Job Preemption with BLCR

Paul H. Hargrove
With Eric Roman and Jason Duell

http://ftg.lbl.gov/checkpoint

checkpoint@lbl.gov
Job Preemption

- Job preemption stops in-progress work to run higher priority work
  - Scheduled maintenance and job reservations
  - Low-priority job class to run when no other work available (or no other work “fits”)
  - High-priority job class to run immediately
Job Preemption by Killing Running Jobs

• Simplest mechanism to deploy (manual or automatic)
• Lose work performed since last checkpoint (if any)
• May refund user’s account, but can’t rebuild “good will” (e.g. if user misses a paper deadline)
Job Preemption via SIGSTOP

- Can send SIGSTOP to all process in a running job
- Job will no longer consume CPU cycles or network bandwidth
- No loss of partial work
- Will still consume memory (physical or swap) and scratch disk on nodes
- Residual resource usage may prevent the incoming job from running
Job Preemption via Checkpoint/Restart

- Preemption via Checkpoint/Restart aims to reduce impact on system utilization
- Partial results are preserved, regardless of periodic checkpoints (fault-tolerance)
- All resources saves to (presumably dedicated) space on disk
  - No conflict with incoming job
• Checkpoint-based preemption useful even in the absence of priority
  – Avoid long queue-draining times prior to maintenance or large jobs, by allowing more jobs than backfill alone
  – Can run large (e.g. full configuration) jobs during only designated hours

• Fault Tolerance, of course
• BLCR = Berkeley Lab Checkpoint/Restart
• Began in 2002 with the goal of providing Linux clusters with a C/R implementation approaching the quality of that on the Cray T3E
• System-level (in kernel) preemptive checkpointing
  – Records kernel view of process state
  – No virtualization or system call intercept
    • Thus no runtime overhead for presence of BLCR
• We realized that we couldn’t do it all
  – TCP/IP might be possible
    • But would be a terrible restriction on MPIs
  – We could never expect to save/restore state of all high-speed network drivers (InfiniBand, Myrinet, Quadrics, etc.)
  – We could become experts in maybe one MPI implementation, but not all
• Chose to write an *extensible* single-node checkpointer of most POSIX-defined resources
• Inter-node communication was “somebody else’s problem”
  – BLCR provides a callback-based mechanism to extend capabilities
  – MPI is most obvious “somebody”
• More on this later…
BLCR Coverage

- Handle most POSIX-specified resources
- Handle processes, process groups and sessions (later 2 are recent additions)
  - Single and multi-threaded (pthreads) apps
- Still some key exceptions
  - Shared memory support in progress right now
  - No socket support (TCP/IP, etc.)
  - Terminal I/O not supported (no emacs or vi)
  - SysV IPC not supported
The S/W Ecosystem

- Applications
- Libraries (non-communication)
- Communication Libs (MPI)
- Batch system

MANY Want to leave them unchanged
FEW Willing to see these modified
MPI Coverage

• Available today
  – OSU’s MVAPICH2 over InfiniBand “gen2”
  – LAM/MPI 7.x over sockets and GM
  – MPICH-V 1.0.x over sockets (MPICH 1.2 ch_p4 derived)

• The future
  – OpenMPI (succeeds LAM/MPI, FT-MPI, LA-MPI & PACX-MPI)
    • IIRC: Hope for 1.3 release around SC07
  – MPICH2 over sockets and over GM
    • Some work done by MPICH-V folks and at ANL (status?)
  – Cray over portals (for NERSC procurement)
    • Will support for XT4 + CNL est. Mid ’08 (Kramer@SC06)
  – At least one other commercial vendor
  – At least one other academic project
Batch System Coverage

- **TORQUE prototype (predates sessions)**
  - “engineering support” from Cluster Resources
  - Full support in TORQUE for SC07?
  - Expect “ports” to OpenPBS and PBS Pro
  - Also needed for Cray’s deliverables to NERSC
- **SGE “how to” report (predates sessions)**
  - New SGE-work in progress (external)
- **Cobalt (ANL)**
  - Work to be done within CIFTS funding
- **At least one commercial vendor**
- **I know of no work for RMS or LSF**
• Not quite there yet, but getting close
  – Look for BLCR+TORQUE+OpenMPI at SC07?

• Could benefit from new collaborators
  – Anybody want to work on LSF+MVAPICH2?
    (TACC Lonestar)
Other Implementations

- **Condor (OpenSource)**
  - [http://www.cs.wisc.edu/condor/checkpointing.html](http://www.cs.wisc.edu/condor/checkpointing.html)
  - User-level preemptive checkpointing
- **Déjà Vu (commercial)**
  - Also provide a preemptive scheduler, DQ
- **Meiosys MetaCluster (commercial)**
  - Acquired by IBM in Jun 2005
  - I don’t know what has become of the technology
- **(Para-)Virtualization (e.g. Xen or VMWare)**