Status and Direction of Kernel Development

Andrew Morton
<akpm@linux-foundation.org>
<akpm@google.com>

Japan Linux Symposium 2007
March 2007

Recap from June 2006

- NUMA work: ongoing
- Memory/NUMA-node hot-add: merged
- Memory hot-remove: no progress
- Memory anti-fragmentation: in progess
- Pagetable sharing: stalled
- Hugetlb work: slow progress
- Security: proceeding steadily
- OS virtualisation: proceeding slowly
- Hardware virtualisation: complete by mid-year
- Reiser4: stalled
- AIO: stalled
- Lockless pagecache: no progress

Recap from June 2006 (cont'd)

- Ext4: slow progress
- eCryptfs: merged
- GFS2: merged
- fscache/cachefiles: stalled
- Per-task delay accounting: merged
- Statistics infrastructure: stalled
- Perfmon: little progress
- Kprobes: in maintenance
- Userspace probes: stalled
- Kdump: in maintenance
- Lock validator: merged
- CKRM: stalled/dead

Recap from June 2006 (cont'd)

- Drivers: steady progress
- Driver hardening: no progress

What's new?

- Hardware virtualisation
  - KVM was merged quickly, is progressing
  - Core paravirtualisation was merged
  - VMWare VMI interface is merged
  - Xen domU support will probably be in 2.6.22

OS virtualisation

- Progress is slow and steady
- Most work involves virtualisation of the kernel's global namespaces
  - utsname, PIDs, UIDs, shm IDs, mounts, netdevices, etc
- The core structure is the nsproxy
- We are merging it one component at a time
- The code at present isn't useful – more work needs to be done before we make it available to userspace

Containerisation (resource management)

- Still no overall plan on how to do this
- Best prospect is generalisation of cpusets
  - cpusets are presently a container for CPUs and NUMA nodes
  - Rename cpuset to “container”, permit containment of other resources
- The big ones are CPU resources and memory
  - Also net bandwidth and disk bandwidth
- Per-container memory limitation is a big problem
  - We don't even have a usable design for this
- Memory containment can be implemented using existing cpusets and fake NUMA
- Help is needed with containerisation

Memory management

- Mel Gorman's anti-fragmentation and ZONE_MOVEABLE work is in -mm
  - Possibly useful for memory hot-unplug
  - It is unclear how useful this will be for per-container memory limitation
- There are reports of scalability problems with large systems
  - Possibly due to cacheline-capturing effects on multicore
- There are reports of memory reclaim inefficiencies under database workloads

Filesystems

- XFS remains the filesystem of choice for high-end applications, due to superior scalability
- But vendors shy away from XFS for supportability reasons
- Vendors seem to be converging on ext4 due to widespread support and stability
- Ext4's roadmap looks good, but progress is slow
- Additional resources here will help
- NFS4 progressing steadily

AIO

- Filesystem AIO is presently supported for direct-IO only
- AIO patches for buffered filesystem AIO are mature
  - But might not be merged due to the syslets proposal
- syslets will make potentially all syscalls asynchronous
- Hence the present AIO code could be removed – applications simply do an asynchronous read()

kevent

- Linux lacks a unified event delivery framework
- Pipes, ttys, sockets, AIO, signals, futexes, etc
- We should be able to wait upon any kernel completion in a single syscall in a unified fashion
- Kevent is a proposed framework for doing this
- But progress is slow due to lack of external review and testing

Misc other new work

- Utrace: new process tracing infrastructure
  - Ptrace becomes layered on top of utrace
  - Utrace is basicaly a complete rewrite of ptrace
- New page replaccement algorithms are being discussed
- Kernel trace infrastructure using static markers
- Technology is being steadily moved over from the realtime kernel

Kernel processes

- We use a two-week merge phase, followed by a two-month stabilisation phase, followed by a release
- This process has been stable for several years and appears to be working OK
- Our weakness is, still, an inadequate amount of work put into resolving regressions
- Our tracking of bugs and regressions is also haphazard
- But there is little point in tracking bugs if there is nobody to fix them
- We devote too few resources to reviewing patches