

{BSDBootCamp}

Testing Ports with redports.org

Bernhard Fröhlich
decke@FreeBSD.org

Grazer Linuxtage 2012

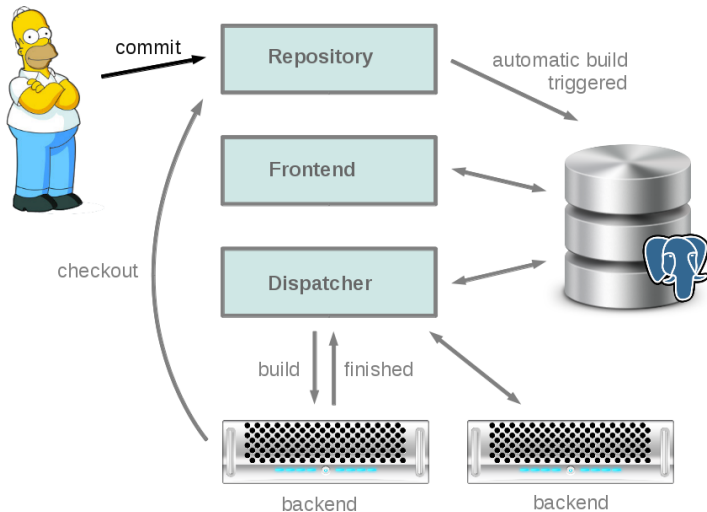
28. April 2012

Why would anyone need that?

Problems:

- ▶ Mistakes in a Port update cause delays
- ▶ Port maintainer is able to test on all supported FreeBSD platforms
- ▶ No dedicated hardware required for testing
- ▶ Better utilization of available hardware

Architecture



{BSDBootCamp}

Repository

- ▶ Own Subversion subtree per user
(eg: <http://svn.redports.org/decke/multimedia/mythtv>)
- ▶ Automatic triggered jobs per commit

Frontend

The screenshot shows the RedParts website interface. At the top, there's a navigation bar with 'HOME', 'GUIDE', and 'QUEUE' buttons. Below that, a 'Buildgroups' section contains a table with columns for Status, Priority, Name, Backends, Queued, and Description. The table lists various FreeBSD buildgroups with their respective backends and queued counts. To the right of the table is a 'JOIN BUILDGROUP:' form with dropdown menus for 'Buildgroup' and 'Priority', and a 'JOIN' button. Below the buildgroups table is a 'Buildqueue' section with a table showing the current build queue, including port names, buildgroups, status, buildtime, priority, user, and links.

Status	Priority	Name	Backends	Queued	Description
		10-CURRENT/amd64	1	7	FreeBSD 10-CURRENT/amd64
		10-CURRENT/i386	1	5	FreeBSD 10-CURRENT/i386
LEAVE	normal	9.0-RELEASE/amd64	1	5	FreeBSD 9.0-RELEASE/amd64
LEAVE	normal	9.0-RELEASE/i386	1	3	FreeBSD 9.0-RELEASE/i386
		8.2-RELEASE/amd64	1	2	FreeBSD 8.2-RELEASE/amd64
		8.2-RELEASE/i386	1	2	FreeBSD 8.2-RELEASE/i386
		7.4-RELEASE/amd64	1	1	FreeBSD 7.4-RELEASE/amd64
		QATty	1	1	Custom LOCALBASE and PREFIX

Port	Buildgroup	Status	Buildtime	Priority	User	Links
finance/R-cran-PerformanceAnalytics 3.0.4.4	10-CURRENT/i386	building	4 hours	highest	tota	
finance/R-cran-PerformanceAnalytics 3.0.4.4	10-CURRENT/amd64	building	3 hours	highest	tota	

- ▶ Trac as the base (Repository browser, Bugtracker, User management, ...)
- ▶ Own plugin for additional functionality

{BSDBootCamp}

Dispatcher

- ▶ Communication with backend
- ▶ Dispatch jobs on all available build machines
- ▶ Fault tolerant
- ▶ Transfer logfiles/packages back to frontend

Technology:

- ▶ daemon forks childs
- ▶ Statemachine (1 step = 1 child process = 1 transaktion)
- ▶ Database locks for synchronisation
- ▶ ANSI C
- ▶ libpq (PostgreSQL C API)
- ▶ libcurl (https)

Backend

- ▶ Based on Marcuscom's Ports Tinderbox
- ▶ Checkout user portstree and merge with official tree
- ▶ package building
- ▶ Build environments (FreeBSD 7.4/8.3/9.0/10-CURRENT on i386/amd64, CLANG, ...)

Technology:

- ▶ CGI shellscripts for automated tasks
- ▶ ZFS snapshot/rollback for portstree merging
- ▶ tmpfs for better performance

{BSDBootCamp}

Future

- ▶ Buildbot for automated portupdates
- ▶ Ports PR buildbot
- ▶ QAT support
- ▶ Patch support
- ▶ OPTIONS support
- ▶ poudriere based backend
- ▶ more Power!

Thanks a lot for listening!
Are there any questions?

Thanks to:

- ▶ My girlfriend (time)
- ▶ Baptiste Daroussin (poudriere)
- ▶ Mark Linimon (hardware)
- ▶ George Neville-Neil (hardware)

[http://people.freebsd.org/~decke/talks/
201204_glt12-bsdbootcamp_redports-en.pdf](http://people.freebsd.org/~decke/talks/201204_glt12-bsdbootcamp_redports-en.pdf)

{BSDBootCamp}