Towards Green Computing using Diskless High Performance Clusters

K. Salah[1], R. Al-Shaikh[2], M. Sindi[2]
[1] Computer Engineering Department, Khalifa University of Science, Technology and Research, Sharjah, UAE
[2]EXPEC Computer Center, Saudi Aramco, Dhahran, Saudi Arabia
khaled.salah@kustar.ac.ae, {raed.shaikh,sindimo}@aramco.com

Abstract – **In recent years, significant research has been conducted to boost the performance and increase the reliability of high performance computing (HPC) clusters. As the number of compute nodes in modern HPC clusters continues to grow, it is critical to design clusters with low power consumption and low failure rate. In particular, it is widely known that the internal disk drives of compute nodes (in the case of diskfull clusters) are a major source of failures. In addition, these diskfull HPC clusters tend to require more power and cooling requirements compared to diskless clusters. In this paper, we propose and implement a large-scale Infiniband-based diskless HPC cluster. The paper presents the cluster configuration and evaluates its performance using various High Performance LINPACK (HPL) benchmarks. The performance is measured in terms of the overall efficiency, speed in Giga-Floating Point Operations per Second (GFLOPS), and HPL execution time. We also measure temperature and power consumption. We compare the performance measurements of our diskless cluster to its diskfull counterpart. For our measurement and comparison, we consider three cluster sizes of 32, 64, and 126 compute nodes.**

Index Terms – **Cluster Computing and Architecture, Green Computing, Linux, Performance Evaluation.**

I. INTRODUCTION

In recent years, we have witnessed a growing interest and research in improving the performance and reliability of the high performance computing components and infrastructure. In addition, power and cooling have become a major issue in designing High Performance Computing (HPC) solutions. Green Top500 [1] was established primarily to address this concern. For all those reasons, many HPC providers and HPC centers are striving to attain all these goals with the least amount of side-effects possible. One of these attempts is researching the diskless HPC systems.

Diskless HPC clusters consist of compute nodes with no local disks. Instead, the compute nodes get their OS image during boot-up by using a centrally located device (or disk node) over a local LAN. In some designs, an internal network (e.g. 1 Gbps Ethernet) is used to provide not only inter-processor communications (IPC) among compute nodes but also a medium for booting and file transmission. In other advanced designs, as shown in this paper, the IPC communication is carried out on a separate extremely high-speed interconnect technology such as Infiniband or Myrinet.

There is a number of obvious advantages to diskless clusters. First, the cost per cluster node becomes lower. Nowadays, the average cost of a server-level disk is about $200 [3]. This translates to $102,400 for a 512 nodes cluster. Second, diskless clusters have smaller footprints, i.e, lower power and cooling requirements. Third, cluster configuration and setup are consistent. In a diskfull cluster, system administrators spend considerable amount of time in developing and running script to ensure identical installations of OS images and files for all individual cluster nodes. In diskless cluster, since all nodes bootup over a network from a centralized disk server, identical OS images and installation files are ensured, thereby achieving system and file consistency across all compute nodes.

The real advantage to diskless clusters, however, is the reduced maintenance, or downtimes. With diskless systems, all mechanical parts – apart from the internal fans – are eliminated. For example, the mean time between failures (MTBF) of an internal disk is reported to be 300,000 hours, or 34 years of continuous operation [2]. Thus, if there is a cluster of 100 nodes, 3 to 4 disks will be replaced every year. If there is a cluster with 12,000 nodes, then on average, a disk fails every 25 hours, or around every day.

On the other hand, there are clearly obvious drawbacks associated with diskless HPC. The most obvious drawback is the added network traffic. Since the compute nodes load their OS image by using a centrally located device over a local LAN, a diskless HPC cluster configuration generates more network traffic than a diskfull HPC cluster by reading the image over LAN. Moreover, if the network connection or the centralized OS image is not available, none of the compute nodes will be accessible. Solutions exist for these drawbacks [3], such as creating a RAM disk on each compute node by allocating part of the compute node's main memory as a partition for the file system. The RAM disk will be used for storing the most frequently accessed files. Therefore, the compute node can access some files from local memory instead of through the network.

The rest of the paper is organized as follows. In Section II, we present our test environment and detailed configurations. In Section III, we describe our benchmark methodology and the tools used to conduct the experiments. The performance and experimental results are discussed and analyzed in Section IV. Section V concludes the study and identifies future work.

As described, many benchmarks were done in the past to measure the performance of diskless HPC systems. Most these benchmarks were done using the Ethernet LAN as a local cluster interconnect for communication as well as for loading the diskless computes with the OS image.

Contributions. Our primary contributions in this paper are as follows. **First**, our diskless experimental setup and configuration are unique from prior experiments that exist in the literature. Our proposed cluster is more practical as it is using state-of-the-art hardware for nodes and advanced interconnect technology. **Second**, our testbed out-scale other prior testbeds. Our cluster consists of a 126 compute nodes, with each node having a quad-core processor. These nodes with multi-core processors would impose high demands on the communication network. **Third**, in sharp contrast to other related and prior experimental work, we study and measure the performance of diskless clusters in terms of a variety of key metrics and measures of engineering and design importance. **Fourth**, the temperature and power consumption are also measured and reported in order to quantify the benefit of using diskless clusters in terms of power saving.

II. THE CLUSTER DESIGN

To perform our diskless vs. diskfull benchmark evaluation, a DELL cluster of PowerEdge M610 Blade Servers was used. As shown in Figure 1, the cluster consisted of 126 nodes with dual sockets and Intel QuadCore X5570 (Nehalem) 2.93GHz processors. The operating system running on the nodes was RedHat Enterprise Linux Server 5.3 with the 2.6.18-128.el5 kernel. Each node was equipped with an Infiniband Host Channel Adapter (HCA) supporting 4x Double Data Rate (DDR) connections with the speed of 16Gbps, and 1Gbps Ethernet connection. The Infiniband connection was used for the actual inter-process communication while the Ethernet connection was mainly used for the OS image boot-up and remote access. Each node also had 12 GB (6 x 2GB) DDR3 1333MHz of memory, therefore, the total amount of memory the system had was around 1.5 TB.

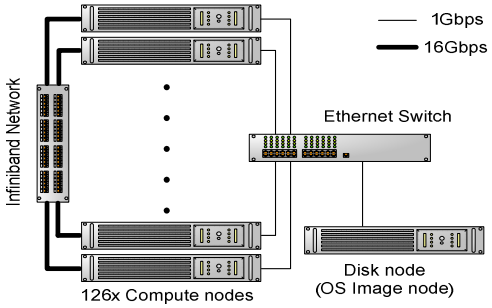

Fig 1. Experimental setup and communication

The physical layout of our cluster consists of six racks, each rack contains two chassis, and each chassis can host up to 12 blade nodes. That is, each rack supports 24 nodes. From each node we had a 4x-DDR Infiniband connection going to a central 144-port Qlogic Infiniband switch. Figure 2 shows the Infiniband interconnection design as described. It is important to mention that this design is considered non-blocking as each node guarantees to have the full 4x DDR 16Gbps interconnect speed. This fast interconnect would drive the cluster to a higher utilization, which in theory, may affect the diskless concept.

Our Infiniband interconnect topology uses three switches. A top-level switch that connects two leaf switches. Each leaf switch can support up to 72 nodes, as it connects 3 racks with each rack supporting 24 compute nodes. Under this configuration, IPC communication among nodes of 32 and 64 clusters is localized to one leaf switch, but for the cluster of 128 nodes, the top-level switch is involved to support more nodes.

Figure 3 shows an example of the HPL input file that was used for our 126-node benchmark runs with the tuned input parameters. Two other HPL input files were generated using the same techniques above for choosing the values of *P, Q* and *N*, one to be run on a system with 32 nodes and the other to be run a system with 64 nodes.

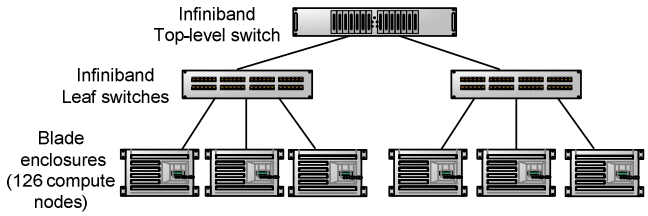

Fig 2. The DDR Infiniband interconnect topology of a 126 nodes cluster

III. EVALUATION METHODOLOGY

In this section, we present our configuration to setup and perform (HPL) benchmark for both diskless and diskfull nodes. To measure and compare performance, we use "LINPACK" benchmark. LINPACK is one of the standard benchmarking tools for HPC, is a collection of Fortran subroutines that analyze and solve linear equations and linear least-squares problems. The package solves linear systems whose matrices are general, banded, symmetric indefinite, symmetric positive definite, triangular, and tridiagonal square. In addition, the package computes the *QR* and singular value decompositions of rectangular matrices and applies them to least-squares problems. LINPACK uses column-oriented algorithms to increase efficiency by preserving locality of reference" [4].

The last parameter that we discuss is "*NB*", which is the block size in our grid. Usually block sizes giving good results are within the (96, 104, 112, 120, 128, …, 256) range, and from our experimental runs, the value of 224 for *NB* has shown to give the best results compared to the various test runs we did with other values of *NB*.

```
HPLinpack benchmark input file
Innovative Computing Laboratory, University of
Tennessee
HPL.out output file name (if any)
file   device out (6=stdout,7=stderr,file)
1      # of problems sizes (N)
414400 Ns
1      # of NBs
224    NBs
0 PMAP process mapping (0=Row-,1=Column-major)
1   # of process grids (P x Q)
16     Ps
63     Qs
16.0   threshold
1      # of panel fact
0      PFACTs (0=left, 1=Crout, 2=Right)
1      # of recursive stopping criterium
4      NBMINs (>= 1)
1      # of panels in recursion
2      NDIVs
1      # of recursive panel fact.
0      RFACTs (0=left, 1=Crout, 2=Right)
1      # of broadcast
0   BCASTs (0=1rg,1=1rM,2=2rg,3=2rM,4=Lng,5=LnM)
1      # of lookahead depth
0      DEPTHs (>=0)
2      SWAP (0=bin-exch,1=long,2=mix)
128    swapping threshold
0   L1 in (0=transposed,1=no-transposed) form
0   U  in (0=transposed,1=no-transposed) form
1      Equilibration (0=no,1=yes)
8      memory alignment in double (> 0)
```

Fig 3. The HPL file configuration for a 126 nodes cluster with N value set to 92% of available memory

To evaluate the performance of our system, the theoretical peak execution speed of the system had to be calculated in GFLOPS to know the maximum theoretical speed which cannot be exceeded. In the Top500 supercomputers terminology [1], the maximum theoretical system speed is referred to as "Rpeak", while "Rmax" is the actual speed obtained when HPL is run. The "Efficiency" of the system is the ratio of Rmax to Rpeak (Rmax/Rpeak). The efficiency can be affected by the underlying interconnect technology used for IPC communication among compute nodes, the amount of RAM available for individual compute nodes, as well as the MPI implementation used for communication among cluster nodes of the system. [4]

For example, to calculate the theoretical Rpeak value for a system that consists of 126 nodes each with 8 Nehalem cores capable of 4 operations per cycle with a speed of 2.93GHz per core, the following formula is used:

$$Rpeak = CPU\ Speed(GHz) \times Total\ Cores \times Ops/Cycles$$
$$= 2.93 \times (8 \times 126) \times 4$$
$$= 11813\ \text{GFLOPS}$$

For our test, each node had 12GB for memory and 8 physical cores, and each launch on the cluster was utilizing all cores available. The HPL problem size "N" was set to 92% of the total memory size of the system in each case.

IV. EXPERIMENTAL RESULTS

In this section, we present and discuss our experimental results based on the benchmark methodology explained in Section III. All measurements reported in this section are the average readings of three runs. The performance is measured and compared for both diskless and diskfull clusters while varying the cluster size. We study and measure the performance in terms of HPL efficiency, GFLOPS, HPL runtime and IO node disk and network utilizations. We also examined the disk swapping effect on the diskless high performance cluster.

It is observed from Figure 6 that the execution time of HPL increases when the cluster size increase, although larger cluster size would have more memory and more processing speed (i.e. GPFLOPS). The reason for this increase is due to the way that LINPACK benchmark works. LINPACK provides three separate benchmarks that can be used to evaluate the performance of a dense system. The first is computing a 100 by 100 matrix, the second is for a 1000 by 1000 matrix, while the third benchmark, of a particular interest, is dependent on the algorithm chosen by the manufacturer and the size and speed in addition to the available memory of the system being benchmarked [22]. The third benchmark was the one used. In other words, the benchmark execution size is made proportional to the size of the cluster in terms of memory, GFLOPS, and nodes. Large clusters will have larger benchmarks to run. This clearly explains the increase of execution time exhibited under clusters of 64 and 126 nodes.

Another experiment was performed where the cluster nodes were forced to swap to disk, by increasing the HPL required memory (i.e. *N* as an HPL input value) to 95%. The intension of this experiment was to measure the effect of disk swapping on the diskless cluster. While the diskfull system continued to run with typical swapping activities, Out-of-Memory (OOM) process was seen on the diskless compute nodes, causing the nodes to kill system processes randomly when they ran out of memory. The diskless HPC did not succeed running the benchmark when swapping is needed. Obviously, that is one limitation of running diskless HPC cluster.

Table 1 Temperature and power consumption (per a single node) for diskfull vs. diskless HPC

#Nodes/State	Avg. Temp (^{O}C)	Avg. Power (Watts)
126 Nodes/diskfull	18	280
126Nodes/diskless	18	277

We also measured the Gigabit Ethernet network utilization and disk I/O activities at the disk node throughout the experiment run time. Figure 7 illustrates the disk IO activity on the image node while the diskless nodes are booting. As shown, the first read burst at 29s was caused by loading the kernel image into the diskless nodes, while the second burst at approximately 70s was caused by the start of actual loading OS files. Beyond 146s, the OS image was entirely loaded into memory and minimal disk reads were taking place. On the other hand, disk writes continued as the diskless nodes were writing their states on the disk node, such as system and kernel logs (e.g. /var). These writes, however, did not exceed 8MB/s aggregate.

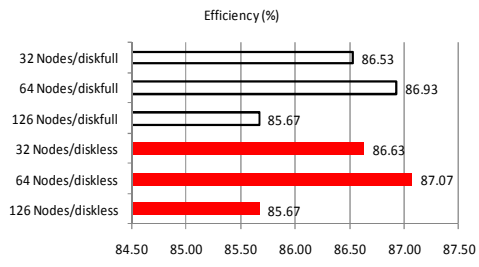

Fig 4. HPL effeciency

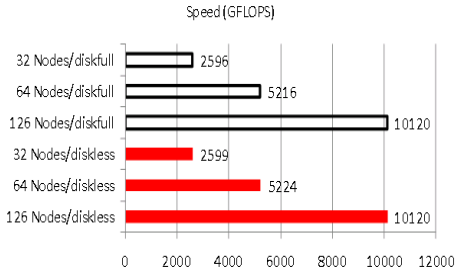

Fig 5. Execution speed in terms of GFLOPS

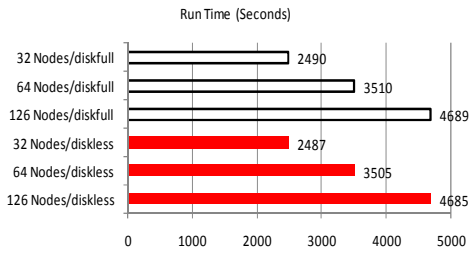

Fig 6. HPL execution time

Figures 4, 5 and 6 show the HPC system efficiency, execution speed in GFLOPS and execution time (in seconds) of both diskfull and diskless configurations. LINPACK efficiency is obtained by dividing the theoretical peak speed (Rpeak) by the maximal LINPACK speed achieved (Rmax). As shown in Figure 4, diskless cluster provides comparable efficiency to the diskfull. And as exhibited in both Figures 5 and 6, diskless slightly outperforms diskfull in terms of execution speed and run time.

During the time of HPL run on all 126 nodes, the temperature of both CPUs, the mother board's temperature, and the power consumption for all 126 nodes were monitored while running diskless and on disk. DELL's version of Intelligent Platform Management Interface (IPMI) tool [17] was used to collect such readings from all 126 nodes while the benchmarks were running on the nodes and fully utilizing the CPU and memory. In terms of temperature and heat dissipation, the diskfull and diskless readings

were about the same at 18°C while performing the HPL test. In terms of power consumption, however, the diskless nodes operated with an average of 277 Watts per node, compared to 280 Watts per node for the diskfull configuration. This difference in power saving matches the hardware specifications of the published DELL internal disks power consumption [18] where they consume around 5 Watts per node. According to the United States' Department of Energy statistics for 2009, the average price for electricity in the USA is 10.01 cents per kW hour [23]. This would translate to an annual saving of U.S. $31,567 for a diskless cluster consisting of 12,000 nodes compared to a diskfull cluster of the same size.

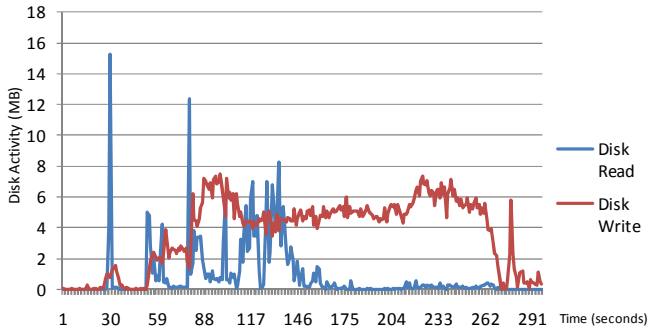

Fig 7. Disk I/O measured at the disk node during the bootup of diskless compute nodes

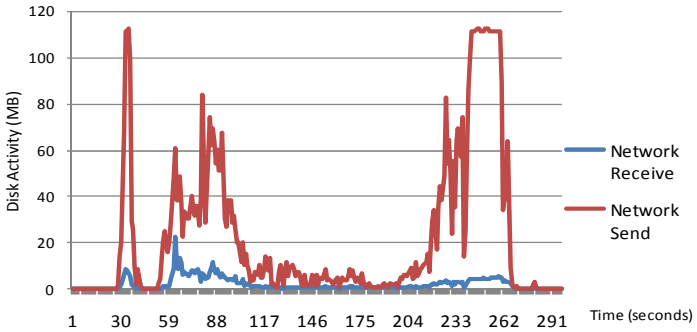

Fig 8. Network activities measured at the disk node during the bootup of diskless compute nodes

V. CONCLUSION AND FUTURE WORK

Diskless HPC clusters are becoming a compelling alternative with greater benefits when compared to diskfull clusters, particularly in terms of reducing power consumption and failure rate. In this paper, we have presented a design and a configuration of a state-of-the-art diskless cluster using Infiniband-interconnect technology. Our cluster consisted of 126 compute nodes equipped with quad-core processors. We measured and evaluated the performance of such a cluster in terms of key metrics which include overall efficiency, execution speed (in GFLOPS), and execution time. We also measured temperature and power consumption. These measurements of diskless cluster were compared to its respective diskfull cluster, considering three cluster sizes of 32, 64, and 126 compute nodes. Our results show that diskless clusters yield comparable performance to diskfull clusters, and in some cases outperform the diskfull. In terms of power consumption, diskless clusters clearly win with power significant saving of at least 3 Watts per node. On the other hand, diskless clusters have shortcomings. For one, diskless clusters require ample of RAM. It was demonstrated that if compute nodes are forced to perform disk swapping by decreasing their available memory, the compute nodes will freeze. Another obvious shortcoming is that the disk node in a diskless cluster can be a single point of failure. However, these two shortcomings can be alleviated by increasing the RAM of compute nodes and by having more reliable disk nodes that use advanced network storage technologies such as NAS and RAID technology.

REFERENCES

[1] The Green Top500 List. Available at: http://www.green500.org/

[2] J. Sloan, "*High performance Linux clusters with OSCAR, Rocks, openMosix, and MPI*", O'Reilly Publication, 2005.

[3] J. Laros and L. Ward, "*Implementing scalable diskless clusters using the network file system*", Proceedings of the Los Alamos Computer Science Institute (LACSI) Symposium 2003, USA, October, 2003.

[4] B. Guler, M. Hussain; T. Leng, and V. Mashayekhi, "*The Advantages of Diskless HPC Clusters using NAS*", DELL Inc., Nov. 2002.

[5] C. Yang and Y. Chang, "*A Linux PC Cluster with Diskless Slave Nodes for Parallel Computing*", High-Performance Computing Laboratory, Department of Computer Science and Information Engineering, Tunghai University, Jan, 2003.

[6] C. Engelmann, H. Ong and S. Scott, "*Evaluating the Shared Root File System Approach for Diskless High-Performance Computing Systems*", Proceedings of the 10th LCI International Conference on High-Performance Clustered Computing (LCI-09), Colorado, 2009.

[7] Terry Jones, Andrew Tauferner, Todd Inglett, et al., "*HPC Colony: Linux at Large Node Counts Report from Experiments Conducted on Sixth BGW Day*", August 10, 2007

[8] J. Laros, C, Segura and N. Dauchy, "*A Minimal Linux Environment for High Performance Computing Systems*", The 10th World Multi-Conference on Systemics, Cybernetics and Informatics, Florida, July 2006, pp.130-138.

[9] C. Lu., "*Scalable Diskless Checkpointing for Large Parallel Systems*", MSc. Thesis, University of Illinois at Urbana-Champaign, 2002.

[10] B. Maher, "*Techniques to Build a Diskless Boot Linux Cluster of JS21 Blades*", IBM Red Book, 2006.

[11] T. Morgan JR., "*DRBL: Diskless Remote Boot in Linux*", Master's Capstone Project on High Performance Computing, April, 2006.

[12] Z. Chen, G. Fagg, E. Gabriel, J. Langou, T. Angskun, G. Bosilca, and J. Dongarra. "*Building fault survivable MPI programs with FT-MPI using diskless* Checkpointing", Proceedings of the 10th ACM SIGPLAN symposium on Principles and Practice of Parallel Programming (PPoPP), Chicago IL, June 2005, pp.213-223.

[13] TOP500 Supercomputers. Available at: http://www.top500.org

[14] HPL - High-Performance Linpack Benchmark. Available at: http://www.netlib.org/benchmark/hpl

[15] S. Frank and R. Haskin, "*GPFS: A shared-disk file system for large computing clusters*", Proceedings of 1st Conference on File and Storage Technologies (FAST), USA, Jan., 2002, pp. 231–244.

[16] P. Reisner and L. Ellenberg, "*Replicated storage with shared disk semantics*", Proceedings of the 12th International Linux System Technology Conference (Linux-Kongress), Germany, Oct, 2005, pp.111-119.

[17] DELL IPMI. Available at: http://linux.dell.com/ipmi.shtml

[18] DELL Blades Server for HPC M610. Available at: http://www.dell.com/us/en/enterprise/servers/server-poweredge-m610.

[19] C. Juszczak, "*Improving the Write Performance of an NFS Server*", Proceedings of the USENIX Winter 1994 Technical Conference, USENIX, Association Berkeley, CA, USA, pp. 20-20, 1994.

[20] Red Hat Knowledge Base: The Optimal Number of nfsd Threads. Available at: http://kbase.redhat.com/faq/docs/DOC-2237

[21] Pallas Benchmarking tools. Available at: http://people.cs.uchicago.edu/~hai/vcluster/PMB/

[22] J. Dongarra, J. Luszczek, and A. Petitet, "*The LINPACK benchmark: past, present and future*", in the Journal of Concurrency and Computation: Practice and Experience, 2003, pp. 803-820.

[23] Energy Information Administration, USA Department of Energy http://www.eia.doe.gov/cneaf/electricity/epm/table5_6_b.htm