Chrome to Chrome PouchDB

Randall Leeds
“tilgovi”

Overview

- Background
- Motivation
- Approach
- Future

Overview

- Background
- Motivation
- Approach
- Future

The Peer-to-Peer Web

The Peer-to-Peer What Now!?

“A peer-to-peer (abbreviated to P2P) computer network is one in which each computer in the network can act as a client or server for the other computers in the network”

– https://en.wikipedia.org/wiki/Peer-to-peer

The (Early) Peer-to-Peer Web

- Early Web was peer-to-peer (kinda)
  - Universities own and manage Web content
  - University researchers access and publish content
  - Mainframe is server is client

What the %$*$ happened?

“The Cloud” happened...

Why clouds?

- Server software got complicated
- Server software got expensive
- The edge of the network is flakey
- Collective Intelligence, Big Data, blah blah... aggregate all the things... whatever...

The Cloud is a Hack

The Cloud is a Hack

- Connectivity
  - We're too far from one another!
- Scale
  - We want to aggregate our data!
- Management
  - We don't want to do any work!

Fog Computing

"... and Its Role in the Internet of Things"

Bonomi, Milito, Zhu, Addepalli

Cisco Systems, Inc.

2013

Fog Computing

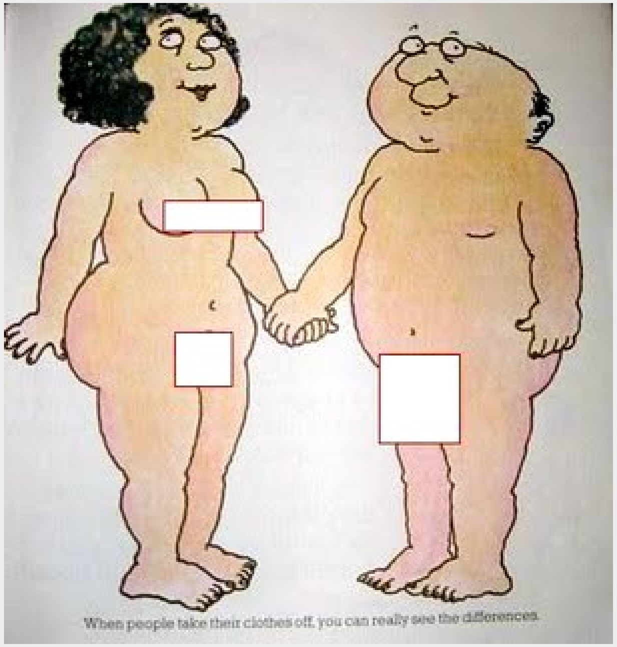
- Low latency and location awareness
- Wide-spread geographical distribution
- Mobility
- Very large number of nodes
- Predominant role of wireless access
- Strong presence of streaming and real time applications
- Heterogeneity

Overview

- Background
- Motivation
- Approach
- Future

Where did I come from?

Where did I come from?



When people take their clothes off, you can really see the differences.

Where did I come from?

- CouchDB Cluster/Partition GsoC
  - Proposal no longer available online...
  - ... which is a good thing.
- Meebo / CouchDB-Lounge
  - No longer developed...
  - ... which is a good thing.

Hey, wait!

That's all “cloud stuff”!

The “Indie Web”

(You've probably never heard of it...)

The “Indie Web”

Own your data.

Rather than posting content on third-party silos of content, we should all own the content we're creating. Publish on your own domain, and syndicate out to silos.

This is the basis of the "Indie Web" movement.

Source: http://indiewebcamp.com/

It's hard being cool

It's hard being cool

- The Indie Web has fallen behind
- Running a server is hard.
  - Like, really, really hard.
  - No, really.

Example: Diaspora

Preparing your system

In order to run Diaspora, you will need to install the following dependencies (specific instructions follow):

- Build tools - Packages needed to compile the components that follow.
- Ruby - The Ruby programming language. (1.9.2 or later).
- RubyGems - A package manager for Ruby code that we use to download libraries ("gems") that Diaspora uses.
- Bundler - A gem management tool for Ruby projects.
- MySQL - Backend storage engine.
- Or: PostgreSQL - Backend storage engine.
- SQLite3 - Relational database management system
- OpenSSL - An encryption library.
- libcurl - A library to make HTTP requests (and much more).
- ImageMagick - An image processing library we use to resize uploaded photos.
- Git - A version control system, which you will need to download the Diaspora source code from GitHub.
- Redis - A persistent key-value store that we use via Resque for background job processing.

We Can Do (Much) Better

CouchApp (Persistence + Application Server)
PouchDB (CouchDB in the browser)

Introducing: PouchDB Server

The Couch is the Browser is the Couch
This ultimate “unhosting”

PouchDB Server

Overview

- Background
- Motivation
- Approach
- Future

PouchDB Server

- Chrome 24 (currently dev) Packaged Apps
  - chrome.experimental.socket.*
  - chrome.experimental.udp.*
- PouchDB

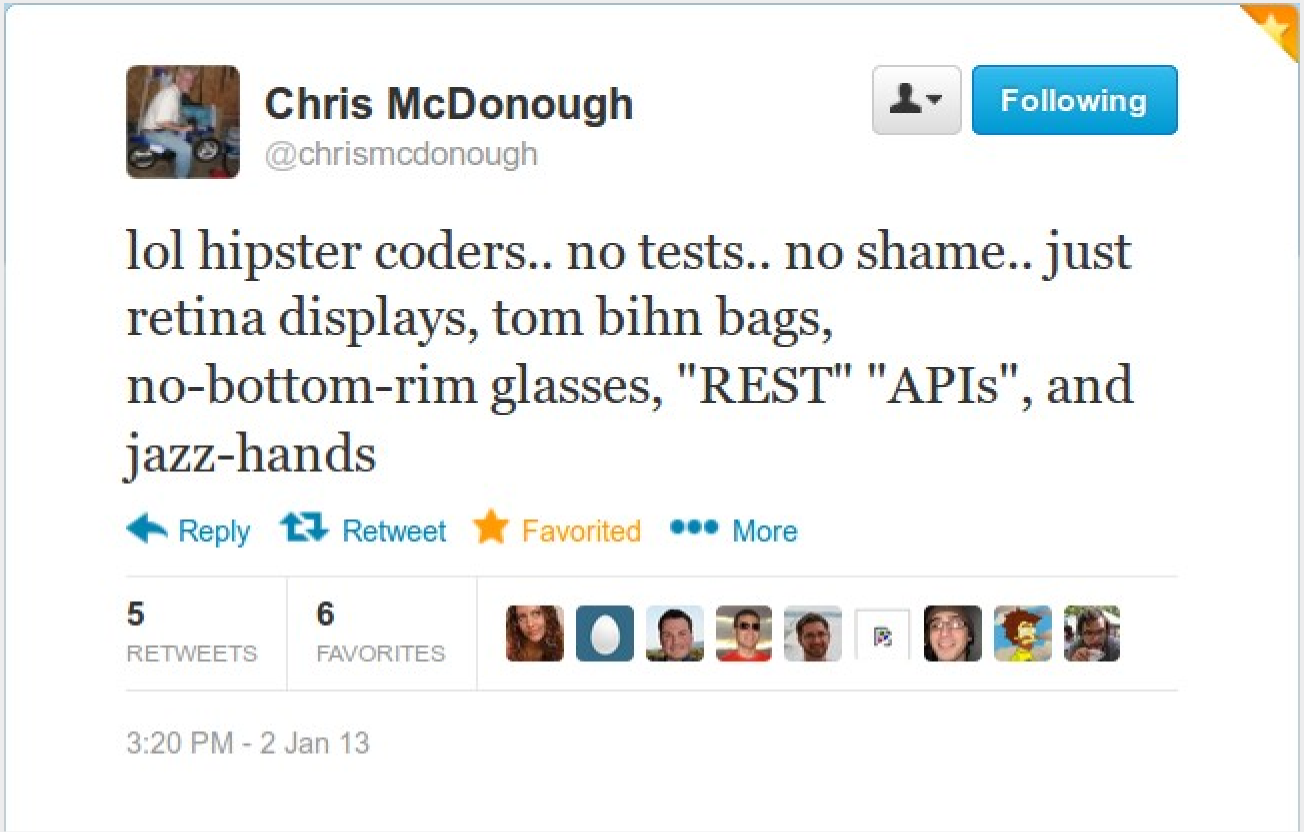
How is this possible?

Mostly, it's kinda easy.

(For Some Value of Easy)

Protip: Steal everything

- Don't reinvent the wheel
- Use what people love
- There are no tests like the tests someone else wrote for you



Chris McDonough
@chrismcdonough

Following

lol hipster coders.. no tests.. no shame.. just retina displays, tom bihn bags, no-bottom-rim glasses, "REST" "APIs", and jazz-hands

5 RETWEETS

6 FAVORITES

3:20 PM - 2 Jan 13

Moar Javascript

- Node.JS has a stable HTTP and Socket API
- Node.JS is JavaScript

Can we put Node.JS
in the browser???

WTF is this?!?
ROFLRAZZI.COM

What can we reuse?

Node.JS is largely written *in JavaScript!*

What can we reuse?

net.js and http.js

Enter Browserify

Browserify

- Write code like you are using node modules
- Run a server-side build step
- Get a bundled JavaScript file

<Jazz Hands>

Problem

Doesn't include 'net' or 'http'

Introducing: chromify

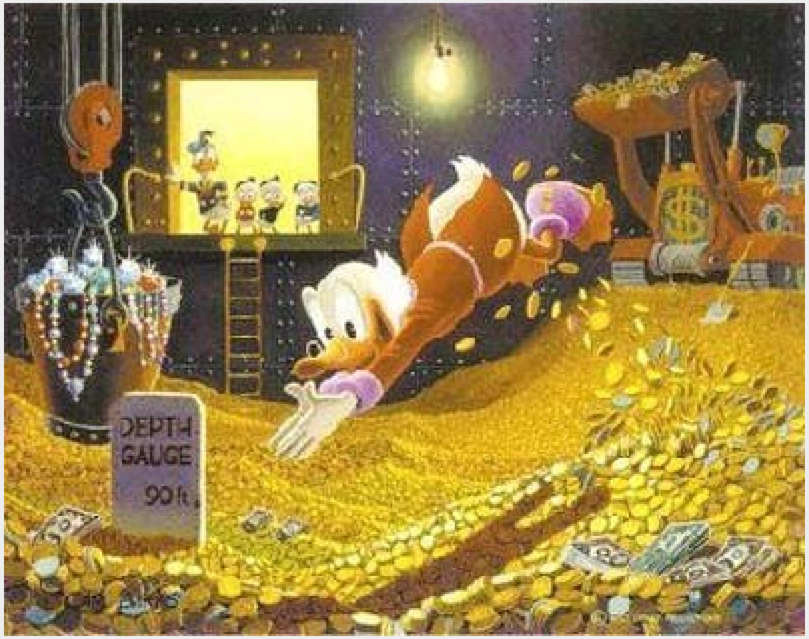
- Browserify plugin adds node's 'net' and 'http' modules
- Unmodified code from node.js core!
- Fake C bindings!
- But... we need an HTTP parser!

Enter Emscripten

- LLVM → JS compiler
  - (Really, how cool is that?)
- joyent/http-parser
  - No dependencies
  - No system calls

Game Plan

- Step 1:
  - Monkey patch process.binding()
    - Fake 'tcp_wrap', 'cares_wrap' in JS
    - Make 'http_parser' wrapper around emscripten parser
- Step 2: <hella jazz hands>
- Step 3: profit!


DEPTH
GAUGE
90 ft.

Overview

- Background
- Motivation
- Approach
- Future

What's next?

- Fix all the super-jank
- More exposure
  - TCP over UDP/STUN
    - Overlays
  - Proxy gateways
- Other browsers
- PouchApps
- Parties. All the time.

Thank You

https://github.com/tilgovi/chromify

https://github.com/tilgovi/pouchdb

(pouch-server branch)

https://github.com/tilgovi/http-parser

(es-parser branch)