

# Lessons from running a web site

## A short talk in two acts

Roy Smith

S7 Labs

[roy@s7labs.com](mailto:roy@s7labs.com)

# Act 1: Tracking data flow in songza.com

Roy Smith

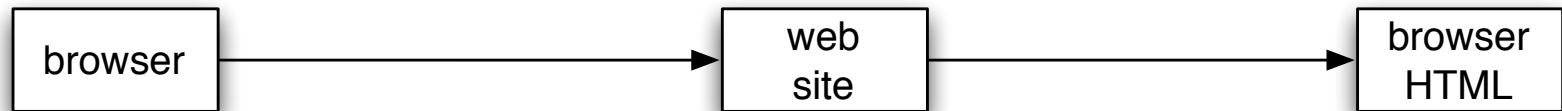
S7 Labs

roy@s7labs.com

# Songza.com architecture

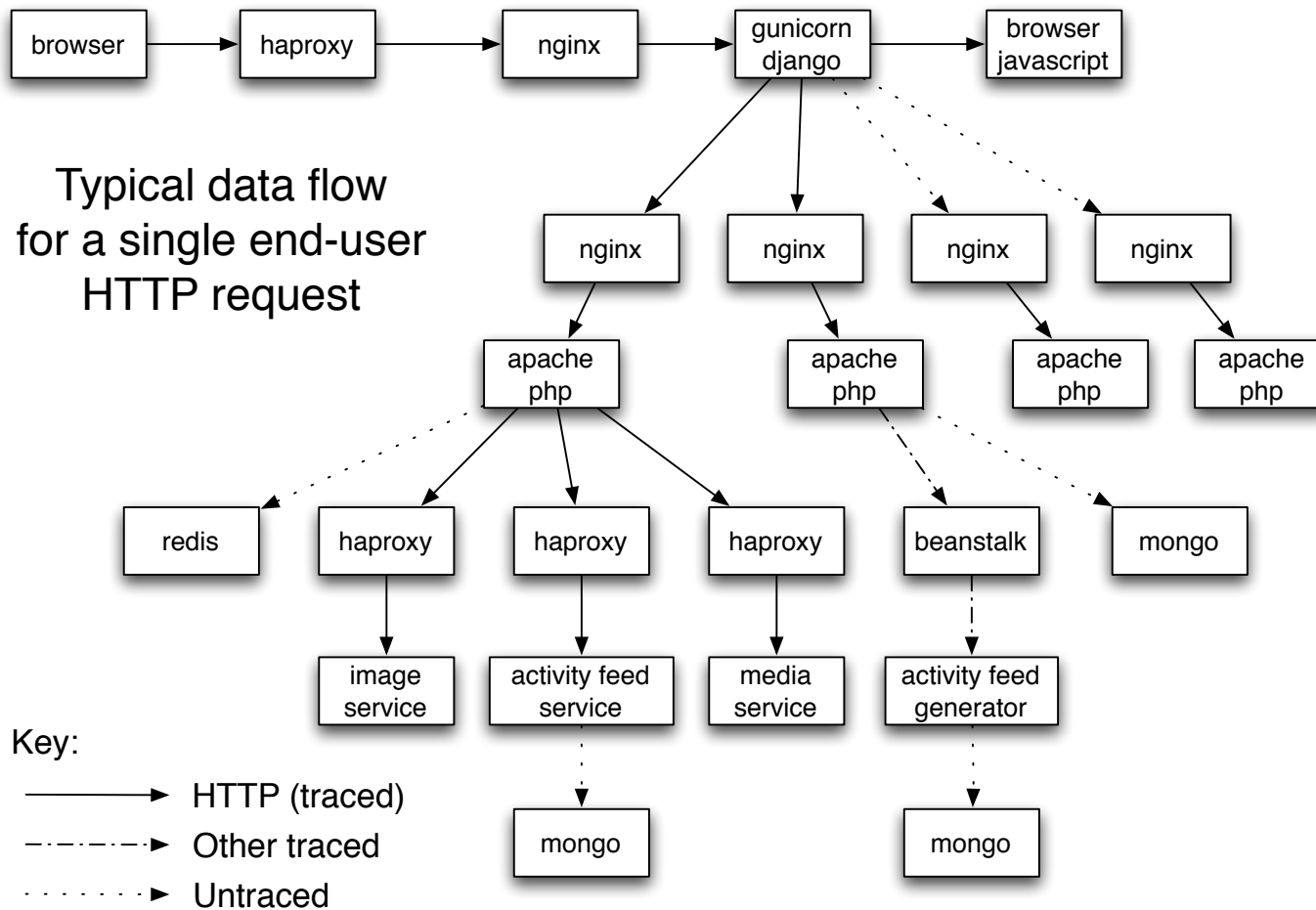
- Like most non-trivial sites, songza.com is not a single process or machine.
- A given page view from a web browser could involve a dozen or more intercommunicating heterogeneous processes.
- HTTP makes convenient duct-tape to hold it all together.

# What the user sees



Typical data flow  
for a single end-user  
HTTP request  
(user perspective)

# What's really happening



# Major components

- haproxy
- nginx
- apache / php
- gunicorn / django
- mongodb
- client-side javascript
- python / tornado services
- beanstalk queue
- memcache
- redis

# Problem:

- Each of these processes logs information about what it's doing and what has gone wrong.
- It can be nearly impossible to track which of gazillions of lines of log output correspond to any single page view.

# Solution:

- Implement unique id tracking.
- haproxy (the normal initial entry point for all traffic) inserts a X-Unique-Request-Id header on all incoming requests.
- If there is already one, it is left alone.



# Implementations

- We're running a custom version of haprox, available at <https://bitbucket.org/wesc/haproxy-xuri>
- Middleware in django pulls id out of request, creates it if missing, and adds it to response.
  - Also shoves it into thread-local storage for the benefit of logging code that doesn't have access to the request object.

- Most major components (apache, django) have similar logic, so requests that bypass haproxy also get assigned ids.
- Everything that knows a request id includes it in every log message it produces.

# User feedback

- We also push the request id out into the HTTP Response headers, so the client-side javascript can find it.
- We use getsatisfaction.com for user feedback, working on a way to tag feedback with the id for end-to-end tacking.

# ID format drivers

- There are many competing design drivers for what an ID should look like. They must be:
  - Unique
  - Compact
  - Easy to compute
  - Can be implemented with portable code
  - Easy to parse (or grep) out of log files
  - Some feel they should encode useful data

- We ended up with:  
CA0A4BD6-4E9F4258-198691
- Which is a reasonable compromise on most of the drivers.
- It's mostly a hash of some stuff, with a counter appended to it. The hash is recomputed when the counter hits some pre-set limit.
- Can be a religious issue.

# The world is not all HTTP

- For things that don't talk HTTP, we find ways to shove the id string into the datastream somehow.
- Example: our beanstalk queues all pass JSON objects, so we just add a `unique_id` field to each queued object.

# Outliers

- Still working on a way to transmit ids to mongo (so far, this has eluded us).
  - Finding a way to do this would be a major win
- No tracing through memcache or redis.
  - These are less of an issue.
  - Unlikely to be a solution due to the nature of these services.

Act 2:  
WTF is running on port 9119?  
Confessions of an HTTP server

Roy Smith  
S7 Labs  
roy@s7labs.com



# Keeping track of services

- We've got a lot of little services running on various ports
- Sometimes it gets confusing
- We have all of our HTTP services respond to a /stats route and confess a bunch of interesting information about themselves:

# Output for Humans

```
$ curl 'localhost:9120/stats?format=text'  
{'base_dir': '/home/roy/songza',  
  'build_type': 'development',  
  'command': ['./feed_mill.py', '-v'],  
  'cwd': '/home/roy/songza/feed',  
  'host': 'dev1.songza.com',  
  'pid': 21911,  
  'sock_addr': '127.0.0.1:9120',  
  'user': 'roy'}
```

# Output for computers

- Default output format is JSON, so tools can do neat stuff:

```
$ ./service-check.py
```

```
FAIL - emproxy (localhost:9195): <urlopen error [Errno 111] Connection refused>
```

```
OK - fb_import_d (localhost:9121): (127.0.0.1:9121, from /home/roy/songza)
```

```
OK - xaps (dev1.songza.com:9196): (10.203.66.57:9196, from /home/roy/songza)
```

```
OK - quad (localhost:9119): (127.0.0.1:9119, from /home/roy/songza)
```

```
OK - suggest (localhost:9198): (127.0.0.1:9198, from /home/roy/songza)
```

```
OK - feed_mill (localhost:9120): (127.0.0.1:9120, from /home/roy/songza)
```

```
OK - medianet (app3.songza.com:9194): (10.209.26.95:9194, from /home/songza/  
deploy/rel-2011-10-14)
```

```
OK - digest (localhost:9199): (127.0.0.1:9199, from /home/roy/songza)
```

```
OK - redis (localhost)
```

```
OK - feed_mill (seems to be talking to the right database)
```

# Who is S7 Labs?

- Hybrid software consultancy / incubator.
- We do most of the development work for songza.com.
- Other work in progress
  - <http://treshr.com>
  - <http://remixt.us>
- We're hiring!
- <http://www.s7labs.com/>

