Introducing FreeBSD 7.0

Kris Kennaway
The FreeBSD Project
kris@FreeBSD.org

October 20, 2007

Introducing FreeBSD 7.0

- FreeBSD 7.0 will be the next release of FreeBSD, and the first major release in 2 years.
- Due out some time later this year (currently in pre-release and available for testing)
- FreeBSD 7.0 brings major changes to the BSD and open source operating system landscape.

Outline:

I The SMPng project: a 7 year journey
  - "Symmetric Multi-Processor, next generation"

II Major new features debuting in FreeBSD 7.0

III What the future holds for FreeBSD

Part I: The SMPng project: A 7 year journey

> *"Last week, approximately 20 BSD developers got together and discussed how to move FreeBSD's SMP support to the next level. Our effort will be largely based on the work that has been done in BSD/OS, which should make things go much more smoothly than they otherwise might, but we still expect -current to be destabilized for an extended period of time."*

– Jason Evans, email to freebsd-current list, 19 June 2000

Multiprocessor support, old and new: FreeBSD 4.x

FreeBSD 4.x is a single-threaded kernel with limited multiprocessor support.

- Able to run user code on multiple processors
- Only one process at a time can execute in the kernel ("Giant lock" around entire kernel)
- Device interrupts may be processed in parallel, subject to some constraints

The historical BSD kernel architecture worked very well for single-processor systems. It fundamentally does not scale to multi-processor systems, which are now becoming universal.

Multiprocessor support, old and new: The SMPng project

Goal: Re-design the FreeBSD kernel as a multi-threaded system, for “next generation” SMP support

- Multiple CPUs must be able to execute kernel code in parallel
- Serialization of shared data structures via various locking primitives
- Balance the performance needs of Uni-Processor (UP) and SMP systems (not always different needs)
- A major challenge...
- ...now complete

SMPng and the Universal Development Model

The SMPng project followed a simple 3 step process:

SMPng, step 1: First, make it work; FreeBSD 5.x

FreeBSD 5.0-5.2.1 (2003-01-17 - 2004-02-22)

- Debut of the new architectural model for symmetric multiprocessor support in FreeBSD.

FreeBSD 5.3 (2004-11-06), 5.4 (2005-05-09)

- The fundamental architectural changes were largely in place
- Some initial progress with kernel parallelism by 5.3 and 5.4 (network stack, virtual memory, ...)

SMPng, step 2: Then make it work *well*; FreeBSD 6.x

FreeBSD 6.0 (2005-11-01), 6.1 (2006-05-08), 6.2 (2007-01-15)

- Stabilized the work of the 5.x branch
- Performance benefits from subsequent development work
  - e.g. Virtual File System (VFS) and Unix File System (UFS) now allow parallel access
- Large parts of the kernel may now operate in parallel, with significant performance gains on many common workloads

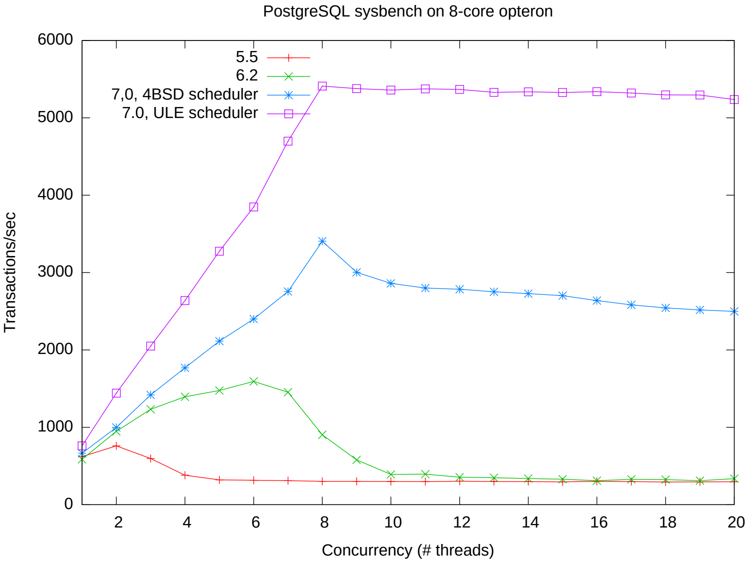
SMPng, step 3: Then make it **fast; FreeBSD 7.0**

- The goals of the SMPng project have been achieved
- The FreeBSD 7 kernel is a fully parallel system
  - The “Giant lock” is no longer present on almost all possible workloads
- Major shift of focus from correctness to optimization, with impressive results

A case study: SQL database performance

- "Online transaction processing" benchmark; `/usr/ports/benchmarks/sysbench`
- Complex transaction-based queries
- Read-only: no disk access to avoid benchmarking disk performance
- Clients and servers on the same system
- PostgreSQL 8.2.4 (process-based + System 5 Inter-Process Communication (IPC)
- MySQL 5.0.45 (thread-based)
- Test hardware:
  1. 4 * 2-core Opteron (amd64 mode)
     - 2.2GHz CPUs, 4 GB RAM
  2. 2 * 4-core Xeon E5320 (i386 mode)
     - 1.8GHz CPUs, 3.5GB RAM

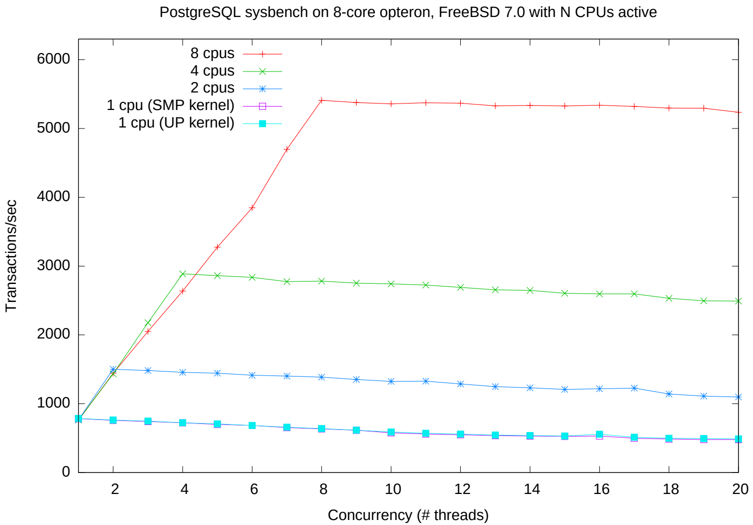
FreeBSD PostgreSQL performance: 5.5, 6.2 and 7.0



Performance of PostgreSQL

- The ULE scheduler has significantly better performance than 4BSD (historical BSD scheduler)
  - 4BSD will probably remain the default in 7.0, changing in 7.1
  - You can easily switch to ULE by recompiling your kernel
- ULE has linear scaling to 8 CPUs and minimal degradation at higher loads; close to ideal performance from the hardware.
- No significant performance problems in the FreeBSD 7 kernel on this workload

FreeBSD 7.0: Scaling with varying number of CPUs

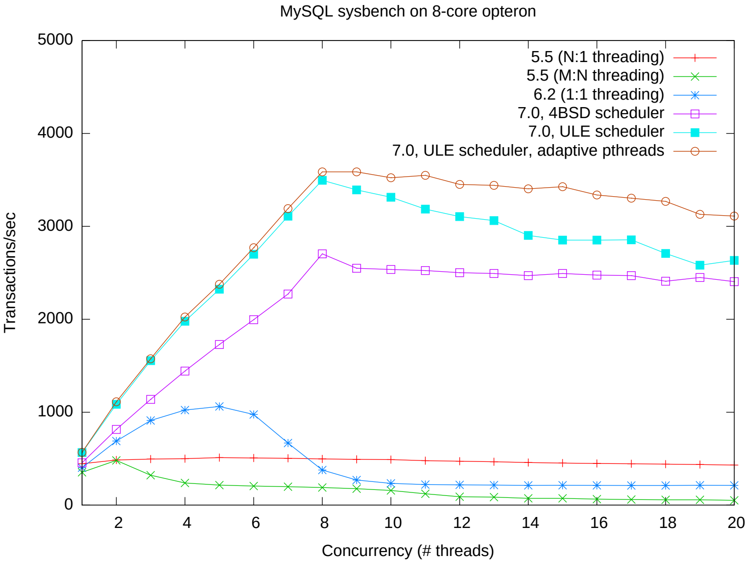


FreeBSD 7.0: Scaling with varying number of CPUs (2)

Notes:

- Performance from 1 → 2 → 4 → 8 CPUs scales linearly
- Consistently stable performance at high loads
- No significant overhead from SMP kernel on UP system

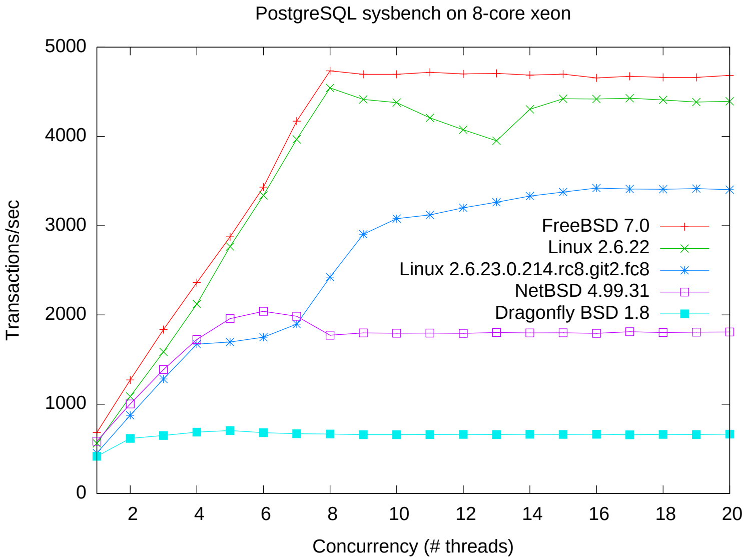
FreeBSD MySQL performance: 5.5, 6.2 and 7.0



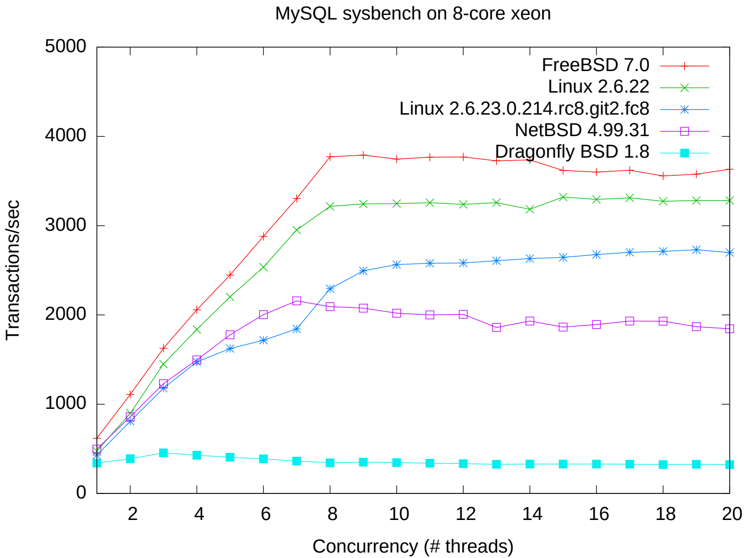
Understanding MySQL performance

- Again, linear scaling up to 8 client threads (= # CPUs)
- The degradation above 8 threads is due to scaling problems within MySQL (not a FreeBSD kernel issue)
- Heavy contention on pthread mutexes within the application
- The graph includes a libpthread patch that reduces the severity of this problem
  - a hack that is also present in glibc, and is used by MySQL
  - the patch will be in 7.0-RELEASE, but it is ultimately an application problem
- NB: On this benchmark PostgreSQL is 35% - 45% faster than MySQL at all loads

FreeBSD vs other operating systems: PostgreSQL



FreeBSD vs other operating systems: MySQL



Comments on other operating systems

Linux

- Some improvement since we initially publicized our benchmarks 6 months ago
- With the 2.6.20.1 kernel, performance drops rapidly above 8 threads and approaches single threaded performance above 14 threads
- 2.6.20.1 still very recent (2007-02-20), so many Linux distributions still shipping older kernels
- Publication of FreeBSD 7.0 performance comparisons motivated improvements in Linux
- 2.6.22 is still 15% slower than FreeBSD 7.0
- The new CFS scheduler in 2.6.23 is "Completely Fair"...to FreeBSD

NetBSD

- Good initial progress on SMP support

Part II: New features debuting in FreeBSD 7.0

FreeBSD 7.0 brings updates to almost every part of the operating system (more than 18000 code changes), as well as several major new features.

1. Filesystem/storage
2. Networking
3. New CPU architectures
4. Security systems
5. User applications
6. Developer tools

Filesystem and storage subsystem changes

New filesystems

- ZFS
  - Sun's amazing new filesystem moves the goalposts. Stay tuned for more in the presentation from Pawel.
- unionfs: overlay multiple filesystem hierarchies into one. Broken for many years but now usable again.
- XFS support (read-only)
- CODA distributed filesystem support fixed
- UFS quotas are now parallelized
- NFS client and server parallelized
- Performance improvements for NFS client

Storage subsystem changes

- SCSI layer (CAM) is now parallelized, including many drivers. Performance benefits for SCSI device access.
- iSCSI initiator (in base system) and target (in ports), allowing remote exporting and local mounting of SCSI devices over TCP/IP

New GEOM (pluggable storage layer) modules

- gjournal; block level journalling provider (can be used with UFS for journalling support)
- gvirstor; virtualized storage provider (create a huge disk image sparsely populated with disks, add more later)
- gcache; read cache for storage layers with small request sizes
- gmultipath; support for multiple paths to the same storage provider (fibre channel, etc)
- gpart; virtualized partitioning support (GPT, APM, ...)

Network stack changes

- Complete elimination of giant lock from network stack
- On-going cleanup and development work
- Socket buffer automatic sizing; dynamically responds to network conditions for improved throughput
- SCTP (Stream Control Transmission Protocol)
- Migration from KAME IPSec to Fast IPSec
  - Improved performance
  - Hardware acceleration with cryptographic accelerators
  - Both IPv4 and IPv6

Network stack changes (2)

- Direct dispatch of inbound network traffic
  - Avoids context switching, improves CPU cache locality, allows concurrency
  - Significant performance benefits on many workloads
- Optional in-kernel Just-In-Time compiler for Berkeley Packet Filter (BPF) programs (tcpdump, etc)
- In-kernel Network Address Translation (NAT) modules for natd(8)
- Link aggregation (create virtual interfaces for fault tolerance and higher capacity)
- Rapid spanning tree protocol support

Network drivers

- Support for commonly encountered 10 gigabit ethernet drivers: Chelsio (cxgb), Intel (ixgbe), Myricom (mxge), Neterion (nxge)
- Transmit Segmentation Off-load (TSO)/Large Receive Off-load (LRO); off-load send/receive into the ethernet driver
- New devices supported

Wireless (802.11); much improved in 7.0

- Wireless 802.11 layer is stable
  - high power ath cards (Senao, Ubiquiti, Wistron)
  - 900MHz ath cards (Ubiquiti, Zcomax)
  - ath (Atheros), iwi, ral (Ralink), ural (RT2500USB) drivers are high quality
- New drivers
  - rum (Ralink RT2500USB, RT2601USB)
  - Intel wireless drivers: ipw (Intel PRO/Wireless 2100), iwi (2200BG/2225BG/2915ABG)
    - Works out of the box
  - ZyDAS ZD1211/ZD1211B

Wireless (802.11), continued

- WPA (Wifi Protected Access) support stable
- New scanning support (background scanning, roaming)
- Atheros protocol extensions
  802.11n support (forthcoming standard)
  - higher performance: up to 135 Mb/sec, channel bonding, improved range, etc
  - drivers not yet committed
- Preparation for future changes (virtual access points, etc)

New CPU architectures

- Improved support for ARM architecture
  - Improved AT91RM9200 (Atmel) support
  - support for Avila Gateworks Xscale boards was added, including a rewrite of the Intel code
  - permission from Intel to bundle μ-code
  - Boot loader can load from Secure Digital (SD) flash cards
  - FreeBSD/ARM used as the basis for growing number of embedded devices
- Sun Ultrasparc T1 (preliminary)
  - 8 cores, 4 threads per core = 32 logical CPUs per package
  - A very interesting new CPU architecture, and one to watch in the future
  - T2: 8 threads * 8 cores = 64 logical CPUs per package!
- X-box!

Security subsystems

Audit subsystem

- fine-grained, configurable logging of security-relevant events
  - System calls, application and user space activities
- Now available by default in GENERIC kernel
- Originally developed for Mac OS X, ported and enhanced. A nice example of code-sharing in the Apple → FreeBSD direction
- Builds on the other advanced security features developed by the TrustedBSD project for FreeBSD

priv(9) API

- common interface for kernel privilege checking
- Privilege model can be modified by Mandatory Access Control (MAC) modules

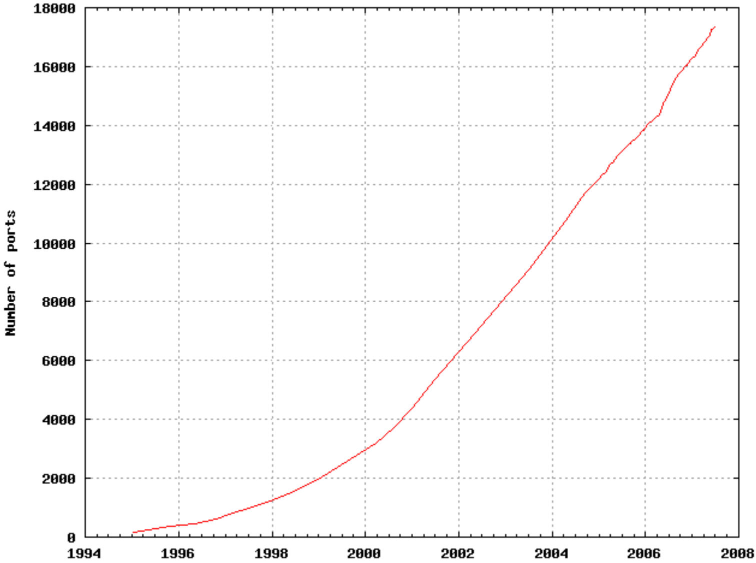
User-level changes

- Many updates to system applications and utilities
- cached: caches queries to nsswitch ("name service switch": user/group/host lookups) for improved performance

Ports collection

- Currently contains 17692 ported third-party applications (1774 more than 6.2)
- Major changes since 6.2:
  - X.org 7.3 (many improvements, e.g. working composite support for improved visual effects)
  - KDE 3.5.7
  - GNOME 2.18.3
  - More than 24000 other changes and updates

Growth of FreeBSD Ports Collection



Performance

- Performance optimizations throughout the system
- The ULE scheduler is now recommended instead of the historical 4BSD scheduler
  - Better interactive performance on desktop systems
  - Significantly better performance on SMP systems
  - 4BSD will remain the default scheduler in 7.0 to be conservative, but likely to switch for 7.1
- If you find a workload that FreeBSD 7.0 performs poorly on, we want to hear about it!

Other kernel changes

- Partial linux 2.6.16 emulation support (not enabled by default)
- Support for Message Signaled Interrupts (MSI) and Extended Message Signaled Interrupts (MSI-X)
- IPMI (Intelligent Platform Management Interface); monitoring system hardware.
- Improved support for legacy-free hardware (e.g. MacBook pro)
- FireWire support for the boot loader
- Asynchronous I/O (AIO) support is parallelized
  - used by e.g. qemu
- New pseudo-tty system, allocates on demand without built in limits, and without requiring root privilege

Developer tools/internals

- GCC 4.2.1
- Improvements to hwpmc (CPU performance counters)
- symbol versioning added to several libraries
- New scalable malloc(3) (jemalloc)
- Optimized kernel locking primitives (sx, rwlocks)
- POSIX message queues
- contigmalloc(9) with buddy allocator
- kernel malloc(9) red zone debugging support
- Improved kernel lock profiling infrastructure
- mini-dumps

Part III: What the future holds for FreeBSD

- The FreeBSD 7.x tree has been branched as part of the release process, and development is also beginning on FreeBSD 8.0-CURRENT, due some time in 2009 (maybe)
- Some of the features that seem to be lurking on the horizon:
  - Continued performance optimization, also targetting 16-core systems (AMD/Intel)
  - Improved network performance on parallel workloads
  - Improved filesystem performance
  - Virtualization support: xen, network stack virtualization, ...
  - BLUFFS: BSD Logging Updated Fast File System. UFS with filesystem-level journalling.
  - Serial Attached SCSI, SATA integrated under CAM (storage layer also used for SCSI)
  - DTrace support from Sun; powerful and extensible debugging and system analysis framework
  - Stuff we haven't even thought of yet!

Summary

- FreeBSD 7.0 brings FreeBSD back to the forefront of OS performance on modern hardware (it's good to be back).
- Providing advanced features not available in other open source operating systems
- An attractive platform for both high end and embedded hardware.
- A new foundation for the years ahead.

Where can I get FreeBSD 7?

- Download the latest snapshot (pre-release) ISO (CD image):

  `ftp://ftp.freebsd.org/pub/FreeBSD/snapshots/200710/`

- Join the `freebsd-stable@FreeBSD.org` mailing list to participate in testing of the 7.0 pre-release and beta series