CRD

BERKELEY LAB

An Overview of Berkeley Lab Checkpoint/Restart (BLCR) for Linux Clusters

Paul Hargrove with Jason Duell and Eric Roman

http://ftg.lbl.gov/checkpoint

Office of Science
U.S. DEPARTMENT OF ENERGY

March 18, 2008

BLCR is…

- BLCR is…
  - Berkeley Lab Checkpoint/Restart
  - System-level preemptive checkpointer
  - Linux specific
  - Single-node, multi-process
  - Extensible for multi-node (e.g. MPI)
  - Kernel module + stub library
  - x86, x86-64, ppc64 and ARM

Outline

- Project goals / motivation
- System design
- Extension interface
- Current status
- Future work

Uses of Checkpoint/Restart

- Gang scheduling
  - No queue drain for maintenance, policy change
  - Higher utilization and/or more flexible scheduling
- Process migration
  - Save job if node failure imminent
  - Pack jobs for optimal network performance
- Periodic backup
  - Not our main focus
  - Application can always do more efficiently
  - But may be useful for systems with long jobs, fast I/O, and/or high node failure rates

Implementation Strategies

- Application-based checkpointing
  - Efficient: save only needed data as step completes
  - Good for fault tolerance: bad for preemption
  - Requires per-application effort by programmer
- Library-based checkpointing
  - Portable across operating systems
  - Transparent to application (but may require relink, etc.)
  - Can't (generally) restore all resources (ex: process IDs)
  - Can’t checkpoint shell scripts (children, etc.)
- Kernel-based checkpointing
  - Not portable, and harder to implement
  - Can save/restore (nearly) all resources

Design Goals

- Target: parallel scientific applications
  - MPI is a must
  - But allow support for other programs/models, too
  - Esoteric features (ptrace, Unix domain sockets) have lower implementation priority
- Implementation: Linux kernel module
  - Lower barrier to adoption than kernel patch
  - Allows upgrades, bug fixes, without reboot
  - No interpose = no added runtime overheads

Design Goals II

- Provide 'toolkit' for distributed C/R
  - We provide single node checkpoint/restart
  - We don't support distributed operating system features
    - No built-in support for TCP sockets, bproc namespaces, etc.
  - We provide hooks to allow parallel runtimes/libraries to implement distributed checkpoint/restart
    - So the MPI library needs to know about checkpointing, but user applications don't

BLCR Extensibility 1

- 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

BLCR Extensibility 2

- 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…

Extension Interface

- Callback functions
  - Registered at start-up (or as needed)
  - Run at checkpoint time, then resume at restart/continue
  - Handle parallel coordination and/or unsupported objects
- Two types of callbacks
  - Signal handler context
    - No thread-safety needed
    - But callback limited to calling signal-safe functions (small subset of POSIX)
  - Separate thread context
    - Can call any function
    - But code needs to be thread-safe
- Critical sections
  - Protect uncheckpointable sections of code

Status: BLCR Coverage

- Handle most POSIX-specified resources
- Handle processes, process groups and sessions
  - Single and multi-threaded (pthreads) apps
  - Pipes, sharing and parentage restored
- Still some key exceptions
  - No socket support (TCP/IP, etc.)
  - Terminal I/O not supported (no emacs or vi)
  - SysV IPC not supported

Status: 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
  - 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

Status: Batch System Coverage

- TORQUE prototype
  - Now in Cluster Resources' SVN repo
  - 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

Future Work

- Continue to update w/ Linux Kernel
- More integration w/ batch systems
- Continued and improved MPI support
- Additional files support
- Additional POSIX resource support