

NetBSD 5.0

Introducing NetBSD 5.0

Andrew Doran
ad@netbsd.org

What is NetBSD?

A mature Unix-like operating system.

Focus areas:

- Server, embedded, and hobby applications.

- Scaling: from mobile phones to large servers.

- Providing a platform for education and research.

Development coordinated by one of the first truly collaborative Open Source projects to emerge on the net.

What is NetBSD 5.0?

The first NetBSD release to see full-time development funded by donations.

The largest set of changes to the kernel since the inception of the project.

A seven million line patch against 4.0.

Over 2 years in development.

Embedded role

An important market segment that helps to drive the system's development, so I'll mention it here.

Popular because the system scales well, the price and license are right, and it's easy to build upon.

Build any NetBSD platform from anywhere with a POSIX environment and C/C++ compiler: from Solaris, Linux, Windows (Cygwin), ...

```
$ uname -s -m
```

```
Linux i686
```

```
$ cd netbsd-src
```

```
$ ./build.sh -m sparc64 release
```

New in 5.0

Massive performance improvements on multicore hardware.

Journaling file system.

Power management: reliable suspend and resume.

Robust 1:1 threading implementation.

POSIX real-time extensions.

Processor sets, affinity, and offline/online.

Full kernel preemption for real-time threads.

File systems (including FUSE) running in user space.

Multiprocessor optimized malloc() for applications.

New in 5.0 (continued)

Integrated LDAP (OpenLDAP).

POSIX asynchronous I/O and message queues.

Modularized thread scheduler.

Writable file systems on DVDs (UDF).

Extensive bug audit of core kernel code.

Many new runtime tuneables.

And much more!

New for workstation users

Fast boot: ~20 seconds from loader to GDM login prompt on modern PCs.

Suspend and resume many laptops reliably, even multi-core.

Threaded applications can exploit the power of all CPUs. Notable examples: video and audio transcoding.

Unified kernel image for native x86. There is now rarely any need to compile a custom one.

Self-optimizing kernel: no penalty running on a single CPU machine.

Xorg to replace XFree86.

.. for servers and embedded apps

Performance and reliability improvements almost everywhere.

Scale from 1 to 32 CPUs with no tweaks on qualified workloads.

Partition CPU resources as needed. Example: dedicate CPUs to your realtime applications.

Realtime applications take advantage of kernel preemption ($\sim 5\mu\text{s}$ typical response on modern x86).

Logging file system: no more waiting for file system checks after an unscheduled system shutdown.

.. for developers and hobbyists

Many new application facilities and APIs. Example: Solaris-compatible atomic operations, with the same API in both kernel and userspace.

Native Java on i386 and soon amd64.

`cc -m32`: develop and test 32 bit applications on 64-bit hosts.

Extensive updates to the kernel programming guide - “section 9” of the system manual.

ATF: the Automated Testing Framework, for unit tests.

Better insight into how the system is performing and operating, with new tools such as `lockstat` (kernel lock statistics), `tprof` (performance-counter based kernel profiling), and mini crash dumps on i386 (no more 4GB kernel crash dumps).

Benchmarks

To illustrate performance improvements in 5.0.

Release versions of each OS installed and are:

- “out of the box”, with no tweaking.

- using hand-compiled software where important for fairness.

- using similar file system configurations: 4GB /local with journaling or soft updates, tmpfs /tmp

- using identical application configurations

Your results may differ, terms and conditions apply, etc.

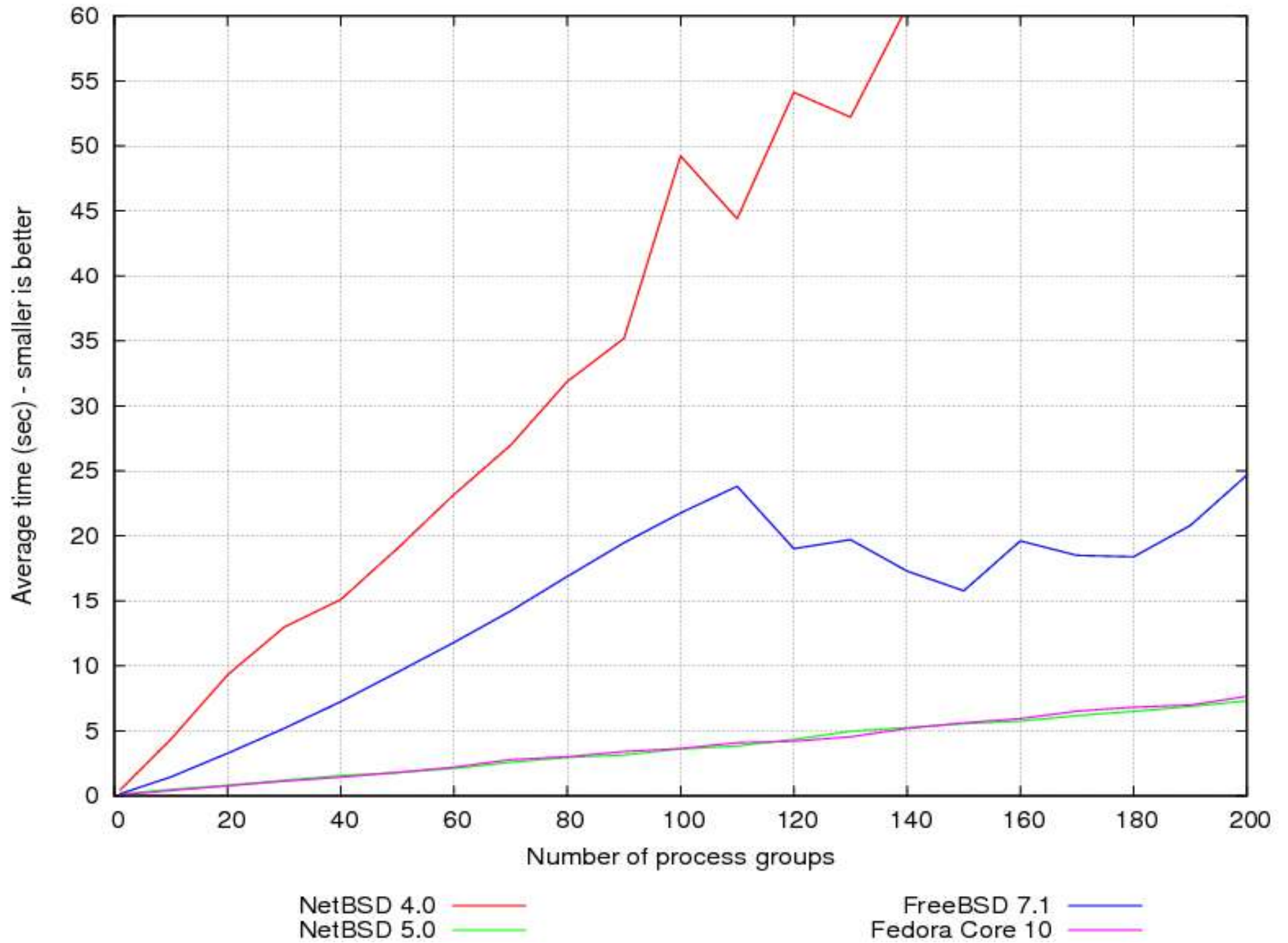
Benchmark: hackbench

Tests efficiency and scalability of the scheduler and IPC mechanisms.

Created by and popular with Linux kernel developers.

Exposes problems with SMP scaling, and with high numbers of active tasks.

hackbench IRC server simulation, 8 core Xeon E5320



Benchmark: MySQL sysbench

Simulates an OLTP style workload.

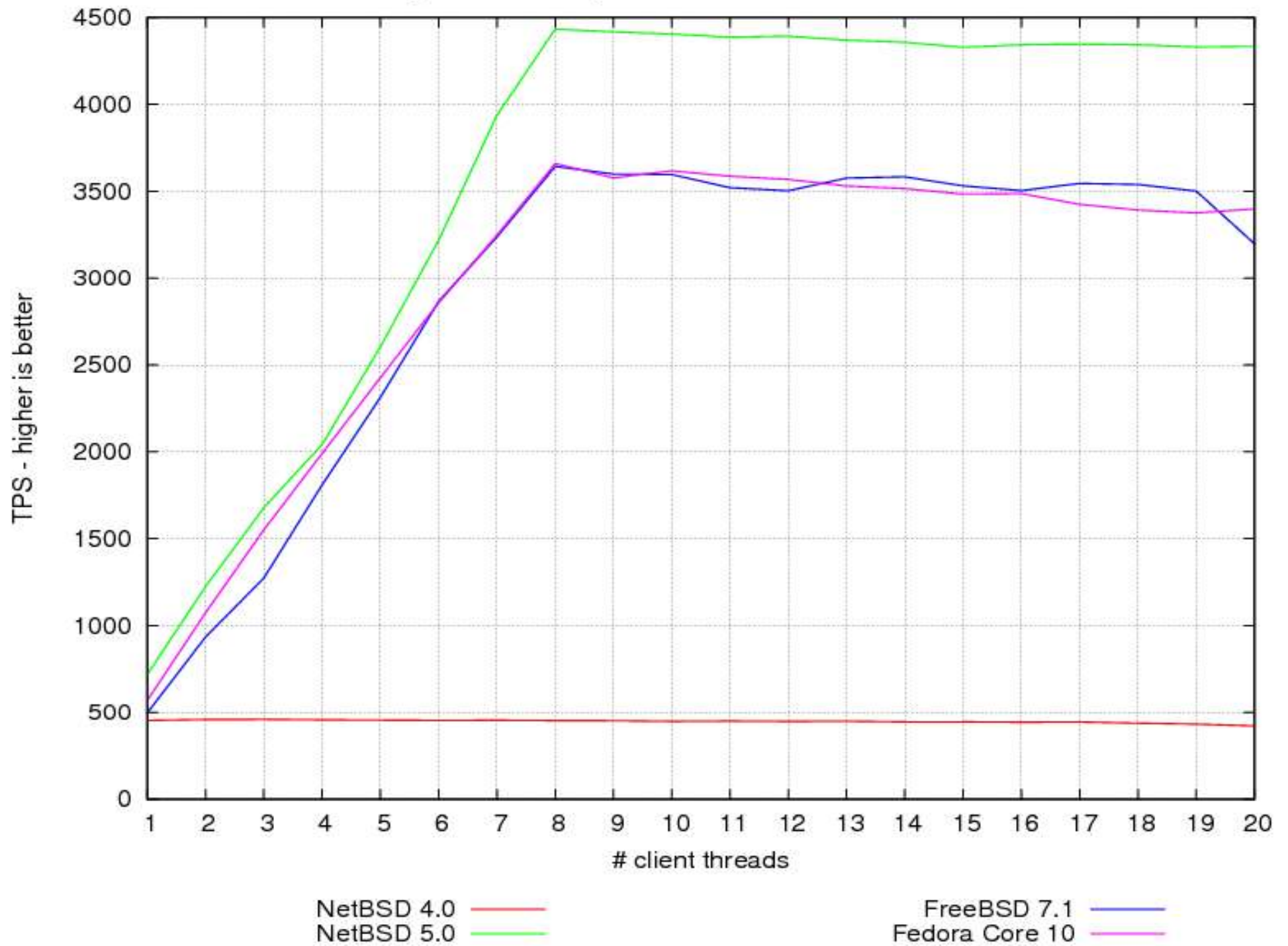
Tests efficiency and scalability of pthread library, malloc(), the scheduler and IPC mechanisms, among others.

Used extensively during development of 5.0.

Run here in read-only configuration: write performance of the DBMS and disk subsystem is a limiting factor we are not interested in for this particular test.

my.cnf available on request.

MySQL 5.0.51a sysbench OLTP, 8 core Xeon E5320



Benchmark: build.sh

NetBSD's build system.

A “software development” workload.

All tools bootstrapped: compiler, binutils, make, etc. The only host tool that sees extensive use is /bin/sh.

In this test, a hand built gcc is used to bootstrap (for fairness).

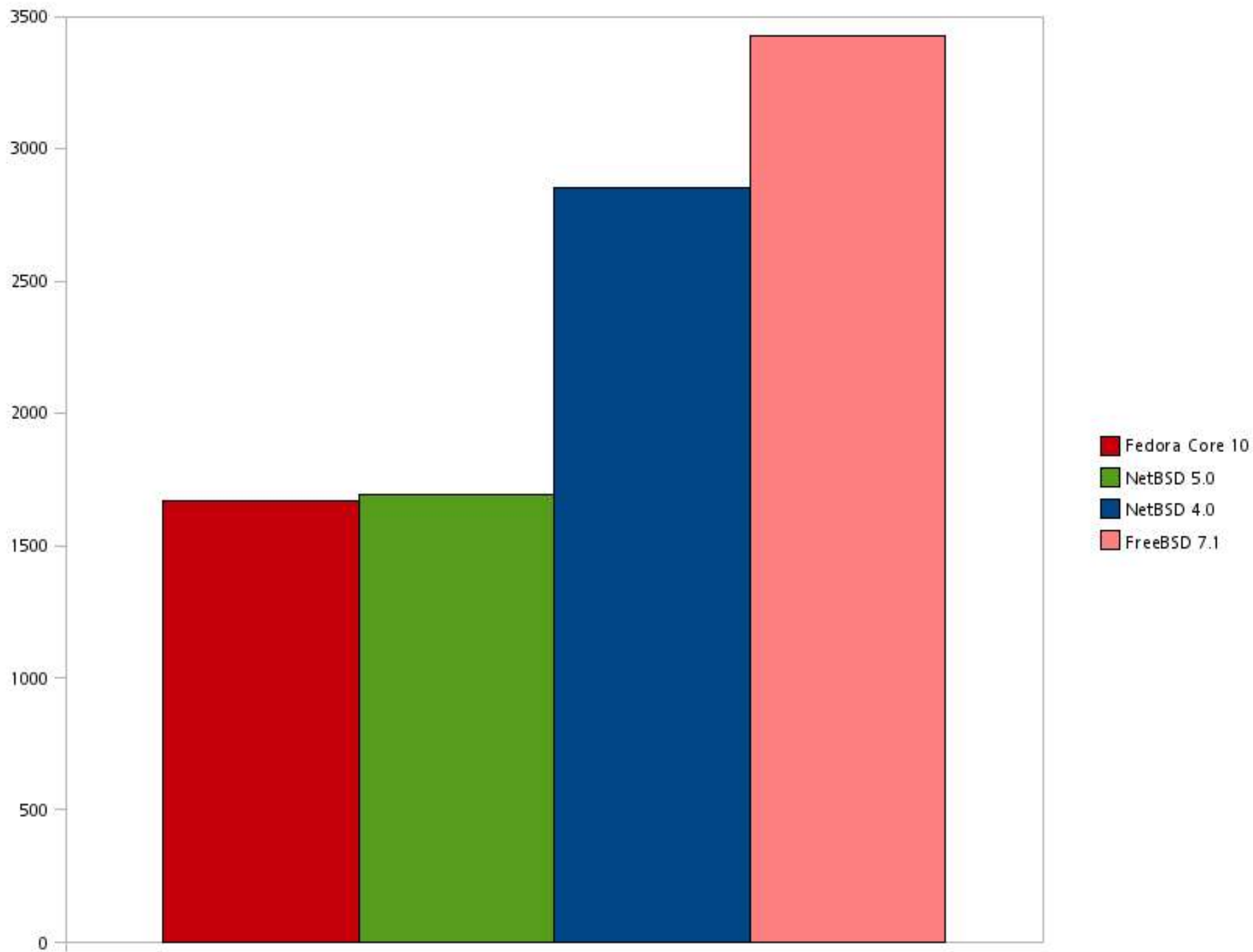
Measures time taken to build a release of NetBSD in seconds: smaller is better.

Fedora excels due to massive investment in improving efficiency and scalability of Linux over the last few years. Our hats are off to the GNU/Linux/Fedora developers.

We want to reverse the position with 6.0.

“build.sh -j 16 -m i386 release”, 8 core Xeon E5320

Time in seconds



The future: possible features for 6.0

Concurrent network stack.

255 core support on x86.

ZFS, Sun's new file system.

15 minutes from install CD to desktop with web and productivity apps.

Further performance and scalability improvements to file and virtual memory subsystems.

Modularized kernel on PC and server systems.

NUMA awareness.

Binary patches and upgrades.

NFS client performance improvements.