Petascale Plan 9 on Blue Gene

Eric Van Hensbergen
bergevan@us.ibm.com
IBM Research

Charles Forsyth
forsyth@vitanuova.com
Vita Nuova

Jim McKie
jmk@plan9.bell-labs.com
Bell Labs

Ron Minnich
rminnich@gmail.com
Sandia National Labs

Research Summary

By the end of this decade petascale computers with vastly more computational power than any in current use will be vital tools for expanding the frontiers of science. These systems will have tens to hundreds of thousands of processors, an unprecedented level of complexity, and will require significant new levels of scalability and fault management. Their potential will be difficult to exploit with existing runtimes and operating systems. The current trend has been to adopt Linux and Unix derivatives that hark back to a 30-year old design predating such parallel systems.

Systems with 100,000 cores more closely resemble a distributed system than a single computer in scale and reliability. Even so, we think that at this scale the system must give an application a single consistent interface to resources and services, both local and remote. The system must obviously not require a complete reboot every time one CPU or network link fails. These factors have lead us to apply and evaluate the technology and principles of a distributed operating system, namely Plan 9, to the largest scale system available to us - the Blue Gene supercomputer.

Blue Gene has many *compute nodes*, which do the computation, connected to far fewer *I/O nodes*, which concentrate access to traditional Ethernet and to file systems. There might be one I/O node for every 8 to 64 compute nodes. The existing run-time environment for Blue Gene uses Linux on the I/O nodes, and a custom O/S kernel on the compute nodes. The latter provides a single execution thread scheduled per core (in virtual node mode) or a single thread per node (in communication co-processor mode), and runs MPI jobs. Our solution will instead distribute a Plan 9 system across all nodes.

Designed as a distributed system, Plan 9 plays to Blue Gene's strength: its interconnects. Plan 9's fundamental protocol, 9P, runs over any in-order reliable network. Blue Gene's two principal internal data networks, the torus and the class-routed-network, both provide reliability and in-order guarantees, removing the need for TCP/IP. Ordinary Plan 9 devices can also multiplex direct access to those networks by both applications and kernel at once, so as to eliminate middleware layers (such as MPI) and their associated overhead.

In Plan 9, user-mode and kernel-mode components can be substituted for each other at will, and components can be remote or local, transparent to the application. Some services are built into the kernel as drivers, and some are provided by servers outside the kernel. Applications, however, access them all the same way, and indeed they can be composed in arbitrary ways: an out-of-kernel server can provide an interface to hardware and, in turn, an in-kernel TCP/IP stack can be layered on top of that out-of-kernel server. This provides great flexibility. For example, the existing software stack on Blue Gene demands a choice between complete kernel management of the network to run TCP/IP, or complete user management so that programs can bypass the operating system code path. Our torus driver instead can multiplex both the kernel's TCP/IP stack and applications sending packets directly, avoiding that stark choice.

Plan 9's abilities can even simplify its own deployment: we wrap a synthetic file system layer around the original command-line based front-end control interface to the Blue Gene. This synthetic file system gives the feel of direct interaction with each node, and plugs nicely into existing Plan 9 tools, such as the Acid parallel debugger, allowing cluster-wide debugging of both applications and system software.

Blue Gene's hardware imposes a particular structure on the network, and we keep its distinction between I/O and compute nodes. Since the Ethernet connection to the I/O node is the only link to external resources, I/O node kernels typically act as file servers, network gateways, and front-end nodes for user access. Depending on the number of I/O nodes available in a particular configuration, kernels with specialized configurations may readily be deployed to optimize the services provided by a particular I/O node. Compute nodes may also run specialized Plan 9 configurations providing monitoring, management, or cacheing services. Unlike the standard Blue Gene environment, however, all nodes can run multiple processes and can be accessed directly by normal system calls (not just via MPI).

Mixing many independent processors with highly-structured networks leaves existing designs for conventional clusters or amorphous peer-to-peer networks less than ideal. Plan 9 gives us a firm foundation on which to build and explore new, effective petascale systems.

Poster and Demo Details

Our poster will highlight our baseline experimental platform, illustrating our bring-up environment, initial cluster configuration and the initial Plan 9 interconnect interfaces. We will also present initial performance measurements and analysis comparing a Plan 9 deployment to a traditional Blue Gene system software stack.

If possible at the venue, we will present a demo of the Plan 9 bring-up and debug environment on Blue Gene. This demo will highlight the use of name space and synthetic file system to allow cluster-wide debug using Plan 9's Acid parallel debugger. It will also detail our tight integration with the Blue Gene JTAG network. We'll demonstrate Plan 9's execution environment by stepping through the provisioning, launching, and management of distributed applications on Blue Gene using Plan 9's standard cluster workload commands.

For more information and details regarding Plan 9, visit http://plan9.bell-labs.com/plan9/.