Condor on Dedicated Clusters

Peter Couvares and Derek Wright

Computer Sciences Department

University of Wisconsin-Madison

pfc@cs.wisc.edu, wright@cs.wisc.edu

http://www.cs.wisc.edu/condor

Condor on Dedicated Clusters: Overview

- Existing Implementations & Capabilities
  - NCSA Origin 2000
  - UW-Madison Compute Cluster
  - Condor Features, Tools, Tricks

Condor on Dedicated Clusters: Overview (cont.)

- Future Directions
  - Resource Reservations
  - Parallel Schedulers
  - Two Scheduling Models in One Pool

NCSA Origin 2000 and Condor

- Massively parallel individual machines (64 to 128 CPUs each)
- LSF scheduler used to schedule dedicated jobs on multiple CPUs
- Condor ready to "backfill" dedicated schedule gaps with opportunistic jobs

NCSA Origin 2000 and Condor (cont.)

- When LSF sees a scheduling gap, it can dynamically inform Condor of the number of available CPUs
  - If more nodes are free, they are advertised to a local Condor pool
  - If more nodes are needed, Condor uses job state to intelligently choose which CPUs to return to LSF

UW-Madison's Compute Cluster

- Built from Low-Cost Commodity Parts
  - 64 Nodes
  - Each with dual-550mhz Xeon CPUs, 1GB RAM, 100Mb NIC
  - Dedicated resources
- Condor runs on all nodes unless specifically disabled for "clean-host" experiments

UW-Madison's Compute Cluster (cont.)

- Supported by 2 quad-CPU fileservers
  - 50GB storage each, 1Gb NICs to cluster
  - Dual-homed with 100Mb NIC to high-speed EMERGE research network
    - Connected to SRB, HPPS, and other data storage systems
    - Network has quality of service functionality
      - Can reserve bandwidth for transfers

Condor & Clusters: PVM & MPI

- PVM works well with Condor's traditional opportunistic model
  - Can dynamically adjust to # of available nodes
- MPI doesn't work as well
  - Requires fixed number of nodes
  - More suited to dedicated resources

Condor & Clusters: Features & Tools

- Parallel checkpointing
  - Problem:
    - Traditional Condor pools often can't assume each machine has adequate disk/network to checkpoint jobs
    - Central checkpoint server is a *potential* bottleneck

Parallel Checkpointing (cont.)

- Observation:
  - Clusters often have adequate resources
    - Great network between nodes
    - Lots of local disk
- Solution:
  - Checkpoint to multiple servers
  - Servers run on cluster nodes themselves
  - Scales very well: You can have a checkpoint server on every node!

Future Directions

- Reservations
- Parallel Scheduling
- Dual Scheduling Across 1 Pool
  - Dedicated scheduling
  - Opportunistic scheduling

Future Directions: Reservations

- Two kinds of reservations: interactive users & future jobs
- Use by resource owner not an issue
- Guaranteed reservations needed
- Need two more entities in Condor:
  - Reservation manager
  - Reservation enforcer

Reservations (cont.)

- Reservation Manager:
  - Reservations are represented as a ClassAd -- can use all the existing technology:
    - Persistent storage
    - Network communication layer
    - Visualization

Reservations (cont.)

- Reservation Manager (cont.):
  - Support reqests by jobs in the system
  - Supports a GUI for interactive users
- Reservation Enforcer:
  - Uses ClassAd matchmaking technology
  - Vacate nodes in advance to avoid flooding the network
  - Current plan: use the Eventd

Future Directions: Parallel Scheduling

- Co-scheduling of multiple hosts
  - MPI Job ClassAd might require N nodes
    - You want all or nothing, or you can have deadlock
  - Other jobs might require co-scheduling:
    - A multi-threaded application might want to claim multiple CPUs on a single SMP machine
  - Requires "gang-matching"

Future Directions: Dual Scheduling Across One Pool

- Condor's hierarchical and parallel scheduling architecture will enable dedicated and opportunistic schedulers to coexist and efficiently share dedicated and non-dedicated resources alike
- Resources will be able to migrate between scheduling systems

Dual Scheduling Across One Pool (cont.)

- During schedule gaps, dedicated compute machines will become available to the opportunistic scheduler
- However, dedicated machines will always "prefer" the dedicated scheduler and will return as soon as they are needed

Questions
and
Thank You!

www.cs.wisc.edu/condor

Condor