



Condor and Multi-core Scheduling



Dan Bradley

dan@hep.wisc.edu

University of Wisconsin

with generous support from the National Science Foundation
and productive collaboration with Red Hat



What Exists Today



- **Parallel scheduling**
 - Designed for MPI-type jobs
 - No way to limit matchmaking to slots on same machine
- **Custom Batch Slots**
 - One “slot” can be \leftrightarrow one core
 - Slot policy can depend on type of job



Example: Custom Batch Slots



- Slot 1, 2, 3
 - only accept 1-core jobs
- Slot 4
 - when claimed by normal 1-core job, behaves normally
 - when claimed by 4-core job
 - stop accepting jobs for slots 1, 2, and 3
 - suspends 4-core job until slots 1, 2, and 3 drain

Related Condor How-to: <http://nmi.cs.wisc.edu/node/1482>



shortcomings



- Awkward to extend to N-core jobs, but doable
- *Very* awkward to extend to N-core jobs and also support dynamic partitioning of memory, etc.
- Requires custom configuration by admins; no standard JDL for submitting grid jobs to sites
- Accounting does not charge multi-core job at higher rate than 1-core job



in development: dynamic slots



- Start with one “batch slot” representing whole machine
- Trim slot to what job requires (cores, memory, disk, network, etc.)
- Leftovers assigned to new slot



TODO



- improve accounting and any other monitoring to support multi-core slots
- support standard JDL for requesting multiple cores
- deal with preemption and/or mechanism for preventing starvation of multi-core jobs
- speed up dynamic slot creation



Questions



- Is it desirable to have the capability to lock a job to a CPU?
- Does Globus provide RSL attributes with well-defined semantics for multi-core jobs?
 - example: (count=N)(jobtype=single)
 - in my experience, this is *not* well-defined
 - can mean N cores x 1 job or N x job