What's New (and Interesting) in the 2.5 Kernel

Robert Love
<rml@tech9.net>

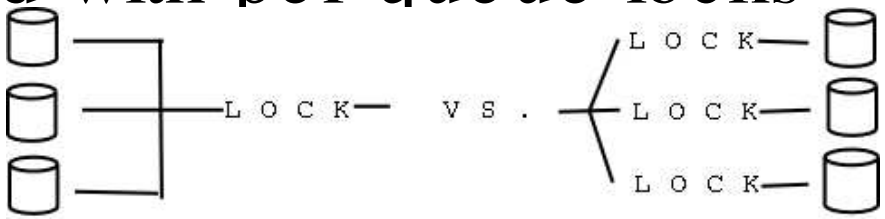
http://tech9.net/rml
http://www.kernel.org/pub/linux/kernel/people/rml
http://www.mvista.com

History thus far of 2.5

- The 2.5 series was started 22 November 2001 when Linus created 2.5.0 as a branch of 2.4.15
- At the Kernel Summit in July 2002, Halloween was picked as the freeze date
- Linus went on vacation on 20 October... but returned in time to release 2.5.45
- No new major new features are to be added (or so it says)

New Block I/O Interface

- Redesigned block I/O layer: the `bio` is now the fundamental data structure
- Bounce buffer no longer needed for block I/O into high memory
- Global request lock (`io_request_lock`) replaced with per-queue locks

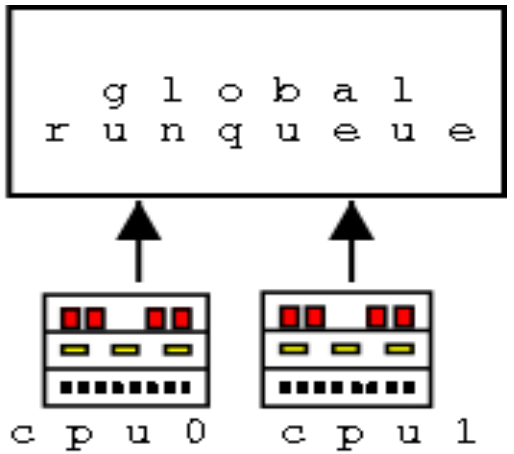


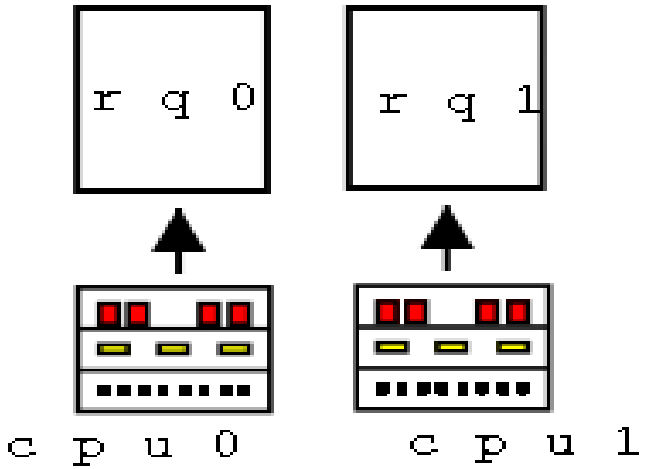
O(1) Scheduler

- Radically redesigned scheduler merged into 2.5.2
- Multi-queue scheduler: per-processor runqueues and locking
- All scheduler algorithms are O(1) – they execute in constant-time regardless of number of processes (or anything else)

Multi-queue scheduler

- The new O(1) scheduler has a unique runqueue for each processor
- Each runqueue has its own lock





Scheduler: O(1) vs. O(n)

- O(n) scheduling:

```
for (each process on the system) {
        find worthiness of this process
        is this the worthiest process yet?
}
run most worthy process
```

- O(1) scheduling:

```
get highest priority level that has
    processes
get first process in the list at that
    priority level
```

Preemptive Kernel

- The Linux kernel was made fully preemptive in 2.5.4
- Previously, kernel code ran cooperatively
  - Source of high scheduling latency
  - Lower priority tasks in the kernel *priority invert* higher priority tasks waiting to run
- Locks are leveraged as preemption markers - kernel now knows when it is safe to preempt and can do so as needed

Improved Latency

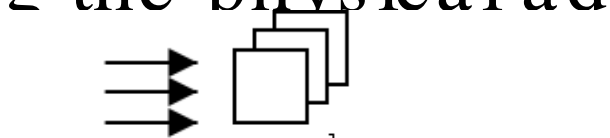
- Kernel preemption provided a base for improving process response
- Now short lock held times => less latency
- Much work in the VM and VFS to reduce latency and prevent catastrophic meltdowns
- Recent LJ tests showed latency decrease of 78.51ms to 0.48ms in Realfeel test

VM

- The VM subsystem is responsible for allocating and managing memory and the process address space
- Reverse-mapping (rmap) VM
- Redesigned, smarter, simpler algorithms
- Provide superior performance in the common case without miserably falling to pieces in the corner cases

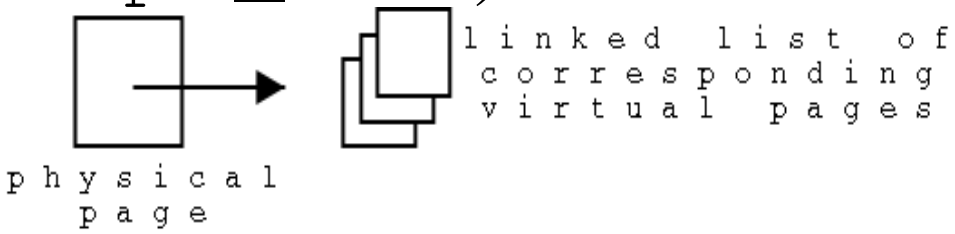
Without rmap

- Linux is a virtual memory OS so your system uses both physical and virtual address
- Virtual addresses are resolved to physical addresses via the page tables (this is easy)
- Physical addresses are resolved to virtual address by looking at *each* page table entry and comparing the physical address (gross)



Reverse mappings

- rmap provides reverse mappings: physical to virtual lookup without scanning page table entries for corresponding physical addresses
- Instead, the physical page (`struct page`) itself contains a list of all matching page table entries (the `pte_chain`)



Keeping Disks Saturated

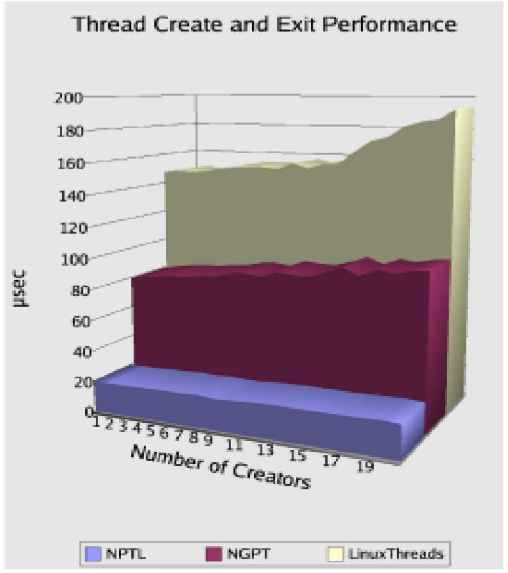
- `pdflush` replaces `kupdated` and `bdflush`
  - `pdflush` avoids the situation where `bdflush` gets stuck on a particular device, starving others
- New Deadline I/O scheduler
  - Gives I/O requests a deadline: ensures disk requests are serviced in a given time
  - Favors reads over writes to prevent writes-starve-reads issues
- VM changes

Threading Changes

- New POSIX Threading Library (NPTL) in glibc 2.3 – entirely redesigned pthreads – replaces LinuxThreads (old library)
- Kernel changes
  - Fast user-space mutexes (futexes)
  - O(n) `exit()` replaced with O(1) `exit()`
  - Proper thread local storage (TLS) support
  - Thread CPU affinity system calls
  - VM changes

Threading Improvements

- Order-of-magnitude improvement in thread performance
  - NPTL is 8x faster than LinuxThreads
  - NPTL can start and stop 100,000 threads in only two seconds. Prior, this would take l
  - <1 micro-second context switch time
  - exit() performance improved 400x in the case of 4000 threads
  - Wow!



2.6 vs. 3.0

- 2.6: 2.4 was a bigger change, "2.6" sounds more stable, who cares?
- 3.0: this kernel is a big change, "3.0" sounds more feature-packed, who cares?
- 5.4: why the petty hang ups over version numbers, anyhow?
- Currently, 2.6 looks to be the choice...

Release Dates?

- Of course, 2.6/3.0 will be released when its done
- But, we feature froze on 31 October 2002
- And, Linus commented on his cruise he wanted to code freeze early 2003
- So, perhaps summer of 2003?

What does the future hold?

- 2.6/3.0 stable kernel
- 2.7/3.1 development kernel
  - tty layer rewrite?
  - SCSI rewrite?
  - Object-based reverse-mapping VM?
  - Better NUMA support
  - Whatever the itch is...

What's New (and Interesting) in the 2.5 Kernel

Robert Love
<rml@tech9.net>

http://tech9.net/rml
http://www.kernel.org/pub/linux/kernel/people/rml
http://www.mvista.com