GPU Behavior on a Large HPC Cluster

Nathan DeBardeleben, Sean Blanchard, Laura Monroe, Phil Romero,
Daryl Grunau, Craig Idler, and Cornell Wright

Los Alamos National Laboratory,
High Performance Computing Division*, Los Alamos NM 87544, USA
{ndebard,seanb,lmonroe,prr,dwg,cwi,cornell}@lanl.gov

Abstract. We discuss observed characteristics of GPUs deployed as accelerators in an HPC cluster at Los Alamos National Laboratory. GPUs have a very good theoretical FLOPS rate, and are reasonably inexpensive and available, but they are relatively new to HPC, which demands both consistently high performance across nodes and consistently low error rate.

We modified a standard acceptance procedure to test GPU performance, error rate and reliability characteristics, and ran the test suite on a Fermi HPC cluster at LANL. We discuss here our methodology for this testing, and present results relevant to the deployment of GPUs in an HPC environment.

In this paper we show performance variability, power usage variability (possibly related), and some reliability concerns on the GPUs tested. We argue for rigorous testing of these devices in deployment as a way of characterizing their behavior.

Keywords: graphics processing units, high performance computing, reliability, acceptance testing, fault-tolerance, resilience, error correction.

1 Introduction

Graphics Processing Units (GPUs) are a promising technology in the field of high-performance computing (HPC). GPUs are highly parallel co-processors, providing FLOPS at low power consumption. They were developed as co-processors for the graphics and gaming industry and were co-designed to fit the needs of these real-time applications, using the fast SIMD parallel pipeline that addressed graphics needs. This SIMD parallel architecture is also suited to many scientific problems, so research and development on the use of GPUs for scientific simulation and HPC intensified in the 2000s.

GPUs are a fairly new technology in HPC, so many interactions of GPUs with HPC systems are not yet fully characterized. These include such questions as the

* A portion of this work was performed at the Ultrascale Systems Research Center (USRC) at Los Alamos National Laboratory, supported by the U.S. Department of Energy contract DE-FC02-06ER25750. The publication has been assigned the LANL identifier LA-UR-13-24171.

D. an Mey et al. (Eds.): Euro-Par 2013 Workshops, LNCS 8374, pp. 680–689, 2014.

reliability of the GPU, the patterns of errors that do occur and the performance of GPUs as an ensemble within an HPC system. These questions are especially important in a highly coupled high-performance computing system, in which performance or reliability problems on a single GPU might affect the entire calculation.

In this paper, we investigate these questions on two large-scale Fermi systems sited at Los Alamos National Laboratory (LANL), and describe our observations testing real systems in the field. We also investigate the error-correction capabilities of the GPU. This is especially interesting in that main-memory DRAM uses chipkill-correct ECC, whereas GPUs use only Single-Error Correct Double-Error Detect (SECDED). SECDED provides a lesser amount of protection against memory errors, so it is of interest to understand how the reliability of GPU SDRAM compares to that of main memory DRAM.

2 High Performance Computing

HPC combines multiple computers together via a high-performance interconnect to work in parallel on problems that are too large to be solved by individual computers. Such HPC clusters are built from thousands to tens of thousands of individual nodes that communicate with one another via a high-speed interconnect. Current top HPC clusters have as many as 1.5 million processors and up to 1.5 Petabytes of memory. These numbers have increased significantly in recent years; further significant increase is expected in the future as HPC systems evolve to handle ever larger problems.

2.1 Node Performance and Reliability Effects on HPC Calculations

The applications running on many HPC systems, such as those at LANL, are tightly coupled simulations of physical phenomena. As the simulation evolves, nodes communicate with each other to propagate information representing the physical phenomena that affect adjacent regions such as temperature, pressure, electrical charge, particle migration and so forth. The simulation progresses in small increments of simulated time, known as time-steps. Information from adjacent regions is required for each time-step, so high-speed, low latency communication between nodes is required for efficient use of the HPC system's processors.

Importantly, delays in the completion of a time-step on one processor will often mean that the other nodes which need adjacent region information to proceed will also be delayed. This can cause a cluster to run overall at the speed of its slowest node.

The sheer quantity of hardware in an HPC system magnifies the effects of the failure rate of its individual components. A cluster with 10,000 nodes, with a mean time between failure (MTBF) for each node of 10 years, would have an MTBF of under 9 hours, when considered as a cluster. Other non-node-based failure modes would drive the real-world MTBF even lower.

3 GPUs in HPC

With the co-development of general programming languages and GPU hardware in the late 2000s, the GPU pipeline became much more accessible for general scientific programming, and the GPU became attractive as an accelerator in HPC architectures. The first GPU-based heterogeneous cluster to make the Top 500 list [12] was Tokyo Tech's TSUBAME in November 2008. The November 2012 list contains 53 GPU-based computers, including the number 1 entry, Titan.

The GPU is a massively parallel SIMD architecture; each GPU has many streaming multi-processors (SMs), each having many cores. The GPU has a hierarchical memory structure. There is a small amount of SRAM shared memory per SM that can be accessed across all cores in the SM. Part of this shared memory can be used as cache, as on CPUs. There is a larger amount of GDDR5 global memory available to all SMs on the GPU. Finally, main memory on the node can be accessed through the limited bandwidth of a PCIe bus between the GPU and the motherboard.

3.1 GPU Memory and Error Protection

The Fermi generation GPU tested here is the first to support Error Correcting Code (ECC) memory protection [8]. Each M2090 has 6GB of GDDR5 memory. GDDR5 memory is based on DDR3 technology but uses synchronous DRAM (SDRAM).

While the Fermi's Single-Error Correct Double-Error Detect (SECDED) error protection is certainly a major improvement over uncorrected errors for HPC loads, this error protection is not as effective as the standard on non-GPU HPC systems. Most of these systems depend on main-memory DRAM equipped with chipkill-correct ECC [4]. A recent study of the Jaguar supercomputer at Oak Ridge National Laboratory (ORNL) showed that chipkill-correct ECC reduces the detectable uncorrectable error rate of memory by 42X compared to SECDED [11]. Because of this, we believe that a greater understanding of the reliability of GPU memory is needed for their effective use in the HPC environment.

3.2 GPU Strengths and Weaknesses

GPUs are very attractive for HPC for several reasons. They provide high FLOPS, as well as a good FLOPS/watt ratio. GPUs are readily available and reasonably inexpensive, in terms of the compute power they provide. They are expected to be available and supported for some time to come. This future support is important, since code developed for these clusters may be in use for decades, and the clusters themselves may be in use for five or more years.

However, GPUs have some drawbacks. The memory hierarchy can make it difficult to get optimized results, and sometimes presents a bottleneck that cannot be avoided. GPUs do require an in-depth understanding of the hardware to get the best results. The uncertainty around reliability should be investigated further in view of the difference between the error-correction used on the GPU

and that used on main memory. Finally, the technology is under rapid development and so needs testing with each hardware and software change, as different releases can lead to quite different performance and reliability results.

4 Testing Methodology

All LANL HPC systems undergo a lengthy test and verification process before hand-over to the target user community. The goal is to provide a reliable, high performing, well prescribed system to the users in as timely a manner as possible after the system has been constructed at the vendor's site.

At LANL, testing is conducted under the Gazebo[6] framework which submits jobs to keep the system busy to a desired level. Gazebo can identify system utilization and node coverage, provide test pass/fail summaries, and basic job results analytics. For the new GPU-based systems, we extended our testing framework to examine GPU performance and reliability, in addition to the standard CPU tests performed on every new system.

Tests used for this paper included Oak Ridge National Laboratory's Scalable HeterOgeneous Computing (SHOC) Benchmark Suite [2], NVIDIA's High-Performance Linpack (HPL) [5], NBody [10], and GPU Bandwidth tests, and Laboratoire d'Informatique de Robotique et de Microelectronique de Montpellier's GPUburn [3]. All low-level hardware (level 0) SHOC tests were run to identify low-level/hardware performance variations across nodes. Most of the results presented in Section 5 are from single-node experiments performed on these systems. This was because these experiments were scavenging free nodes of the system in production while regular users were utilizing the rest of the cluster.

We tested a large-scale cluster, Moonlight, whose nodes were equipped with PCI-attached NVIDIA Fermi GPU accelerators, which were top-of-the-line in their generation of GPUs. Moonlight is a 308-node, dual-socket, 8-core Intel Sandy Bridge cluster that has 32GB main memory and two NVIDIA Tesla M2090 GPGPUs [9] per node. It is interconnected with a QLogic Infiniband fabric operating at 40 Gbit/sec (4x QDR). While the cluster is equipped with PCIe-3.0, the M2090 GPU is PCI2-e.

While Kepler, the next generation of NVIDIA's GPU, is out, LANL does not yet have access to large-scale Kepler clusters for testing. Furthermore, we wanted to test hardware in production use by scientists, and there is a latency between the time systems are procured and the time they are brought into production.

5 Results

For our experiments, we tested using all of the software packages mentioned in Section 4. We focus in this section on issues that surfaced during the SHOC and HPL runs. In particular, we saw unexpected variation in GPU-to-CPU transfer rate on the SHOC tests, which might lead to lowered performance across the cluster on our highly coupled simulations. We noticed an unusual distribution

of the GPU-only HPL performance results, which could indicate unusual performance distributions on our simulations. We also saw low power draws on one of the GPUs that led to low performance and which relate somewhat to the odd distribution of the GPU-only HPL performance results. Finally, we saw an inordinate number of high residuals on HPL, which might indicate susceptibility to silent data corruption (SDC) during actual production use of the machine. In the following sections we detail these experiments.

The performance tests in Section 5.1 and 5.2 were run on a system in production usage at LANL running CUDA 4.1. Later, CUDA was upgraded to 5.0, and some of these problems improved. The reliability tests in Section 5.3 were run on the same system after the upgrade to CUDA 5.0.

5.1 GPU-to-CPU Data Transfer Variation

We used the SHOC BandwidthTest to conduct tests on data transfer rates from the CPU to the GPU and vice versa. We performed 200 tests at transfer sizes from 1KB up to 512MB. These tests were done for transfers from CPU to GPU and from GPU to CPU.

There were no significant variations in CPU-to-GPU transfers found. However, for GPU-to-CPU transfers, we saw major slowdowns in the transfer rates for some data of size greater that 4 MB. Figure 1 shows how the transfer rate of data from the GPU to CPU varies as the size of the transfer increases. Notice the transfer rate generally increases as the transfer size increases.

However, for some sets of the largest data, we saw data transfer rates around 13x slower than more typical transfers of the same data size. These fluctuations can be seen in the bottom right of Figure 1. This kind of fluctuation could have very negative effects on tightly coupled scientific applications that perform at the speed of the slowest component.

This strange behavior disappeared once we upgraded from CUDA 4.1 to CUDA 5.0. Driver problems are not uncommon on cutting-edge systems such as these, however, this demonstrates the need for rigorous testing on future versions of CUDA to ensure that this kind of anomalous behavior doesn't occur in different releases. It is encouraging that our cycle-scavenging and acceptance testing framework was able to find this problem so that it could be addressed.

5.2 HPL Performance Variation across Nodes

High Performance Linpack was used to characterize performance with and without the GPUs. An HPL test that used only the CPUs on a single node was run hundreds of times across all nodes (for coverage) and was found to produce a near normal performance distribution as can be seen in Figure 2. The CPU-only HPL runs show little change in performance from run to run and is consistent with our previous experiences running this benchmark.

This was not true when the HPL test utilized both the GPUs and CPUs on a single node, as shown in Figure 3. The combined GPU and CPU test shows very wide variation from run to run with distinct peaks at a variety of performance

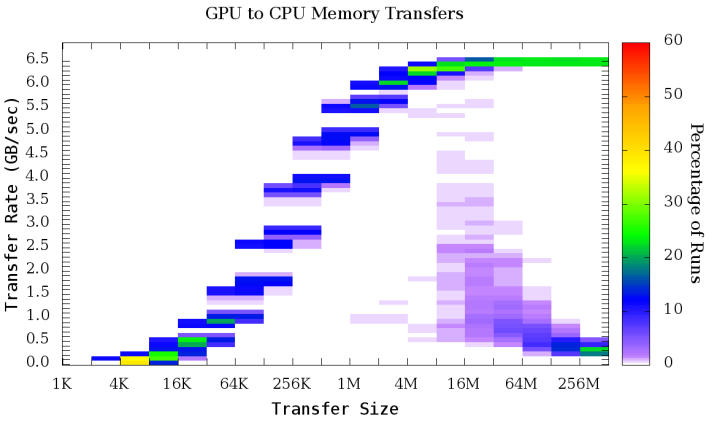

Fig. 1. The distribution of data transfer rates from GPU to CPU colored by percentage of runs of a given size at a given rate. The fluctuations in transfer rate can be seen in the data at the lower right of the graph. We saw a significant fraction of these larger datasets being transferred at rates less than 1 GB/sec, much slower than the usual rate of 6+ GB/sec shown in the upper right of the graph. This effect disappeared after upgrading to Cuda 5.0, underscoring the requirement for rigorous testing.

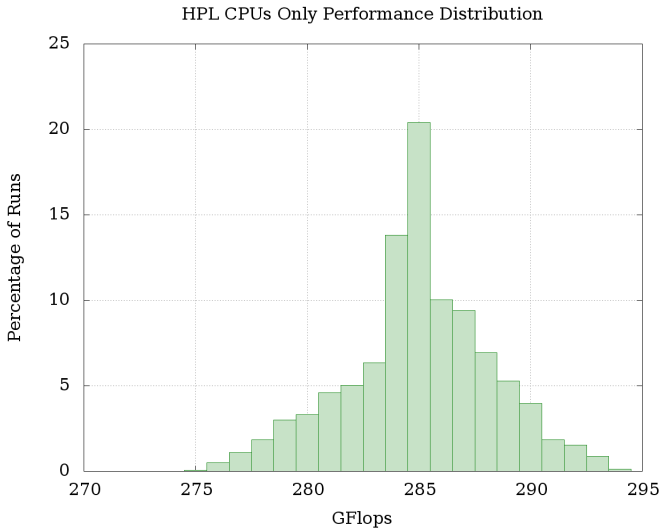

Fig. 2. CPU-only HPL performance shows a reasonable fit to a normal distribution with a standard deviation of only 3.2 GFlops

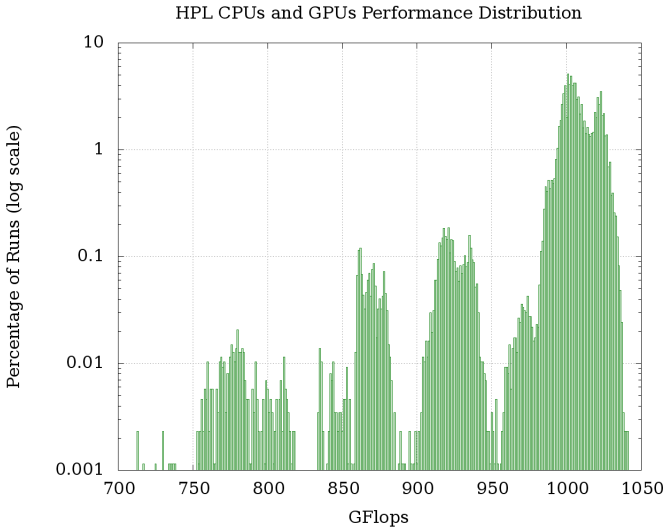

Fig. 3. GPU/CPU HPL performance has a distribution that is definitely not normal and has a standard deviation of 28.2 GFlops. In particular, this distribution shows two distinct peaks, and a long tail to the left with two small peaks visible representing low-performing tests. Please note the log scale.

levels. Inconsistencies in performance of this nature are of particular concern to the HPC community as tightly coupled applications tend to run at the speed of the slowest processing unit.

In an effort to explain the variations observed in the combined GPU-CPU HPL test, we looked at the power used by the GPUs under load. HPL was run more than 134,000 times and the maximum power usage was recorded for each run. The results of these runs appear in Figure 4 where each data point represents the HPL performance and the associated power usage of each GPU on that node.

One would expect the GPUs working in tandem would have similar power usage and this would be represented as a circle in the plot. However, GPU0 was observed to draw a wide range of power over the runs. GPU1 shows much less variability. Lower HPL performance correlates with those runs in which GPU0 draws low power. This behavior possibly indicates a hardware problem on the Moonlight nodes, which is not at this time resolved.

In addition to the obvious concerns about variable performance and variable power usage shown in Figure 4, there is a question about the total power usage of each GPU. The PCI specification is for 225 Watts, and it is evident the GPU counters are reporting a power draw above this value. We do not have a feeling for sensor accuracy and it is possible that these counters may be inaccurate. However, it is well understood that hardware running at the extremes (both upper and lower) of specification can incur reliability and dependability

problems. There is also an interesting anomaly in GPU1 power draw (which becomes bi-modal) when GPU0 draws around 240W. The cause of this is still undetermined at this time.

These tests consumed a great many system and analyst cycles and, after upgrading to CUDA 5.0, we were unable to revisit them as the NVIDIA monitoring library (nvml) is broken. The tool does not return information about the power usage of the second GPU and, consequently, it fails to provide the information necessary to run the power experiment.

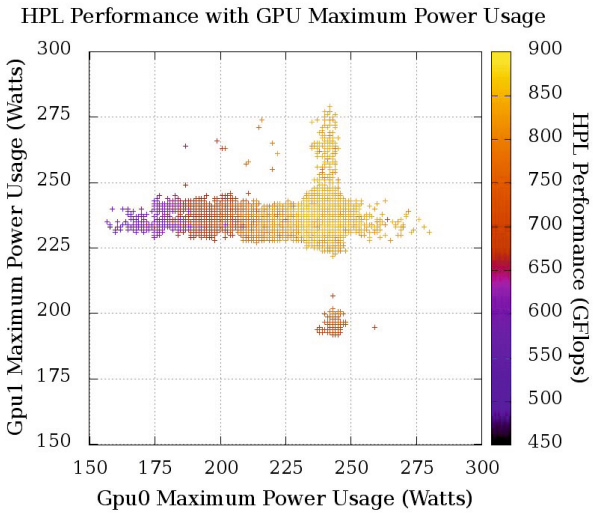

Fig. 4. GPU-only performance colored by HPL performance. The points in this graph represent (GPU0, GPU1) power draws, and do not represent the number of nodes showing each power draw pair. The great majority of the nodes show both GPUs falling into the 225-250 watt range. As might be expected, low power draw on either GPU is associated with low performance. GPU0 shows greater variability in power usage than GPU1, and all of the lowest-performing runs are associated with low power usage on GPU0. GPU0 power use variability is an unresolved question on this machine.

5.3 Residuals and Double Bit Errors

To look for potential reliability problems we examined a single month of HPL output after upgrading to CUDA 5.0. HPL computes $Ax - b = 0$ and calculates *residual errors*. Residuals are places where the calculation is not zero due to rounding. Small rounding errors are common across platforms. However, the residuals should be small when compared to the numerical accuracy of any particular platform. As seen in Table 1, the results from a single month of HPL testing contained many excessively large residuals.

Table 1. Results of HPL runs including excessive residuals found and double bit errors that occurred during HPL runs

Total Runs	Total Hours	Bad Residuals	DBEs
40,171	16,738	1624	27

Approximately 4% of our HPL runs resulted in errors. Of the 1,624 excessive residuals, 27 of them corresponded with logged reports of at least one double-bit error (DBE) occurring on one of the GPUs being used. These measured events lead to fault rates of 0.097 per hour per node for excessive residuals and 0.00161 per hour per node for double bit errors. On a machine the size of the Moonlight cluster, 308 nodes, this would result in 30 bad residuals per hour and 0.5 double-bit errors per hour when extrapolated to the entire machine. These rates seem inordinately high and are of great concern.

This suggests an application checkpoint frequency of less than an hour as a way to mitigate double-bit errors alone. Checkpoint frequencies on current HPC systems are on the order of 16-24 hours and exascale predictions are for frequencies anywhere from tens of minutes to several hours [7,1].

There are several worrisome points here. We found approximately 1,600 instances of excessive residuals that did not correspond with logged DBEs. It is not clear whether there are problems with the DBE reporting of these GPUs or whether we are seeing a large number of multi-bit errors (MBEs). This makes us wonder if SECDED ECC is sufficient or if a higher level of memory error protection is needed on the GPU. Another possibility is that there is something else going on that we do not yet understand that is unrelated to memory.

6 Conclusions

We have presented a study of a large number of tests performed on a GPU cluster are Los Alamos National Laboratory. For some of the tests we found no problems, but for certain applications we found some problematic results. It is not uncommon in HPC systems to have problems exposed only by tests that exercise the system in a particular manner.

The types of problems we discovered required a large commitment of resources and were found on the cluster when in production. The system had already passed standup and acceptance testing and was in use by scientific teams. We posit that these problems were obscure and unlikely to have been found on small systems and stand-alone machines but clearly have performance and reliability impacts. While some of these problems went away after upgrading driver versions, it is important to recognize that software upgrades are often held off until fully evaluated for performance, correctness, and security testing. As such, a driver upgrade may not be a simple undertaking on a production system.

The addition of memory error protection on the GPUs is a welcome and clearly required addition in the HPC arena. However, there appear to be problems

associated with high error rates and/or error reporting. Additionally we found performance variability that seems to be associated with power usage variability on systems at LANL. Whether this is also the cause of the reliability problems has not been determined.

GPUs are a relatively new addition to the HPC market and were not created initially with scientific computing in mind. For these reasons, we think it prudent to scrutinize all aspects of the technology. We have shown the importance of rigorous (and ongoing) testing and examination of logs, application results, and internal counters on the GPUs, and have shown the importance of maintaining current versions of device driver hardware where possible. In the future, we hope to explore whether these effects persist into later generations of NVIDIA GPUs.

References

1. Cappello, F., Geist, A., Gropp, B., Kale, L., Kramer, B., Snir, M.: Toward exascale resilience. International Journal of High Performance Computing Applications 23, 374–388 (2009)
2. Danalis, A., et al.: The scalable heterogeneous computing (shoc) benchmark suite. In: Proceedings of the 3rd Workshop on General-Purpose Computation on Graphics Processing Units, GPGPU 2010, pp. 63–74. ACM, New York (2010), http://doi.acm.org/10.1145/1735688.1735702
3. Defour, D., Petit, E.: Gpuburn: A system to test and mitigate gpu hardware failures. In: Proceedings of the International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation (May 2013)
4. Dell, T.J.: A white paper on the benefits of chipkill-correct ecc for pc server main memory (1997)
5. Fatica, M.: Accelerating linpack with cuda on heterogenous clusters. In: Proceedings of 2nd Workshop on General Purpose Processing on Graphics Processing Units, GPGPU-2, pp. 46–51. ACM, New York (2009), http://doi.acm.org/10.1145/1513895.1513901
6. Idler, C.: Gazeo (2013), http://www.github.com/losalamos/Gazebo
7. Kogge, P., et al.: Exascale computing study: Technology challenges in achieving exascale systems (2008)
8. NVIDIA: NVIDIA's Next Generation CUDA Compute Architecture: Fermi (2009), http://tinyurl.com/ykawxw9
9. NVIDIA: Tesla m2090 dual-slot computing processor module (June 2011), http://tinyurl.com/3wbxd46
10. Nyland, L., Harris, M.: 31 Fast n-body simulation with cuda, ch. 31
11. Sridharan, V., Liberty, D.: A study of dram failures in the field. In: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, SC 2012, pp. 76:1–76:11. IEEE Computer Society Press, Los Alamitos (2012), http://dl.acm.org/citation.cfm?id=2388996.2389100
12. top500: Top500 supercomputer sites (November 2012), http://www.top500.org/lists/2012/11/