WWW.DRAGONFLYBSD.ORG

What Is DragonFly?

- Forked off of FreeBSD-4.x ~2 years ago.
- A ground-up reworking of traditional BSD programming models to achieve our goals.
- A new approach to SMP scaleability in the BSD world.
- Designing for long-term code maintainability and robustness.
- Designing for long-term forwards and backwards compatibility.
- Uber Goal is transparent, natively-supported, fully cache coherent single-system-image clustering, with all the trimmings.
- A parallel effort is being undertaken to improve userland.

DragonFly
Kernel Side Programming Methodologies

- A Threaded, cpu-localized approach to parallelism
- Very little differentiation between UP and SMP coding practices.
- Minimal UP vs SMP tradeoffs.
- Concept of natural ownership of data structures (either by a cpu or by a thread) for which no synchronization is necessary to access.
- Focus on greatly simplifying the programming model.
- But the best algorithms are not always quite that simple.
- Synchronous cross-cpu data access more expensive, but there are plenty of ways to remove the issue from the critical path and the APIs can be very simple.

DragonFly
User side system methodologies

- Target mainstream installs, assume that large scale customization will use a different installation and upgrade mechanism.
- Make a clear distinction between files owned by the installation, and files that can be adjusted by the system operator to simplify the upgrade regimen and reduce confusion.
- Ultimate goal is to provide production-level support by emphasizing forwards and backwards compatibility.
- Biggest single problem is the package management system.
- Second biggest problem is kernel data structure visibility in userland.
- Third biggest problem is kernel syscall compatibility over time.

DragonFly
Packaging System Requirements

- Our Requirements:
  - Track requested installs verses side effects.
  - Remove side effects that are no longer referenced.
  - Aggressive proving of build and run-time dependancies.
  - Multi-version installs not just supported, but required.
  - Ability to easily compartmentalize environments.
  - Ability to downgrade as well as upgrade.
  - Primary focus on binary packages.
  - What about value-add features such as automatic live-service updates?
- No current packaging system does it all properly, but a desire to leverage existing work. Use of VarSyms looks promising.

DragonFly 1.2 Released
Now that you know what it is...

- 1.2 Released April 8 2005:
  - 6 month release schedule. 1.0, 1.2, 1.4, etc.
  - Now officially branching our releases.
  - Release branches will contain only bug and security fixes.
- Live Filesystem based Releases a huge success:
  - Allows pre-testing without messing with the hard drive.
  - Complete base system, MFS mounts for ease of use, recovery, etc.
  - No sources, ports, and only a few pre-built packages are on the CD.
  - Netboot Server feature.
  - Several ways to customize or auto-install.
  - A quick install & reboot has advantages.
  - Room to grow (230MB).

The DragonFly 1.2 Release

- This will be the last release to use the Big Giant Lock in the critical pat
- All recent work stabilized, network and namecache work in particular.
- All major subsystem threading goals reached except for VFS threading.
  - Journaling backend threaded (not ready for prime time yet).
  - Interrupt threading proved out and working well.
  - Network protocol stack threading and cpu localization proved out.
  - VFS ops can now be issued by pure threads.
  - VFS no longer needs a proc, except when dealing with user UIOs.
  - VFS precursor work for per-mount threading about 80% complete.
- NFSv3 passes fsx tests, again (twas broke for a long time).
- NFSv3 bug and performance fixes for TCP mounts.
- NDIS support from FreeBSD integrated.

The DragonFly 1.2 Release

- A ton of driver updates. Improved USB keyboard handling.
- More kernel subsystems using [M]SFBUFs, buffer cache remains.
- Thread Local Storage (TLS) support implemented.
- Userland thread libraries are far better abstracted, but needs more wor
- Minix MINED editor brought into /bin for single-user mode (37K)
- VM_MAP data structure now indexed with a red-black tree.
- ALTQ and PF (Packet Filter) have been integrated.
- Gauntlet thrown down, all kernels now build with full debug info.
- MBUF allocator rewritten to use the slab allocator but
  is not yet cpu-localized.
- Core callout() infrastructure rewritten and cpu-localized.

The DragonFly 1.2 Release
Networking

- TCP pure window updates greatly reduced:
  - No longer sends silly window updates when the receiver knows that the sender still has plenty of buffer space to send data in.
- Experimental TCP ACK aggregation for fast (GiGE) LANs:
  - Takes advantage of packet aggregation done by ethernet hardware.
  - Greatly reduced (4:1) packet rate in the ACK direction at full bore.
  - Greatly reduced TCP processing protocol overheads on both ends.
  - Timeliness of ACKS and window updates not sacrified.
  - Disableable for very large bandwidth-delay products (WAN/SAT). (due to smaller effective exponent for congestion window expansion)
  - Load Beneficial rather then Load Adverse.

The DragonFly 1.2 Release
Networking

- TCP SACK support is well tested and considered very stable now.
- TCP header prediction is fixed (other BSDs check your hit rates!) (broken on and off since inception, noted by Stevens).
- Fast tail-append implemented for socket buffers for TCP and UDP, following other BSDs.
- Replicated (per-cpu) wildcard socket topology now considered stable.
- A ton of precursor cleanup work on the route table in preparation for per-cpu replication has been done.
- MTU Discovery has been fixed, possibly broken on other BSDs too. Other BSDs should make sure that a low default MSS isn't masking a problem. Known broken in FreeBSD-4, known working in FreeBSD-6.

DragonFly Kernel Subsystems
Major Work In-Progress

- Converting the buffer cache clean/dirty lists into red-black trees.
- Formalize a new algorithm for handling list ripouts during scans that block.
- Implement partial fsyncing by the filesystem syncer.
- Get rid of the write_behind heuristic that kills us on so many benchmarks.
- Optimize IPI Messaging (avoid unnecessary IPIs and add a passive send variant).
- Make malloc/free (the slab allocator) 100% MP safe by moving the BGL past the slab layer and into the VM layer.
- Per-cpu localization of the mbuf allocator to make it 100% MP safe.
- Per-cpu localization of the route table (route table replication).

DragonFly Kernel Subsystems
Major Work In-Progress

- Per-cpu localization of the IPF (packet filter) module.
- Turning off the BGL in the network protocol stack threads.
- Turning off the BGL in the NETIF interrupt handlers.
- Add POSIX signal sharing and other mechanisms required to support POSIX threading.
- Major rev bump for all shared libraries to support ABI changes.
- Add a Kernel syscall library layer which runs in userspace and handles ABI compatibility issues rather then building system calls into libc and creating a mess inside the kernel with renumbered system calls.
- Continued investigation of packaging systems (such as pkgsrc)

DragonFly Kernel Subsystems
Sampling of Goals for 1.4

- Threaded VFS (encompasses a number of related sub-projects).
- Remove the Big Giant Lock (BGL) from all remaining critical code paths.
- Come up with a Solution to the BIOS interrupt misrouting problem.
- Complete the POSIX thread library project and any remaining TLS issues.
- Decide on a packaging system.

And Ultimately

- A cache coherency layer above the VFS layer to support clustering.
- Process Migration.
- Single System Image clustering support with all the trimmings.
- Port to other architectures.

The DragonFly BSD Project
April 2005

Joe Angerson
David Xu
Matthew Dillon
Craig Dooley
Liam J. Foy
Robert Garrett
Jeffrey Hsu
Douwe Kiela
Sascha Wildner
Emiel Kollof
Kip Macy
Andre Nathan
Erik Nygaard
Max Okumoto
Hiten Pandya
Chris Pressey
David Rhodus
Galen Sampson
YONETANI Tomokazu
Hiroki Sato
Simon Schubert
Joerg Sonnenberger
Justin Sherrill
Scott Ullrich
Jeroen Ruigrok van der Werven
Todd Willey

and Fred -->

WWW.DRAGONFLYBSD.ORG