New Features in BulkTracker

pkgsrcCon 2015

Benny Siegert ‹bsiegert@NetBSD.org›

Outline

1. Introduction
2. Statistics
3. New features
4. Embracing JavaScript and JSON
5. Future work
6.

What is BulkTracker?

Web app to track pkgsrc bulk build status.

“Which package fails to build on which platform?”

Built in Go, running on Google App Engine.

http://bulktracker.appspot.com/

github.com/bsiegert/BulkTracker

How does it work?

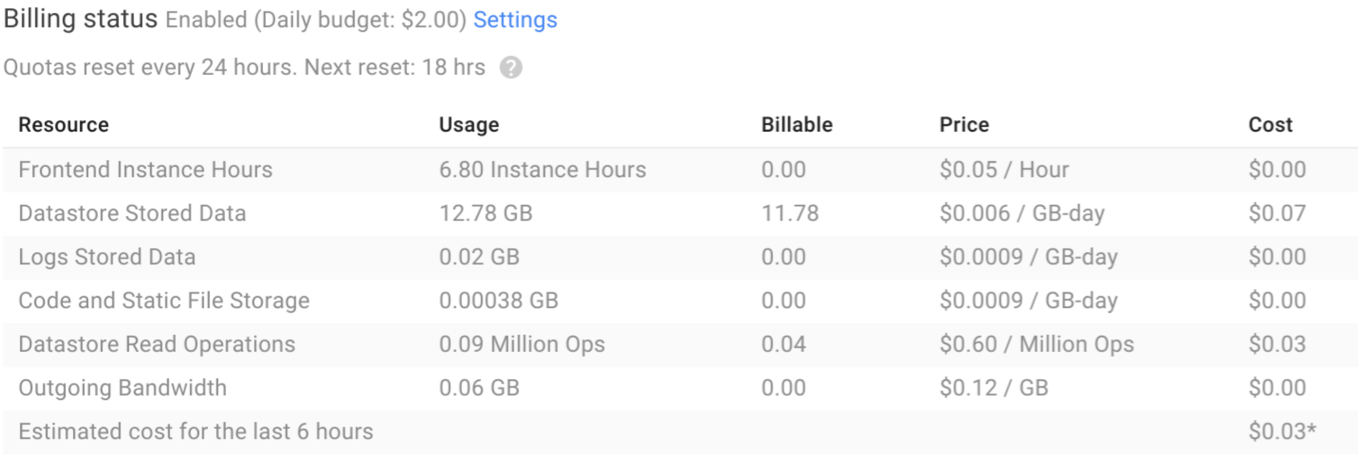
Subscribed to mail to pkgsrc-bulk@.

- Parses incoming reports, downloads “machine-readable” version.
- Generates 1 record per build + 1 per package result.

Some statistics

768 builds, ~4M individual package results

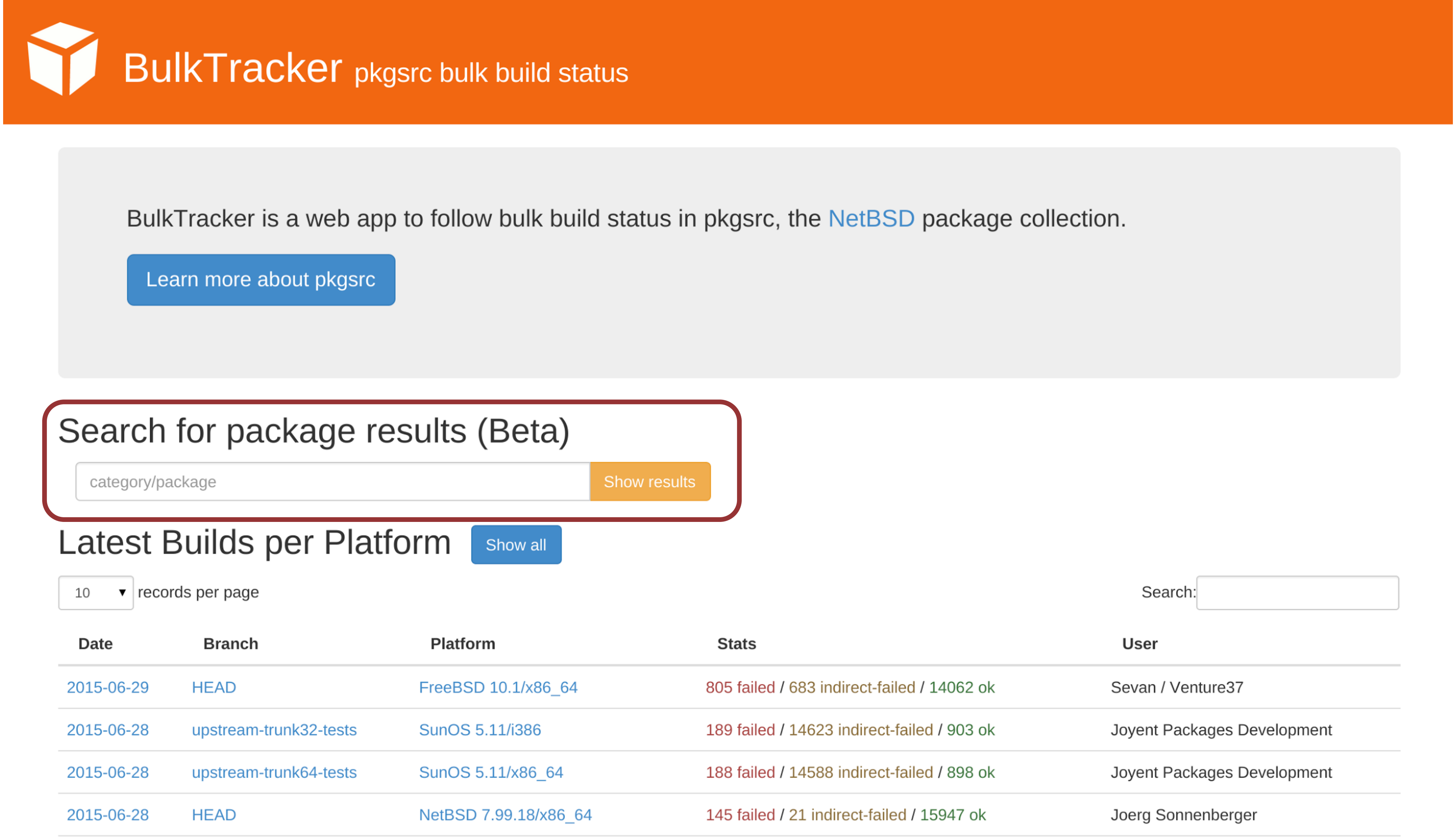
800M of data (plus 12G of indexes …)



Billing status Enabled (Daily budget: $2.00) Settings

Quotas reset every 24 hours. Next reset: 18 hrs

Resource	Usage	Billable	Price	Cost
Frontend Instance Hours	6.80 Instance Hours	0.00	$0.05 / Hour	$0.00
Datastore Stored Data	12.78 GB	11.78	$0.006 / GB-day	$0.07
Logs Stored Data	0.02 GB	0.00	$0.0009 / GB-day	$0.00
Code and Static File Storage	0.00038 GB	0.00	$0.0009 / GB-day	$0.00
Datastore Read Operations	0.09 Million Ops	0.04	$0.60 / Million Ops	$0.03
Outgoing Bandwidth	0.06 GB	0.00	$0.12 / GB	$0.00
Estimated cost for the last 6 hours				$0.03*



BulkTracker is a web app to follow bulk build status in pkgsrc, the NetBSD package collection.

Search for package results (Beta)

Latest Builds per Platform

Date	Branch	Platform	Stats	User
2015-06-29	HEAD	FreeBSD 10.1/x86_64	805 failed / 683 indirect-failed / 14062 ok	Sevan / Venture37
2015-06-28	upstream-trunk32-tests	SunOS 5.11/i386	189 failed / 14623 indirect-failed / 903 ok	Joyent Packages Development
2015-06-28	upstream-trunk64-tests	SunOS 5.11/x86_64	188 failed / 14588 indirect-failed / 898 ok	Joyent Packages Development
2015-06-28	HEAD	NetBSD 7.99.18/x86_64	145 failed / 21 indirect-failed / 15947 ok	Joerg Sonnenberger

New feature: latest build per platform

Collapse repeated builds into one:

- Default “builds” view on home page only shows one build for each {branch, platform, user}

Also the default set for package results.

New feature: package results

Enter category/pkgname, get build results

Currently: for latest build for each platform

Need UI for selecting builds / versions to show

Originally planned: grid view – useful?

Embracing JavaScript

Most pages don’t use JavaScript (just DataTables), all logic is in Go code + templates.

Package results is static HTML + JSON.

It turns out that some things *are* easier in JavaScript. (Using the DataTables API.)

JSON API = clean backend

Clean separation of logic and presentation.

Can store the entire result in memcache.

Shared memcache on App Engine is free!

Go can directly marshal structs ↔JSON.

If people are interested, could become an API.

Thinking about the data model

(too) many writes to the datastore.

Use Cloud SQL? (probably more expensive)

Use blobstore and store all build details as a single blob?

- will make cross-correlations slow.

Remaining pain points

What if fetching machine-readable report fails?

UI needs improvement.

- surface builds with no detail
- make “retry index” feature more interactive

Other requirements

PRD in the wiki for such a tracking tool,
not all of the requirements have been addressed.

- keep track of updates to pkgsrc [how?], surface updated packages that have not been built
- annotate build failures with explanation
- “results from packages you maintain”

Conclusion

http://bulktracker.appspot.com/

Looking for people interested in joining
or sending pull requests!