Status and Direction of Kernel Development

Andrew Morton
<akpm@osdl.org>

OSDL-Japan Linux Symposium
June 2006

Looking Forward

- Overview of recent and pending kernel features which are relevant to Enterprise computing
- A walkthrough of the various kernel subsystems, looking at what is coming
- Will also review current and potential problems with getting these features merged into the public kernel

Memory Management

- NUMA work
  - NUMA node-aware memory placement (page allocator, slab)
  - Inter-node page migration APIs
  - Proceeding steadily – Christoph Lameter is doing good work here
- Memory/NUMA-node hot-add
  - Proceeding steadily - Yasunori Goto is leading this
- Memory hot-remove
  - This is really hard. Some attempts have been made, but no apparent progress in about a year
  - Linus says “do it in hardware”

Memory Management (cont'd)

- Fragmentation avoidance in the page allocator
  - Improved success rate for atomic higher-order allocation attempts (gigabit networking)
  - Perhaps improved page coloring
  - Perhaps will permit dynamic allocation of hugetlb pages
  - Developer: Mel Gorman. Possibly will be merged this year
- Pagetable sharing
  - Important feature for some databases (great reduction in kernel memory usage)
  - Dave McCracken continues to work on it, but it is complex and progress is uncertain

Memory Management (cont'd)

- Ongoing hugetlb work
  - Many groups are interested in hugetlb pages
  - Recent improvements in faulting/mprotect
  - People are interested in being able to use hugepages as general-purpose memory
    - For shared library text: feasible, but I'm unaware of serious work happening at present
    - For malloc(): this is hard

Security

- Ongoing work with address-space/mmap randomization to make attacks harder
  - Features are being merged slowly but steadily, mainly from Red Hat's “execshield” product
- SELinux continues to be well-supported
  - The userspace policies are causing some problems, but Red Hat are persisting

OS Virtualization

- This refers to the ability to run multiple instances of userspace on the same kernel
- Several groups have similar products – most prominent is Vserver
- They appear to be cooperating well and work in the public kernel is ramping up
- Resource isolation between instances is a problem
  - Should be able to leverage CKRM infrastructure
- Migration of OS instances between machines requires kernel object serialization and might be messy
- We're not yet at a merge decision-point

Hardware Virtualization

- Xen, Vmware, etc
- Distribution support is ramping up, but work in the public kernel is slow, and there are several process problems
- We don't yet know what the eventual paravirtualization support in Linux will look like
- If it differs significantly from today's Xen then the vendors may have some back-compatibility problems
- We urgently need to set our direction and get the work done

VFS (namespace plane)

- The Virtual Filesystem's current namespace capabilities are sufficient for OS virtualization
- Some work is being done to support read-only bind mounts
- The activity level in this part of the VFS is relatively low

VFS (data plane)

- The second rewrite of the 2.6.x readahead code is in -mm kernels
  - Quite complex and intrusive
  - No immediate plans to merge this
- The reiser4 team are working on improving the performance of the core write() handling
- The direct-io code is stable, but has become complex and it somewhat inefficient for some workloads
- The AIO code was never completed. There are additional out-of-tree patches but we are uncertain whether to proceed with them

VFS (data plane) (cont'd)

- “lockless pagecache” (Nick Piggin)
  - Reduce contention/traffic on the radix_tree lock via RCU
  - Somewhat intrusive, but we might merge this if the benchmark results are good

CPU Management

- CPU scheduler enhancements continue to be merged at a steady rate
- Lead times for scheduler features are long
  - The code is complex and is sensitive
  - Discovery of performance regressions tends to take a long time
- Multicore-awareness has been merged and work on that continues
  - People are looking into moving some power-management awareness into CPU scheduler decisions
- The priority-inheriting futex feature will be merged in 2.6.18 (for POSIX PTHREAD_PRIO_INHERIT and PTHREAD_PRIO_PROTECT)

Filesystems

- Reiser4
  - Development continues
  - Merge is stalled due to lack of OSD interest and lack of review resources
  - We need to find a way to help it along
- Ext4
  - Improved performance and large-device scalability
  - Extents, delayed allocation, multiblock allocation, 48-bit block numbers, etc
  - A team has been formed and is actively working this
- eCryptfs
  - Possibly a 2.6.18 feature

Filesystems (cont'd)

- GFS2
  - Probably a 2.6.18 feature
  - GFS2 also is somewhat stalled by lack of expert reviewers
- fscache/cachefs/cachefiles: local disk-backed caching for network filesystems (NFS, AFS)
  - Large, complex
  - Perhaps will be merged in the 2.6.19 timeframe if we work on it

Surveillability

- Per-task delay accounting
  - Plan to merge for 2.6.18
  - Provides extensible netlink-based mechanism for passing per-task accounting up to userspace
  - Future accounting enhancements should be based on this
- Statistics infrastructure
  - For non-task-associated accounting (eg, I/O accounting)
  - Probably will not merge for 2.6.18
  - It is unclear whether this will meet the future accounting requirements of other subsystems
  - Needs more review, community feedback

Surveillability (cont'd)

- Perfmon
  - Generic access to CPU-specific performance counters
  - Large, complex, mature product with significant existing user base
  - There are concerns that it might be overdesigned
  - We're having trouble getting momentum behind this feature

Diagnostics

- Kprobes
  - Well supported, features continue to be merged
- Userspace kprobes
  - Ability to insert a probe point on a userspace instruction
  - This feature had significant design issues. They appear to be unresolvable
- Kdump
  - Well supported, slow but steady improvement
  - Progress in userspace has been disappointing. We need all OSDs to support kdump out-of-the-box so that community-based testers can easily send dumps to developers

Diagnostics (cont'd)

- Lock validator
  - Runtime validation of kernel locking consistency/correctness
  - Recently merged into -mm kernels
  - Is expected to provide considerable assurance of the correctness of new development
  - Fairly intrusive and large
  - Quite a large number of lockdep-specific annotations are needed to suppress false positives
  - A 2.6.19 merge is quite likely

Manageability

- CKRM
  - The current CKRM core is well-implemented and is acceptable for a merge
  - But some of the CKRM controllers are more problematic
  - Cannot proceed with merging CKRM core until the CKRM controllers are deemed acceptable
  - CPU controller: needs work, but will be OK
  - I/O controller: has not been published yet
  - Memory controller: very problematic
  - Unless/until we can sort out the design of the CKRM memory controller, CKRM is blocked

Drivers

- Infiniband drivers continue to be merged at a high rate
- Serial Attached SCSI drivers are proceeding gradually
  - Possibly a 2.6.18 feature, maybe 2.6.19
- SCSI target patches are being worked on – perhaps a 2.6.18/2.6.19 feature

Driver Hardening

- Consists of things such as
  - Regularization of diagnostic messages
  - Improved APIs for managing and delivering driver diagnostic messages
  - Improved accounting/performance metrics
    - Use the proposed statistics APIs?
  - Fault injection framework and implementation
- There has been no visible activity on any of this in the past 1-2 years
- There is no fundamental objection to these features – someone needs to do the work to prepare an acceptable implementation