Early Linpack Performance Benchmarking on IPE Mole-8.5 Fermi GPU Cluster

Xianyi Zhang [1),2)] and Yunquan Zhang [1),3)]
1) Laboratory of Parallel Software and Computational Science, Institute of Software, Chinese Academy of Sciences, Beijing,100190, China
2) Graduate University of Chinese Academy of Sciences, Beijing,100190, China
3)State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences , Beijing,100190, China
xianyi@iscas.ac.cn , zyq@mail.rdcps.ac.cn

1.Introduction

Linpack is a de facto standard benchmark for supercomputer. Based on Linpack benchmark, TOP500 website [1] lists top 500 high performance computers in whole world every half year.

NVIDIA Fermi GPU [2] achieves a significant improvement in general purpose computing. Especially, faster double precision performance, ECC and cache are very important features for scientific computing.

According to the above advantages, IPE [3] used Fermi GPUs to build their heterogeneous supercomputer named Mole-8.5, though it is mainly designed for multi-scale discrete simulations that are most suitable for fine-grain massive parallel process, it is still interesting to see its Linpack performance. We carried out an early benchmarking of the system and have got 207.3TFlops, No.19 on Top500 2010 June list.

In the following, we will introduce the architecture of computing node in IPE Mole-8.5 Cluster. Then, we will briefly review HPL on Fermi package modified by NVIDIA and the tuning tips. Next, we will analyze the data transfer between CPU and GPU. At last, it's the result and conclusion.

2.The Architecture of IPE Mole-8.5 Cluster

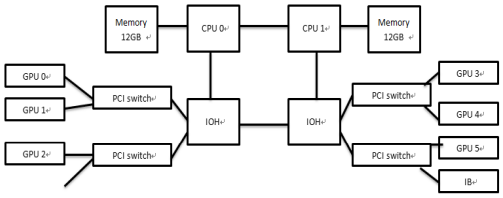

Fig. 1. The architecture of computing node in IPE Mole-8.5 Cluster

Figure 1 shows the architecture of computing node in IPE Mole-8.5 Cluster. There are 2 Intel quad cores CPUs and 2 IOH chips, which are connected with QPI. In each IOH chip, there is 2 PCI switches which provide 2 PCIe X16 slots. Therefore, each IOH chip provides 4 PCIe X16 slots. One IOH Chip equips 3 Fermi GPUs and the other IOH Chip equips 3 Fermi GPUs and 1 Infiniband HCA. Table 1 lists the configuration details.

Table 1. The configuration of computing node (At first, it had 24 GB memory. In 288 and 320 nodes Linpack runs, we enlarged the memory to 48GB.)

CPU	2-ways Intel Xeon E5520 (Quad Cores) 2.26GHz
Memory	24GB,48GB DDR3
GPU	6 NVIDIA Tesla C2050 Cards
OS	CentOS 5.4 Linux Kernel 2.6.16
Compiler	GCC 4.1
CUDA SDK	3.0
NVIDIA driver	195.36.20
MPI	OpenMPI 1.4.2
BLAS	GotoBLAS2-1.13

3.HPL on Fermi package and tuning tips

NVIDIA provided modified HPL package for Fermi. The method is similar with reference [4] and the main modifications are listed as follow.

•Implement CPU/GPU hybrid DGEMM function as shown as Figure 2. Automatically split the work load between GPU and CPU. Use optimized DGEMM function for Fermi

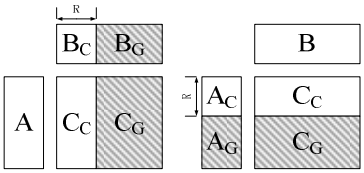

Fig.2. CPU/GPU hybrid DGEMM (The shadow matrix is running on GPU)

- Implement CPU/GPU hybrid DTRSM function. The method is similar with DGEMM
- Use CUDA stream to overlap the data transfer and computing.

In tuning Linpack performance, DGEMM/DTRSM split ratio, problem size(N), block size (nb) and grid size (PxQ) are the key factors. You should firstly optimize these. Although there is many GPUs and CPUs in single node, we find that binding the process to the neighboring CPU and GPU has the better performance.

4.Analyzing data transfer between CPU and GPU

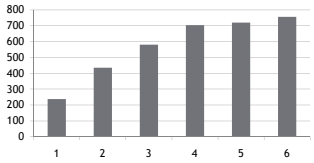

Fig.3. Linpack performance (Gflops) on single node (Use 1-6 GPUs)

As Figure 3 shown, the speedup drops significantly on 5-6 GPUs. Obviously, the bottleneck is data transfer between CPU and GPU.

To investigate the requirement of data transfer bandwidth via PCIe, we logged the amount of data. In single GPU, when N equals 51200 and NB is 1024, the amount of data is 371GB from CPU to GPU. In backward direction, it's 337GB. The Linpack running time is about 379 seconds. Therefore, the bandwidth per GPU needs above 2GB/s. In 6 GPUs, we developed a rough trace tool to replay the data transfer behavior. In the replay running, data transfer costs about 54.9s whereas Linpack costs 61.9s.

Therefore, the bottleneck indeed is PCIe bandwidth in this system. The PCI switch could not support enough bandwidth for Linpack benchmark.

5.Linpack result

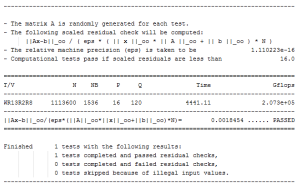

Fig.4. Linpack performance (Tflops) on 80-320 nodes and final output

Figure 4 depicts Linpack performance from 80 nodes to 320 nodes. According to the Linpack results, there is good scalability in multi IPE Mole-8.5 nodes. Meanwhile, the ECC of Fermi smoothes our Linpack runs in large scale.

6.Conclusion

In short, we got 207.3TFlops on 320 nodes (1920 NVIDIA Fermi GPUs). The bottleneck of Linpack benchmark on this system is data transfer between CPU and GPU via PCIe. We advice that, for better Linpack efficiency, single node equips less than 4 GPUs. Nevertheless, for Mole-8.5, the emphasis is on multi-scale discrete simulations which has lighter load on PCIe, so current architecture may still be justified.

Acknowledgement

This work is partly supported by Ministry of Finance under the Grant (No. ZDYZ2008-2), the National 863 Plan of China (No.2006AA01A125, No. 2009AA01A129, No. 2009AA01A134).

Reference

[1] http://www.top500.org
[2] http://www.nvidia.com/object/fermi_architecture.html
[3] http://www.ipe.ac.cn/csms
[4] M.Fatica, "Accerlerating linpack with CUDA on heterogeneous clusters," in Proc. Of 2nd Workshop on General Purpose Processing on Graphics Processing Units, 2009