VMs for Resource Multiplexing

Low-Power Computing
Carnegie Mellon University
David Andersen

HPC Background

- Types of hardware/clusters
- Types of workloads
- Management systems (condor, etc)
- Programming them
- Challenges

HPC Clusters

- Older: Sandia Red Storm: Cray XT3/4
  - 13,000 nodes
  - each w/2.4Ghz AMD Opteron, 2-4GB RAM
  - Cray SeaStar network interface - 2GB/s (That's bytes...)
  - 100 GB/s to 1159 TB of parallel disk
  - 50 GB/s of external network b/w

Newer: Ranger

- SunBlade x6420
- 3,936 nodes / 62,976 cores (Q core, Q proc)
- 123TB memory (32GB per node)
- 1.73PB shared disk, 31.4TB local
- 579.4 TFlops

Local:

- PSC's "Salk" cluster
  - 36 bldes -- dual proc, dual core
  - Itanium2, 8GB local memory
  - NUMAlink interconnect - shared memory (previous ones were message-passing)

Evolution of HPC

- In the old days: Supercomputers. Vector supercomputers.
- Then: Shared-memory MP machines
- Now: Clusters of commodity nodes
- Tomorrow?

HPC Frontiers

- Multiprocessor all along
- Multicore yesterday
- Tons of cores today (bluegene-L; reading next week - uses Cell processor)
- Doubly tons of specialized cores tomorrow (NVidia Tesla, Intel Larrabee - massive cores + vector proc)
  - Have to map compute to them, but large benefits iff you can

HPC vs "normal" cluster?

- Typically the interconnect
  - Infiniband, etc. - very low-latency, high bandwidth switched networks
  - e2e latency is in microseconds
- Cray used to make their own, etc.

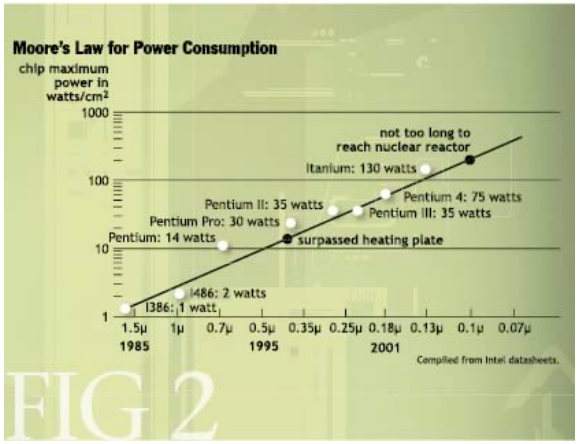
HPC Challenges

- Reliability, reliability, reliability
  - When you have 13,000 nodes, *something*'s going to crash. Soon.
  - Checkpoint + restart is the usual answer.
  - Time to checkpoint becomes reliability limit.
- Data storage, I/O, reliability (see Schroeder/Gibson)
- Heat & Power
- Programming the damn thing!

HPC Power

- 1991: Cray C90, 600 sq ft, 500kW
- 1991: Intel Delta, 512 CPU (nearly as fast as C90), 53 kW, 200 sq ft
- 2002: ASCI Q machine: 17,000 sq feet, 3MW of power.
  - Performance grew 2000x since 1991
  - But only 65x per square foot
  - And only 20x per watt

More power



Source: Wu-chun Feng, ACM Queue Article

HPC Workloads

- CPU-bound: finite element simulations, computational astrophysics/chemistry, etc.
  - Common theme: Interactions between (many!) particles, tiny timesteps, figure out local changes, iterate.
- I/O: Loading models, storing results

Benchmarks

- Standard but not always helpful: LINPACK, etc. (Linear algebra kernels, etc.)
- Better: NAS Parallel

[illegible]	[illegible]	0.2	7.2	24
1 solver (BT)	12^3	0.3	7.2	34

NAS Parallel Benchmarks Sample Code Statistics

- Best: Your own codes...

Table from NASA NAS parallel benchmark specification

What's the real ?

- Given a workload that (usually) runs on multiple machines,
- Where the workload is divided into units that can be run {somewhere}
- How to allocate that workload onto physical machines?
- Complications:
  - Time-varying workload per unit
  - Do workloads compose linearly?? (Cache; Disk sharing)

Interesting observations

- Running full-bore
  - Power goes up as workload leaves cache; goes down as memory unable to saturate CPU
  - This kind of result likely to be *very* workload dependent.

IBM paper results

- Cache-aware packing is critical. Heuristics:
  - If WSS << cache,
    - Pack such that sum(WSS) <= cache
  - If WSS >> cache
    - Pack with other >> cache apps, take tons of memory (They're slow anyway)
  - In the middle -- your choice. Fewer machines vs. performance.

VMs and HPC

- Earlier work by HP: Consolidation has benefits - many jobs are idle sometimes; some jobs are full-bore (testing & devel vs. production runs)
- Huge performance fear - HPC workloads often super-optimized...

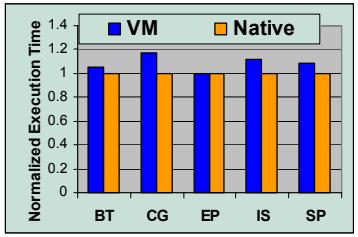
VMs

- Reason 0: Customize OS used on the nodes.
  - Mayyybe: Faster OS (but VM...)
  - Definitely: Usability (but maybe slower); Security
- Option 1: Consolidation
  - Do any jobs use << CPU time than machine time?
- Option 2: Migration

Virt overhead for HPC

- Most virt runs native machine instructions
  -

But Performance?



	Dom0	VMM	DomU
CG	16.6%	10.7%	72.7%
IS	18.1%	13.1%	68.8%
EP	00.6%	00.3%	99.0%
BT	06.1%	04.0%	89.9%
SP	09.7%	06.5%	83.8%

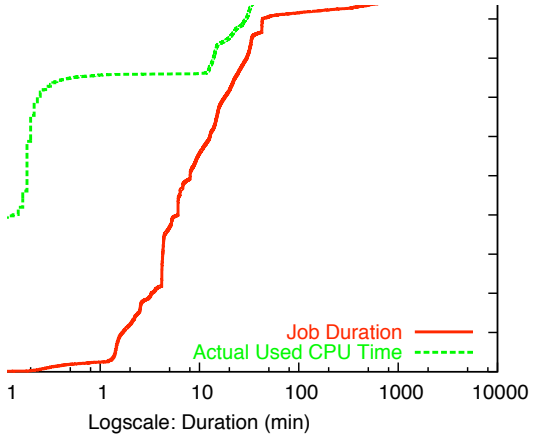
- NAS Parallel Benchmarks (MPICH over TCP) in Xen VM environment
  - Communication intensive benchmarks show bad results
- Time Profiling using Xenoprof
  - Many CPU cycles are spent in VMM and the device domain to process network IO requests

ICS'06 -- June 28th, 2006

Slide from Wei Huang ICS '06 talk, "A Case for High Performance Computing with Virtual Machines"

HPC & VMs

- Data from RRC Kurchatov Institute (Moscow) HPC cluster - 100 nodes, 2.8 Ghz Xeon, 2GB, 80GB disks
- Comparison: Actual time (ACT) vs. Wall-clock (WCT)



(a)

Source: *Optimizing Grid Site Manager Performance with Virtual Machines*, Cherkasova, Gupta, et al.

Job Distribution

- Long jobs (> 1 day) consume 80% of the CPU resources
- 2% of jobs last longer than 3 days, but consume 42% of the CPU resource

50% of jobs use less than 2% of their WCT

Source: *Optimizing Grid Site Manager Performance with Virtual Machines*, Cherkasova, Gupta, et al.

Whole-DC Power Management

- Qs: Model <X> vs power, or dynamically measure?
  - Generality vs. (possibly) response time vs. (possibly) correctness
- Scaling & stat mux -- P2 had a very stat-mux-like flavor (increasing time-scales at increasing granularity)
- Only 2 p-states needed? (recall earlier "dominant p-states" thoughts) -- VM consolidation might help here by shifting machines more towards "full" or "off"