SC19 SCC Final Architecture Proposal

Team GeekPie_HPC
ShanghaiTech University

1 HARDWARE CONFIGURATION

1.1 Configuration Overview

To better balance the performance and power consumption in each scenario, we adopt the design of heterogeneous configurations. Our cluster consists of six nodes: four compute nodes equipped with GPU accelerators and two storage nodes. The following table summarizes the system configuration details. We describe the reasons for choosing the hardware and the power evaluation method in the following sections.

Table 1: Parameters of Hardware Components

Type	GPU node	Storage node
Processor	2 × Intel Xeon Gold 6240	
Chipset	Intel C620 series chipset	
Memory	12 × 16GiB, DDR4 RDIMM, 2666 MHz	
Storage	TBD	4 × 480GB Intel SSD
GPU Card	4 × NVIDIA Tesla V100	/
Fans	8 × redundant hot-swap 8025 fans	
HCA Card	Mellanox IB ConnectX-3 HCA card, FDR	

1.2 Power Consumption

In this section, we list estimations of each hardware component and calculate the total power consumption of each configuration in some common scenarios.

Though CPUs and GPUs will reach a peak of 150W and 250W accordingly, we can reduce clock frequencies of CPUs and GPUs to sustain the total power within limits. As the HPL benchmark intent to test the peak performance of CPUs and GPUs, we only get 12 GPUs involved in the benchmark. Since the HPCG benchmark does not thoroughly test GPUs, we involve all the CPUs and GPUs for the benchmark.

Table 2: Power estimation in benchmarks

Scenario CPNT	HPL	HPCG
Processor	8 × 90W + 4 × 15W	
Chipset	~ 4 × 15W	
Memory	~ 4 × 27W	
Storage	≤ 6 × 5W + 8 × 5.2W	
GPU Card	12 × 115W + 4 × 15W	16 × 92W
Fans	6 × 40W	6 × 37W
HCA Card	4 × 8W	
PSU Overhead	~ extra 4% cost	
FDR-IB Switch	1 × 130W	
Sum	~ 2971W	~2986W

When running scientific applications, GPUs usually are not able to get full utilization. Besides, the power consumption of CPUs and GPUs are partly overlapped since the program will idle the CPUs when waiting for data to be returned from GPUs.

Table 3: Power estimation running applications

Scenario CPNT	CPU Heavy	GPU Heavy
Processor	12 × 150W	8 × 105W + 4 × 15W
Chipset	≤ 6 × 15W	~ 4 × 15W
Memory	≤ 6 × 27W	~ 4 × 27W
Storage	≤ 6 × 5W + 8 × 5.2W	
GPU Card	16 × 15W	16 × 85W
Fans	6 × 40W	
HCA Card	6 × 8W	
PSU Overhead	~ extra 4% cost	
FDR-IB Switch	1 × 130W	
Sum	≤ 2886W	~2993W

The estimated power of processors, memory, GPU cards, and ventilators is referred to as IPMI monitoring data in test environments. A part of the power estimation of GPU cards is from our experience in other supercomputing competitions. The values of chipsets, SSDs and the switch in the table are maximum power according to their official datasheet[5][4][3]. The power of chipset and memory in idle states is negligible thus eliminated in the calculation.
Power consumptions in all scenarios have been evaluated.

1.3 Choice of Acceleration Cards

After analyzing the traits of applications, we decide to use GPU to accelerate VIPC, while FPGA cards are proved less suitable for the computing scenario in this competition.

For maximum performance, we will deploy up to 16 NVIDIA V100 GPUs in our system. Since installing over 4 Nvidia GPUs requires extra NVLink and potentially unbalances the workload among the nodes, we assign four GPU cards to each GPU node. We have a total of 16 GPUs in our configuration.

1.4 Choice of CPUs

1.4.1 CPU Quantity. With similar overall performance, we prefer a smaller number of nodes as this arrangement can save more energy from chipsets and voltage regulators and reduce communication overheads amongst servers. The idle power of a node that equips four GPU cards is around 260 Watts. Though fewer nodes take better power efficiency, the 4-node configuration may “waste” the power limits as it is not practicable to install CPUs that cost more than 250Watts. Modern commercial CPU usually is power efficient.

Thus we deploy six nodes in our cluster in which case each CPU is expected to cost 130 to 150 watts.

1.4.2 CPU Model. Referring to Intel SKUs[1] and vendor's list, Xeon Gold 6240 was the most suitable CPU that we could choose. Within 150W TDP, it can run at 3.4GHz on all 18 cores (non-AVX)[8], thus providing satisfying performance.

According to our vendor, Inspur NF5280M5 supports dual Intel Xeon Scalable processors. Our previous experience teaches us that six NF5280M5 with 12 Xeon Gold 6132 (140W TDP) cost about 2800 Watts. Thus a CPU with higher TDP could be a good choice to achieve better performance within the power limits. According to Intel SKUs[1] and vendor's list, Xeon Gold 6240 fits our needs. Furthermore, Xeon Gold 6240 can offer a higher performance when it is running at 3.4GHz on all 18 cores (non-AVX)[8].

1.5 Choice of Storage

It is also important to optimize the storage for IO-500 benchmark and potential I/O bounded applications. We use the two nodes without GPU cards as storage nodes and change their configuration to contain external NVMe SSDs. The vacant PCIe lanes are connected to 4 Samsung 970 PRO SSDs in the rear rack on each storage node.

It is important to notice that Samsung 970 PRO features 2bit V-NAND technology, which reduces the impact of performance after running out of cache under long term workload. Considering that it claims the unparalleled sequential read/write speeds of up to 3,500/2,700 MB/s respectively and random performance of up to 500,000 IOPS for the both[6], with the help of parallel file system, our storage nodes can provide outstanding capabilities to serve given applications.

2 SOFTWARE SYSTEM

The goal of our software configuration is to provide the best convenience possible to boost our workflow. Good configuration can save us a ton of time from typing the verbose command line and doing the debugging process. Besides, we provide a dynamic power adjustment on some components to achieve the best performance within the power limits.

2.1 OS Selection

We are going to deploy CentOS 7.6 as our operating system for all our nodes. CentOS is a community-driven Linux distribution that tracks just ahead of Red Hat Enterprise Linux (RHEL), a disposiposition for the commercial market [2]. It is designed as an OS for servers, including HPC systems. It provides great consistency and management tools. Besides, CentOS also has abundant driver supports. Therefore it is suitable for our contest need.

The latest version of CentOS is 8.0, but it has only been released for several months. We have been testing on our training cluster for quite a long time. We will temporarily not bump to the higher version, since the software package support may not be sufficient and the system environment may differ from the current one.

2.2 Drivers & Tools

Our cluster hardware setup includes NVIDIA GPU and Mellanox InfiniBand. Both of them need extra drivers to be recognized and to function. We will install the latest drivers from both vendors. In addition, we will install the CUDA Toolkit for development. With this tool, we can also try to port CPU applications onto GPU.

Besides GPU and IB, we identify CPU and main memory as another two major sources of power consumption. For NVIDIA GPU, we can use `nvidia-smi` command line tool to adjust its power limit. For CPU, we will use `cpupower` command provided by `kernel-tools` package to adjust its power limit. Aside from those commands, we will also use `ipmitool` to retrieve sensor readings like fan speed, temperature and power consumption of each component. With these useful data, we can dynamically adjust system parameters to provide best compute capacity under the restriction of 3,000 Watt total power consumption.

2.3 File System Selection

To share data between multiple nodes, we need some kind of distributed/parallel file system. Since we have an IO-500 challenge in this contest, we also need to provide high performance in parallel IO scenarios. Our choice is either NFS or some parallel file system, or use both of them for different mount points.

For NFS, the main purpose is to share common software packages and configuration files. InfiniBand enables us to use NFS with RDMA (Remote Direct Memory Access) support. Therefore we will deploy NFS over RDMA. Directories including `/home`, `/opt` will be shared using NFS so that all the user data files and customized software packages can be synchronized instantly without manual intervention.

For parallel file systems, we have some choices such as OrangeFS [10] and Lustre [9]. We still need further investigation to decide on which file system to use. Parallel file systems are mainly used for IO-intensive tasks in our configuration. In this contest, we have not yet found any IO-intensive applications other than IO-500 challenge, so parallel file system selection may mainly focus on improving IO-500 performance. If we find any IO-intensive tasks, we will then consider porting it from native IO to MPI-I/O.

2.4 Software Management

To better manage the system environment, we will install all the custom programs used by the applications in the `/opt` directory as a convention. General softwares include compilers, mathematics libraries, and various utilities. The subdirectories will be named after software name, version, compiler version and compile flags so we allow multiple versions of the same software to coexist. In order not to interfere with other softwares in the shell environment, we choose to use the `modules` tool to manage the paths and flags. Environment variables like `PATH` for the needed libraries can be automatically imported after invoking `module load` on the given module name. This gives us a lot of flexibility in choosing dependency libraries.

Besides customized software, we also saved a copy of the open-source software mirrors in the local disk. With this local mirror, we can install missing packages instantly, even without Internet connection.

2.5 Job Management

Some applications typically run for a long period of time. Therefore, running jobs in the background becomes a must-have requirement. We choose Slurm [7] to fulfill our need. Slurm supports partitioning and scheduling based on resource requirements. It also provides convenient job management like starting or canceling jobs.

We will configure several resource partitions for Slurm and also prepare job submission scripts for each application in advance.

2.6 Account Management

For easier administration, we choose to create one Linux user for each problem so that the users' data file won't mess up and the shell environment can be isolated. Benchmarks will take place in the same user account. Besides, we also have an `admin` account for system management like installing software, monitoring system status and adjusting power limit. This user will be granted with `sudo` permission to perform these tasks.

REFERENCES

[1] [n. d.]. 2nd Gen Intel Xeon Scalable Processor SKUs. https://www.intel.com/content/dam/www/public/us/en/documents/product-briefs/2nd-gen-xeon-scalable-processors-brief.pdf

[2] [n. d.]. CentOS Project. https://www.centos.org Accessed: 2019-10-05.

[3] [n. d.]. ConnectX-3 Adapter User Manual. http://www.mellanox.com/related-docs/user_manuals/ConnectX-3_Pro_Ethernet_Single_and_Dual_QSFP+_Port_Adapter_Card_User_Manual.pdf

[4] [n. d.]. Intel C621 Chipset Specification. https://ark.intel.com/content/www/us/en/ark/products/97338/intel-c621-chipset.html

[5] [n. d.]. Intel SSD datasheet. https://www.intel.com/content/dam/www/public/us/en/documents/product-specifications/ssd-530-sata-specification.pdf

[6] [n. d.]. Samsung V-NAND SSD 970 PRO. https://www.samsung.com/semiconductor/global.semi.static/Samsung_NVMe_SSD_970_PRO_Data_Sheet_Rev.1.0.pdf.

[7] [n. d.]. Slurm Support and Development. https://www.schedmd.com Accessed: 2019-10-05.

[8] [n. d.]. Xeon Gold 6240 - Intel. https://en.wikichip.org/wiki/intel/xeon_gold/6240

[9] Daniel Pilaud, N Halbwachs, and JA Plaice. 1987. LUSTRE: A declarative language for programming synchronous systems. In *Proceedings of the 14th Annual ACM Symposium on Principles of Programming Languages (14th POPL 1987). ACM, New York, NY*, Vol. 178. 188.

[10] Robert B Ross, Rajeev Thakur, et al. 2000. PVFS: A parallel file system for Linux clusters. , 391–430 pages.