NODE.JS IN PRODUCTION

Dienstag, 31. August 2010

COMMUNITY

- lomography.com (shameless plug)
- they upload their photos (~5000 per day)
- we show them on our site

MANY DIFFERENT SIZES

- we need photos in all kind of sizes (thumbnail, grid, detail etc.)
- we don’t know all needed sizes in advance
  - not every photo in every size
- we need to resize them on demand

ON DEMAND RESIZING

- 1st attempt aka “lets start”
  - sinatra/passenger application
    - queue resize job and process in background
    - show “smaller version”
      - sometimes thumbnail

ON DEMAND RESIZING

- 2nd attempt aka “the worst idea i’ve ever had”
  - sinatra/passenger application
    - resize photo and deliver
      - blocks the entire server fo

40 instances

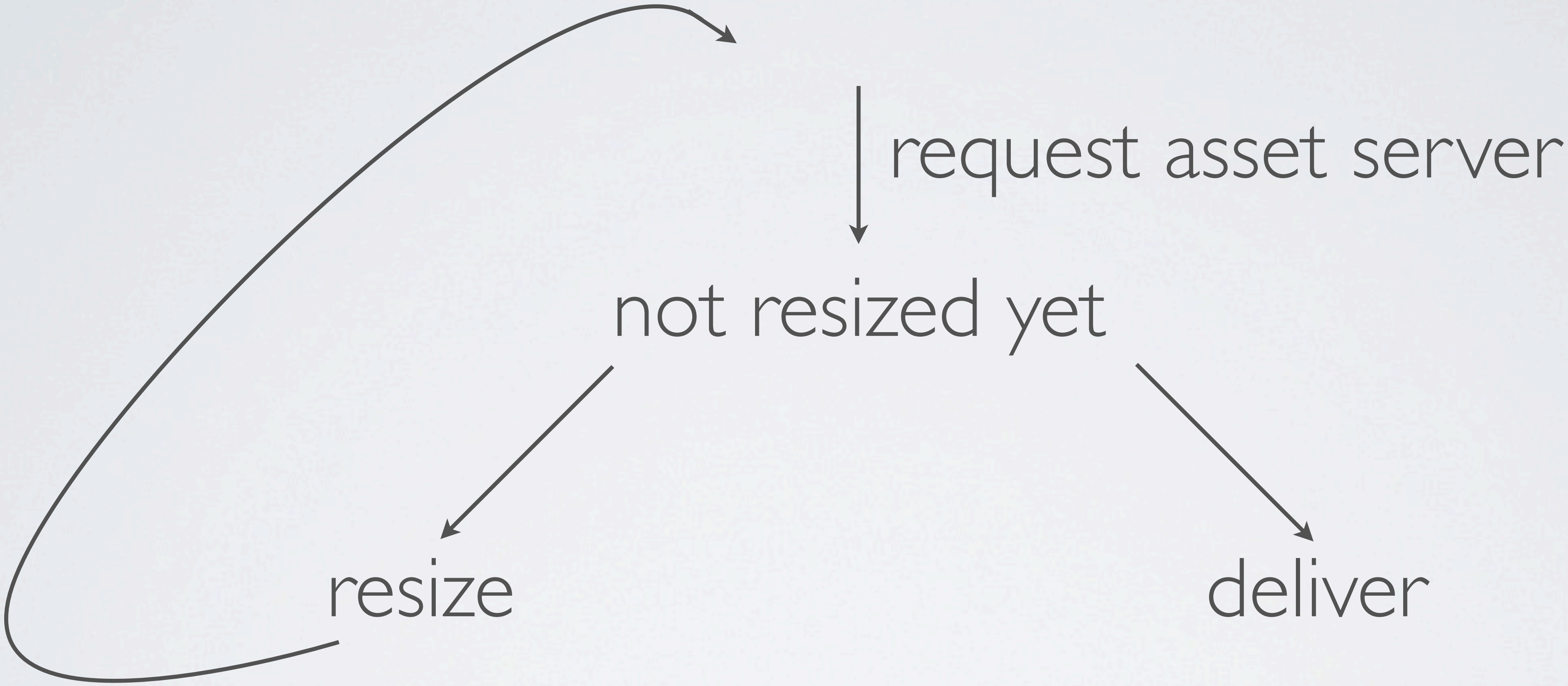
blocks normal phot
delivery

ASYNC TO THE RESCUE

- i tried async sinatra with eventmachine
- can’t read my own code after 6 hours
- node to the rescue
  - easier to write async code in javascript

ON DEMAND RESIZING

- 3rd attempt aka “node rules them all”
    - node application
        - redirect to node server
        - resize photo
        - redirect back



request asset server

not resized yet

resize

deliver

WHY NODE?

- a shitload of IO
  - check if file exists on disk (shared network device) (200 ms)
  - check if file exists on amazon S3 (upto 1 sec)
    - download original from Amazon S3 (upto 2 sec)

SLOW START

- installed node
- built first prototype (happy face)
- tried some test "suites"
  - unable to install a single one
- plain javascript file test.js
  - "node test.js"

GETTING READY “TESTING”

- installed on linux (our servers run linux)
- crashed immediately (sad face)

m(

WHATS WRONG?

```
process.addListener('uncaughtException', function (err) {

  console.log('Caught exception: ' + sys.inspect(err));

  response.simpleText(500, "Internal Error");
});
```

READ THE CHANGELOG

- mac homebrew version 0.1.100
- debian package version 0.1.99
- “Introduce console.log() and friends.”

m(

CALCULATE SHA1

```
var calculateSha1ForFile = function(filename, callback) {
  var readStream = fs.createReadStream(filename);
  var sha1ForFile = crypto.createHash('sha1');

  readStream.addListener('error', function(err) {
    callback(err);
  });
  readStream.addListener('data', function(data) {
    sha1ForFile.update(data);
  });
  readStream.addListener('close', function() {
    callback(false, sha1ForFile.digest('hex'));
  });
};
```

CONVERT BINARY TO BINARY

```
readStream.addListener('data', function(data) {
  sha1ForFile.update(data);
});
```

```
readStream.addListener('data', function(data) {
  sha1ForFile.update(data.toString('binary'));
});
```

```
      m(
```

TOO MANY OPEN FILES

- http request library does not close connections (on linux)
- node IPv4 TCP **unknown protocol** (ESTABLISHED)

REVERSE PROXY

- no security analysis (until now)
- do not connect node directly to the net
- use nginx or lighttpd (or some other evented webserver)
  - do some filtering

RELEASE CYCLE

- new version every 1-2 weeks
- upgrade is a must
  - a lot of bugfixes
- you need tests! really!

MONITORING

- node crashes sometimes (i don’t know why)
- install some kind of monitoring (god or monit)
  - restart node after crash

WAS IT WORTH?

- YES! node rocks!
- hard to write synchronous code
- resize photos and delivery normal thumbnails without 10000 instances aka “less memory”
- able to resize 5 photos per second
    - cpu utilization ~90%
- only 120 lines of javascript

@beanieboi

- http://www.lomography.com/photos/6035921
- http://www.lomography.com/photos/11460174
- http://www.lomography.com/photos/6177726