Linux Kernel Scalability

New features in Linux kernel v2.6
for high-end computing

Jes Sorensen
Wild Open Source Inc.
jes@wildopensource.com
http://www.wildopensource.com/

Agenda

- What is the 2.5+ fuzz all about?
- For what kinds of applications will 2.6 be a major improvement?
- Kernel improvements for scalability
- Q&A

2.5+ and all the fuzz

- Linux 2.5.x is a development kernel, not intended for production use
- Linux 2.6 which will be the first major stable kernel release since 2.4.0 - (January 7th, 2001)
- Current distributions all ship 2.4.x
- Wide adoption will happen when the major distributions start shipping 2.6

What applications will gain from 2.6

- 2.6 will not be stunningly faster than 2.4 for normal workloads on normal machines
- It should be more pleasant to use on the desktop due to improved fairness and lower latency
- Major improvements in the areas where 2.4 was sub-optimal:
  - large number of processors
  - large amounts of memory
  - large number of threads and/or processes
  - large disks and number of disks
  - high performance networking
  - improved error handling

New O(1) scheduler

- Scheduling performance almost independent of number of processes and threads in the system
- Scales better on SMP systems - per CPU run-queues
- Interactive tasks receive special treatment, resulting in better interactive performance under high load
- New improved thread implementation, NPRT (Native POSIX Threading Library)

SMP locking

- All global locks removed from VM layer (pagemap_lru_lock and pagecache_lock)
- Block layer lock removed
- cli()/sti() (global interrupt disable) has been eliminated
- Developers are running Linux on 32 CPU machines (and bigger!)

Virtual Memory layer improvements

- Virtual Memory layer has been improved substantially
- RMAP (reverse map) provides virtual->physical and physical->virtual mappings
- Allows for more intelligent decisions on what memory to swap out/in to/from disk.

Improved block I/O layer

- Block layer has been rewritten
- Much improved error handling
- io_request_lock is gone - global lock that was used by all block device drivers and the block layer
- “biobufs” (block I/O buffer) allowing I/O requests larger than PAGE_SIZE
- 64 bit DMA to HIGHMEM
- SCSI is undergoing major improvements (after several years of staleness)

File Systems

- New high performance file systems: XFS, JFS
- Large File System support
  - 2.4: 1TB or 2TB
  - 2.6: 10^18 bytes for XFS
- Larger files:
  - 16TB with a 4KB PAGE_SIZE (ia32)
  - 64TB with a 16KB PAGE_SIZE (ia64)

Questions?