Analyzing Millions of GitHub Commits

*what makes developers happy, angry, and everything in between?*

Brian Doll  briandoll@github.com  @briandoll
Ilya Grigorik  igrigorik@google.com  @igrigorik
<facepalm>
"Keeping up with 3000+ open-source projects is not easy... If only there was a better way!"

Ilya, circa early 2012
(Ilya's) Burning questions...

- What were the hot new projects today?
  - In Ruby land...
  - In JavaScript land...
  - Globally?
- Did anyone commit something interesting or controversial?
- For the people I follow, which projects did they follow or contribute to?
- What are the emerging projects, or languages?
- ...

@briandoll  @igrigorik
GitHub is *kinda a big deal* in open-source...

**Activity stats:**
- **Max:** 184,570 events / day
- **Avg:** 125,970 events/day
- **1~2 events / second!**

2,348,118 people hosting over 4,048,538 repositories

jQuery, reddit, Sparkle, curl, Ruby on Rails, node.js, ClickToFlash, Erlang/OTP, CakePHP, Redis, and many more

@briandoll    @igrigorik
The "aha" moment:

It's not my timeline, it's the **global timeline** that **contains the answers**.

Now if only we had access to the GitHub archive...

(one weekend later...)
Open-source developers all over the world are working on millions of projects: writing code & documentation, fixing & submitting bugs, and so forth. GitHub Archive is a project to record the public GitHub timeline, archive it, and make it easily accessible for further analysis.

Looking for the daily top & watched repository reports? Sign up here.

GitHub provides 18 event types, which range from new commits and fork events, to opening new tickets, commenting, and adding members to a project. The activity is aggregated in hourly archives, which you can access with any HTTP client:

http://www.githubarchive.org  collector code @ https://github.com/igrigorik/githubarchive.org/
Anatomy of an event

- CommitCommentEvent
- CreateEvent
- DeleteEvent
- DownloadEvent
- FollowEvent
- ForkEvent
- ForkApplyEvent
- GistEvent
- GollumEvent
- IssueCommentEvent
- IssuesEvent
- MemberEvent
- PublicEvent
- PullRequestEvent
- PullRequestReviewCommentEvent
- PushEvent
- TeamAddEvent
- WatchEvent

18 event types. JSON payload, meta-data rich.
{  
  + actor_attributes: { ... },  
  actor: "raziel23x",  
  - repository: {  
      created_at: "2012-10-09T09:11:41-07:00",  
      url: "https://github.com/MotorolaSpyder/android_local_spider",  
      description: "Local Manifest for CM10/AOSP on Motorola Droid RA2R",  
      stargazers: 0,  
      owner: "MotorolaSpyder",  
      has_issues: false,  
      open_issues: 0,  
      pushed_at: "2012-10-10T11:45:05-07:00",  
      forks: 0,  
      organization: "MotorolaSpyder",  
      has_downloads: true,  
      fork: true,  
      size: 208,  
      master_branch: "jellybean-cm-stock",  
      name: "android_local_spider",  
      id: 6143640,  
      homepage: "http://apkmultitool.com",  
      private: false,  
      watchers: 0,  
      has_wiki: true  
  },  
  url: "https://github.com/MotorolaSpyder/android_local_spider/compare/53a53da7d6...476c157eba",  
  public: true,  
  type: "PushEvent",  
  - payload: {  
      size: 1,  
      ref: "refs/heads/jellybean-cm-stock",  
      head: "476c157eba52d65793a33954fae127863a75148e1",  
      - shas: {  
          "476c157eba52d65793a33954fae127863a75148e1",  
          "raziel23x@gmail.com",  
          "Update local_manifest.xml"  
  }  
}
## GZIP archive(s)

<table>
<thead>
<tr>
<th>Query</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity for April 11, 2012 at 3PM PST</td>
<td><code>wget http://data.githubarchive.org/2012-04-11-15.json.gz</code></td>
</tr>
<tr>
<td>Activity for April 11, 2012</td>
<td><code>wget http://data.githubarchive.org/2012-04-11-{0..23}.json.gz</code></td>
</tr>
<tr>
<td>Activity for April 2012</td>
<td><code>wget http://data.githubarchive.org/2012-04-{01..31}-{0..23}.json.gz</code></td>
</tr>
</tbody>
</table>

- Raw JSON data
- Hourly archives
- Easy access
- Uploaded every hour

- Tool agnostic
  - Lots of work
  - Non-interactive
  - Hard to analyze large ranges

Hmmmm...
"Dremel is a scalable, interactive ad-hoc query system for analysis of read-only nested data. By combining multi-level execution trees and columnar data layout, it is capable of running aggregation queries over trillion-row tables in seconds. The system scales to thousands of CPUs and petabytes of data, and has thousands of users at Google."

developers.google.com/bigquery
GitHub Archive =
  JSON data
  Meta-data rich

BigQuery =
  Interactive ad-hoc analysis
  Trillion-row tables
  Table scan friendly (no indexes)
  Column storage for efficient access
  ...

BigQuery + GitHub = Profit *

* still working on the profit part
Data import in 3 commands - *automation ftw!*

1. $ wget http://data.githubarchive.org/2012-04-11-15.json.gz
2. $ ruby flatten.rb 2012-04-11-15.json.gz > flat.csv.gz
3. $ bq load github.timeline flat.csv.gz

Hourly cron-job to import flattened CSV **
A RegExp against entire table? Why not...

Compose Query

```
SELECT
curse.repository_language, language,
total.count total_commits,
curse.count curse_commits,
curse.count/total.count * 100 curse_percentage
FROM
{}
SELET
repository_language, COUNT(*) as count
FROM
[pubdata:samples.github_timeline]
WHERE
REGEXP_MATCH(payload_commit_msg, r'\[S]ucks\[D]s[mr]n')
AND
repository_language != ''
GROUP BY
repository_language
ORDER BY
count DESC
as curse
JOIN
{}
SELET
repository_language, COUNT(*) as count
```

RUN QUERY Show previous query results

Speaking of interactive, ad-hoc analysis..
- BigQuery <3 table scans
- What’s an index? **Table scans are no slower** than any other query...

https://gist.github.com/671fe0d3cb5e669a4fd6
Not your ....'s SQL language

Timestamp Functions
- FORMAT.UTC_USEC
- PARSE.UTC_USEC
- UTC_USEC_TO_DAY
- ...

Aggregate Functions
- AVG, COUNT
- STDDEV, VARIANCE
- QUANTILES
- TOP, ...

String Functions
- CONTAINS
- SUBSTR
- CONCAT, RPAD, LPAD
- ...

SQL bread and butter
- JOIN
- HAVING
- GROUP BY
- ORDER BY
- ...

Nested Record Functions
- WITHIN
- FLATTEN
- Scoped aggregation...

Other Functions
- CASE
- IF
- HASH
- ... and many others

https://developers.google.com/bigquery/docs/query-reference

@briandoll       @igrigorik
GitHub Daily (email) reports!

Speaking of scratching an itch...

https://www.githubarchive.org/
GitHub Daily: GitHub + BigQuery + MailChimp

1. Cronjob
   a. Run query via `bq`
   b. Export JSON
   c. Render HTML template
   d. Email via MailChimp

2. ~30 line of code

http://www.githubarchive.org/

@briandoll @igrigorik
GitHub Daily = GitHub Archive + BigQuery + MailChimp

```
SELECT repository_name, repository_language, repository_description, COUNT(repository_name) as cnt,
repository_url
FROM github.timeline
WHERE type="WatchEvent"
  AND PARSE_UTC_USEC(createdAt) >= PARSE_UTC_USEC("#{yesterday} 20:00:00")
  AND repository_url IN (  
    SELECT repository_url
    FROM github.timeline
    WHERE type="CreateEvent"
      AND PARSE_UTC_USEC(repository_created_at) >= PARSE_UTC_USEC('#{yesterday} 20:00:00')
      AND repository_fork = "false"
      AND payload_ref_type = "repository"
    GROUP BY repository_url
  )
GROUP BY repository_name, repository_language, repository_description, repository_url
HAVING cnt >= 5
ORDER BY cnt DESC
LIMIT 25
```
GitHub Data Challenge

Analyze with BigQuery, submit your entries...

https://github.com/blog/1112-data-at-github
octoboard.com - stats since March 11, 2012

- **Global Activity**: 28,240,341 (up 124,957/day)
- **Created Repositories**: 1,264,750 (up 5,596/day)
- **Open Sourced Repositories**: 24,361 (up 108/day)
- **Forked Repositories**: 918,792 (up 4,063/day)

Yesterday:
- **Global Activity**: It was a **good** day, they were up 32% above average.
- **Created Repositories**: It was a **very good** day, they were up 55% above average.
- **Open Sourced Repositories**: It was a **very good** day, they were up 48% above average.
- **Forked Repositories**: It was a **very good** day, they were up 43% above average.

- **Opened Pull Requests**: 448,530 (up 1,985/day)
- **Opened Issues**: 873,284 (up 3,864/day)
- **New GitHub Pages**: 59,856 (up 265/day)
- **Created Gists**: 839,692 (up 3,715/day)

Yesterday:
- **Opened Pull Requests**: It was a **very good** day, they were up 49% above average.
- **Opened Issues**: It was a **very good** day, they were up 36% above average.
- **New GitHub Pages**: It was a normal day, they were down 7% below average.
- **Created Gists**: It was a **bad** day, they were down 33% below average.

Denis Roussel [https://github.com/KuiKui/Octoboard](https://github.com/KuiKui/Octoboard)
~108 private repositories released to the public / day

Active JavaScript and Ruby communities on GitHub.
~2000 Pull requests / day - which languages?

2x the activity on weekdays than on weekends! Saturday's are the slowest.
Emotional impact of programming languages...

http://geeksta.net/geeklog/exploring-expressions-emotions-github-commit-messages/

Ramiro Gomez
https://github.com/yaph

@briandoll       @igrigorik
Emotional impact ... example query for "joy"

```
SELECT repository_language, COUNT(*) as cntlang
FROM [githubarchive:github.timeline]
WHERE repository_language != ''
AND payload_commit_msg != ''
AND PARSE_UTC_USEC(created_at) < PARSE_UTC_USEC('2012-05-09 00:00:00')
AND REGEXP_MATCH(payload_commit_msg,
    r'(\?i)\b(yes|yay|hallelujah|hurray|bingo|amused|cheerful|excited|glad|proud)\b')
GROUP BY repository_language
ORDER BY cntlang DESC
```

Table-scans for the win!

https://github.com/yaph/gh-emotional-commits

@briandoll       @igrigorik
Emotional impact: anger

- VimL takes the top spot
- C makes more people angry than Java? Interesting!
- Python makes more people angry than Ruby... But we all knew that! :-)

http://geeksta.net/geeklog/exploring-expressions-emotions-github-commit-messages/

@briandoll  @igrigorik
Emotional impact: amusement

- Ruby takes #1
- What's so amusing about C#??? :)

Regexp:

```
(?i)\b(ha(ha)+|he(he)+|lol|rofl|lmfao|lulz|lolz|rotfl|lawl|hilarious)\b
```

http://geeksta.net/geeklog/exploring-expressions-emotions-github-commit-messages/

@briandoll  @igrigorik
Emotional impact: surprise

- Perl, of course...
- Or, if it has a /C/ as part of the name

Regexp:

```
(?i)b
(yikes|gosh|baffled|stumped|surprised|shocked)b
```
Emotional impact: swear word inducing...

- If it has a `/C/` as part of the name, it'll make you swear.

Regexp:

(snip) :-)
How do they stack up?

- **PHP, Objective-C** and **C#** are net positive
- **Java, Shell** and **C** are fairly even while **VimL** is just bad news
Disable 'showmatch' option Matching parens are highlighted even without this option; what it does is jump the cursor to the matching paren which is insane.

10/23/12 3:28 AM
styles everywhere, tutorial for first user login, fixing some css

10/23/12 2:57 AM
again

10/23/12 2:30 AM
more

10/23/12 2:29 AM
Security worked out

10/23/12 2:07 AM
travis ci
A **Ruby** programmer is *very likely to know JavaScript*, while a **Perl** programmer is not.

**Java** is a popular language, but stands primarily alone.

[GitHub link](https://github.com/mjwills/ProgLangVisualise)
Programming Language Popularity

StackOverflow Questions Tagged vs. Projects on Github

Popularity Rank on StackOverflow (by # of tags)

Popularity Rank on Github (by # of projects)

Corr = 0.78

http://www.drewconway.com/zia/?p=2892

@briandoll  @igrigorik
There is a lot of existing VimL, common lisp and visual basic code, but everyone is afraid to ask questions about them?
Repository activity by language

Mapping organizations with 250+ projects on GitHub to their respective programming languages

http://zoom.it/kCsU
GitHub activity by country

Commits per 100k people

http://bl.ocks.org/2727882

@briandoll       @igrigorik
Projects using the fork to pull paradigm...

1. homebrew
2. bootstrap
3. rails
4. gitignore
5. ...

Number of Fork2Pull for April, 2012 by projects

- mxcl/homebrew
- twitter/bootstrap
- rails/rails
- github/gitignore
- hackdaymanifesto/hackdaymanifesto.github.com
- meteor/meteor
- sarahelizgray/cit594_git_assignment
- reedlaw/ruby-mmo
- leachim6/hello-world
- jeffhung/osdc-2012-hackathon

Jean-Noël Avila

Joined on Nov 20, 2009

https://gist.github.com/2623537
Pull request latency!

- 50%+ pull requests come in within 1 hour of the fork
- 80%+ pull requests come in within 1 day of the fork

1/2 minute? Spelling mistakes, etc!

https://gist.github.com/2623537
Pull request latency: the query...

```sql
SELECT COUNT(DISTINCT ForkTable.url) AS f2p_number,
       FLOOR(LOG2((PARSE_UTC_USEC(PullTable.created_at) - PARSE_UTC_USEC(ForkTable.created_at))/3000000)) AS f2p_interval_log_2_minute
FROM
  (SELECT
        url,
        repository_url,
        repository_language,
       MIN(created_at) AS created_at
  FROM [githubarchive:github.timeline]
  WHERE type='ForkEvent'
  AND PARSE_UTC_USEC(created_at) >= PARSE_UTC_USEC('2012-04-01 00:00:00')
  AND PARSE_UTC_USEC(created_at) < PARSE_UTC_USEC('2012-05-01 00:00:00')
  GROUP BY repository_language, repository_url, url)
AS ForkTable
INNER JOIN
  (SELECT...
  )
AS PullTable
ON
  ForkTable.repository_url=PullTable.repository_url AND
  ForkTable.url=PullTable.payload_pull_request_head_repo_html_url
WHERE PARSE_UTC_USEC(PullTable.created_at)>PARSE_UTC_USEC(ForkTable.created_at)
GROUP BY
  f2p_interval_log_2_minute
ORDER BY
  f2p_interval_log_2_minute ASC
```

https://gist.github.com/2623537#file_fork2_pull_request_by_latency.sql
Does the eye of the public make for better and well tested code?

Just by watching how other, more senior project members behave, they learn what a good commit looks like.

Infrastructure with low barriers seems to be very important in getting developers to test their contributions.

Because contribution has become so easy, project owners reported seeing what they called drive-by commits.

http://blog.leif.me/2012/09/github-testing/
Research: Analysis of OSS development using DNA sequencing tools
by Aron Lindberg and Tim Henderson at Case Western Reserve University

What is the "social DNA" of successful open source projects?
Research: Analysis of OSS development using DNA sequencing tools by Aron Lindberg and Tim Henderson at Case Western Reserve University

Overall activity levels are tightly coupled with commit levels

Success breeds success; i.e. communities that are growing or declining are likely to continue the trajectory that they have started (An object in motion...)

Don’t ignore those who commit infrequently or only report bugs: growing a leadership pipeline through quickly establishing a broad base of developers supports long-term success
Moar & Better Data!

*Import in progress...*
SELECT expr1 WITHIN RECORD, expr2 WITHIN node_name...

Support for nested (JSON) data in BigQuery!  New import in process...

http://googledevelopers.blogspot.com/2012/10/got-big-json-bigquery-expands-data.html

@briandoll  @igrigorik
Kudos to GitHub....

Github Archive data now goes back to **Feb 12, 2011**

- **Feb 12, 2011 - Now!**

- Raw JSON data for 2011:
  - `wget http://data.githubarchive.org/201{1,2}-{01.12}-{01..31}-{0..23}.json.gz`
SELECT questions
FROM audience

Brian Doll  briandoll@github.com  @briandoll
Ilya Grigorik  igrigorik@google.com  @igrigorik