Managing Software Tools in High-Performance Computing Environments

David R. Montoya, Georgia A. Pedicini, David O. Gunter, HPC-4

Past high-performance computing (HPC) environments were based on a singular architecture with extensive support from the vendor. Examples at Los Alamos National Laboratory (LANL) include the Crays (1970s–1980s), Blue Mountain (1990s), and the Q clusters (2000). During the earlier days of supercomputers there had to be a close collaboration between the vendor and the implementation site personnel. Computers such as these could not be built with only local skills, and much of the software was custom-developed for the machine. Overall this was an expensive proposition. In those early days the Advanced Simulation and Computing (ASC) program provided an environment to build products that could support those new-generation machines. The ASC PathForward program, begun in 1998, helped tie universities, commercial vendors, and national laboratories into partnerships to develop the supporting software infrastructure in support those new machines. Capabilities and skill bases were created, which in turn fed new products into the HPC economy.

More recently, the open-source movement developed a process where software could be free and user maintained, allowing intellectual capital to grow based on the interest that could be generated in a given area or product. For user software, this movement fostered an explosion of products, driving both vendor-supported and open-source software products. Another development area is Linux, a low-cost operating system (OS). Ron Minnich (formerly of CCS Division) further developed this OS as BProc/Clustermatic, which emphasizes a lightweight kernel on compute nodes of a complex Linux cluster. This was the advent of what we call capacity HPC.

This phase of computing has moved into production capacity within the past 3–4 years, creating both opportunities and challenges. Moore's Law, describing growth in hardware capability, continues to provide an indicator for new generations of hardware. Computational clusters are just reaching production quality when new hardware is already being procured for the next capacity cluster. Newer machines have different capabilities, may run newer versions of the Linux kernel, have different network fabric implementations, or have different parallel file systems. The result is a loss of consistency in the software and user environments. The challenge is to manage consistency to the best level possible. Tools to manage and plan for a diverse environment with different software requirements and configurations are essential.

Environment

The details of each cluster that makes up the current heterogeneous HPC environment must be understood to grasp what impact its specific hardware has on the software environment. Table 1 describes the current inventory of HPC clusters at LANL. Footnotes to the table indicate differences among machines that are nominally of the same class.

From a software health perspective, attributes of interest are as follows:

- Number and type of segments describe the individual units that make up a portion of the larger machine. On some clusters, different segments provide different functionality and must be viewed as separate machines. Front-end and compile nodes, not reflected here, add another dimension of complexity. Taking these all into account, we monitor more than 50 machines.
- Processor and OS provide a base for the core software architecture of a machine. Subcharacteristics that have a large impact are kernel version, glibc version (the library that defines system calls), and interconnect. The variant of the OS

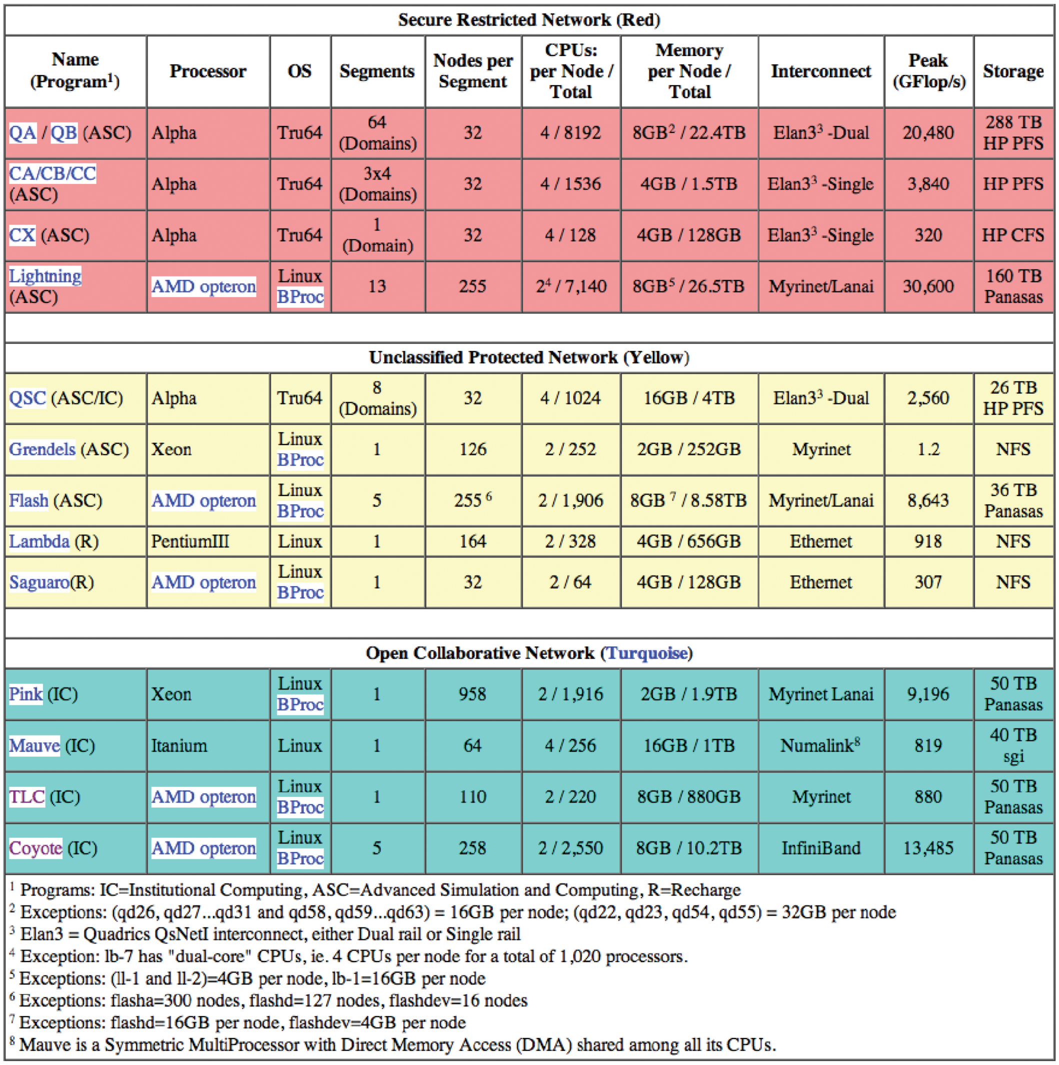
has a huge impact when using software products that are tightly coupled to the hardware. Many of our systems have a BProc implementation. This is a concern with software when it needs to tap into specific libraries with a known implementation (e.g., BProc/Myrinet and BProc/InfiniBand). Any tools that deal with process management and migration need to be rewritten to accommodate the BProc paradigm. Specifically, we had to modify MPI (message-passing interface) implementations and debuggers to run under BProc.

- Interconnect and storage systems come into play when we look at processor communication and I/O. Message Passing Interface (MPI) implementations, and software interacting with file systems, may need to be retuned for each implementation.

There are trade-offs when it comes to supporting software on current and emerging HPC platforms. As long as the growth curve is steep for architectures that provide increasing computational power, we need to invest in the infrastructure that must follow it to make it useable. This investment will have to grow as we move to the new accelerator and hybrid architectures for which there currently are no software tools.

For more information contact David R. Montoya at dmont@lanl.gov.

Funding Acknowledgements
This research was supported by the NNSA tri-Lab Advanced Simulation and Computing Program.



Name (Program[1])	Processor	OS	Segments	Nodes per Segment	CPUs: per Node / Total	Memory per Node / Total	Interconnect	Peak (GFlop/s)	Storage
Secure Restricted Network (Red)									
QA / QB (ASC)	Alpha	Tru64	64 (Domains)	32	4 / 8192	8GB[2] / 22.4TB	Elan3[3] -Dual	20,480	288 TB HP PFS
CA/CB/CC (ASC)	Alpha	Tru64	3x4 (Domains)	32	4 / 1536	4GB / 1.5TB	Elan3[3] -Single	3,840	HP PFS
CX (ASC)	Alpha	Tru64	1 (Domain)	32	4 / 128	4GB / 128GB	Elan3[3] -Single	320	HP CFS
Lightning (ASC)	AMD opteron	Linux BProc	13	255	2[4] / 7,140	8GB[5] / 26.5TB	Myrinet/Lanai	30,600	160 TB Panasas
Unclassified Protected Network (Yellow)									
QSC (ASC/IC)	Alpha	Tru64	8 (Domains)	32	4 / 1024	16GB / 4TB	Elan3[3] -Dual	2,560	26 TB HP PFS
Grendels (ASC)	Xeon	Linux BProc	1	126	2 / 252	2GB / 252GB	Myrinet	1.2	NFS
Flash (ASC)	AMD opteron	Linux BProc	5	255[6]	2 / 1,906	8GB[7] / 8.58TB	Myrinet/Lanai	8,643	36 TB Panasas
Lambda (R)	PentiumIII	Linux	1	164	2 / 328	4GB / 656GB	Ethernet	918	NFS
Saguaro(R)	AMD opteron	Linux BProc	1	32	2 / 64	4GB / 128GB	Ethernet	307	NFS
Open Collaborative Network (Turquoise)									
Pink (IC)	Xeon	Linux BProc	1	958	2 / 1,916	2GB / 1.9TB	Myrinet Lanai	9,196	50 TB Panasas
Mauve (IC)	Itanium	Linux	1	64	4 / 256	16GB / 1TB	Numalink[8]	819	40 TB sgi
TLC (IC)	AMD opteron	Linux BProc	1	110	2 / 220	8GB / 880GB	Myrinet	880	50 TB Panasas
Coyote (IC)	AMD opteron	Linux BProc	5	258	2 / 2,550	8GB / 10.2TB	InfiniBand	13,485	50 TB Panasas

[1] Programs: IC=Institutional Computing, ASC=Advanced Simulation and Computing, R=Recharge
[2] Exceptions: (qd26, qd27...qd31 and qd58, qd59...qd63) = 16GB per node; (qd22, qd23, qd54, qd55) = 32GB per node
[3] Elan3 = Quadrics QsNetI interconnect, either Dual rail or Single rail
[4] Exception: lb-7 has "dual-core" CPUs, ie. 4 CPUs per node for a total of 1,020 processors.
[5] Exceptions: (ll-1 and ll-2)=4GB per node, lb-1=16GB per node
[6] Exceptions: flasha=300 nodes, flashd=127 nodes, flashdev=16 nodes
[7] Exceptions: flashd=16GB per node, flashdev=4GB per node
[8] Mauve is a Symmetric MultiProcessor with Direct Memory Access (DMA) shared among all its CPUs.

Table 1. *Current High-Performance Compute Clusters managed by HPC Division (December 2006).*