Chapter 2
Platforms

Abstract The ability to quantify power and energy use is critical to understanding how power and energy are currently being used. A measurement capability is also necessary to measure the effect of tuning or modification of platform parameters, CPU frequency, and network bandwidth for example. In later chapters, these effects will be evaluated based on energy savings versus performance impact. Simply stated, power and energy measurement first require hardware support. This chapter will outline the hardware architectures and some of the significant systems software of the platforms used in the experiments detailed in this book.

2.1 Hardware Architecture

The experiments covered in this book were all accomplished on some variant of the Cray XT architecture. To our knowledge, this is the only platform that exposes the ability to measure current draw and voltage, in situ, as described in Chap. 3. Both the idle cycle and CPU frequency scaling experiments required specific operating system modifications to the Catamount [1] light-weight kernel (LWK). Catamount, authored by a team at Sandia National Laboratories, was the first production operating system available on the Cray XT line of supercomputers. Catamount support is currently limited to the Cray XT3 and XT4 variants of the architecture. The Cray XT architecture also affords the rare ability to tune performance parameters of other components. This capability is exploited to tune network bandwidth while measuring the effect on application energy in Chap. 6. The following sections will describe the specific test platforms and configurations in more detail. It is important to note that obtaining the dedicated time on production High Performance Computing (HPC) platforms to conduct experiments of this type is extremely difficult and very expensive.

J. H. Laros III et al., *Energy-Efficient High Performance Computing*,
SpringerBriefs in Computer Science, DOI: 10.1007/978-1-4471-4492-2_2,

2.1.1 Red Storm

Red Storm, the first of the Cray XT architecture line, was developed jointly by Cray Inc., and Sandia National Laboratories. The Cray XT architecture has been installed at numerous government and commercial sites including Oak Ridge National Laboratory. Red Storm is currently a heterogeneous architecture containing both dual and quad core processors. Both variants are used in the experiments discussed in this book. All nodes, dual and quad, are connected via a Seastar 2.1 network interface controller/router (Seastar NIC) in a modified mesh (mesh in X and Y directions and a torus in the Z direction).

Dual Core Nodes

The network bandwidth experiments described in Chap. 6 were accomplished on the dual core (XT3) partition of Red Storm. The XT3 partition contains 3,360 AMD 64 bit 2.4 GHz dual-core processors with 4 GB of DDR memory (2 GB per compute core). Each XT3 node is connected to the network via a Seastar NIC. The ability to manipulate the network bandwidth of the platform is equivalent on both the XT3 and XT4 partitions. The primary driver of using the XT3 partition for the network bandwidth experiments was simply the availability of this partition. The idle experiments were conducted on both the dual and quad core partitions of Red Storm.

Quad Core Nodes

Red Storm's XT4 partition utilizes AMD 64 bit 2.2 GHz quad-core processors with 8 GB of DDR2 memory (2 GB per compute core). Red Storm has 6,240 quad-core compute nodes, each connected to the network via a Seastar NIC. The frequency scaling experiments described in this book were conducted solely on the quad-core processors of either Jaguar or Red Storm due to the Advanced Power Management (APM) architectural requirements. The method of exploiting APM features will be discussed in Chap. 6. Some of the applications used in this research are export controlled and could not be executed on Jaguar (an open platform). Since the architectures and software stacks used were identical, we simply maximized our use of each platform based on application requirements and test platform availability.

2.1.2 Jaguar

Use of Jaguar was granted through the Department of Energy's Innovative and Novel Computational Impact on Theory and Experiment (INCITE) program. Jaguar, located at Oak Ridge Leadership Computing Facility (OLCF), was used for a portion of the frequency scaling experiments outlined in Chap. 6. The XT4 partition of Jaguar was specifically employed since it was both easier to gain dedicated access to and the architecture supported Catamount with much less up front effort than the XT5 partition would have required. Dedicated access was necessary for a number of reasons, primarily driven by our requirement to run Catamount (no longer a Cray supported operating system for the XT5 architecture and beyond). The XT4 partition of Jaguar contains 7,832 64 bit quad-core AMD Opteron processors (or nodes). Each core executes at 2.2 GHz and has access to 8 GB of DDR2 memory (2 GB per compute core). Each node on Jaguar is connected to the network via a Seastar NIC. The network topology of the Jaguar XT4 partition is a 3D torus. Jaguar's network topology, differs somewhat from Red Storm's modified mesh. These differences are not significant to the experiments conducted and had no affect on the results.

2.2 Operating System

Serial number one of the Cray XT architecture employed a light-weight kernel operating system named Catamount. For approximately four years, Catamount was delivered by Cray Inc. as the production operating system for the XT3 platform line. Catamount, at the time, was the latest in the lineage of light-weight operating systems authored, or co-authored, by Sandia National Laboratories.[1] Catamount was designed to get out of the way of the application. Important hardware abstractions and memory management are all provided with performance being the primary design consideration. When a parallel application is launched, Catamount provides the initial setup for the application, including contiguous memory allocation, and then suspends itself other than handling interrupt driven tasks such as those from network devices (Seastar NIC). This is a simplistic description but sufficient for the purposes of this book. Catamount is a small deterministic operating system in contrast with general purpose operating systems such as Linux. While it has proven to be a successful production operating system, Catamount has also proved invaluable for operating system and systems software research at Sandia National Laboratories.

The first experiment conducted was directed at saving energy during idle cycles since early versions of Catamount ignored this as a design consideration. The design of Catamount preceded many of the APM capabilities found on recent processor architectures. Once successful, later experiments explored further power efficiencies by leveraging more advanced APM features such as frequency scaling. The simplicity and deterministic nature of Catamount greatly aided in conducting these experiments. More detail on specific modifications will be included in our coverage as necessary.

---

[1] Sandia's most recent effort is the Kitten light-weight kernel [2].

2.3 Reliability Availability and Serviceability System

Historically, Reliability Availability and Serviceability (RAS) systems were commonly provided by vendors on mainframe class systems. Today, RAS systems look very different and are generally unique to high end custom HPC class architectures (Cray XT/XE and IBM BlueGene L/P and Q, for example). It is hard to define a clear line where cluster management systems become RAS systems. Generally, cluster management systems consist of a loose collection of open source utilities that are each individually designed for a narrow purpose. They are seldom well integrated and can often be intrusive to the primary purpose of the platform, computation. At small scale, the level of interference is often acceptable. RAS systems, in general, are typically more intentionally designed and integrated, often specific to a single architecture (in our opinion one of the failures of RAS system designs historically [3]).

The following are excerpts from the requirements for Cielo [4], a recent capability class procurement by the Alliance for Computing at Extreme Scale (ACES), a collaboration between Sandia National Laboratories and Los Alamos National Laboratory.

1. *To achieve delivery of the maximum continuous system resource availability, the RAS system must be a well engineered, implemented, and integrated part of the proposed platform.*
2. *There shall be a separate and fully independent and coherent RAS system.*
3. *The RAS system shall be a systematic union of software and hardware for the purpose of managing and monitoring all hardware and software components of the system to their individual potential.*
4. *Failure of the RAS system (software or hardware) shall not cause a system or job interrupt or necessitate system reboot.*

While this is only a small portion of the requirements that described and specified the RAS system for Cielo, it suggests some differentiating characteristics between a generic cluster management system and what is considered a RAS system. For these experiments, one of the most important characteristics is the separation but close integration of the RAS system in relationship to the underlying platform. This allows for the out-of-band scalable collection of current and voltage data necessary for all experiments included in this book. Out-of-band, in this context, means that control and monitoring of the platform, in general, is accomplished without affecting the platform or the software running on the platform. Measuring current and voltage data, for example, does not require an operating system interrupt using our methodology. It is very important that experimental methods do not, or minimally, affect the normal activity of the system. In a related work [5], laptops were employed to measure power using the operating system ACPI interface. This method causes an operating system interrupt each time a measurement is requested. While the interruption is minimal, at large scale this type of measurement could introduce the equivalent of operating system noise [6, 7]. There is no such interruption during the measurements used in in our experiments. The separate RAS network additionally allows the collection of these measurements in a scalable manner.

References

1. S.M. Kelly, R.B. Brightwell, Software Architecture of the Light Weight Kernel, Catamount, in *Cray User Group*, CUG, 2005
2. Kitten Light Weight Kernel, Sandia National Laboratories. Available https://software.sandia.gov/trac/kitten
3. J.H. Laros III, A Software and Hardware Architecture for a Modular, Portable, Extensible Reliability Availability and Serviceability System, in *IEEE Proceedings of the Workshop on High Performance Computing Reliability Issues*, 2006
4. Cielo, Sandia National Laboratories and Los Alamos Laboratory. Available http://www.lanl.gov/orgs/hpc/cielo/
5. R. Ge, X. Feng, K.W. Cameron, Performance-Constrained Distributed DVS Scheduling for Scientific Applications on Power-aware Clusters, in *Proceedings of the International Conference on High Performance Computing, Networking, Storage, and Analysis (SC)*, ACM/IEEE, 2005
6. K.B. Ferreira, R. Brightwell, P.G. Bridges, Characterizing Application Sensitivity to OS Interference Using Kernel-Level Noise Injection, in *Proceedings of the International Conference on High Performance Computing, Networking, Storage, and Analysis (SC)*, ACM/IEEE, 2008
7. F. Petrini, D. Kerbyson, S. Pakin, The Case of the Missing Supercomputer Performance: Achieving Optimal Performance on the 8,192 Processors of ASCI Q, in *Proceedings of the International Conference on High Performance Computing, Networking, Storage, and Analysis (SC)*, ACM/IEEE, 2003

http://www.springer.com/978-1-4471-4491-5

Energy-Efficient High Performance Computing
Measurement and Tuning
Laros III, J.H.; Pedretti, K.; Kelly, S.M.; Shu, W.; Ferreira, K.; Van Dyke, J.; Vaughan, C.
2013, XIV, 67 p. 19 illus., 8 illus. in color., Softcover
ISBN: 978-1-4471-4491-5