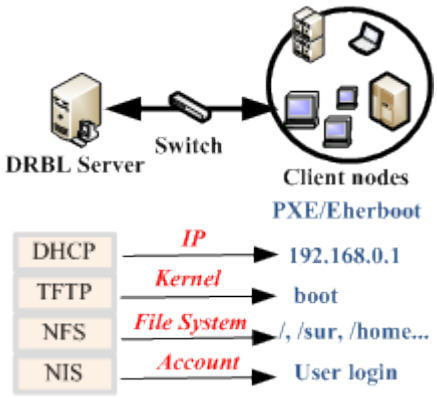

Figure 1. DRBL Operation [3]

In DRBL environment, client machines will boot via PXE/Etherboot (diskless) [3], and users can build a modified Kernel image or edit DRBL-related configuration file for specific usage such as SSI Cluster. Hard drives become optional for DRBL client machines. Nevertheless, it still allows hard drives to be used as swap space. That's why DRBL can save lots of time on SSI-Related System deployment by configuring client machines through DRBL server (boot server) in a centralized boot environment.

In this case, it focuses on effective configuration and deployment for Cluster environment; the package-based deployment tools are the best choice. Both FAI and DRBL are package-based deployment tools, but FAI need complex pre-configure. Hence, this case chooses DRBL for its simple configuration and amazed deployment speed. DRBL also provides Cluster management tools for manage client nodes.

3.2. Kerrighed

Kerrighed [9] is a Single System Image operating system for Clusters. Kerrighed can merge machines into a virtual SMP (Symmetric multiprocessing) machine and also merge distributed memory into a virtual shared memory. In the parallel computing, it can run OpenMP applications and MPI applications. The latest version of Kerrighed is 2.3.0 and it based on Kernel 2.6.20.

The main feature of Kerrighed are (1)Cluster wide process management, (2)support for Cluster wide shared memory, (3)Cluster file system, (4)transparent Process Checkpointing, (5)high availability for users applications, (6)customizable Single System Image features

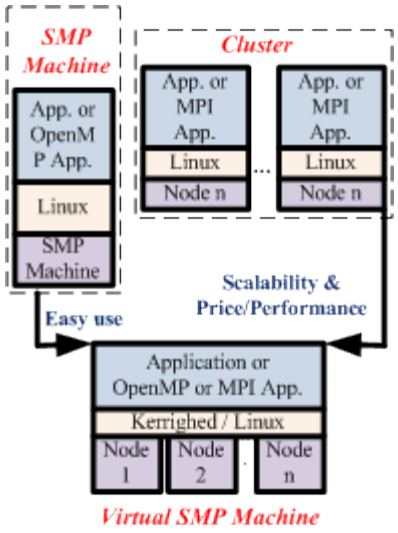

Figure 2. Philosophy of Kerrighed [9] [10]

Figure 2 is simple philosophy of Kerrighed. Kerrighed merge scalability of Cluster and easy use of SMP advantages. It makes all nodes become a virtual SMP machine. Users can use this virtual SMP to run a kind of applications or science computing.

3.3. Implementation

This section described how to use DRBL to deploy Kerrighed nodes. Figure 3 is DRBL-Kerrighed architecture. DRBL Server is the central management machine. Nodes (krg01~krg15) are diskless machines, nodes get Kernel image (patched by Kerrighed), user account, IP address, and file system through DRBL mechanism. After nodes boot, Kerrighed nodes is done. Then, nodes loaded Kerrighed modules and startup. It can use Linux commands (*ps, top*...) to check this virtual SMP machine information. Users will find that all of these CPUs and memory of nodes will be merged; the distributed nodes will become a virtual SMP.

In this test-bed, each node equipped with Intel® Core™ 2 Quad 2.40GHz and 2GB DDR667 RAM. When DRBL environment is done and deploy Kerrighed nodes successfully, these 16 nodes will become a virtual SMP machine. This virtual Kerrighed SMP has 64 CPUs and 32GB memory. If some processes running on it, total nodes can see processes' global process id, and if applications have multi-processes, the Kerrighed scheduler will migrate processes to idle CPUs automatically. Kerrighed Kernel responsible for load-balance of virtual SMP

system, it offered a Kernel-based migrate mechanism for dynamic processes migration.

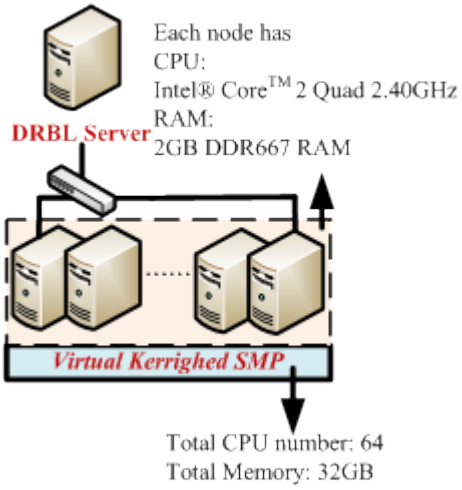

Figure 3. DRBL with Kerrighed Architecture

The software stack of architecture (see Figure 4) is divided into two parts: Kernel space and User space. The Kerrigehd Kernel and modules are in the Kernel space; DRBL and some daemons (TFTPd, NFSd, DHCPd, and NISd) run in the User space. DRBL is responsible for Cluster management and service management (TFTPd, NFSd, DHCPd, NISd). Clonezilla is an additional mechanism of DRBL, it clones system image for system backup.Our implement procedure is divided into following:

(1) Kerrighed Installation: After installing a kind of Linux distribution, just download the Kerrigehd tar-ball and vanilla Kernel from Kernel.org. Follow the Kerrighed installation guide to patch Kernel, build modules and install it.

(2) DRBL Installation: First step, it has to install DRBL software in server. If system OS is Debian or RPM package system, it just installed l package form DRBL website. Then it needs to execute DRBL configuration command "*drplsrv –i"* to choose your Kernel version for nodes and automatic installs the packages that DRBL required, such as DHCP, NFS, NIS and TFTP. Then, using DRBL deployment command "*drblpush –i"* to push Kerrighed environment to all nodes. DRBL offers interactive dialog to help users to build DRBL environment and it automatic configured and started all the services required to make the Cluster work. It automatically detects the network interfaces that have private IP addresses assigned to them and asks users how many clients want to set up. DRBL provides two methods for nodes IP address: (1) fix IP address (binding MAC address): this feature is useful to setting up system for security; (2) dynamic IP address (range of IP address) in the open environment where anyone can add a new machine dynamically.

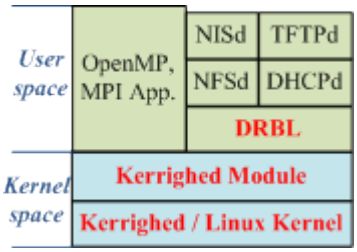

Figure 4. DRBL with Kerrighed Architecture

(3) DRBL-Kerrighed environment configuration and management: When above steps accomplished, then the DRBL with Kerrighed is complete. For specific purposes, some system configuration and environment tuning maybe necessary. The command "*dcs"* of DRBL pops out DRBL management graphic user interface to manage nodes.

In this case, it is suitable to choose DRBL to deploy nodes massively. DRBL can fast and efficiently deploy nodes, the deployment procedure just need two commands (*drplsrv –i, drblpush –i*). DRBL can automatically setup required services (DHCP, NFS, NIS and TFTP). DRBL offers central management interface to effective managed and configured nodes. In addition, DRBL also offers a lot of management commands.

4. Discussion

From section 2.1, it is known that E. Imamagic' and D. Mihajlovic' divided the automatic installers into two different groups, one is image-based and the other is package-based. The major difference between these two different groups is the store of software stack on server computer [4]. DRBL belongs to the package based automatic installers because it has packed several necessary packages for users to install easily by running the DRBL scripts. Users can easily update or upgrade those installed software packages of the clients even if additional software packages installation, but the package dependencies should be concerned carefully to fit actual requirement. In addition, most of these popular GNU/Linux distributions whether it's RPM-based System or Debian-based System all have

well-known package management systems to settle these misgivings. So that may not be a big problem for dealing with package dependencies, and users can also make the most of the DRBL commands to modify package needs or any software changes.

Table 1.Deployment tools comparison

	Distribution	Support Diskless/ Sysmless	Type	Node configuration tools	Cluster management tools	Database installation
SystemImager	ALL	Yes	Image	Yes	No	No
OSCAR	RPM-based System	Yes	Image	Yes	Yes	No
Kadeploy	ALL	No	Image	Yes	Yes	Yes
FAI	Debian-based System	Yes	Package	Yes	No	No
DRBL	ALL	Yes	Package	Yes	Yes	No

DRBL is currently support lots of well-known GNU/ Linux distributions, so that is convenient for users to build up a Cluster from those supported platforms. Users can have more choices on different platforms, and the diskless environment makes Cluster deployment and management easier. DRBL provides lots of practical commands with brief instructions for users to modify those configuration files and necessary changes from client nodes, so that users can control all these client nodes and let them act as users need at the same time through DRBL related commands. That's the reason of choosing DRBL for deployment and management of Cluster environment, because it performs excellent in both node configuration and Cluster management. Above the Table 1 shows the comparison of those popular deployment tools.

5. Conclusion

DRBL can be a convenient Clustering tool for users to deploy SSI Clusters easily and quickly. In addition, DRBL can work on lots of popular GNU/Linux distributions, and it really brings great benefits to Cluster deployment and management. DRBL really helps to save lots of time and costs on hardware expenses and machine maintenance for Cluster user and administrator. Furthermore, DRBL guides users through Cluster installation and deployment with the clear and easy installation wizard, so it really makes Cluster deployment more convenient. That's the reason why we choose DRBL to deploy Kerrighed SSI Cluster.

After combining DRBL and Kerrighed, users can initialize the Kerrighed service in any one of these client nodes and run their HPC-related applications, like OpenMP or MPI. Because Kerrighed is designed for easy use, high performance, high availability, efficient resources management, and high customizability [9]. Kerrighed SSI Cluster make client nodes becomes a virtual SMP [14] machine that build up from several standard PCs. This SSI Cluster deployment solution provides those users who have the requirement of high computing power or Cluster characteristics. In conclusion, the merge of DRBL and Kerrighed brings users a convenient way to have a SMP machine with maximum economical benefits.

6. References

[1] Beowulf Project Overview, http://www.beowulf.org/overview/index.html.

[2] Core OSCAR TEAM. HOWTO: Create an OSCAR package, Jan.2004. http://svn.oscar.openClustergroup.org/trac/oscar/.

[3] Diskless Remote Boot in Linux (DRBL), http://drbl.sourceforge.net/.

[4] E. Imamagic, D. Mihajlovic. “Automatic Inastallers Review,” CARNet Users' Conference, 2004.

[5] Fully Automatic Installer, http://www.informatik.uni-koeln.de/fai/

[6] G. Vallee, Stephen L. Scott, C.Morin, J.-Y. Berthou, and H. Prisker. SSI-OSCAR: a Cluster Distribution for High Performance Computing Using a Single System Image. Proceedings of the 19th International Symposium on High Performance Computing Systems and Applications (HPCS'05) 1550-5243/05, IEEE 2005.

[7] J. Parpaillon. XtreemOS Tutorial Deploying Kerrighed, Oct.2006.

[8] Kadeploy, http://Kadeploy.imag.fr/.
[9] Kerrighed, http://www.Kerrighed.org/wiki/index.php/Main_Page.
[10] KerLabs website, http://www.kerlabs.com.
[11] Wikipedia: Single-system image, http://en.wikipedia.org/wiki/Single-system_image .
[12] Wikipedia: SystemImager, May.2008. http://wiki.systemimager.org/index.php/Main_Page.
[13] SystemImager, Feb.2008. http://freshmeat.net/projects/systemimager. http://gforge.inria.fr/docman/view.php/69/668/xtreemos.pdf.
[14] Wikipedia: Symmetric multiprocessing, http://en.wikipedia.org/wiki/Symmetric_multiprocessingy.