Linux Kernel 2.6 – New Features III: Networking

Jerry Cooperstein

Axian Inc

coop@axian.com

@ OGI 1/07/03

On-line

Download these talks from:

http://www.axian.com/learning.php

Today

- General Overview of 2.6 Features, and Status
- Review of First Two Lectures
- Networking Enhancements
- Other New Items:
  - New Module Implementation and Utilities
  - New System Call Mechanism

Linux Kernel 2.6

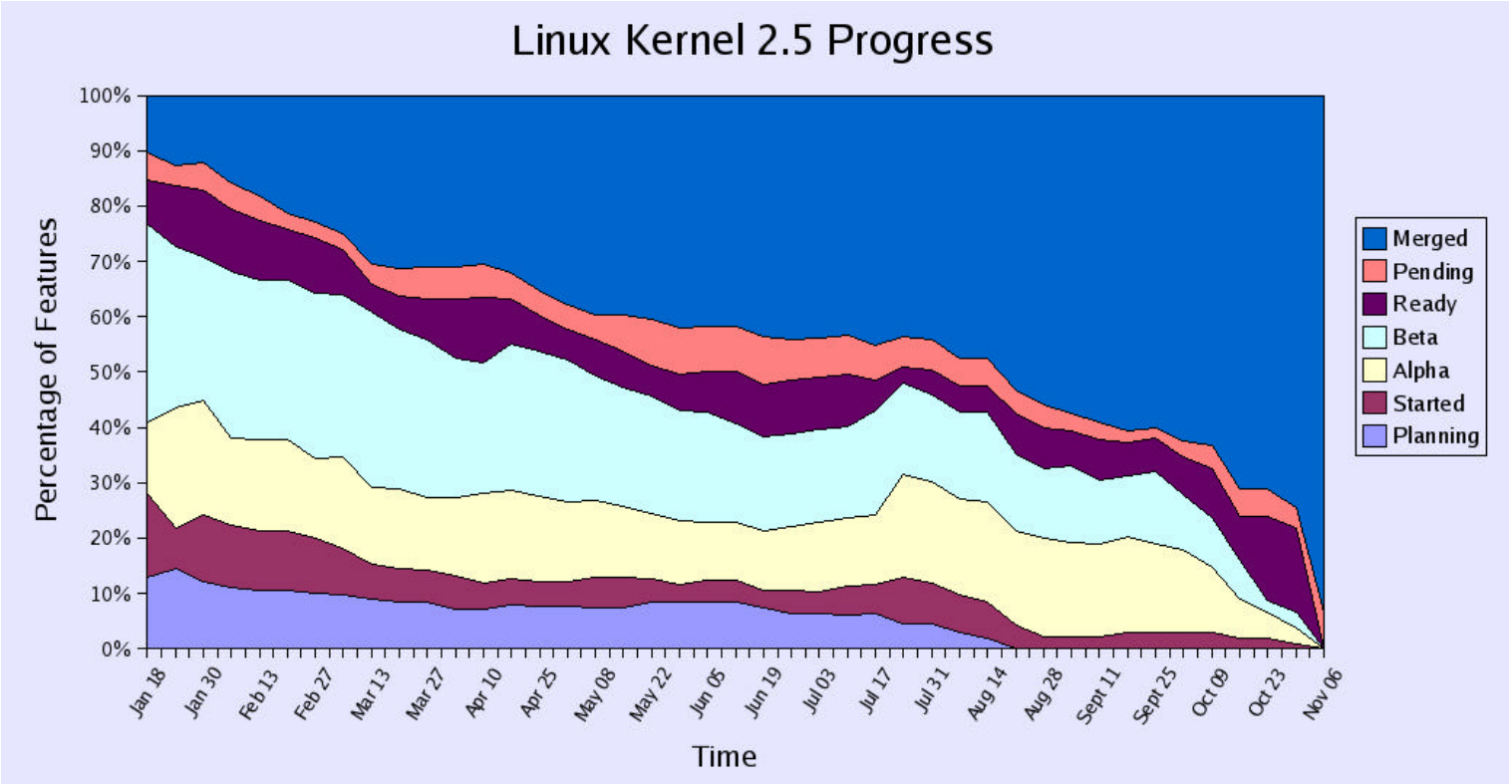
- Feature Freeze: Halloween 2002
- Better Performance, Especially on **SMP**
- Better Scalability
- Better I/O Subsystem, New Filesystems
- Many New Hardware Drivers
- New Platforms
- Many Features Tested as 2.4 Patches

Latest Status

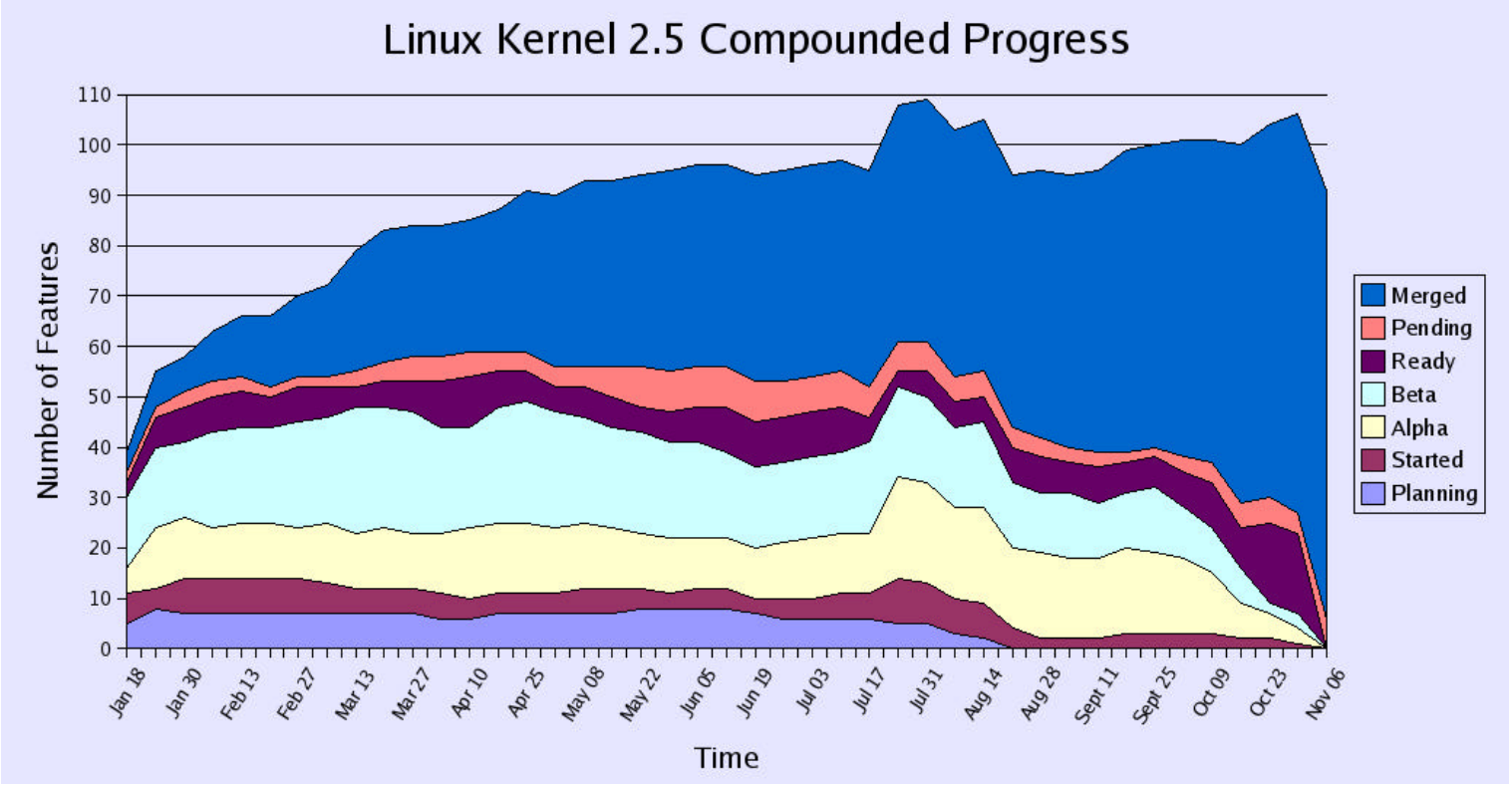
- Guillaume Boissiere maintains a status report, updated weekly at:

http://kernelnewbies.org/status/latest.html

Boissiere's Status Report: Progress



Boissiere's Status Report: Compounded Progress



New Features: General (Review of Lecture I)

- Preemptable Kernel
- O(1) Scheduler
- New Kernel Device Structure (**`kdev_t`**)
- Improved Posix Threading Support (NGPT and NPTL)
- New Driver Model & Unified Device Structure

New Features: General (Review of Lecture I)

- Faster Internal Clock Frequency
- Paring Down the BKL (Big Kernel Lock)
- Better in Place Kernel Debugging
- Smarter IRQ Balancing
- ACPI Improvements
- Software Suspend to Disk and RAM

New Features: General (Review of Lecture I)

- Support for USB 2.0
- ALSA (Advanced Linux Sound Architecture)
- LSM (Linux Security Module)
- Hardware Sensors Driver (lm-sensors)

New Features: Architectures (Review of Lecture I)

- AMD 64-bit (x86-64)
- PowerPC 64-bit (ppc64)
- User Mode Linux (UML)

New Features: General (Review of Lecture II)

- CPU Clock and Voltage Scaling
- Setting Processor Affinity
- Improved **NUMA** Support
- Reverse Mapping VM System (**rmap**)
- Large Page Support
- High Resolution Posix Timers
- New Serial Port Driver Rewrite and API

New Features: Journalling Filesytems (Review of Lecture II)

- Ext3 (already in 2.4)
- ReiserFS (already in 2.4)
- JFS (IBM)
- XFS (SGI)

New Features: I/O Layer (Review of Lecture II)

- Rewrite of Block I/O Layer (BIO)
- Asynchronous I/O
- IDE Layer Update
- ACL Support (Access Control List)
- New NTFS Driver

Removed Features (Review of Lecture II)

- Export of **`sys_call_table`**
- End of **Task Queues**

New Features: Networking (Lecture III)

- Network Programming Difficulties
- Linux Network Programming History
- Network Programming Layers in Linux

New Features: Networking (Lecture III)

- TCP Segmentation Offload
- SCTP Support
  (Stream Control Transmission Protocol)
- Bluetooth Support (no longer experimental)
- NAPI (Network Interrupt Mitigation)
- Zero-Copy Networking and NFS
- NFS v4

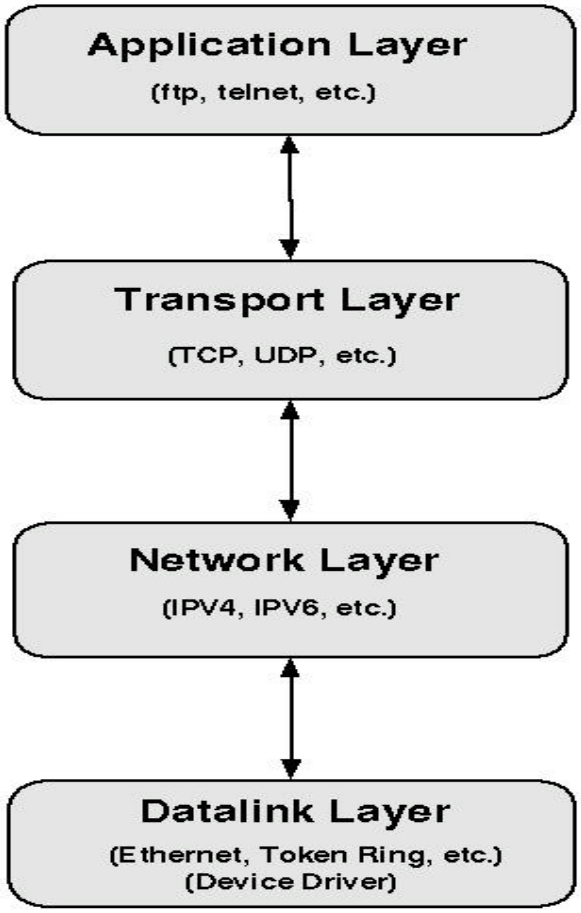
Network Programming Difficulties

- Complexity
- Historical Standards (and Baggage)
- Unpredictability of Traffic and Topology
- Hardware Variety
- Security
- Need to Interface with other OS's

Linux Network Programming: History

- Based on BSD **Sockets**
- All Linux network code and sockets API developed from scratch
  - Not based on Unix source code or AT&T derived code
- Most important standard: Posix 1003.1g
  - http://www.pasc.org/standing/sd11.html

Network Layers



Networking Programming Layers in Linux

- Layered Approach:
  - Application (clients, servers (ftp, http))
  - Transport (data encapsulation (TCP, UDP))
  - Network (routing, data transmission (IPV4))
  - Datalink (device(NIC), method, (Ethernet))
- Each layer independent
  - Object-Oriented Methods

TCP Segmentation Offload (TSO)

- TCP Segmentation:
  - Split large chunk of data into packets
  - Construct headers
  - Perform checksumming
  - Copy data and transfer to NIC

TCP Segmentation Offload (TSO)

- Offload:
  - Pass template headers
  - Pass one large data buffer
  - NIC handles Segmentation
  - One big I/O operation instead of many small ones
  - Must have hardware support in NIC

TCP Segmentation Offload (TSO)

- Results for Intel E1000:
  - Tx/Rx file send long (bi-directional Rx/Tx)
    - w/o TSO: 1500Mbps, 82% CPU
    - w/ TSO: 1633Mbps, 75% CPU
  - Tx TCP file send long (Tx only)
    - w/o TSO: 940Mbps, 40% CPU
    - W/ TSO: 940Mbps, 19% CPU
- **http://lwn.net/Articles/9129 (scott.feldman@intel.org)**

TCP Segmentation Offload (TSO)

- Results:
  - Not much increase in throughput
    - Bandwidth already saturated
  - Up to halving of CPU usage
    - Thus, a worthwhile optimization

Stream Control Transmission Protocol Support (**SCTP**)

- Message oriented, reliable transport protocol
- Direct support for **multi-homing**
- Runs on top of Internet Protocol (**IPV4,6**)
- Connection-oriented data delivery
- Congestion Control
- Intro article by Daisy Chang and Jon Grimm (IBM)
  http://www-124.ibm.com/linux/presentations/lwce2002/Chats/SCTP/SCTP-LWE.pdf

Stream Control Transmission Protocol Support (**SCTP**)

- In addition to **TCP** features:
  - Message Framing
  - Ordered and Unordered Message Delivery
  - Multi-streaming
  - Multi-homing
- Accepted in Kernel 2.5.33
- **http://lksctp.sourceforge.net**

Bluetooth Support (No longer experimental)

- **Bluetooth** standard for low-cost, short-range, small form factor radio-linked PC's, phones, and other devices
- Wireless **USB** and **serial cable** replacement
- TCP/IP can be done using ppp
- Not a complete wireless LAN solution such as 802.11; a cable replacement

Bluetooth Support (No longer experimental)

- In Linux kernel since 2.4.6, but was ***experimental***, i.e., unsafe.
- Upgraded to safe in 2.5.14
- Uses the **BlueZ** protocol stack:
  - **http://bluez.sourceforge.net**
- Handles multiple devices and connections simultaneously

NAPI
(Network Interrupt Mitigation)

- **NAPI** stands for **NEW API**
- Improved handling of high network loads
- Work of Jamal Hadi Salim, Robert Olsson and Alexey Kuznetsov
  - **http://www.cyberus.ca/~hadi/usenix-paper.tgz**
- Accepted in Kernel 2.5.7

NAPI
(Network Interrupt Mitigation)

- Helps eliminate **congestion collapse** due to **interrupt livelock:**
  - High interrupt rate overwhelms CPU
  - Packets can't be processed in timely fashion
  - Backlog queue gets saturated
  - Packets get dropped

NAPI
(Network Interrupt Mitigation)

- Modern NIC's have **ring** of **DMA** buffers
- Incoming packets go into next buffer
- Normally one interrupt per incoming packet
- OK for low loads to deal with one at a time

NAPI
(Network Interrupt Mitigation)

- **NAPI** combines **interrupt** and **polling** mechanisms
- Interrupts are disabled after first one
- Then ring is **polled** periodically and packets are pulled as necessary
- When no new packets, interrupts restarted
- Packet backlog queue eliminated

NAPI

(Network Interrupt Mitigation)

- Requires changes to network drivers to use
- Older drivers will still work
- Most important for high speed (Gigabit) NIC's
- For low loads may be slightly slower due to the cost of polling

Zero-Copy Networking and NFS

- Can do in one system call with:

```
sendfile(sd, fd, &offset, nbytes);
```

- Only one system call
- Since kernel 2.4, **zero-copy** methods have been used for **`sendfile()`**

Zero-Copy Networking and NFS

- Suppose you want to send data from a file out on the network:

```
fd = open("/tmp/file", O_RDONLY);
sd = socket(PF_INET, SOCK_STREAM, 0);
connect (sd, &serv_addr, addrlen);
read(fd, &buf, nbytes);
write(sd, &buf, nbytes);
```

- This requires two system calls and an extra copy; inefficient, wastes memory

Zero-Copy Networking and NFS

- Zero-Copy **NFS** appeared in 2.5.47
- Work of Hirokazu Takahasi of VA Linux (Japan)
- Can use up to 100 % of CPU on **SMP** machine
- So far applies only to **TCP**; **NFS** over **UDP** underway

NFS Version 4

- From RFC 3010, has:
  - Improved access, performance, on Internet
  - Strong security, built-in negotiation
  - Better cross-platform interoperability
  - Permits protocol extensions
  - Operation **coalescing**
  - Integrated file locking
  - Full support of Windoze file sharing semantics

NFS Version 4

- Fully included since kernel 2.5.43
- Open Source Project headed at: **http://www.citi.edu/projects/nfsv4**
- See **... /OLS2001/index.html** for details
- Also **http://www.nfsv4.org**

New Features: Module Implementation and Utilities

- Module loading, unloading, etc, has been moved (mostly) back into the kernel
- New, thinner, set of utilities (**insmod, rmmod, depmod, modprobe**)
  - Older versions still remain (e.g., **insmod.old**)
- Came **after** feature freeze
- Work of Rusty Russel: download from **ftp://ftp.kernel.org/pub/linux/kernel/people/rusty**

New Features: System Call Mechanism

- System calls on P4 are slower than for earlier CPU's
- Old mechanism:
  - use **`int 0x80`** instruction to generate exception
- New mechanism:
  - use **`sysenter`**, is much faster

New Features: System Call Mechanism

- Not all **x86** CPU's support **`sysenter`**
- Monkeys with some CPU registers
- Requires support in **C** library (**glibc**)
- Hard to use with more than 5 arguments
- Speedup:
  - Pentium 4: factor of 2
  - Pentium 3: factor of 1.2

Upcoming Linux Programming Classes at Axian

- RHD143 (Linux Programming Essentials)
  - **April 7-11**
- RHD221 (Linux Device Drivers)
  - **April 14-18**
- RHD236 (Linux Kernel Internals)
  - **April 21-25**
- Linux Kernel Network Programming ***(New)***

Upcoming Linux Programming Classes at OGI

- Linux Kernel Internals
  - **March 17–23, June 16-20**
- Linux Device Drivers
  - **April 14–18, July 14-18**
- Linux Kernel Network Programming ***(New)***
  - **February 10–14, May 12–16, August 11-15**

Upcoming PLUG Meetings

- Thursday, Feb 6, 7PM at PSU, Smith Memorial Center Room 298
- **ALSO:** Monthly PLUG Linux Clinic, Saturday, Jan 18, 1 - 4 pm, at Riverdale HS, 9727 SW Terwilliger Blvd
  `http://server.riverdale.k12.or.us/~danh`