Kernel Summit 2008 Summary

Theodore Ts'o

THE LINUX FOUNDATION

Agenda

- What is the Kernel Summit
- Topics discussed during the 2008 Kernel Summit
- Conclusion, and how to get invited to the kernel summit.

History of the Kernel Summit

- **In 1998, Linux was "discovered" by the mass industry**
  - Articles in Fortune magazine
  - Many new Linux developers and many existing developers now funded to work on Linux full-time
- **Need for face-to-face meetings**
  - E-mail does not have the nuance of face to face discussions. Fierce arguments over e-mail can sometimes be quickly resolved in face-to-face meeting.
  - Previously developers could meet each other when they were invited to speak at conferences such as the Linux Kongress, but as the Linux developer community grew significantly in 1998-2001, this was no longer sufficient
- **So I organized the first Linux Kernel Summit in San Jose in 2001**

Kernel Summit Locations

- **First Kernel Summit was located in San Jose in 2001**
- **Subsequent summits were held just before the Ottawa Linux Symposium in Canada between 2002—2006**
- **In 2007, we moved the Kernel Summit to Cambridge, England, since roughly one-third of the kernel developers were traveling from Europe**
- **In 2008, we held the Kernel Summit in Portland, Oregon**
  - Helped to support a new conference, the Linux Plumber's Conference, which focused on low-level user space and their interfaces with the Kernel
- **In 2009 the Kernel Summit will be held in Tokyo, Japan**
  - Hopefully this will help to start a technical conference in Japan much like OLS and Linux.conf.au. Hopefully it will be attended by not just corporate developers but also from the hobbiest/amateur community.

How the kernel summit is organized

- I serve as the program committee chair; as the chair, I select the program committee, which is composed of senior Linux Kernel Developers.
- Logistical support was provided by Usenix from 2001—2006. In 2007, the Linux Foundation started providing the Logistical support.
- The program committee is responsible for issuing invitations to kernel developers and setting the agenda for the Kernel Summit

The Kernel Summit Invitation Process

- **An initial list of kernel developers assembled by analyzing people who have at least a critical threshold of Developer's Certification of Origin (i.e., “Signed-off-by” headers) in the Linux git repository**
  - This biases the list in favor of subsystem maintainers
  - This list is published on the kernel summit discuss mailing list
- **Anyone can propose additions to this list by nominating other developers or nominate themselves**
- **The program committee considers each developer on this list to decide who should be given an invite.**
  - Each program committee member submits a ballot ranking each potential invitee for suitability from 1-5. This is averaged and used as the basis of discussions on 2-3 conference calls.
  - Goal: invite a good cross-section of kernel developer subsystem maintainers
  - Goal: invite appropriate maintainers from multiple sides of controversial topics to be discussed at the kernel summit
  - 70-75 invitations will be issued, in 2-3 rounds of invites

Other Kernel Summit Attendees

- **Other ways that attendees can receive invitations**
  - Gold and Platinum sponsorship includes 1 or 2 sponsored invites. (Sponsorships are used to fund the kernel summit and to pay travel scholarships for developers who can not get their employers to pay for their travel.)
  - 5 developers from the MAINTAINERS list are selected by lottery (goal: so device driver authors can have a chance to attend)
  - Panel of end users (and in the past, representatives from database vendors and CPU architects) provides input to kernel developers

The Kernel Summit Discuss mailing list

- **Public mailing list which is established each year approximately 9 months before the summit**
  - New list is established each year (i.e., last year's list was ksummit-2008-discuss@lists.linux-foundation.org)
  - Announced on last year's discuss list and on the Linux Kernel Mailing List (LKML)
- **Purpose of the list**
  - Announcements for the nomination/invitation process
  - Suggestions for improving the logistics of the Kernel Summit
  - Suggestions for topics to be placed on the Kernel Summit Agenda

Kernel Summit Topics

- **Kernel summit topics are suggested on the discuss list**
  - Some topics are resolved via e-mail, either on the discuss list or on LKML
  - The agenda is not finalized until 3-4 weeks before the Kernel Summit
- **Approximately half of the topics are technical**
  - Generally when there is no consensus over multiple approaches to addressing a problem or new feature
  - Often the issue at hand impacts multiple subsystems
- **The other half of the topics tend to be process-oriented**
  - Questions of how the kernel development process can be fine-tuned
  - For example, the change in the 3 month 2.6 stable kernel releases was made during the 2004 Kernel Summit
- **“Hallway track” discussions**
  - Many important technical discussions take place informally between developers during the breaks between the formal sessions.

2008 Kernel Summit Topics

- Linux 3.0 (should we dump old code)
- Minisummit reports
- When should drivers be merged?
- Filesystem and block layer interaction
- Cross-subsystem issues
- The Linux Patchwork tool
- Standardizing initrd's
- Kernel quality and release process
- Kernel Tracing
- Documentation
- Kerneloops.org bug-fixing session
- More mini-summit reports
- Projects with large userspace components
- The new suspend/resume infrastructure
- Fixing the Kernel Janitors project

Linux 3.0

- **Proposal from Alan Cox (who was not present) to remove old legacy device drivers and/or system calls**
- **After discussion, conclusion that cost of maintaining old drivers/system code is minimal**
- **Linus was strongly against this idea**
  - "We should be proud that binaries from 1991 still run on modern kernels"
- **Cristoph Hellwig: "Would be a marketing disaster to create a major release that only removed features."**

Minisummit Reports

- **In-depth technical discussions which are specific to a single subsystem take place at Mini-summits**
  - These are usually scheduled at various Linux conferences
  - Relatively small; usually no more than a dozen developers or so
- **Mini-summits reported at the 2008 Kernel Summit**
  - Power Management
  - Wireless Networking
  - Containers
  - Virtualization
  - Networking

When should drivers be merged?

- **What quality level should be required before a device driver is merged into the kernel source tree?**
  - Many hardware vendors implement a Linux driver only because they have contracts which require it. Their commitment to a quality driver is sometimes not very high.
  - Once the driver is accepted, kernel developers have very little leverage to encourage them to fix the driver.
- **On the other hand, drivers can get fixed more quickly if the are in the mainline kernel; and a sloppy driver is better than no driver at all**
- **Conclusion**
  - If distributions are shipping the driver, then it should be in mainline
  - If hardware specific knowledge is required to improve the driver, that must come from the hardware vendor
  - Greg K-H, who has been maintaining the staging tree, will start merging drivers which are in the staging tree into mainline. Drivers will be marked experimental and will taint the kernel, until they can graduate as fully supported drivers.

The Patchwork Tool

- Paul Mackerras presented a tool which was developed by OzLabs (the IBM Linux Technology Center in Canberra, Australia), called Patchwork
- The tool monitors a mailing list and collects patches sent to the list to make sure they do not get lost.
- Used by the PowerPC maintainers, who have found it very useful.
- Many kernel developers thought Patchwork was very interesting were interested in trying it out.
- The SCSI and Ext4 subsystems volunteered to experiment using the tool to see how it worked in practice.

Common tools to build Initramfs

- **Initramfs is a bundle of userspace code which is run immediately after the kernel is booted to set up the system**
  - Resume from hibernation if needed
  - Load any block device drivers that might be necessary
  - Initialize software raid and LVM as appropriate
  - Mount the root filesystem
- **Every single distribution does it differently, and in most cases, badly.**
- **Changes in the kernel can break distribution-provided initramfs**
  - So users who try upgrading to a newer kernel.org kernel can find that the new kernel is incompatible with the distribution-built initramfs
- **David Jones is going to try to build a new facility for creating initramfs that is integrated into the kernel build tree.**

Kernel Quality and Release Processes

- **Kerneloops.org**
  - System which reports kernel stack dumps from community distributions and aggregates the reports into a web site
  - Bugs found by kerneloops.org very different from bugs reported by testers to LKML
  - At any point in time 60% of the oops reports are from the top 10 bugs. The top 25 bugs are responsible for 75% of the oops reports. Ergo, fixing a small number of bugs can have a big impact.
- **Regression Tracking**
  - Rafael Wysocki reported on statistics from tracking regressions (new bugs introduced during the development cycle).
  - New regressions can be fit to a logarithmic curve; regression fixes can be fit to a straight line. By plotting where the curves bisect it is possible to project when all of the regressions are fixed. The last few 2.6 kernels have been released 1-3 weeks before that point.
  - One problem which was identified is that regression fixes sometimes spend too much time sitting in linux-next and do not get merged into mainline quickly enough.

Release Timing

- **Matt Mackall suggested changing the release cycle by cutting the merge window in half, thus reducing the amount of change introduced, so that hopefully a new stable 2.6 kernel could be reduced in 6 weeks instead of 3 months.**
  - Require changes to be ready to merge and in the linux-next tree before the merge window opened
- **The idea was very attractive, and this was almost adopted by the kernel developers present. However, Matthew Wilcox pointed out that some regressions, in particular performance regressions take a long time to find and debug, and doubling the number of kernels to test would make it harder to find and fix performance regressions.**

Kernel Tracing

- **System Tap**
  - System Tap has been highly problematic
  - SystemTap requires tight integration with the kernel
  - But very few kernel developers have been able to use SystemTap
    - Focus on usability by enterprise end-users
    - SystemTap developers not kernel developers
    - As a result, not enough tapsets; SystemTap users have to be conversant with Linux kernel sources
  - Much frustration between SystemTap developers and Kernel developers
- **Kernel developers now starting to create their own simple tracing tools**
  - Ftrace based technology based on profiling hooks added by copmpiler
  - Linux Trace Toolkit
  - Addition of static tracepoints instead of depending on dynamic kprobes as used by SystemTap

Conclusion

- **Hopefully this presentation has given you a flavor of the sort of discussions which take place at the kernel summit.**
  - And what to expect for the 2009 kernel summit next year
- **How to get invited to the Kernel Summit**
  - Volunteer!
    - Review patches
    - Help fix bugs
    - Become a subsystem maintainer
  - Do more than just work assigned by your employer
    - Some engineers are allowed by their company to spend part of their work day improving the kernel.
    - Others do this work on their own time.

Legal Statement

- This work represents the view of the author(s) and does not necessarily represent the view of IBM or of the Linux Foundation.
- IBM is a registered trademark of International Business Machines Corporation in the United States and/or other countries.
- Linux is a registered trademark of Linus Torvalds.
- Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.
- Other company, product, and service names may be trademarks or service marks of others.

THE LINUX FOUNDATION