An Englishman's Guide To Queueing

paulhammond.org/2012/queueing

Paul Hammond

paul@paulhammond.org

@ph

“An Englishman, even if he is alone, forms an orderly queue of one”

– George Mikes, How to be an Alien, 1946

DEVOPS OR NOT

devopsornot.dev

DEVOPS OR NOT

Paul Hammond

	Name	Company	Devops	
	DEVOPS BORAT	-	230,561	devup
	Patrick Debois	Jedi BVBA	230	devup
	Artur Bergman	Fastly	229	devup
	Mike Krieger	Instagram	123	devup
	Mike Rembetsy	Etsy	122	devup
	Theo Schlossnagle	OmniTI	116	devup
	John Willis	enStratus	112	devup
	Paul Hammond	Typekit	0	devup
	John Allspaw	Etsy	-32	devup

What should happen when this button gets pushed?
It can’t be that hard, right?

```
# on update
UPDATE users SET devups = devups + 1;

# to display the leaderboard
SELECT * FROM users ORDER BY devups LIMIT 10;
```

“We need an audit trail”

```
# on update
INSERT INTO devups (user_id, voter_id, created)
VALUES (10, 100, "2012-10-04 14:07:23");

# to display the leaderboard
SELECT user_id, count(*) AS c FROM devups
GROUP BY user_id ORDER BY c DESC LIMIT 10;
```

“Great Job!”

“Can we get a page showing the people you’ve voted for?”

```
# to display the people you've voted for
SELECT user_id, count(*) AS c FROM devups WHERE
  user_id IN (SELECT user_id FROM devups AS v
WHERE v.voter_id = 3) GROUP BY user_id ORDER BY
                c DESC LIMIT 10;
```

“Can we change the algorithm?”

```
# generate leaderboard with weighted averages
  SELECT user_id, SUM((SELECT count(*) FROM
devups AS v WHERE v.voter_id = u.voter_id AND
v.user_id = u.user_id) / (SELECT count(*) FROM
 devups AS v WHERE v.voter_id = u.voter_id))
      FROM devups AS u GROUP BY u.user_id;
```

“Can we show your score at the top of every page?”

“Can we get a leaderboard of people that work at Etsy?”

“Can we get a leaderboard of companies?”

“Can we get a leaderboard of people called Mike?”

“Can we get a leaderboard of people called Theo?”

DEVOPS OR NOT

devopsornot.dev/theo

DEVOPS OR NOT

Paul Hammond

	Name	Company	Devops	
	**THEO SCHLOSSNAGLE**	OmniTI	116	devup

“Can we get a leaderboard of people in London?”

“Can we get a leaderboard of people with linked Twitter accounts?”

“Can we get a leaderboard of people with no Twitter account?”

“Can we get a leaderboard of people you follow on Twitter?”

“Why is the site so slow?”

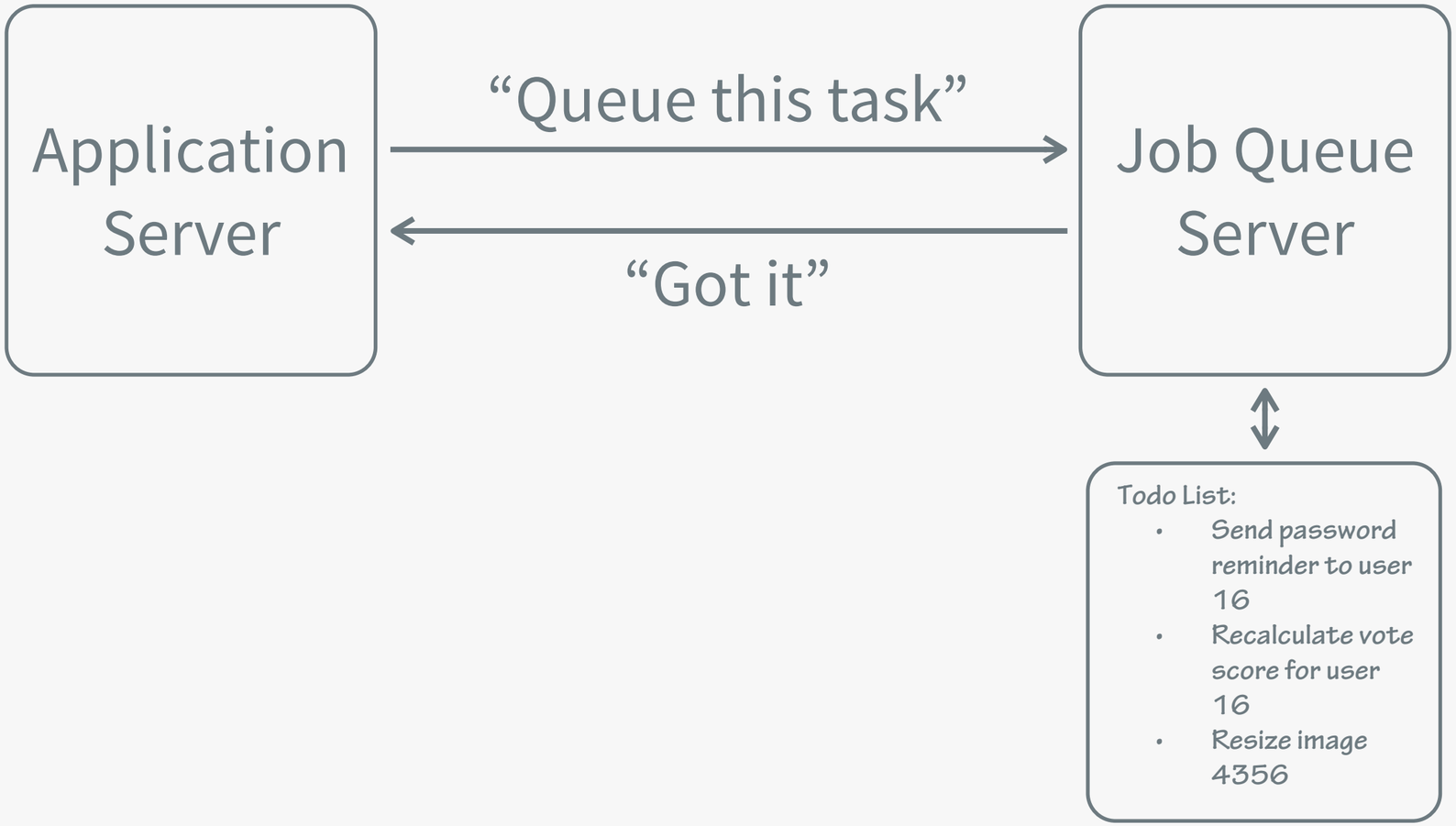
Do slow calculations when inserting the data

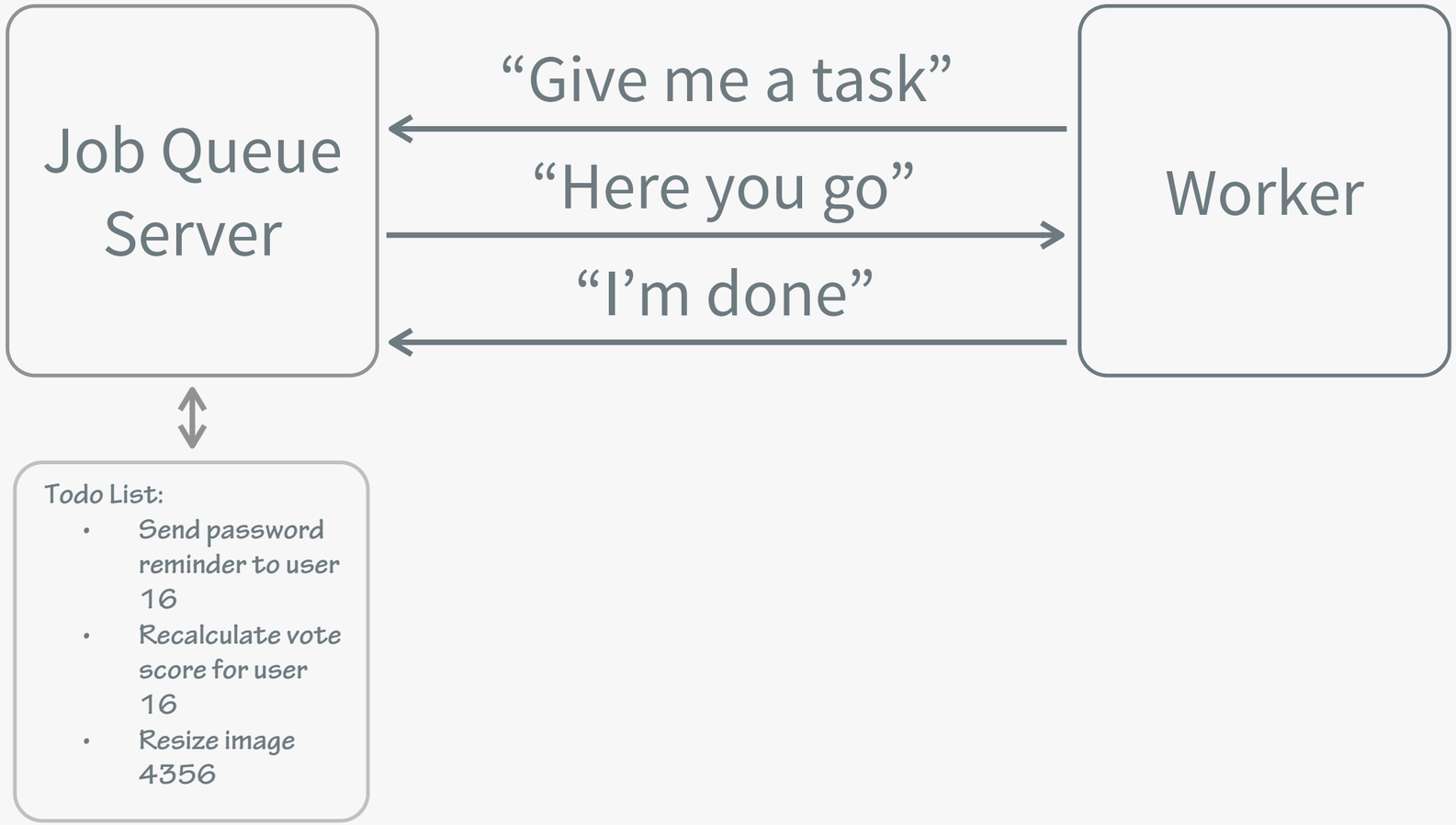
OR

Do slow calculations when displaying the data

Or you can use a Job Queue

Do slow work outside the HTTP Request Handler


Application Server
“Queue this task”
“Got it”
Job Queue Server
Todo List:
Send password reminder to user 16
Recalculate vote score for user 16
Resize image 4356


Job Queue Server
“Give me a task”
“Here you go”
“I’m done”
Worker
Todo List:
Send password reminder to user 16
Recalculate vote score for user 16
Resize image 4356

Gearman

Beanstalkd

Resque

Kafka

Kestrel

Celerey

RabbitMQ

Amazon SQS

Build your own

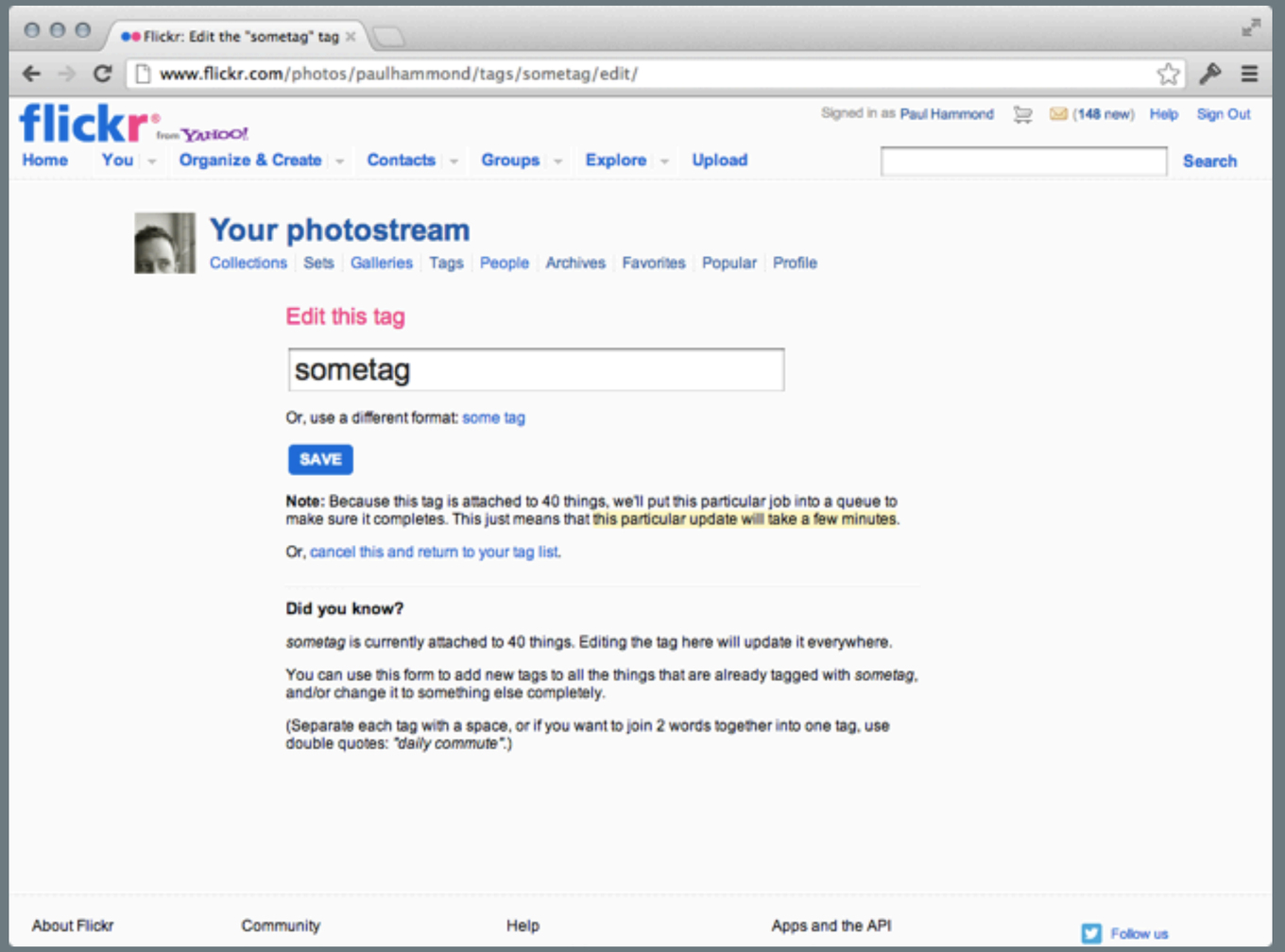
...

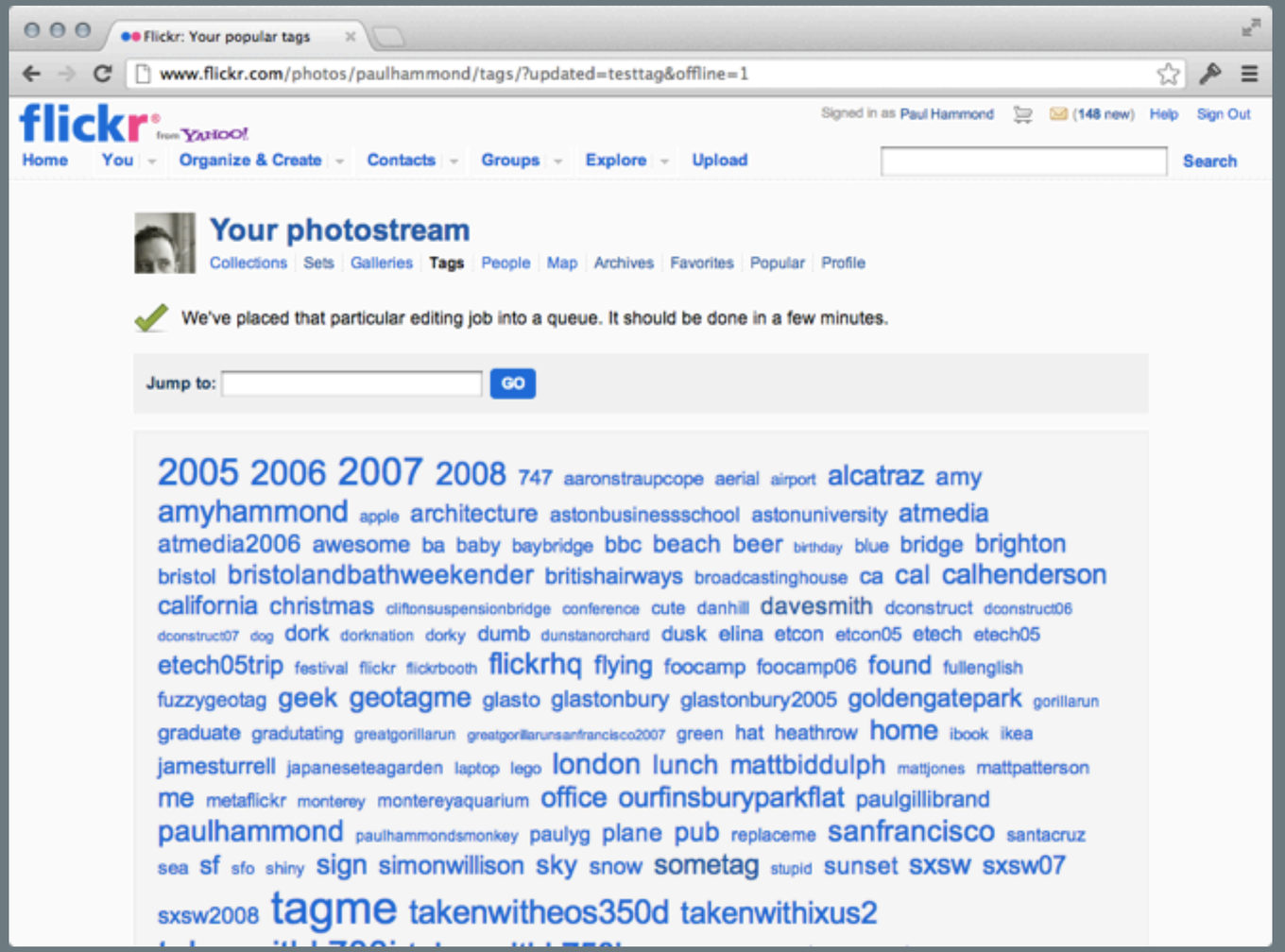
I’m not going to talk about any of these today.

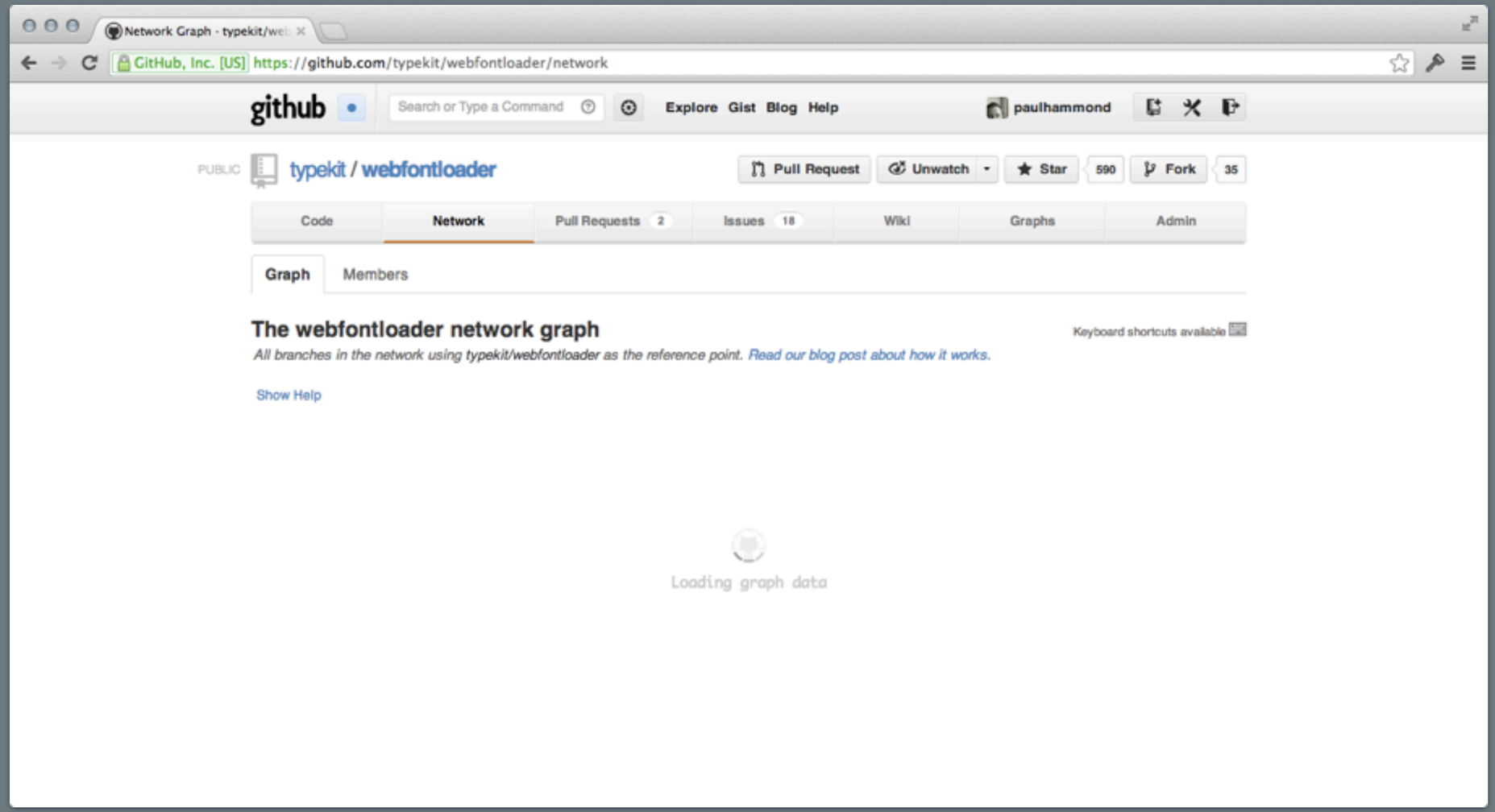
I’m going to talk about everything else you need to know.

#1 TIMTOWTUT

#1 There is more than one way to use them.


Flickr: Edit the "sometag" tag
www.flickr.com/photos/paulhammond/tags/sometag/edit/
flickr from YAHOO!
Signed in as Paul Hammond (148 new) Help Sign Out
Home You Organize & Create Contacts Groups Explore Upload Search
Your photostream
Collections Sets Galleries Tags People Archives Favorites Popular Profile
Edit this tag
sometag
Or, use a different format: some tag
SAVE
Note: Because this tag is attached to 40 things, we'll put this particular job into a queue to make sure it completes. This just means that this particular update will take a few minutes.
Or, cancel this and return to your tag list.
Did you know?
sometag is currently attached to 40 things. Editing the tag here will update it everywhere.
You can use this form to add new tags to all the things that are already tagged with sometag, and/or change it to something else completely.
(Separate each tag with a space, or if you want to join 2 words together into one tag, use double quotes: "daily commute".)
About Flickr Community Help Apps and the API Follow us


Flickr: Your popular tags
www.flickr.com/photos/paulhammond/tags/?updated=testtag&offline=1
flickr from YAHOO!
Signed in as Paul Hammond (148 new) Help Sign Out
Home You Organize & Create Contacts Groups Explore Upload Search
Your photostream
Collections Sets Galleries Tags People Map Archives Favorites Popular Profile
We've placed that particular editing job into a queue. It should be done in a few minutes.
Jump to: GO
2005 2006 2007 2008 747 aaronstraupcope aerial airport alcatraz amy amyhammond apple architecture astonbusinessschool astonuniversity atmedia atmedia2006 awesome ba baby baybridge bbc beach beer birthday blue bridge brighton bristol bristolandbathweekender britishairways broadcastinghouse ca cal calhenderson california christmas cliftonsuspensionbridge conference cute danhill davesmith dconstruct dconstruct06 dconstruct07 dog dork dorknation dorky dumb dunstanorchard dusk elina etcon etcon05 etech etech05 etech05trip festival flickr flickrbooth flickrhq flying foocamp foocamp06 found fullenglish fuzzygeotag geek geotagme glasto glastonbury glastonbury2005 goldengatepark gorillarun graduate gradutating greatgorillarun greatgorillarunsanfrancisco2007 green hat heathrow home ibook ikea jamesturrell japaneseteagarden laptop lego london lunch mattbiddulph mattjones mattpatterson me metaflickr monterey montereyaquarium office ourfinsburyparkflat paulgillibrand paulhammond paulhammondsmonkey paulyg plane pub replaceme sanfrancisco santacruz sea sf sfo shiny sign simonwillison sky snow sometag stupid sunset sxsw sxsw07 sxsw2008 tagme takenwitheos350d takenwithixus2


Network Graph · typekit/web
GitHub, Inc. [US] https://github.com/typekit/webfontloader/network
github
Search or Type a Command
Explore Gist Blog Help
paulhammond
PUBLIC
typekit / webfontloader
Pull Request
Unwatch
Star
590
Fork
35
Code
Network
Pull Requests 2
Issues 18
Wiki
Graphs
Admin
Graph
Members
The webfontloader network graph
Keyboard shortcuts available
All branches in the network using typekit/webfontloader as the reference point. Read our blog post about how it works.
Show Help
Loading graph data

Network Graph · typekit/web

GitHub, Inc. [US] https://github.com/typekit/webfontloader/network

github Search or Type a Command Explore Gist Blog Help paulhammond

PUBLIC typekit / webfontloader

Pull Request Unwatch Star 590 Fork 35

Code Network Pull Requests 2 Issues 18 Wiki Graphs Admin

Graph Members

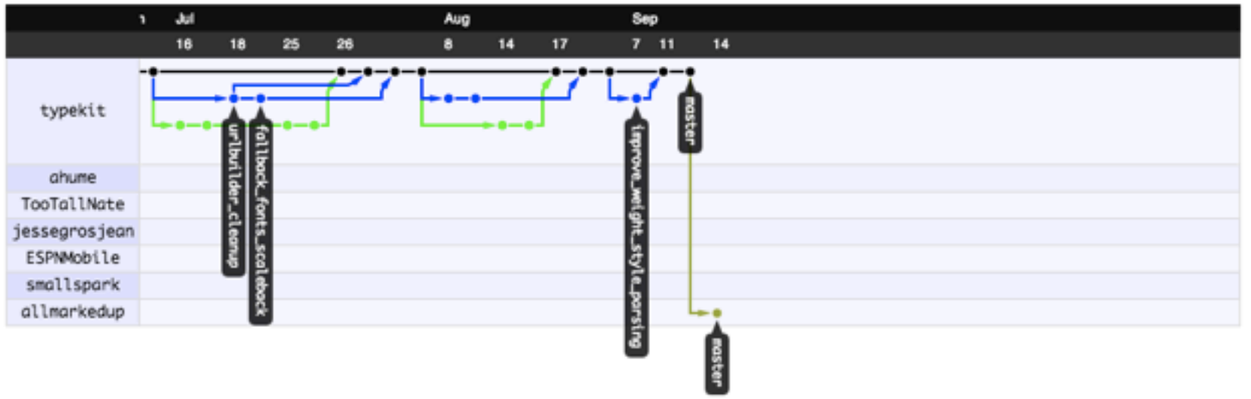
The webfontloader network graph

Keyboard shortcuts available

All branches in the network using typekit/webfontloader as the reference point. Read our blog post about how it works.

Show Help

Last updated: 4 minutes ago



Anything not shown on the
next page load can be silently queued.

Emails, tweets, external apis, webhooks…

User: 50–200 ms

Follower: 1–10 seconds

Stranger: 1–2 minutes

decafbad.com/blog/2008/07/04/queue-everything-and-delight-everyone

Think about the user interface.

#2 Errors happen.

```
class Job::HandleVote
  def perform
    …
    db.select(…)
    …
  end
end
```

What happens when the database connection fails?

```
# deep inside most web frameworks
begin
  handle_http_request
catch exception
  display_internal_error
end
```

If you’re outside the HTTP request handler then you can’t just show an error page.

```
# many queue systems
job = queue.pop
begin
  job.run
catch exception
  log(exception)
  # drop job on floor
end
```

```
# some systems
job = queue.pop
begin
  job.run
catch exception
  log(exception)
  job.requeue
end
```

Errors can occur anywhere

```
# some systems
job = queue.pop
begin
  job.run
catch exception
  log(exception)
  job.requeue
end
```

```
# resilient systems
job = queue.reserve_job
begin
  job.run
  queue.remove(job)
catch exception
  log(exception)
end
```

It is impossible to guarantee a job will be run exactly once.

```
class SendPasswordRemider {
  def perform {
    …
    email.send()
    …
  }
}
```

Job will be run one or zero times

OR

Job will be run one or many times

Reliable job queues are really slow.

Slow reliable queue

AND

Fast unreliable queue

Job will be run one or zero times

OR

Job will be run one or many times

Job will be run one or zero times

OR

Job will be run one or many times

(or possibly zero if things go really wrong)

#3 Idempotency is awesome.

idempotent ≠ nilpotent

```
foo = 1
```

```
foo = foo + 1
```

Reindexing a document
in the search index

Fetching data from a read-only API 

Resizing a photo
(in some cases)

Sending an email 

Posting a tweet 

Writing to an external API

Idempotent means it’s OK if it runs twice

Update canonical source in request.
Then queue an idempotent job to
update denormalized copies.

If a job fails to run it’ll fix itself next time.

#4 Jobs don’t run in order.

No shared state between workers.

“Reindex user 20 from the database”

NOT

“Reindex user 20 with these attributes”

Jobs that create other jobs when finished.

#5 Lock contention hurts.

Sometimes jobs stomp on each other when run in parallel.

So you add locking.

```
class ReindexUser {
  def perform {
    raise LockedError unless lock(user)
    …
  }
}
```

Very common for one user to generate multiple jobs in a short period of time.

Queue churn

Do you really need to lock?

Can your locks be more fine-grained?

#6 Alerting is hard.

Queue length

Jobs run per second

Jobs queued per second

Jobs failed per second

Lock fails per second

(Jobs queued - Jobs run)

Job run time

Job queue time

Job total time

Graph all of these metrics.
They’re really addictive.

Graph all of these metrics.
But don’t alert on most of them.

Job queue alerts aren’t usually actionable.

“Wait for it to go back down”

Thresholds are hard to choose.

Normal usage changes over time.

Job queues provide flexibility.

Time based metrics are most useful.

Find good thresholds through trial and error.

Base thresholds on your SLAs or user experience needs.

“Reindexing jobs cannot queue for more than half a second”

Autoscale capacity based on queue length?

#7 You might not want to use a queue.

If you need a job to run immediately then you don’t need a queue, you need a worker pool.

Connect to a worker over the network

Multi-threaded servers

Event loop based servers

Fork a new process

Needs monitoring and capacity planning

#8 You need to think.

1. There is more than one way to use them.
2. Errors happen.
3. Idempotency is awesome.
4. Jobs don’t run in order.
5. Lock contention hurts.
6. Alerting is hard.
7. You might not want to use a queue.
8. You need to think.

Thank you

Slides will soon be available at
paulhammond.org/2012/queueing