Multicore/Virtualization and Data Management

Peter Elmer
Princeton University
Storage Jamboree, Amsterdam
16 June, 2010

Multicore CPU's

- This workshop is focused on one aspect of the evolution of the computing models of the experiments, namely the evolution of storage and the organization of and access to data
- There is of course another relevant change coming which is relevant for some aspects of the data management, namely the foreseen evolution of the processors to ever larger numbers of cores and the evolution of our software to exploit these processors more sensibly
- Multicore is the topic of the LCG R&D Workshop next week (along with virtualization)
  - My comments should be interpreted as “CMS”
  - A proper discussion with all experiments will happen next week

Multicore CPU's

- The current model for HEP applications to exploit multicore CPU's is very simplistic: we exploit the event level parallelism and (typically) simply launch one application per core
- The local schedulers matched this by configurations to schedule independently (and incoherently) an individual “job” per core
- (Hyperthreading and pilot jobs are only minor caveats to the above)

Multicore CPU's

- The result “works” in that we are able to exploit multicore CPU's for small numbers of cores, but there are many consequences as the number of cores/CPU increases:
  - The memory needs increase with each generation of CPU's
  - The number of independent readers and writers (to local disk, to remote storage) increases with each generation of CPU
  - Ever increasing numbers of independent and possibly incoherent jobs running on any given piece of physical hardware. Each of these running “jobs” commands an ever tinier slice of resources
- And on top of that, there are a number of reasons to believe that this trivial parallelism won't scale efficiently on the CPU's themselves
- Thus the experiments have been preparing various better implementations of “multicore-aware” applications

Multicore Deployment

- We would like proceed with the commissioning and deployment of "multicore-aware" applications in the next six months (i.e. by end of 2010)
  - This implies grid/scheduler configurations to permit us to get something more than "one job slot ~ one core" as well as providing the corresponding accounting metrics (for CPU, memory use, etc.)
- Ideally, the best model would be to go back to scheduling the **entire node** and allowing us to exploit it. (Where "entire node" means "the physical thing running a single unvirtualized copy of the OS".) We do not see a use case for asking for (or limiting ourselves to) an arbitrary number of cores. We believe that we can exploit an entire node efficiently, regardless of the number of cores it has. There is no reason to subdivide it.

How does this affect Data Mgmt?

- The number of schedulable running "entities" within the system will drop by a factor of 4-8 (today) and will become approximately constant in each site going forward
- Changes the system working point by an order of magnitude
- The resources (local disk, memory, etc.) managed by a (pilot) job start to become significant, providing many opportunities for optimization:
  - Stage-in to and mgmt of local disk, suddenly we are "memory rich"
  - coordinated/coherent I/O access across activity "node"
  - reduced external connections&streams
  - local output merging (or direct write of combined file)
  - "backfill" CPU intensive activities if necessary

Virtualization

- A side note on virtualization, as there seems to be a disconnect here:
  - CMS (as an experiment) is not asking for the general introduction of virtualization on WN's. In fact we are specifically requesting that if we say “we can run on the actual OS version of the node” that we be allowed to do that, without any virtualization arbitrarily introduced
- Clearly there are situations (e.g. OS version migration) where we might not be able to do that, and virtualization can help us handle this better than in the past. If we don't *need* it, however, we shouldn't have to pay any cost for it.
- Clearly if we don't own the resources (e.g. cloud computing), the choice is not necessarily ours

Summary

- Storage/data access and their evolution are the themes of this workshop
- The evolution will however include an evolution of what is considered a “job” due to the deployment of multicore-aware applications, with an effect on (and opportunities for better) data management
- Discussion in LCG R&D workshop next week will hopefully result in some single statement for GDB, etc.
- In CMS we would like to begin to deploy and commission multicore-aware applications using a “whole node” granularity this year