4.9pt

The Road Map for 5-STABLE

The FreeBSD Release Engineering Team

$FreeBSD: doc/en_US.ISO8859-1/articles/5-roadmap/article.sgml,v 1.30 2007/09/17 21:05:56 keramida Exp $

This document is now mostly of historical value. It presented a roadmap for the development of FreeBSD’s `RELENG_5` branch. It was originally written in February 2003 (between the 5.0 and 5.1 releases), and was intended to provide a plan for making the `RELENG_5` branch “stable”, both in terms of code quality and finalization of various APIs/ABIs. For a different perspective, the article “Choosing the FreeBSD Version That Is Right For You” (http://www.FreeBSD.org/doc/en_US.ISO8859-1/articles/version-guide) may be of interest. The version-guide article was written in August 2005 (two and a half years later), and it contains a section discussing how these plans and events actually unfolded, as well as some lessons learned.

Table of Contents

1 Introduction and Background

After nearly three years of work, FreeBSD 5.0 was released in January of 2003. Features like the GEOM block layer, Mandatory Access Controls, ACPI, SPARC64® and ia64 platform support, and UFS snapshots, background filesystem checks, and 64-bit inode sizes make it an exciting operating system for both desktop and enterprise users. However, some important features are not complete. The foundations for fine-grained locking and preemption in the kernel exist, but much more work is left to be done. Performance and stability compared to FreeBSD 4.*X* has declined and must be restored and surpassed.

This is somewhat similar to the situation that FreeBSD faced in the 3.*X* series. Work on 3-CURRENT trudged along seemingly forever, and finally a cry was made to "just ship it" and clean up later. This decision resulted in the 3.0 and 3.1 releases being very unsatisfying for most, and it was not until 3.2 that the series was considered "stable". To make matters worse, the `RELENG_3` branch was created along with the 3.0 release, and the `HEAD` branch was allowed to advance immediately towards 4-CURRENT. This resulted in a quick divergence between `HEAD` and `RELENG_3`, making maintenance of the `RELENG_3` branch very difficult. FreeBSD 2.2.8 was left for quite a while as the last production-quality version of FreeBSD.

Our intent is to avoid repeating that scenario with FreeBSD 5.X. Delaying the `RELENG_5` branch until it is stable and production quality will ensure that it stays maintainable and provides a compelling reason to upgrade from 4.*X*. To do this, we must identify the current areas of weakness and set clear goals for resolving them. This document contains what we as the release engineering team feel are the milestones and issues that must be resolved for the `RELENG_5` branch. It does not dictate every aspect of FreeBSD development, and we welcome further input. Nothing that follows is meant to be a sleight against any person or group, or to trivialize any work that has been done. There are some significant issues, though, that need decisive and unbiased action.

2 Major issues

The success of the 5.*X* series hinges on the ability to deliver fine-grained threading and re-entrancy in the kernel (also known as SMPng) and kernel-supported POSIX threads in userland, while not sacrificing overall system stability or performance.

2.1 SMPng

The state of SMPng and kernel lockdown is the biggest concern for 5.*X*. To date, few major systems have come out from under the kernel-wide mutex known as "Giant". The SMP status page at http://www.FreeBSD.org/smp provides a comprehensive breakdown of the overall SMPng status. Status specific to SMPng progress in device drivers can be found at http://www.FreeBSD.org/projects/busdma. In summary:

- VM: Kernel malloc is locked and free of Giant. The UMA zone allocator is also free of Giant. vm_object locking is in progress and is an important step to making the buffer/cache free of Giant. Pmap locking remains to be started.
- GEOM: The GEOM block layer was designed to run free of Giant and allow GEOM modules and underlying block drivers to run free of Giant. Currently, only the ata(4) and aac(4) drivers are locked and run without Giant. Work on other block drivers is in progress. Locking the CAM subsystem is required for nearly all SCSI drivers to run without Giant; this work has not started yet.

  Additionally, GEOM has the potential to suffer performance loss due to its upcall and downcall data paths happening in kernel threads. Improved lightweight context switches might help this.
- Network: Work has restarted on locking the network stack. Routing tables, ARP, bridge, IPFW, Fast-Forward, TCP, UDP, IP, Fast IPSEC, and interface layers are being targeted initially, along with several Ethernet device drivers. The socket layer, IPv6, and other protocol layers will be targeted later. The primary goal of this work is to regain the performance found in FreeBSD 4.*X*. The cost of context switching to the device driver ithreads and the netisr is still hampering performance.
- VFS: Initial pre-cleanup started.
- buffer/cache: Initial work complete on locking the buffer.
- Proc: Initial proc locking is in place, further progress is expected for FreeBSD 5.2.
- CAM: No significant work has occurred on the CAM SCSI layer.
- Newbus: some work has started on locking down the device_t structure.
- Pipes: complete
- File descriptors: complete.
- Process accounting: jails, credentials, MAC labels, and scheduler are out from under Giant.
- MAC Framework: complete
- Timekeeping: complete
- kernel encryption: crypto drivers and core crypto(4) framework are Giant-free. KAME IPsec has not been locked.
- Sound subsystem: complete, but lock order reversal problems seem to persist.
- kernel preemption: preemption for interrupt threads is enabled. However, contention due to Giant covering much of the kernel and most of the device driver interrupt routines causes excessive context switches and might actually be hurting performance. Work is underway to explore ways to make preemption be conditional.

2.2 Interrupt latency and servicing

SMPng introduced the concept of dedicating kernel threads, known as ithreads, to servicing interrupts. With this, driver interrupt service routines are allowed to block for mutexes, memory allocations, etc. While this makes writing drivers easier, it introduces considerable latency into the system due to the complete process context switch which must be performed in order to service the ithread. This is aggravated by the extensive coverage over the kernel by the Giant mutex, and often results in multiple sleeps and context switches in order to service an interrupt. Drivers that register their interrupt as INTR_MPSAFE are less likely to feel these aggravating effects, but the overhead of doing a context switch remains. Interrupt service routines that are registered as INTR_FAST are run directly from the interrupt context and do not suffer these problems at all. However, the INTR_FAST property forces the interrupt line

to be exclusive; no sharing can occur on it. The proliferation of shared interrupts on PC systems makes this undesirable.

Several ideas have been proposed to help combat this problem:

- Special casing ithreads to be lightweight is a possibility. This might involve reducing the amount of saved context for the ithread, stack-borrowing from another kthread, and/or creating a new fast-path to avoid the mi_switch() routine.
- A new interrupt model can be introduced to allow drivers to register an 'interrupt filter' along with a normal service routine. This would be similar to the Mac OS X model in use today. Interrupt filter routines would allow the driver to determine if it is interested in servicing the interrupt, allow it to squelch the interrupt source, and possibly determine and schedule service actions. It would run in the same context as the low-level interrupt service routine, so sleeping would be strictly forbidden. If actions that result in sleeping or blocking for long periods are required, the filter would signal to the caller that its normal ithread routine should be scheduled.

2.3 Kernel-supported application threads

The FreeBSD 5.1 development cycle saw the KSE package jump into a highly usable state. THR, an alternate threading package based on some of the KSE kernel primitives but implementing purely 1:1 scheduling semantics also appeared and is in a similarly experimental but usable state. Users may interchange these two libraries along with the legacy libc_r library via relinking their apps or by using the new libmap feature of the runtime linker. This excellent progress must be driven to completion before the `RELENG_5` branch point so that the libc_r package can be deprecated.

- The kernel and userland components for KSE and THR must be completed for all Tier-1 platforms. The decision on which thread package to sanction as the default will likely be made on a per-platform basis depending on the stability and completeness of each package.

Table 1. KSE Status

Platform	Kernel	Userland	Works?
i386	YES	YES	YES
alpha	NO	YES	NO
sparc64	YES	NO	NO
ia64	YES	YES	YES
amd64	YES	YES	YES

Table 2. THR Status

Platform	Kernel	Userland	Works?
i386	YES	YES	YES
alpha	YES	YES	YES
sparc64	YES	YES	NO
ia64	YES	YES	YES
amd64	NO	NO	NO

- KSE must pass the ACE test suite on all Tier-1 platforms. Additional real-world testing must also be performed to ensure that the libraries are indeed useful. At a minimum, the following packages should be tested:
  - OpenOffice
  - KDE Desktop
  - Apache 2.x
  - BIND 9.2.x
  - MySQL
  - Java™ 1.4.x

3 Requirements for 5-STABLE

The `RELENG_5` branch must offer users the same stability and performance that is currently enjoyed in the `RELENG_4` branch. While the goal of SMPng is to allow performance to far exceed what is found in `RELENG_4` and its sibling BSDs, regaining performance to the basic level is of the utmost importance. The branch must also be mature enough to avoid ABI and API changes while still allowing potential problems to be resolved.

3.1 ABI/API/Infrastructure stability

Enough infrastructure must be in place and stable to allow fixes from `HEAD` to easily and safely be merged into `RELENG_5`. Also, we must draw a line as to what subsystems are to be locked down when we go into 5-STABLE.

- KSE: Both kernel and userland components must reach the same level of functionality for all Tier-1 platforms, in both UP and SMP configurations. The definition of "Tier-1 platforms" can be found in http://www.FreeBSD.org/doc/en_US.ISO8859-1/articles/committers-guide/archs.html. Continued testing against the ACE test suite must be made as the `RELENG_5` branch draws near. KSE must pose no functional regressions for the ongoing Java certification program. Common desktop and server applications must run seamlessly under KSE. A policy must be decided on as to which platforms will enable KSE as the default threading package, how to allow the user to switch threading packages, and how third-party packages will be made aware of these choices.
- busdma interface and drivers: architectures like PAE/i386™ and sparc64 which do not have a direct mapping between host memory address space and expansion bus address space require the elimination of vtophys() and friends. The busdma interface was created to handle exactly this problem, but many drivers do not use it yet. The busdma project at http://www.FreeBSD.org/projects/busdma tracks the progress of this and should be used to determine which drivers must be converted for `RELENG_5` and which can be left behind. No new storage or network drivers shall be allowed into the FreeBSD source tree. Exceptions for other classes of drivers must be justified in public discussion.
- PCI resource allocation: PC2003 compliance requires that x86 systems no longer configure PCI devices from the system BIOS, leaving this task solely to the OS. FreeBSD must gain the ability to manage and allocate PCI memory resources on its own. Implementing this should take into account cardbus, PCI-HotPlug, and laptop docking-station requirements. This feature will become increasingly critical through the lifetime of `RELENG_5`, and therefore is a requirement for the `RELENG_5` branch.

3.2 Performance

Performance hinges on the progress of SMPng infrastructure in the following areas:

- Storage: The GEOM block layer allows storage drivers to run without Giant. All drivers that interface directly with GEOM (as opposed to sitting underneath CAM or another middleware) must be locked and free of Giant in both their strategy and completion paths. Their interrupt handlers must also run free of Giant.
- Network: The layers in the IPv4 path below the socket layer must be locked and free of Giant. This includes the protocol, routing, bridging, filtering, and hardware layers. Allowances must be made for protocols that are not locked, especially IPv6. Testing must also be performed to ensure stability, correctness, and performance.
- Interrupt and context switching: As discussed above, interrupt latency and context switching have a severe impact of performance. Context switching for ithreads and kthreads must be improved. New interrupt handling models that allow for faster and more flexible handling of both traditional and MSI interrupts must be investigated and implemented.

3.3 Benchmarks and performance testing

Having a source of reliable and useful benchmarks is essential to identifying performance problems and guarding against performance regressions. A "performance team" that is made up of people and resources for formulating, developing, and executing benchmark tests should be put into place soon. Comparisons should be made against both FreeBSD 4.*X* and Linux 2.4/2.6. Tests to consider are:

- the classic "worldstone"
- webstone: `www/webstone`
- Fstress: http://www.cs.duke.edu/ari/fstress/
- ApacheBench: `www/p5-ApacheBench`
- netperf: `benchmarks/netperf`
- Web Polygraph: http://www.web-polygraph.org/ Note: does not compile with gcc 3.x yet.

3.4 Features:

- NEWCARD/OLDCARD: The NEWCARD subsystem was made the default for FreeBSD 5.0. Unfortunately, it does not include support for non-Cardbus bridges and falls victim to interrupt routing problems on some laptops. The classic 16-bit bridge support, OLDCARD, still exists and can be compiled in, but this is highly inconvenient for users of older laptops. If OLDCARD cannot be completely deprecated for `RELENG_5`, then provisions must be made to allow users to easily install an OLDCARD-enabled kernel. Documentation should be written to help transition users from OLDCARD to NEWCARD and from pccardd(8) to devd(8). The power management and "dumpcis" functionality of pccardc(8) needs to be brought forward to work with NEWCARD, along with the ability to load CIS quirk entries. Most of this functionality can be integrated into devd(8) and devctl(4).
- New scheduler framework: The new scheduler framework is in place, and users can select between the classic 4BSD scheduler and the new ULE scheduler. A scheduler that demonstrates processor affinity, HyperThreading and KSE awareness, and no regressions in performance or interactivity characteristics must be available for `RELENG_5`.

- GDB: GDB in the base system must work for sparc64, and must also understand KSE thread semantics. GDB 5.3 is available and is reported to address the sparc64 issues.

3.5 Documentation:

- The manual pages, Handbook, and FAQ should be free from content specific to FreeBSD 4.*X*, i.e. all text should be equally applicable to FreeBSD 5.*X*. The installation section of the handbook needs the most work in this area.
- The release documentation needs to be complete and accurate for all Tier-1 architectures. The hardware notes and installation guides need specific attention.

4 Post `RELENG_5` direction

The focus should be bug fixes and incremental improvements, as with all the -STABLE development branches. Following the usual procedure, everything should be vetted through the `HEAD` branch first and committed to `RELENG_5` with caution. New device drivers, incremental features, etc, will be welcome in the branch once they have been tested in `HEAD` and found stable enough.

Further SMPng lockdowns will be divided into two categories: driver and subsystem. The only subsystem that will be sufficiently locked down for `RELENG_5` will be GEOM, so incrementally locking down device drivers under it is a worthy goal for the branch. Full subsystem lockdowns will have to be fully tested and proven in `HEAD` before consideration will be given to merging them into `RELENG_5`.