

Using DRBL to Deploy MPICH2 and CUDA on Green Computing

Jiun-Yu Wu^{1,2}, Yao-Tsung Wang², Steven Shiau², Hui Ching Wang¹

¹Department of Applied Mathematics, National Chung Hsing University, Taichung City, Taiwan.

²National Center for High Performance Computing, Taiwan.

e-mail: adherelinux@hotmail.com, jazz@nchc.org.tw, steven@nchc.org.tw, hcwang@amath.nchu.edu.tw

Abstract—In this paper, an energy efficient architecture for Build Energy Efficient GPU and CPU Cluster Using DRBL is proposed. This architecture helps administrator not only to quickly deploy and manage GPU and CPU Cluster environment, but also bring benefit of energy efficiency in scientific computing. The experiment simulates 3 cases to prove energy efficiency. We will compare GPU and CPU Cluster Using DRBL with the design without DRBL. According to the experiment results, the architecture provides a way to implement a power economization computing architecture that reduces energy consumption.

Keywords—Green Computing; CUDA; DRBL; MPICH2; GPU.

I. INTRODUCTION

HPC (High Performance Computing) is an important research domain for all of human; it helps people to solve science questions massively. On the other side, it also brings energy consumption and environmental problems. In recent years, both green computing and grid computing are putting together for discussion, and energy consumption optimization becomes a crucial issue to grid computing instead of computing performance.

Education institutions or research organizations that demand to adopt Cloud Computing to solve computing and energy saving problems may use our toolkit for practice or experiment. DRBL helps a lot on energy saving and cost down because of the diskless design. The DRBL provides a diskless or systemless environment for client machines. It works on Debian, Ubuntu, Mandriva, Red Hat, Fedora, CentOS and SuSE. DRBL uses distributed hardware resources and makes it possible for clients to fully access local hardware. We will demonstrate how DRBL really works on power saving and how much energy DRBL can save.

Cloud computing is more and more popular recently owing to its characteristics such as distribution file system [2, 3], implementation of large-scale tasks, analyzing very large data sets, etc. As we implement large-scale tasks, more power is consumed. Green computing is a growing topic in recent years. Our objective is to build an energy-saving environment by DRBL software.

Green computing is environmentally responsible use of computers and related resources [6, 7]. The goal of green computing is efficient resource utilization as well as reduced resource consumption [8]. Some suggested practices for Green Computing are the following ways: (1)

find out how much energy in IT system; (2) ensure unused equipment are turned off when it is not being used; (3) educate staff to the benefits of saving energy and recycling; and (4) identify IT management practices that reduce power consumption [6]. Some approaches of green computing are algorithmic efficiency; virtualization, terminal server, and power management (including power supply, storage, and other devices).

The architecture is composed of DRBL (Diskless Remote Boot in Linux) [1] MPICH2 (High-performance and Widely Portable MPI) [5], and CUDA (Compute Unified Device Architecture) [15] software. DRBL server offered clients to use MPICH2 and CUDA software. Clients don't install any software. Clients use PXE to connect the server and manage the deployment operating system. We are necessary installation CUDA driver and MPICH2 library on the server. Figure 1 shows system environment, and clients can run MPI program and CUDA program.

II. BACKGROUND

A. 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 [9]. DRBL supports lots of popular GNU/Linux distributions, and it is developed based on diskless and systemless environment for client machines. Figure 1 shows DRBL system architecture. DRBL uses PXE/Etherboot, DHCP, TFTP, NFS and NIS to provide services to client machines, so it is not necessary to install GNU/Linux on the client hard drives individually. Users just prepare a server machine for DRBL to be installed as a DRBL server, and follow the DRBL installation wizard to configure and dispose the environment for client machines step by step. It's really an easy job to deploy a DRBL environment on clustering systems even for a GNU/Linux beginner. Consequently, cross-platform and user-friendly are the key factors that make the DRBL become a superior clustering tool. DRBL can efficiently deploy diskless or diskfull cluster environment, and manage client. It configures these services (TFTP, NIS, DHCP, and NFS) to build a cluster environment. According to this implementation, an administrator just needs two steps to deploy cluster environment. (1) Step 1:

Installs DRBL packages and generates kernel and initrd for client; (2) Step 2: setup environment parameters, such IP address, and numbers of clients. It also provides cluster management and cluster system transformation (diskfull or diskless system).

B. CUDA

In the recent years, the graphic card has become powerful computing tools. The CUDA software created by NVIDIA Company, CUDA was parallel computing architecture Graphics processing units (GPUs) were originally designed to perform the highly parallel computations required for graphics rendering. The massive parallel computing ability has made GPUs very powerful devices in solving scientific and engineering problems recently. GPU is programmable level upgrade due to the great improvement of Semiconductor Processing technology.

General single GPU (Graphics Processing Unit) usually contains hundreds of programmable processing units, e.g., NVIDIA GEFORCE 9800 GT has 112 processors. It has powerful computation capability for engineering scientific problems. GPUs can handle for acceleration of image processing, linear algebra computations, image processing, molecular dynamics, ray tracing and so on.

In 2006, NVIDIA introduced a general-purpose parallel computing architecture, CUDA (Compute Unified Device Architecture). CUDA is a new parallel programming model that supports C programming language, FORTRAN, and now, OpenCL. C++ will soon be supported in the future according to the announcement of NVIDIA. CUDA gives instruction set for the parallel computation in CUDA GPUs. CUDA can solve data-intensive computing.

C. MPICH2

MPICH2 [4] is a tool of the Message-Passing Interface for CPU. MPICH2 was proposed by Argonne National Lab. MPICH2 is open source which is freely available license. It support system, including Microsoft Windows, UNIX and Linux (Ubuntu, Centos, Fedora, etc.). The latest of version is 2-1.0 that we can download on official website. MPICH2 is implementation for distributed-memory and shared memory in parallel computing. MPICH2 offer parallel programming library which supports C, C++, Fortran language. The MPICH2 offers us some library which uses very convenience. In this paper, we use DRBL deploy to MPICH2 and CUDA software on PC cluster.

III. ARCHITECTURE OVERVIEW FOR GPU AND CPU CLUSTER

A. GPU and CPU Cluster

We have 8 computers in the experiment environment. We combine DRBL with MPICH2, CUDA in my system environment. People can use GPU and CPU clustering quickly because DRBL helps to easy and quickly deploy GPU and CPU clustering environment. The project is suitable for PC classroom. The reasons are as the following. (1) PC Classrooms need the same system and centre management; (2) It allows all the configuration of your client computers by installing just one server machine; (3) Only the server has hard disk, and client has not hard disk; (4) Compared with diskfull design, the diskless design saves power consumption in PC classrooms.

Figure 1 shows system architecture. At least 20 clients in PC classroom can be easily managed by the DRBL server. Each client has 4 cores in CPU cluster. Each client gets a NVIDIA GEFORE 9800GT graphics card which supports CUDA. Clients are diskless machines. Users just prepare a server machine for DRBL installation as a DRBL server. They will boot via PXE/Etherboot. Users will find that all of these CPUs and memory of nodes will merge by DRBL server. We configure installation software in the DRBL server before we arrange clients. Users just prepare a server machine for DRBL installation wizard to configure a push the environment for client machines step by step. DRBL can support user to massively deploy and effectively manage clusters.

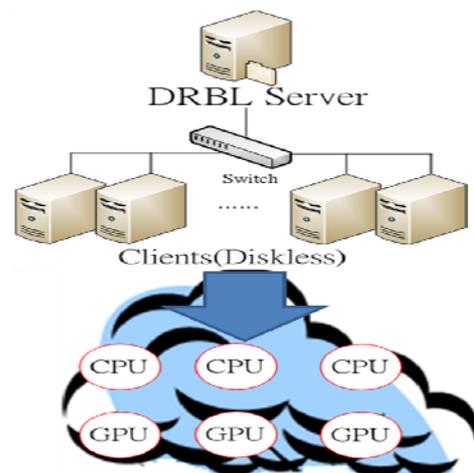


Figure 1. Using DRBL on GPU and CPU Cluster

It is efficient to deploy diskless cluster environment by DRBL component. In the first step, it has to install DRBL software in the server. If the OS is Debian or RPM package system, only 1 package from DRBL website should be installed. Then it needs to execute DRBL configuration command “drplsrv -i” to choose your Kernel version for nodes and automatically installs the packages that DRBL required, such as DHCP, NFS, NIS and TFTP. Then, using DRBL deployment command “drblpush -i” to push system environment to all clients. DRBL offers interactive dialog to help users to build DRBL environment and it automatically configures and starts 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 administrator how many clients will be setup. DRBL provides two methods for nodes IP address: (1) fixed 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 Experiment Cases.

B. Experiment Environment

The experiment uses computers in PC classroom in our research center. One of computers has already DRBL server with software (such as: MPICH2, g++, gcc, NVIDIA-Linux-driver, cudatoolkit, cudasdk and) in the server. It's very flexible to transform between two different modes cluster environment (diskfull and diskless) through DRBL.

The cluster has 1 server, 7 clients, and the PC are equipped with Intel® Core(TM)2 Quad CPU Q9550 @ 2.83GHz. Table I illustrates Hardware specifications and Software list.

TABLE I. HARDWARE SPECIFICATIONS AND SOFTWARE LIST

Hardware (PC)	Software
Intel® Core™ 2 Quad CPU Q6600 2.4Hz	Ubuntu 10.04
8 GB RAM	Kernel 2.6.32.21
160GB Hard disk	DRBL 1.94-27
Intel 82571EB Gigabit NIC	gcc, g++, fort77, MPICH2
Hardware(Network switch)	
Linksys SLM2048 48 port 10/100 Gigabit Switch	

C. Measured Environment

Figure 2 shows our experimental and measured environment. The measured environment includes one DRBL server, seven clients, one network switch, one electricity monitor and one notebook for data collection. The server was installed DRBL and other software. It can

transform between diskfull and diskless cluster environment easily. These one power distribution units (PDUs) supply power to these machines and transport power utilization (ampere) to server.

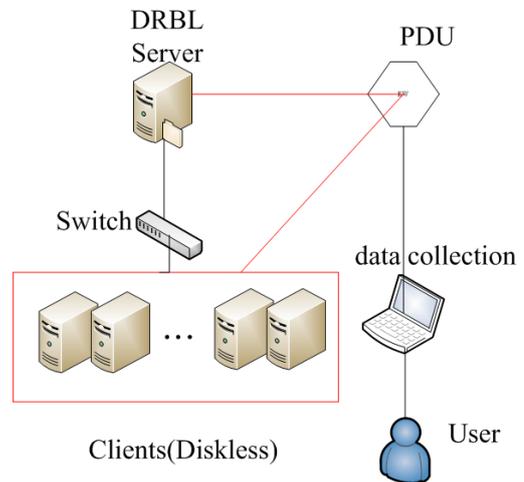


Figure 2. Energy Measured Environment Cluster

The electricity measurement has two categories on my environment. These categories are performed with two evaluation cases: (1) Examples of CUDA (2) Examples of MPICH2. CUDA uses GPU Computing SDK code samples. The MPICH2 solves Laplace equation, Gauss elimination, and performs Matrix multiplication.

IV. POWER CONSUMPTION CALCULATION AND EXPERIMENT DESIGN

A. Amazing PDU Utility

Figure 3 is the power evaluation (Amazing PDU Utility [10]). The major functions of Amazing PDU Utility are as the following: (1) an Ethernet interface for the built-in web server; (2) Audible Alarm (warning and overload); (3) Graphical User Interface; (4) Meter; (5) Calculation of power consumption ; (6) Graphics report output. The PDU has a total current and Kilowatt record every minute. User can observe the current variable and download relative data from electricity monitor.



Figure 3. Energy Measured Environment Cluster

B. Experiment Design

We provide three examples to test. We use GPU Computing SDK code samples. It can be download from the NVIDIA website. We do compile the GPU Computing SDK code samples in ubuntu system. The examples are briefly introduced as follows:

- An NBODY use the gravitation potential to determining inter-atomic forces. The formulation of force present by the following:

$$f_{ij} = G \frac{m_i m_j}{\|r_{ij}\|^2} * \frac{r_{ij}}{\|r_{ij}\|}$$

- PARTICLE systems are used in many simulations from Molecular dynamic and astrophysics simulations and computational fluid dynamics, etc.
- The POSTPROCESSGL example use CUDA to interoperability to post-process an image of a 3D scene generated in OpenGL.

We use these examples to test in MPICH2 cluster. The following examples have paralleled programming. The example is introduced by the following.

- Using finite difference method to solve Laplace equation. The Laplace equation is given by:

$$\nabla^2 \phi = \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0 \tag{1}$$

$$IC : \phi(0,0) = 100$$

$$Bc : \phi(x,0) = \phi(0, y) = \phi(1, y) = \phi(x,1) = x^2 - y^2$$

By numerical discretization, substituting Eqn in the Laplace equation, we defined

$$\phi_{i,j} = \frac{1}{4}(\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1}) \tag{2}$$

This example handles 4096*4096 meshes on 32 processors.

- The Gauss Elimination method solves matrix equations. The algebraic equation is AX=B. This example handles 3200*3200 meshes on 32 processors.
- The MATRIX MULTIPLICATION is important for linear algebra. This example handles 4096*4096 meshes on 32 processors.

C. Power Consumption Result

Table II is Kw/h for CUDA, MPICH2 in power consumption. Table III shows the power saving percentage in power evaluation experiment. Table IV is execution time. NBODY example saves almost 9 % power compared to boot with hard disk, PARTICLE example saves almost 11 % power compared to boot with hard disk, POSTPROCESSGL example saves almost 5.8 % power compared to boot with hard disk, LAPLACE EQUATION example saves 3.8 % power compared to boot with hard disk, GAUSS ELIMINATION example saves 3.4 % power compared to boot with hard disk, MATRIX MULTIPLICATION example saves 4.1 % power compared to boot with hard disk.

TABLE II. ALL EXAMPLES OF POWER CONSUMPTION

CUDA(1-3) MPICH2(4-6)	PXE boot with DRBL server (Kw/h)	Local boot with hard disk (Kw/h)
1.NBODY	0.02889	0.03178
2.PARTICLES	0.02864	0.03225
3.POSTPROCESSGL	0.02586	0.02746
4.LAPLACE EQUATION	0.04069	0.04234
5.GAUSS ELIMINATION	0.02439	0.02525
6.MATRIX MULTIPLICATION	0.03982	0.04153

TABLE III. POWER SAVING PERCENTAGE IN POWER EVALUATION EXPERIMENT

CUDA(1-3) MPICH2(4-6)	Boot with DRBL design VS boot with hard disk (%)
1.NBODY	9.09377
2.PARTICLES	11.1938
3.POSTPROCESSGL	5.826657
4.LAPLACE EQUATION	3.897024
5.GAUSS ELIMINATION	3.405941
6.MATRIX MULTIPLICATION	4.117505

*Boot with DRBL design VS boot with hard disk = (Local boot with hard disk - PXE boot with DRBL server) / Local boot with hard disk * 100%.

TABLE IV. TIME CONSUMING IN POWER EVALUATION EXPERIMENT

CUDA job	PXE boot with DRBL server (s)	Local boot with hard disk (s)
NBODY	60	60
PARTICLES	60	60
POSTPROCESSGL	60	60

TABLE V. TIME CONSUMING IN POWER EVALUATION EXPERIMENT

MPICH2 job	PXE boot with DRBL server (s)	Local boot with hard disk (s)
Laplace equation	75	77
Gauss elimination	42	44
Matrix multiplication	66	70

V. DISCUSSION AND CONCLUSION

We can say that the diskless design of DRBL can bring effect on power saving for CPU and GPU applications. Table III presents all cases. It shows that the power consumption percentage in our experiment. Owing to the bottleneck of high throughput I/O data communication and overhead, the DRBL architecture is not suitable for I/O intensive applications. On the other hand, it brings a great benefit to those CPU intensive applications with RAM disk [11].

In the future, we will research into the relation between power consumption and various CPU, GPU frequencies on Ubuntu. We can use cpufreqd instruction that adjusts CPU frequency under diskfull and diskless environment. CPU frequency is dynamically controlled by cpufreqd. We can use NVCILOCK software to overclock NVIDIA based video cards on the Linux system. GPU frequency is dynamically controlled by NVCILOCK.

We can observe the effects of various CPU, GPU frequencies on power consumption under diskfull and diskless environment. The green computing is more and more important in the future, we still have lots of performance tuning and power measuring work for advanced evaluation. Some of the tools are used for advanced evaluation, such as bootchart [12], lm-sensors [13] and powertop [14].

REFERENCE

[1] Diskless Remote Boot in Linux (DRBL), NCHC. [Online]. <http://drbl.sourceforge.net/> [accessed; Jan, 2011].

[2] J. Cope, M. Oberg, H. M. Tufo, and M. Woitaszek, "Shared Parallel Filesystems in Heterogeneous Linux Multi-Cluster Environments" in Proc. 6th LCI International Conference on Linux Clusters: The HPC Revolution, 2005.

[3] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud Computing and Grid Computing 360-Degree Compared," Grid Computing Environments Workshop, 2008. GCE '08, Vol. 12-16, pp. 1-10, 2008.

[4] DeinoMPI. [Online]. <http://mpi.deino.net/> [accessed; Feb, 2011].

[5] MPICH2: High-performance and Widely Portable MPI [Online].

<http://www.mcs.anl.gov/research/projects/mpich2/> [accessed; Jan, 2011].

[6] Green Computing, Wikipedia. [Online]. http://www.greenlivingpedia.org/Green_computing [accessed; Jan, 2011].

[7] Green Computing definition webpage on SearchDataCenter.com. [Online]. Available: http://searchdatacenter.techtarget.com/sDefinition/0,,sid80_gci1246959,00.html [accessed; Jan, 2011].

[8] Green Computing webpage on Computing. [Online]. <http://www.computing.co.uk/ctg/special/1814241/green-computing> [accessed; Feb, 2011].

[9] W. C. Kuo, C. Y. Tu, and Y. T. Wang, "Deploy Kerrighed SSI Massively Using DRBL," HPC ASIA 2009, 2009.

[10] Amazing PDU Utility [Online]. ftp://ftp.opengear.com/ip-pdu/cd/User_Manual/PDU_Utility_User_Manual_v2.pdf [accessed; Jan, 2011].

[11] Che. Y. Tu, W. C. Kuo, Y. T. Wang, and S. Shiau, "Building Energy Efficient ClassCloud using DRBL", 10th IEEE/ACM International Conference Grid Computing, Vol. 13-15, pp. 189-195, 2009.

[12] The Bootchart website. [Online]. Available: <http://www.bootchart.org/> [accessed; Nov, 2010].

[13] The lm-sensors website. [Online]. Available: <http://lm-sensors.org> [accessed; Nov, 2010]

[14] The PowerTop website. [Online]. <http://www.lesswatts.org/> [accessed; Nov, 2010].

[15] P. Harish and P. J. Narayanan, "Accelerating Large Graph Algorithms on the GPU Using CUDA", In High Performance Computing – HiPC 2007, Vol. 4873, pp. 197-208, 2007.