Performance

CS 130, Lecture 14

Let’s check in

https://forms.gle/vAtgTkuY3LMVS9os7

A word - What’s it like working with other people’s code?

A word - What's the program with the worst performance you have ever used?

A tweet - How do you plan out your own work capacity?

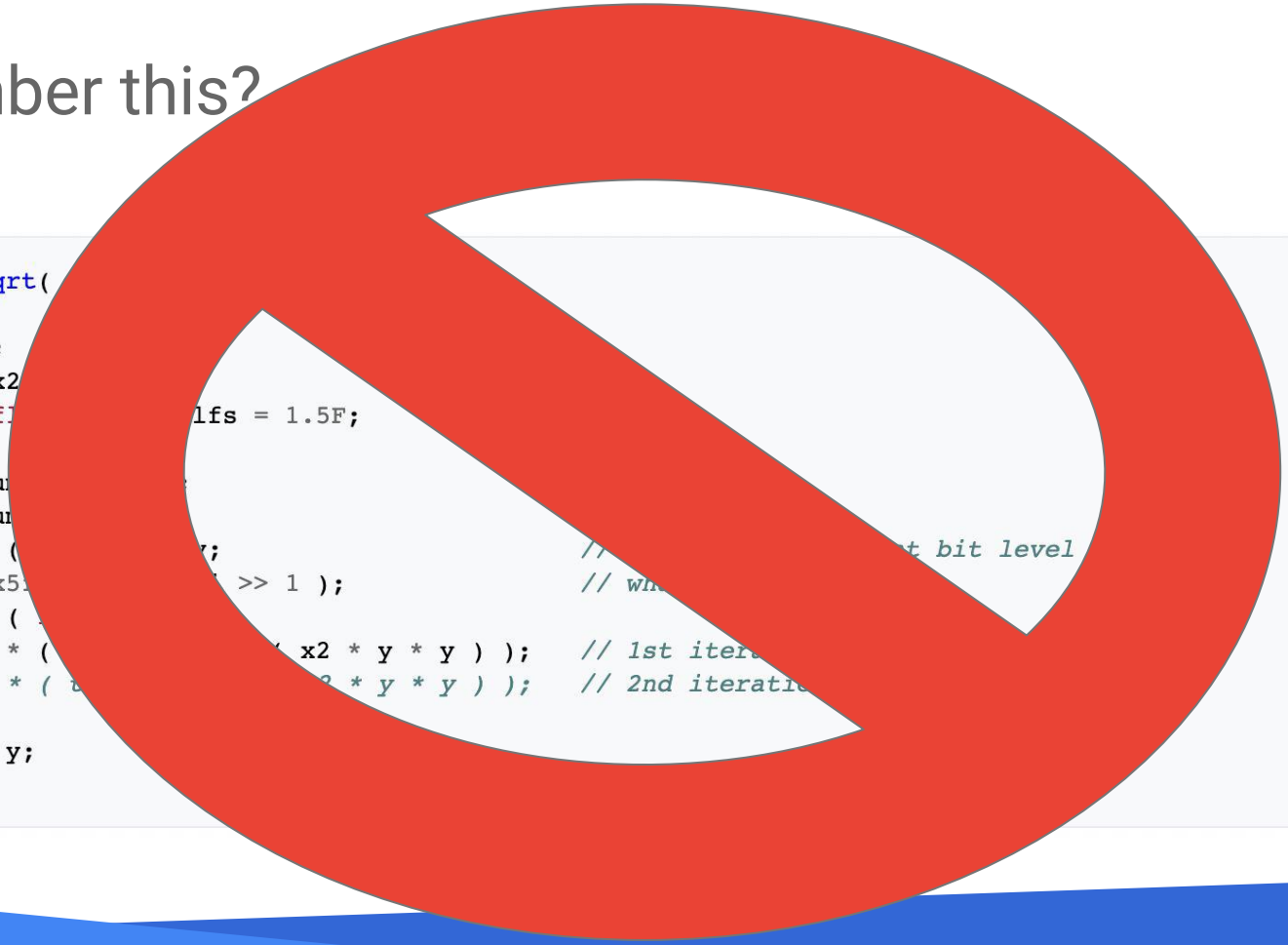
Remember this?

```
float Q_rsqrt( float number )
{
    long i;
    float x2, y;
    const float threehalfs = 1.5F;

    x2 = number * 0.5F;
    y  = number;
    i  = * ( long * ) &y;                       // evil floating point bit level hacking
    i  = 0x5f3759df - ( i >> 1 );               // what the fuck?
    y  = * ( float * ) &i;
    y  = y * ( threehalfs - ( x2 * y * y ) );   // 1st iteration
//  y  = y * ( threehalfs - ( x2 * y * y ) );   // 2nd iteration, this can be removed

    return y;
}
```

Remember this?



```
float Q_rsqrt(
{
    long i;
    float x2
    const fl                lfs = 1.5F;

    x2 = nu
    y  = nu
    i  = * (                ;                    //                    t bit level
    i  = 0x5                >> 1 );              // wh
    y  = * (
    y  = y * (                  x2 * y * y ) );  // 1st iter
//  y  = y * (                  * y * y ) );     // 2nd iterati

    return y;
}
```

Webserver Performance

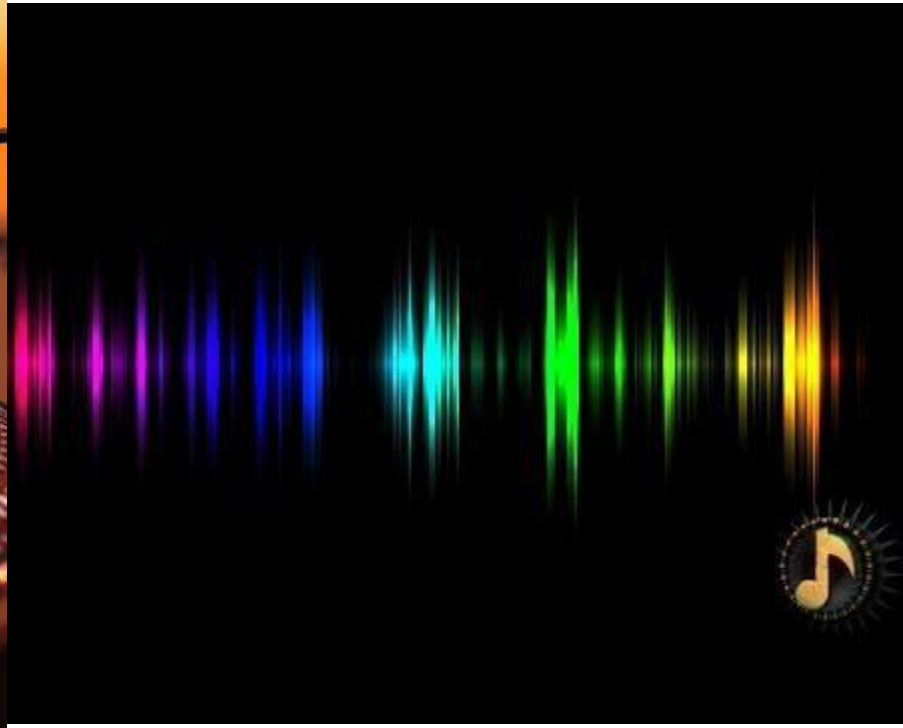
What are the right performance metrics for a webserver?

Webserver Performance

1. QPS
2. Latency
3. Memory footprint
4. Concurrent connections
5. Bandwidth

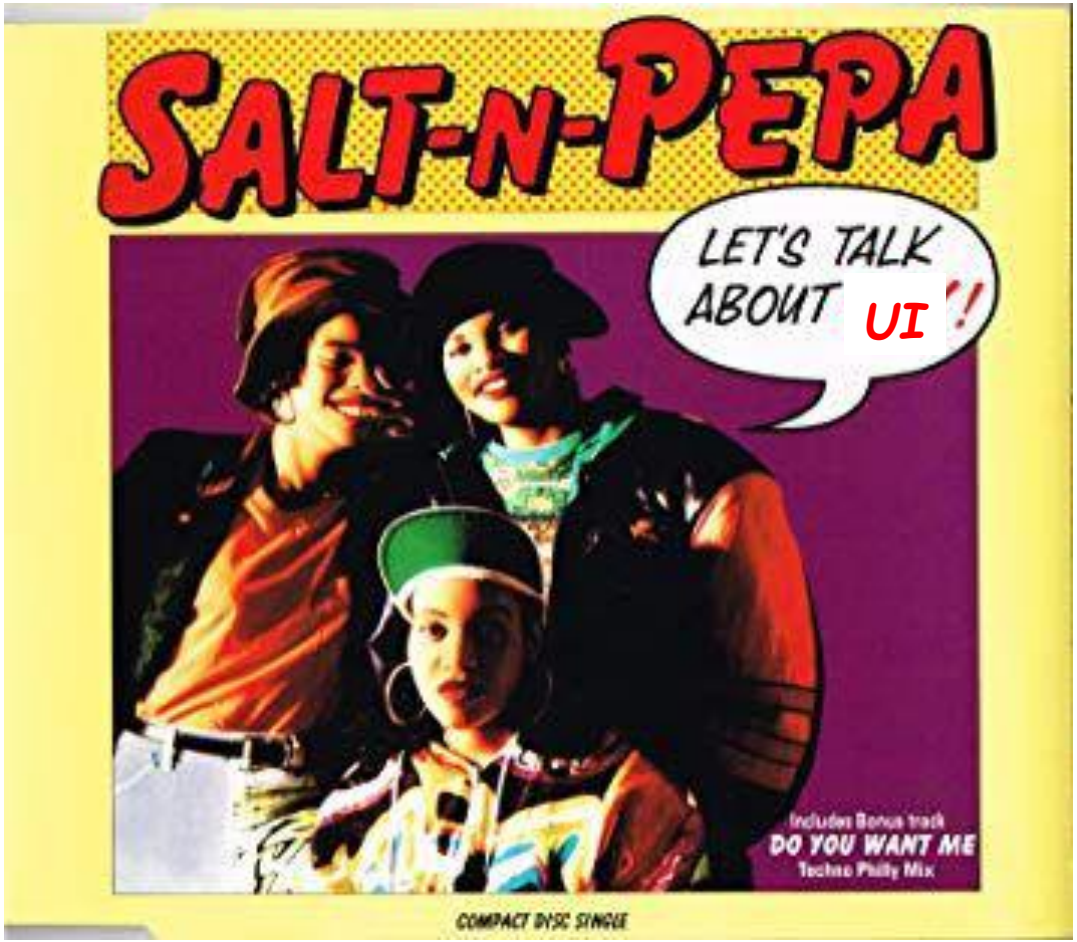
~~Webserver Performance~~

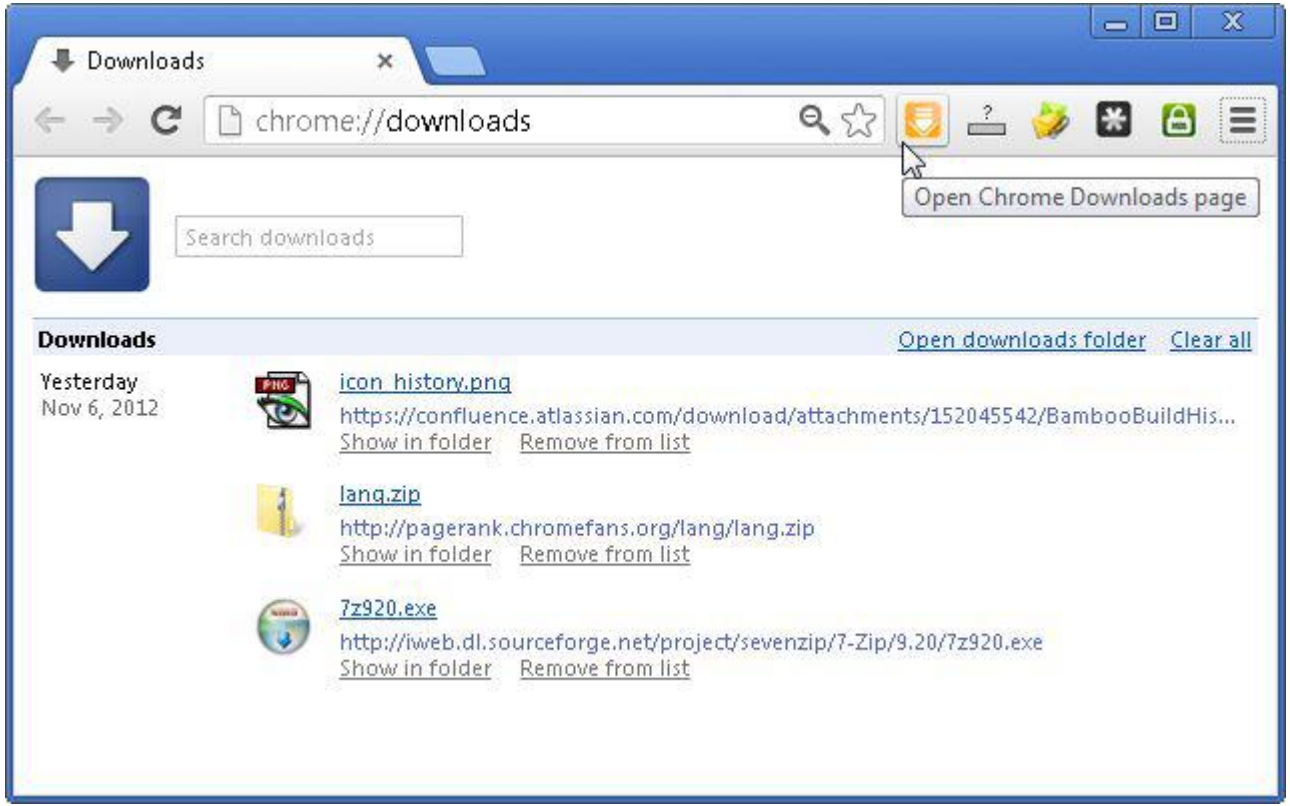
1. ~~QPS~~
2. ~~Latency~~
3. ~~Memory footprint~~
4. ~~Concurrent connections~~
5. ~~Bandwidth~~

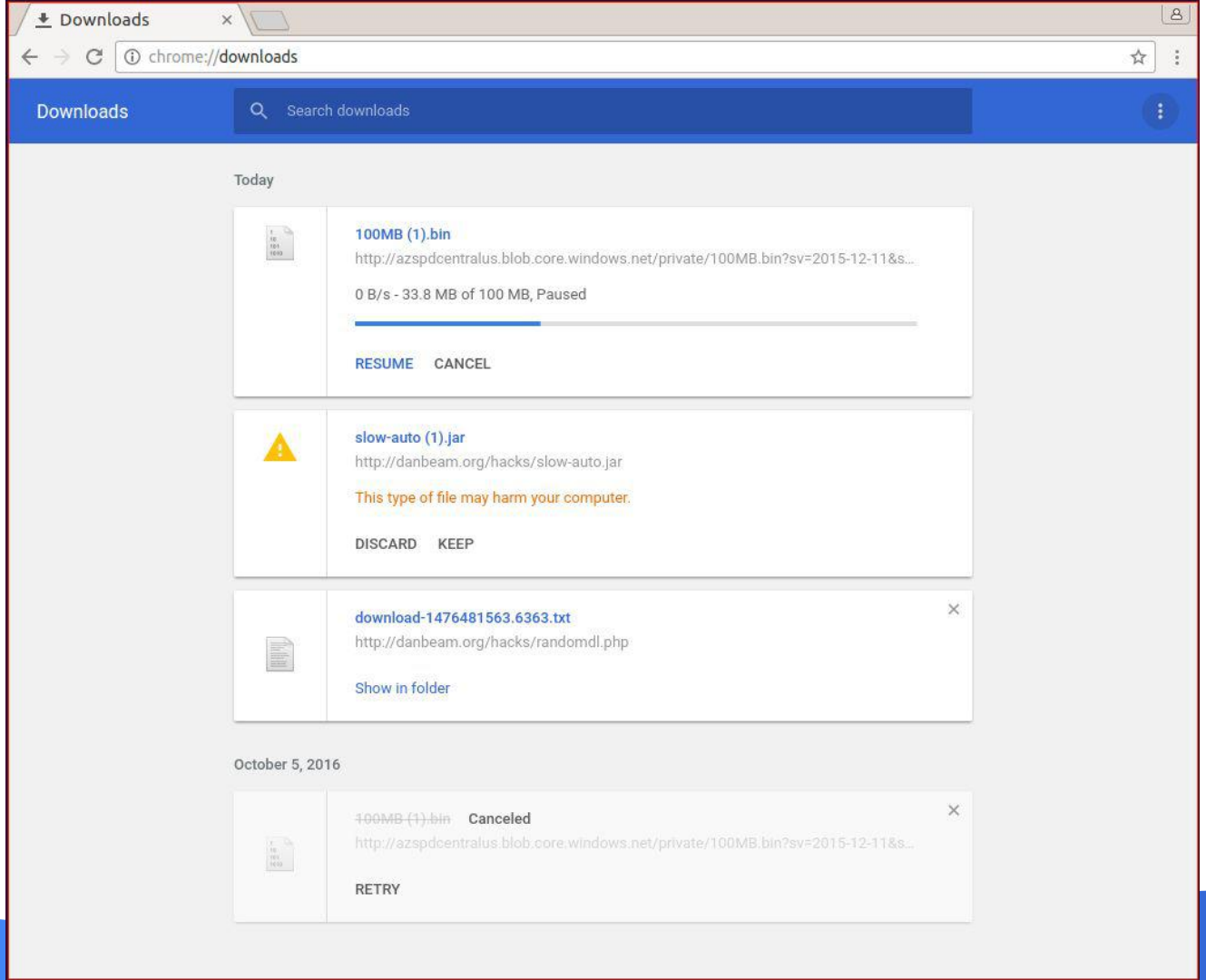


We’re tired of talking about web servers

And there’s not enough time to optimize your web servers anyways, sooo...

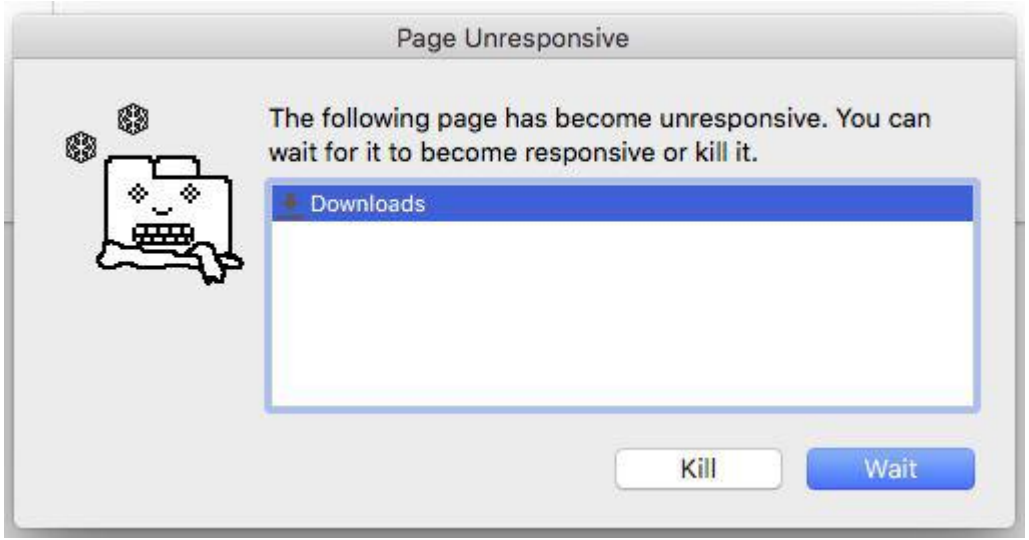

SALT-N-PEPA
LET'S TALK ABOUT UI !!
Includes Bonus track
DO YOU WANT ME
Techno Philly Mix
COMPACT DISC SINGLE


Downloads
chrome://downloads
Open Chrome Downloads page
Search downloads
Downloads
Open downloads folder
Clear all
Yesterday
Nov 6, 2012
icon_history.png
https://confluence.atlassian.com/download/attachments/152045542/BambooBuildHis...
Show in folder
Remove from list
lang.zip
http://pagerank.chromefans.org/lang/lang.zip
Show in folder
Remove from list
7z920.exe
http://iweb.dl.sourceforge.net/project/sevenzip/7-Zip/9.20/7z920.exe
Show in folder
Remove from list


Downloads
chrome://downloads
Downloads
Search downloads
Today
100MB (1).bin
http://azspdcentralus.blob.core.windows.net/private/100MB.bin?sv=2015-12-11&s...
0 B/s - 33.8 MB of 100 MB, Paused
RESUME
CANCEL
slow-auto (1).jar
http://danbeam.org/hacks/slow-auto.jar
This type of file may harm your computer.
DISCARD
KEEP
download-1476481563.6363.txt
http://danbeam.org/hacks/randomdl.php
Show in folder
October 5, 2016
100MB (1).bin
Canceled
http://azspdcentralus.blob.core.windows.net/private/100MB.bin?sv=2015-12-11&s...
RETRY

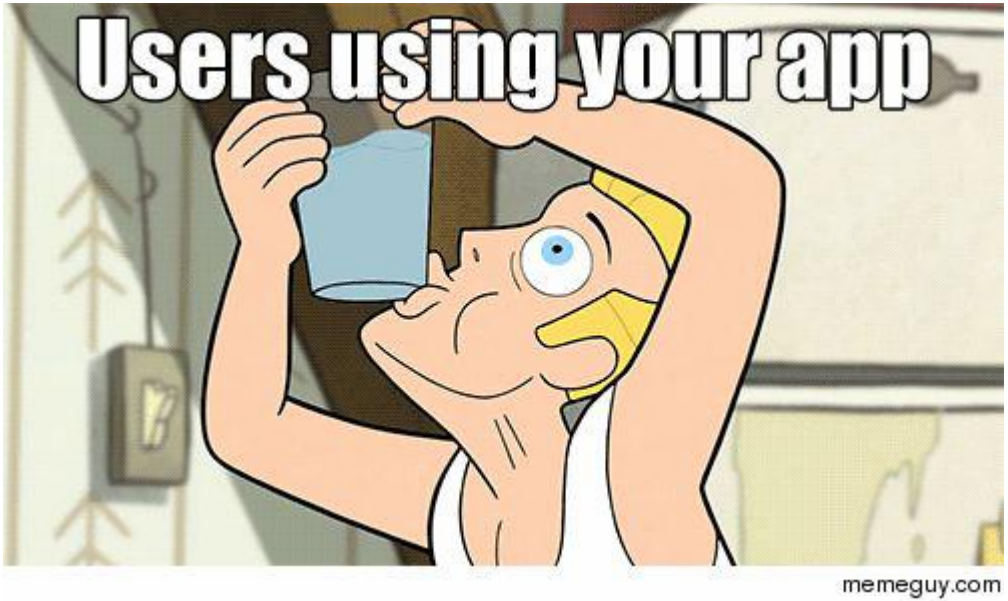
We ported an existing code base

We didn’t see performance problems, but users did




NOW WATCH THE BARS...
You're Holding it
Antenna Problems?
You're holding it wrong.
FULL BARS!
JESUSDIAZ·GIZMODO.COM
THEY'RE ALL GO
COOL HUH?
HOLDING
IT WRONG
LOSES NETWORK SIGNA
ALL THE TIME
YOU'RE HOLDING IT

Because this does seriously happen



We pretty much ignore this, saying
“We’ll make it fast later. How hard could it be?”

Old code

- Was fairly simple
- Used raw HTML/CSS/JS and had no dependencies
- Showed capped/fixed number of downloads
- Loaded/scrolled very quickly

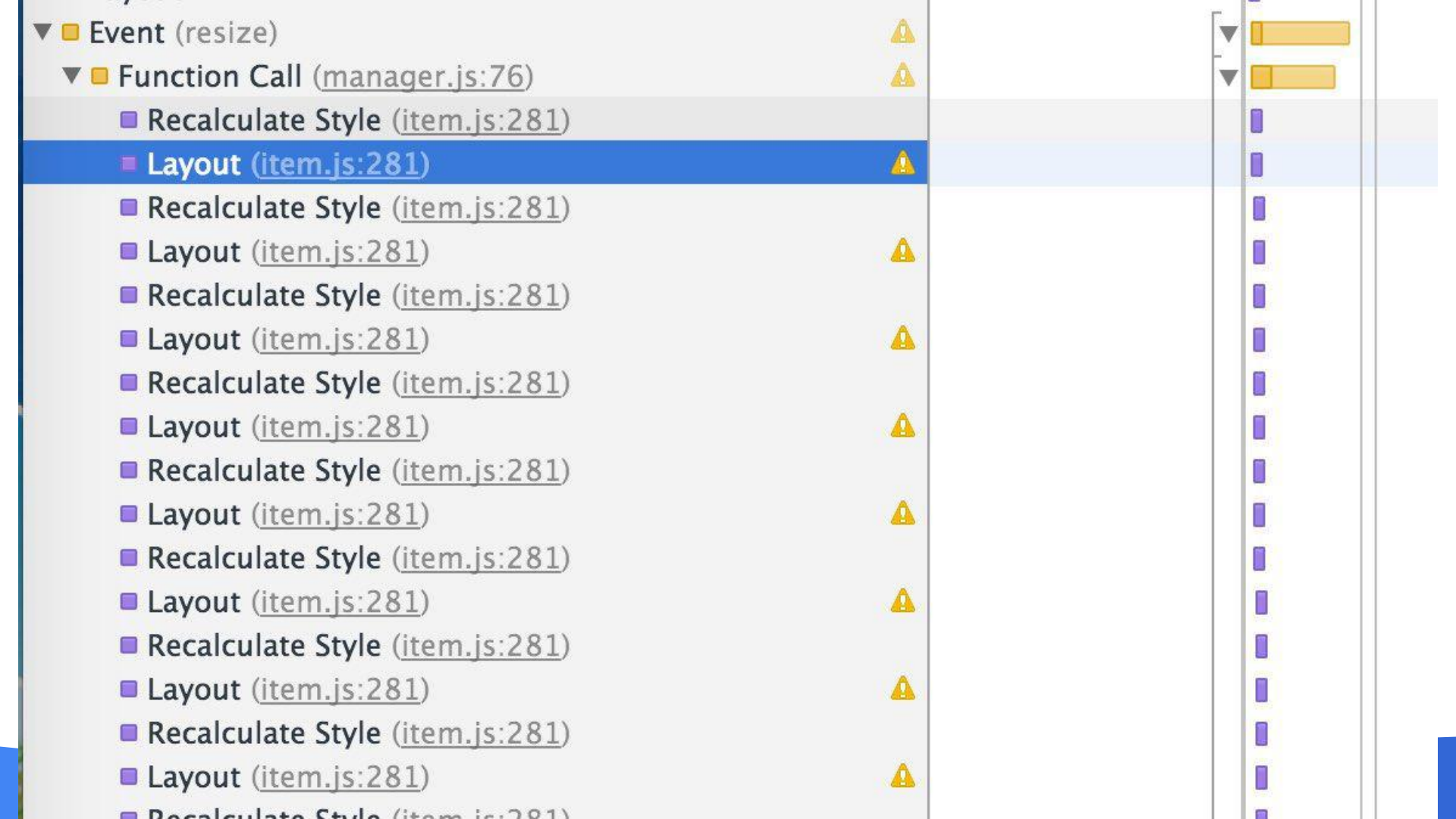
New code

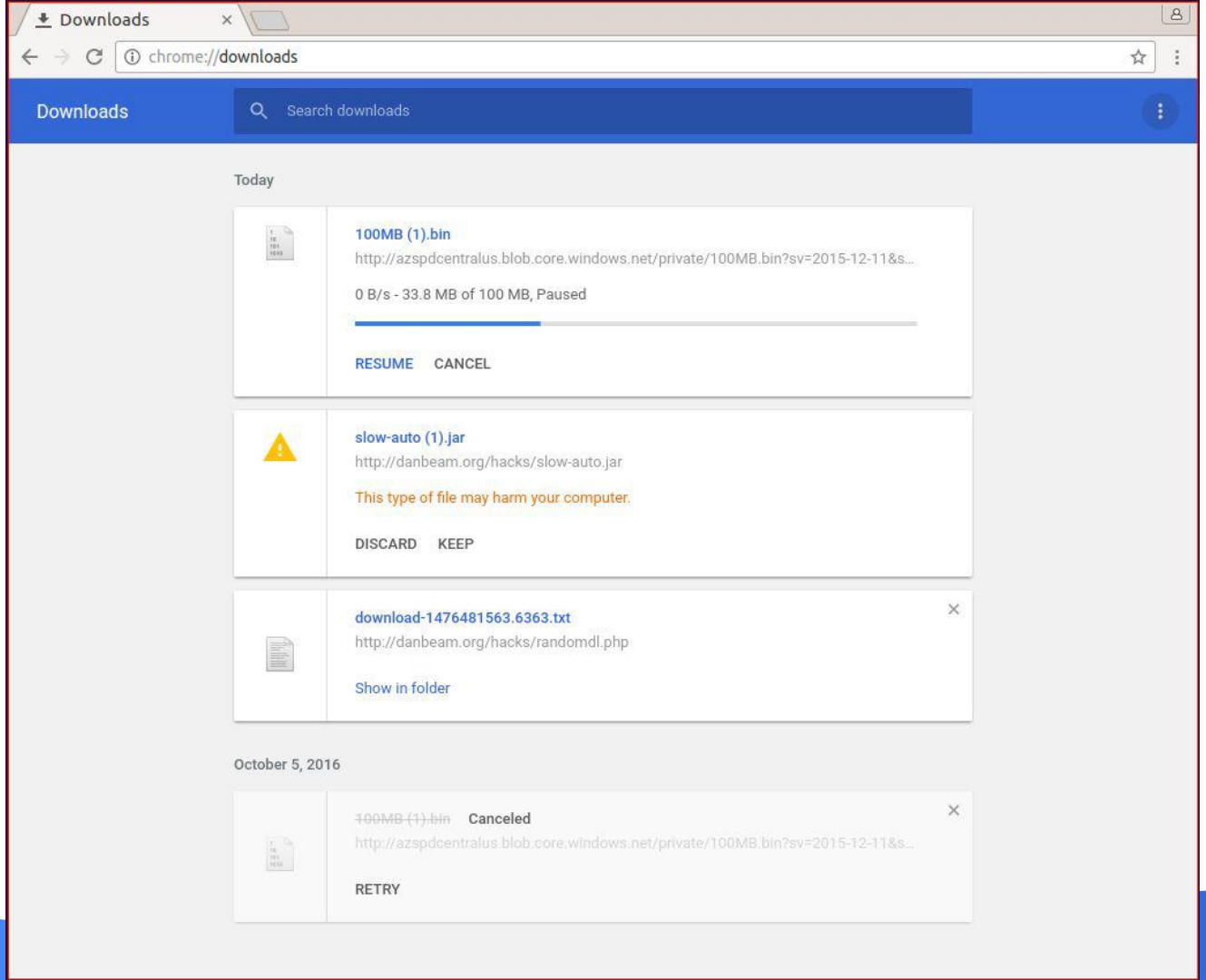
- Was similar complexity at first glance, but...
- Had some dependencies (Polymer library + elements)
- Showed capped/fixed number of downloads
- Loaded quickly for us!

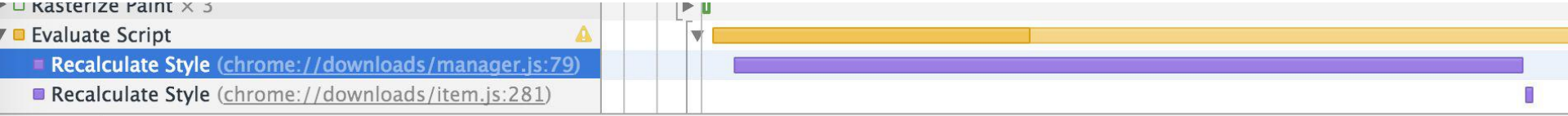
But then...

- One of the lead developers of WebKit (now Blink) filed all these bugs
  - Polymer downloads page takes 800ms to load and then 6 seconds to show content
  - Polymer download scrollbar width code causes flood of style recalcs and layouts
  - Polymer downloads page does huge 500ms-1.8 second style recalcs and runs script for hundreds of ms when using search box

They’re definitely not “holding it wrong”.


Event (resize)
Function Call (manager.js:76)
Recalculate Style (item.js:281)
Layout (item.js:281)
Recalculate Style (item.js:281)
Layout (item.js:281)
Recalculate Style (item.js:281)
Layout (item.js:281)
Recalculate Style (item.js:281)
Layout (item.js:281)
Recalculate Style (item.js:281)
Layout (item.js:281)
Recalculate Style (item.js:281)
Layout (item.js:281)
Recalculate Style (item.js:281)
Layout (item.js:281)
Recalculate Style (item.js:281)
Layout (item.js:281)


Downloads
chrome://downloads
Downloads
Search downloads
Today
100MB (1).bin
http://azspdcentralus.blob.core.windows.net/private/100MB.bin?sv=2015-12-11&s...
0 B/s - 33.8 MB of 100 MB, Paused
RESUME
CANCEL
slow-auto (1).jar
http://danbeam.org/hacks/slow-auto.jar
This type of file may harm your computer.
DISCARD
KEEP
download-1476481563.6363.txt
http://danbeam.org/hacks/randomdl.php
Show in folder
October 5, 2016
100MB (1).bin
Canceled
http://azspdcentralus.blob.core.windows.net/private/100MB.bin?sv=2015-12-11&s...
RETRY



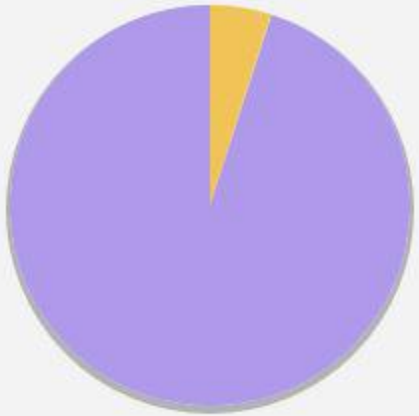
Summary | Aggregated Details

Type	Recalculate Style
Total Time	400.03 ms
Self Time	400.03 ms
Elements affected	1198

Evaluate Script

725.10 ms

161.08 ms



1.76 s

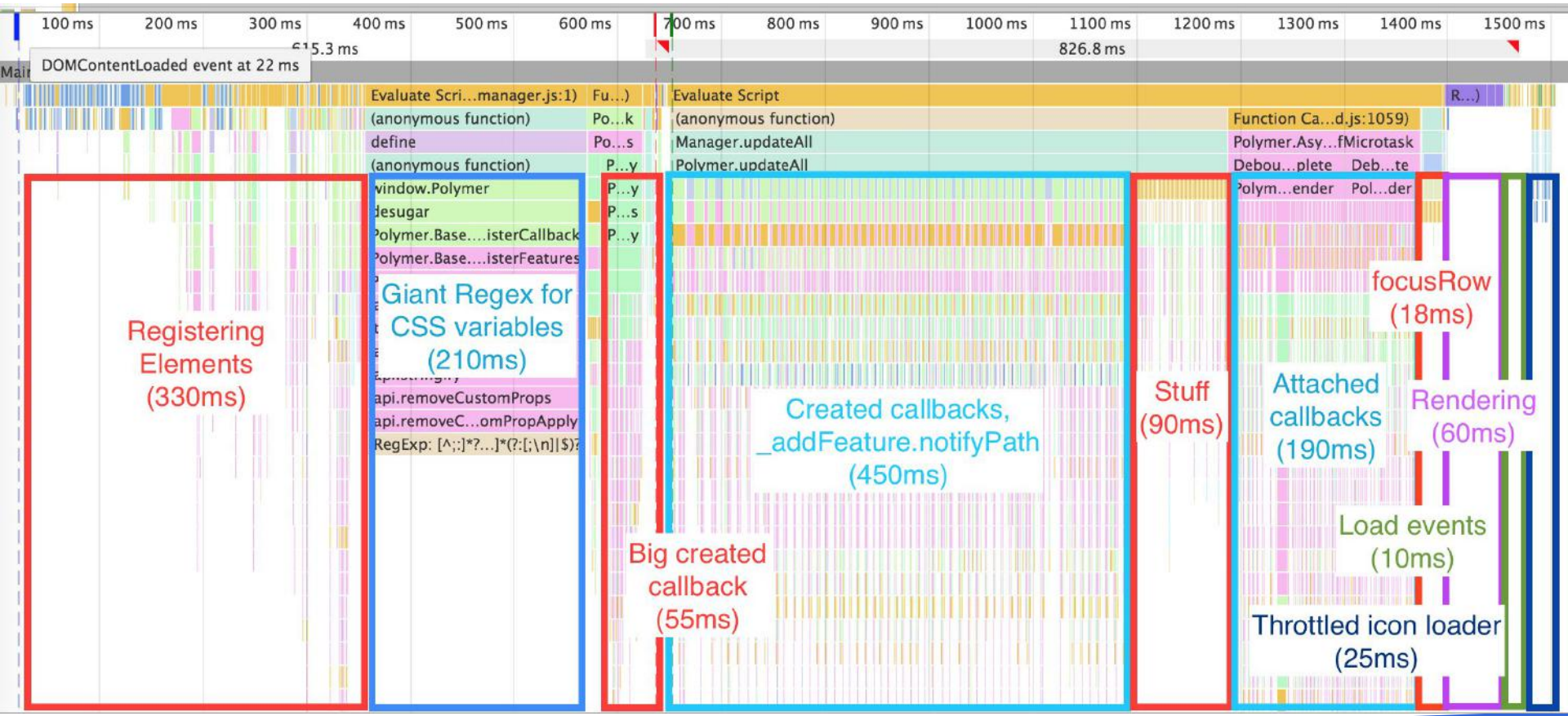
86.13 ms Scripting (Self)

1.57 ms Scripting (Children)

1.67 s Rendering

Scripting (Self)

Rendering


100 ms
200 ms
300 ms
400 ms
500 ms
600 ms
700 ms
800 ms
900 ms
1000 ms
1100 ms
1200 ms
1300 ms
1400 ms
1500 ms
615.3 ms
826.8 ms
DOMContentLoaded event at 22 ms
Evaluate Scri…manager.js:1)
Fu…)
Evaluate Script
R…)
(anonymous function)
Po…k
(anonymous function)
Function Ca…d.js:1059)
define
Po…s
Manager.updateAll
Polymer.Asy…fMicrotask
(anonymous function)
P…y
Polymer.updateAll
Debou…plete
Deb…te
vindow.Polymer
P…y
Polym…ender
Pol…der
desugar
P…s
Polymer.Base….isterCallback
P…y
Polymer.Base….isterFeatures
Giant Regex for CSS variables (210ms)
Registering Elements (330ms)
api.removeCustomProps
api.removeC…omPropApply
RegExp: [^;:]*?…]*(?:[;\n]|$)?
Big created callback (55ms)
Created callbacks, _addFeature.notifyPath (450ms)
Stuff (90ms)
Attached callbacks (190ms)
focusRow (18ms)
Rendering (60ms)
Load events (10ms)
Throttled icon loader (25ms)

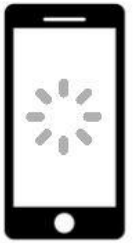

So this is
definitely our
fault
at this point...

Before you start optimizing: when is it fast **enough**?

Determine exit criteria

- It's fast enough when
  - ~~I~~ users can scroll at 60fps
  - It loads instantly (~130ms)
  - It works on crappy hardware

Google's web performance model: RAIL



Response

Animation

Idle

Load

How do we keep it fast?

Performance monitoring

Many ways to do this. They all boil down to:

- Agree upon/create some semi-stable testing environment[s]
- Gather inputs that might affect performance
  - poll version control for new revisions
  - use new versions of libraries
  - anything else that might affect end-user performance (i.e. simulate a bad network)
- On input changes: run the app 1 (or N) times while measuring stuff
  - CPU, GPU, power, time
- Archive and track the results over time
- Know quickly when/if your app got slow (and why)

Reproducibility is a huge problem

(which is why I was ready to ship a slow UI to users without knowing it…)

If you've ever written a UI:
congrats, you are the proud owner of a distributed system!

On the server, you ***can*** just throw money at the problem

You've replaced engineering ingenuity with barrels of cash. Now what?

UIs are a different kind of distributed system

They run on many **very different** machines … that you don't have control over

- Different hardware
  - Drastically different amounts of RAM/CPU/disk
- Different host software
  - OS
  - Current load
- Different user-specific data
  - you probably wont have access to this
  - it might be hard to generate/reproduce
- Different network bandwidth/latency/reliability

Gather information from the “wild” (Remote profiling)

If you can, profile users in the most realistic way possible. Chrome has:

- chrome://tracing
- inspector
- probably lots of other stuff…

But these require us (Google) to be able to reproduce locally (on fast machines/network/with our own data) or users to send us traces.

Chrome created “UMA” to upload metrics collected on local user computers

Gather information from the “wild” (Remote profiling)

So we added remotely logged metrics for load time and other things, and guided our optimizations based on this.

High latency turnaround process (for us it was like ~weeks):

1. Make “total guess” change
2. Launch it
3. Put it front of more and more users
4. Gather data
5. Interpret data
6. Repeat

And lastly, if you can’t make things actually fast...

Pretend they are.

Pretend to be fast

Trick the human brain

- Loading shims
- Progressive enhancement
- Progress bars (meh)
- Humans are only so fast (~16.7ms frames are good enough for average eye)
- Humans detect instant at ~130ms (ish, but this is slowly changing)

Are you actually gonna tell us how you made

Chrome’s UI fast?

POLYMER SUMMIT

JOIN THE DISCUSSION AT

So far so good...

No! Make the **code** lazy! (oh!)

- Virtualization
  - How “optimizable” is your UI? How much is showing on the “critical path”?
- Defer resource allocation until acquisition

This takes work and generally adds complexity.

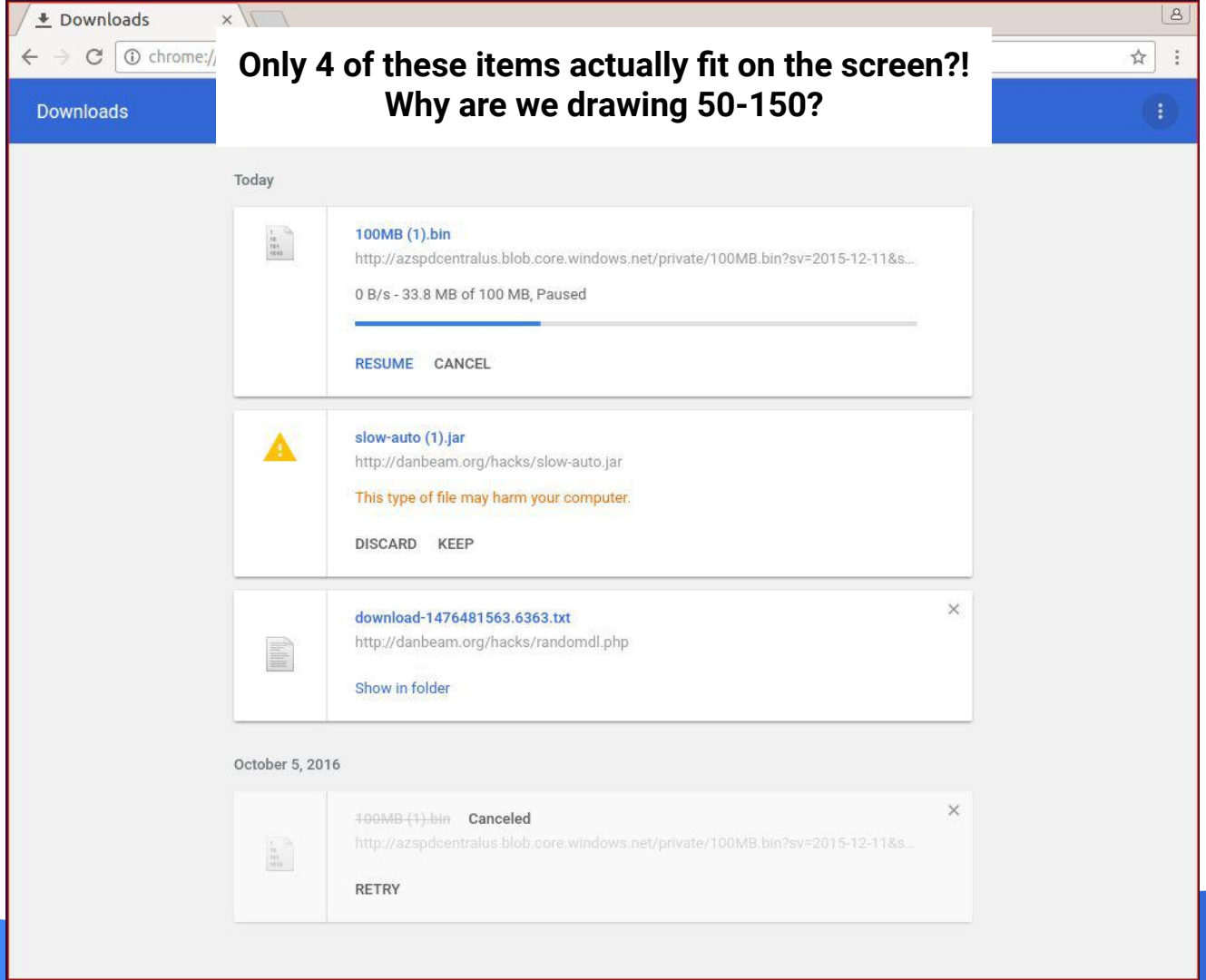
1. Reduce the

Conditional rendering

T = N x K

POLYMER SUMMIT

JOIN THE DISCUSSION AT


Downloads
chrome://
Only 4 of these items actually fit on the screen?!
Why are we drawing 50-150?
Downloads
Today
100MB (1).bin
http://azspdcentralus.blob.core.windows.net/private/100MB.bin?sv=2015-12-11&s...
0 B/s - 33.8 MB of 100 MB, Paused
RESUME
CANCEL
slow-auto (1).jar
http://danbeam.org/hacks/slow-auto.jar
This type of file may harm your computer.
DISCARD
KEEP
download-1476481563.6363.txt
http://danbeam.org/hacks/randomdl.php
Show in folder
October 5, 2016
100MB (1).bin
Canceled
http://azspdcentralus.blob.core.windows.net/private/100MB.bin?sv=2015-12-11&s...
RETRY

Settings

Search settings

People

Dan Beam
Syncing to dbeam@google.com

Turn off

Managed by **google.com**

Sync
On - custom settings

Manage your Google Account

Chrome name and picture

Import bookmarks and settings

Autofill

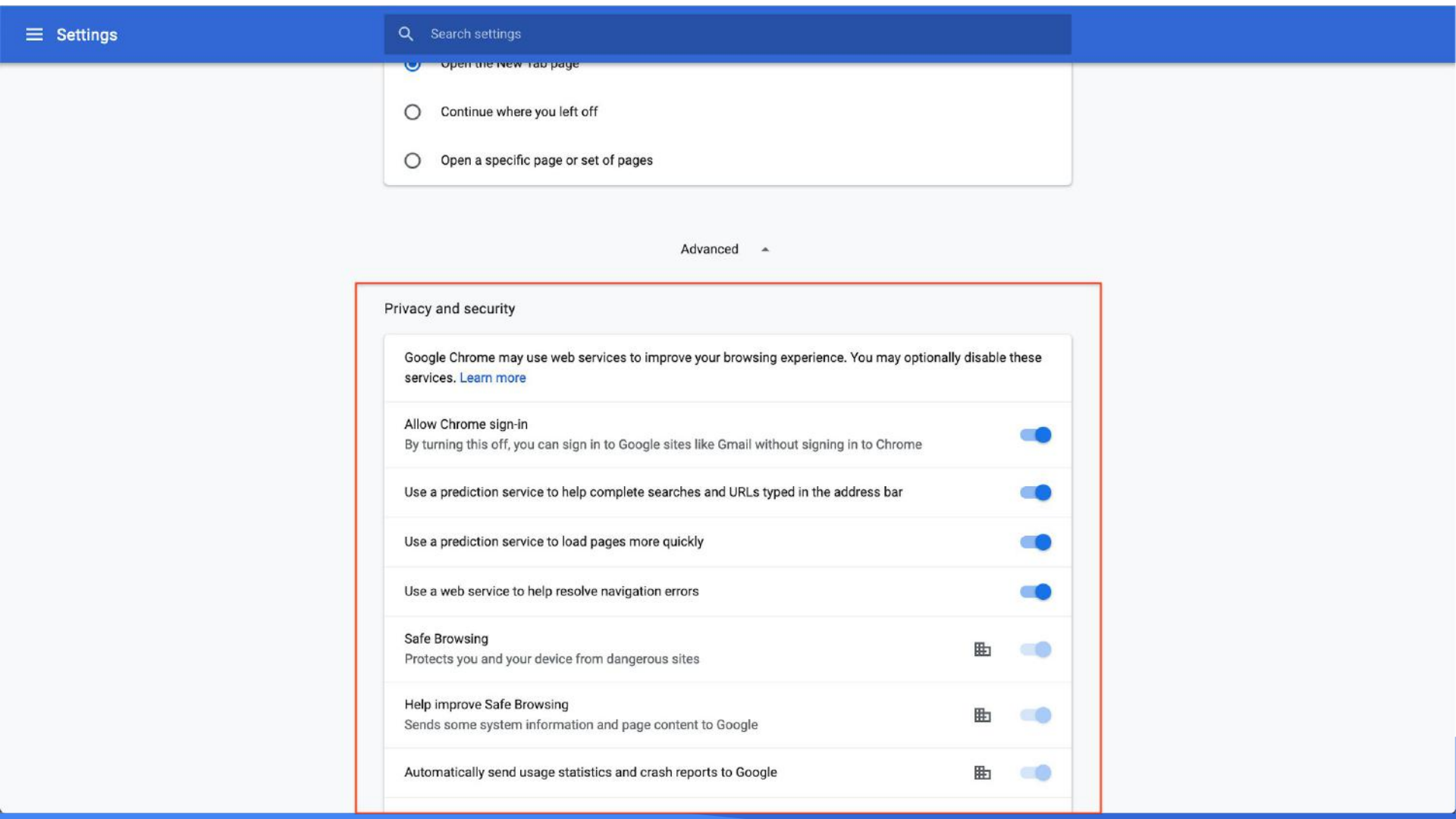
Passwords

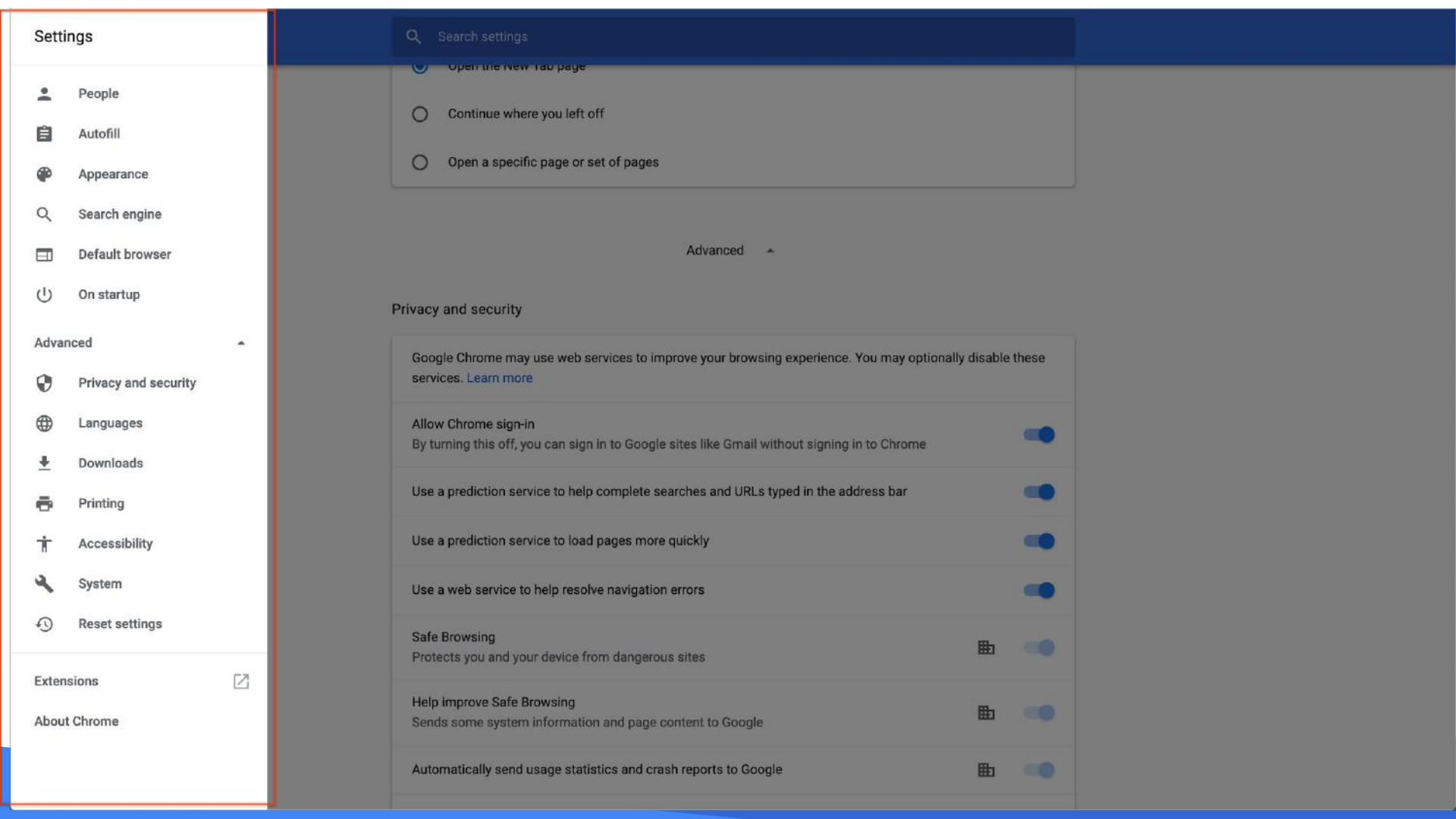
Payment methods

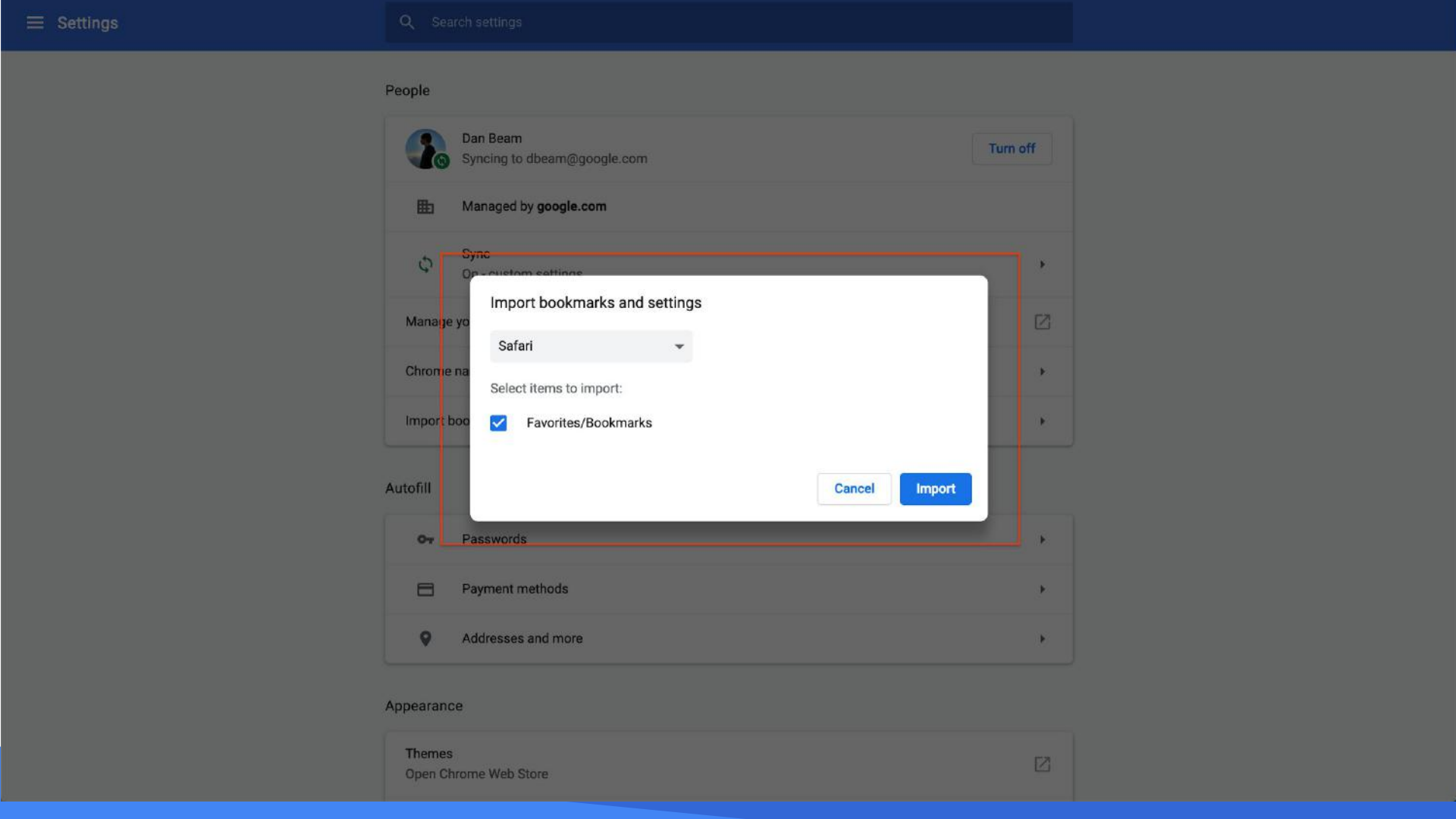
Addresses and more

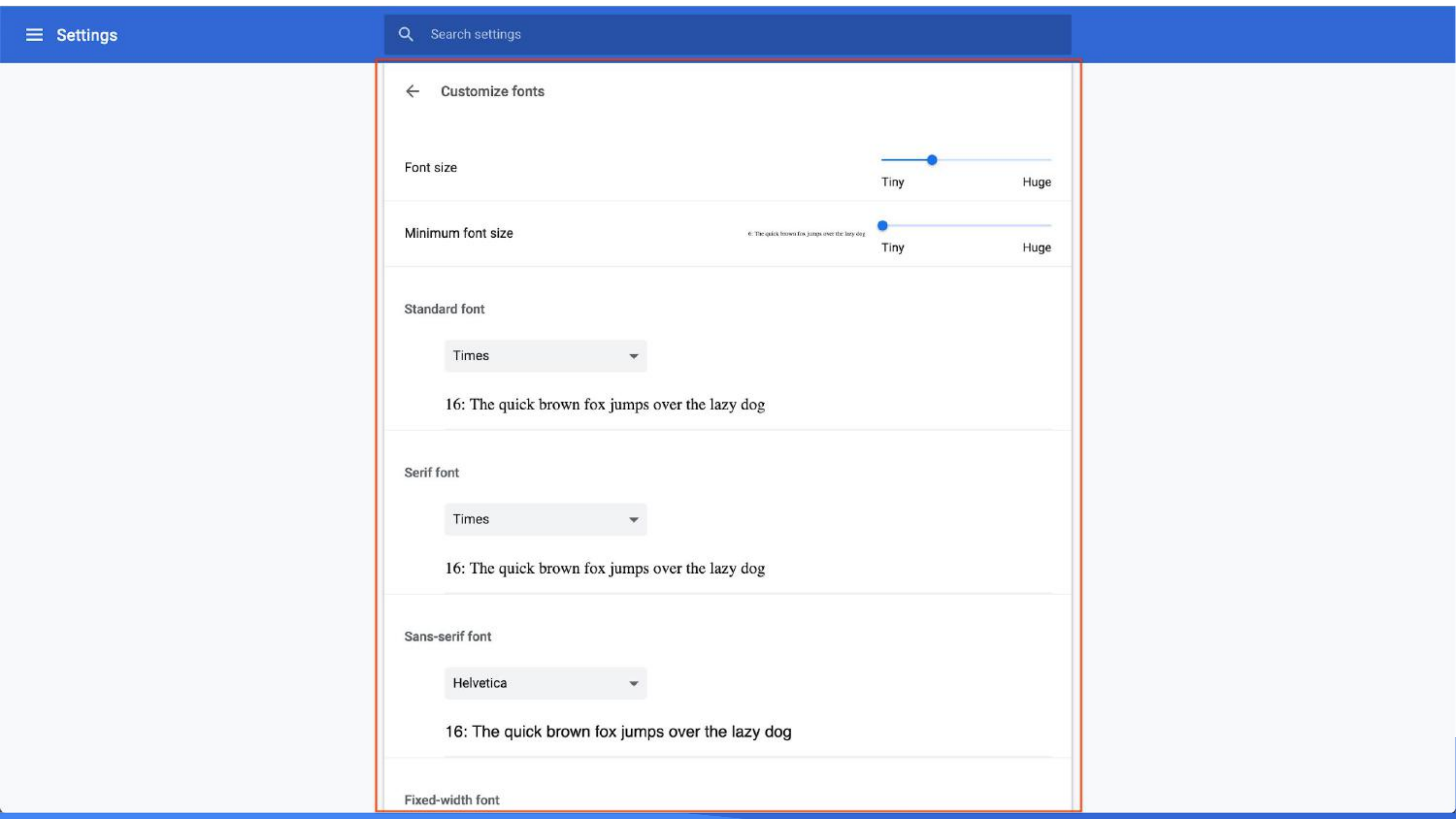
Appearance

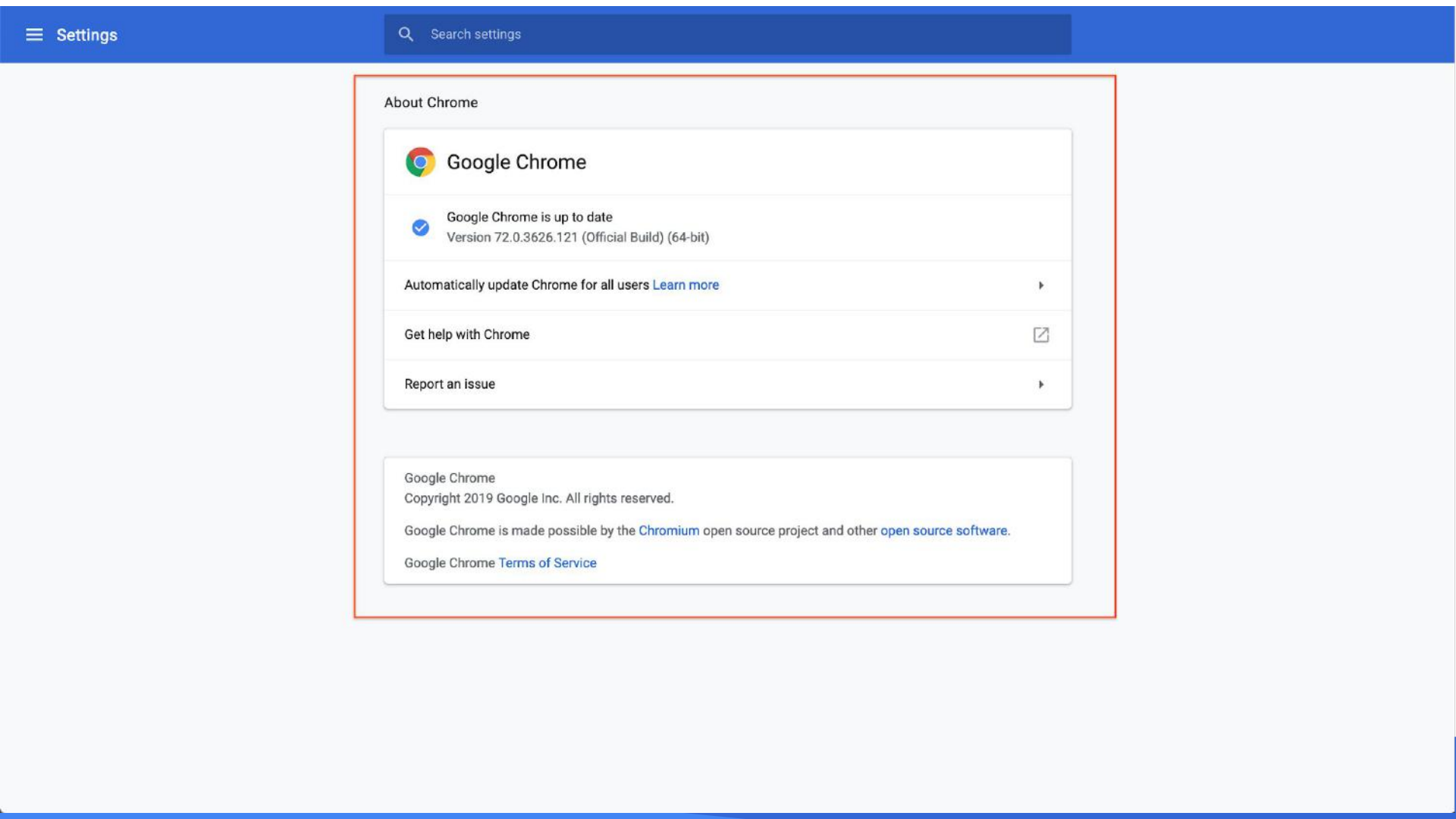
Themes
Open Chrome Web Store


Settings
Search settings
Open the New Tab page
Continue where you left off
Open a specific page or set of pages
Advanced
Privacy and security
Google Chrome may use web services to improve your browsing experience. You may optionally disable these services. Learn more
Allow Chrome sign-in
By turning this off, you can sign in to Google sites like Gmail without signing in to Chrome
Use a prediction service to help complete searches and URLs typed in the address bar
Use a prediction service to load pages more quickly
Use a web service to help resolve navigation errors
Safe Browsing
Protects you and your device from dangerous sites
Help improve Safe Browsing
Sends some system information and page content to Google
Automatically send usage statistics and crash reports to Google


Settings
People
Autofill
Appearance
Search engine
Default browser
On startup
Advanced
Privacy and security
Languages
Downloads
Printing
Accessibility
System
Reset settings
Extensions
About Chrome
Search settings
Continue where you left off
Open a specific page or set of pages
Advanced
Privacy and security
Google Chrome may use web services to improve your browsing experience. You may optionally disable these services. Learn more
Allow Chrome sign-in
By turning this off, you can sign in to Google sites like Gmail without signing in to Chrome
Use a prediction service to help complete searches and URLs typed in the address bar
Use a prediction service to load pages more quickly
Use a web service to help resolve navigation errors
Safe Browsing
Protects you and your device from dangerous sites
Help improve Safe Browsing
Sends some system information and page content to Google
Automatically send usage statistics and crash reports to Google


Settings
Search settings
People
Dan Beam
Syncing to dbeam@google.com
Turn off
Managed by google.com
Sync
Import bookmarks and settings
Safari
Select items to import:
Favorites/Bookmarks
Cancel
Import
Autofill
Passwords
Payment methods
Addresses and more
Appearance
Themes
Open Chrome Web Store


Settings
Search settings
Customize fonts
Font size
Tiny
Huge
Minimum font size
Tiny
Huge
Standard font
Times
16: The quick brown fox jumps over the lazy dog
Serif font
Times
16: The quick brown fox jumps over the lazy dog
Sans-serif font
Helvetica
16: The quick brown fox jumps over the lazy dog
Fixed-width font

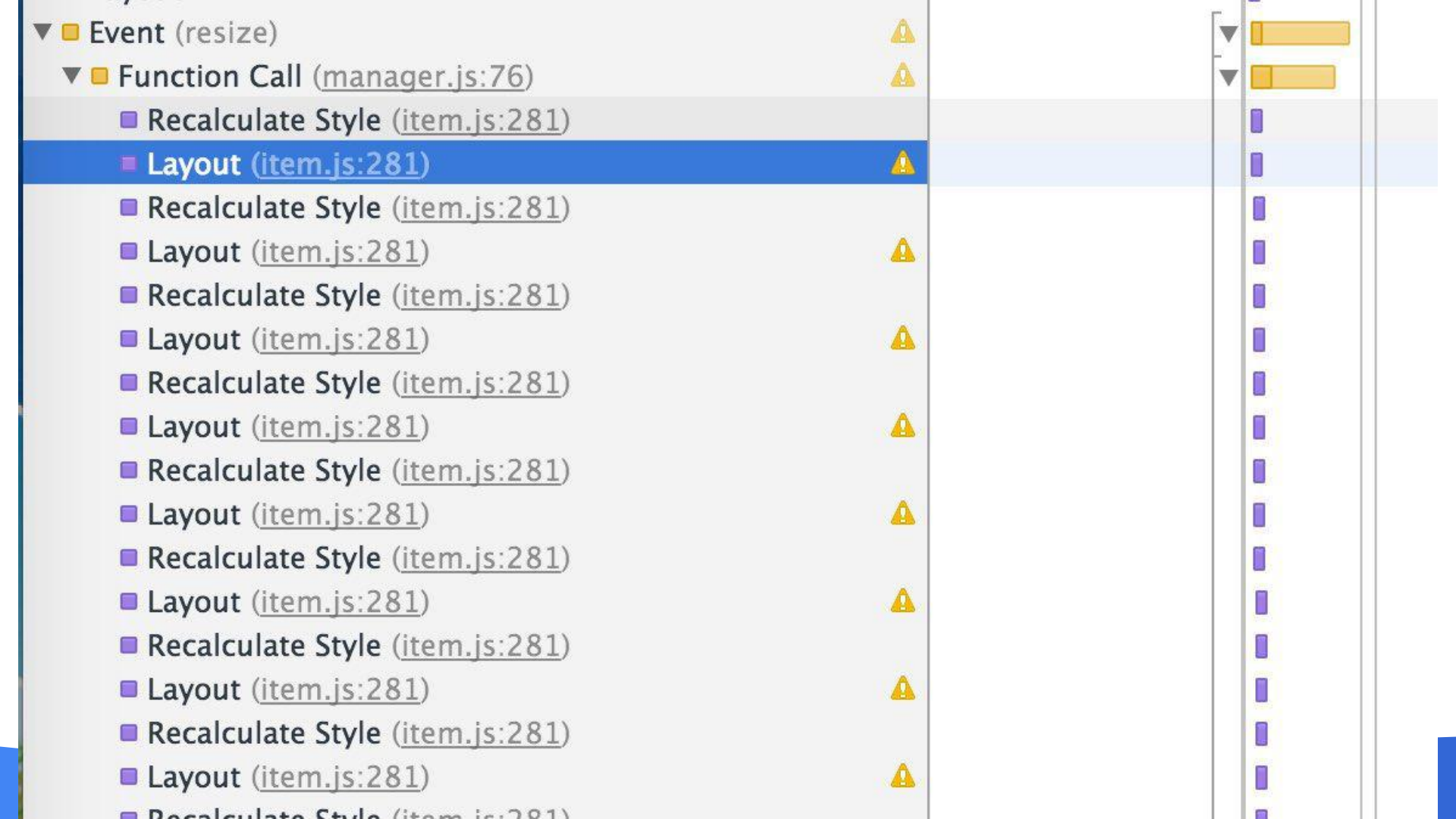

Settings
Search settings
About Chrome
Google Chrome
Google Chrome is up to date
Version 72.0.3626.121 (Official Build) (64-bit)
Automatically update Chrome for all users Learn more
Get help with Chrome
Report an issue
Google Chrome
Copyright 2019 Google Inc. All rights reserved.
Google Chrome is made possible by the Chromium open source project and other open source software.
Google Chrome Terms of Service

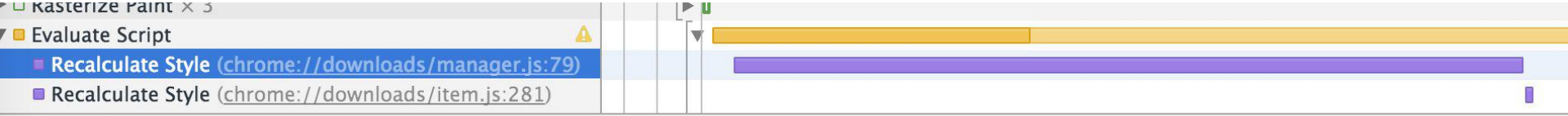
2. Reduce the cost

$T = N \times K$

POLYMER SUMMIT

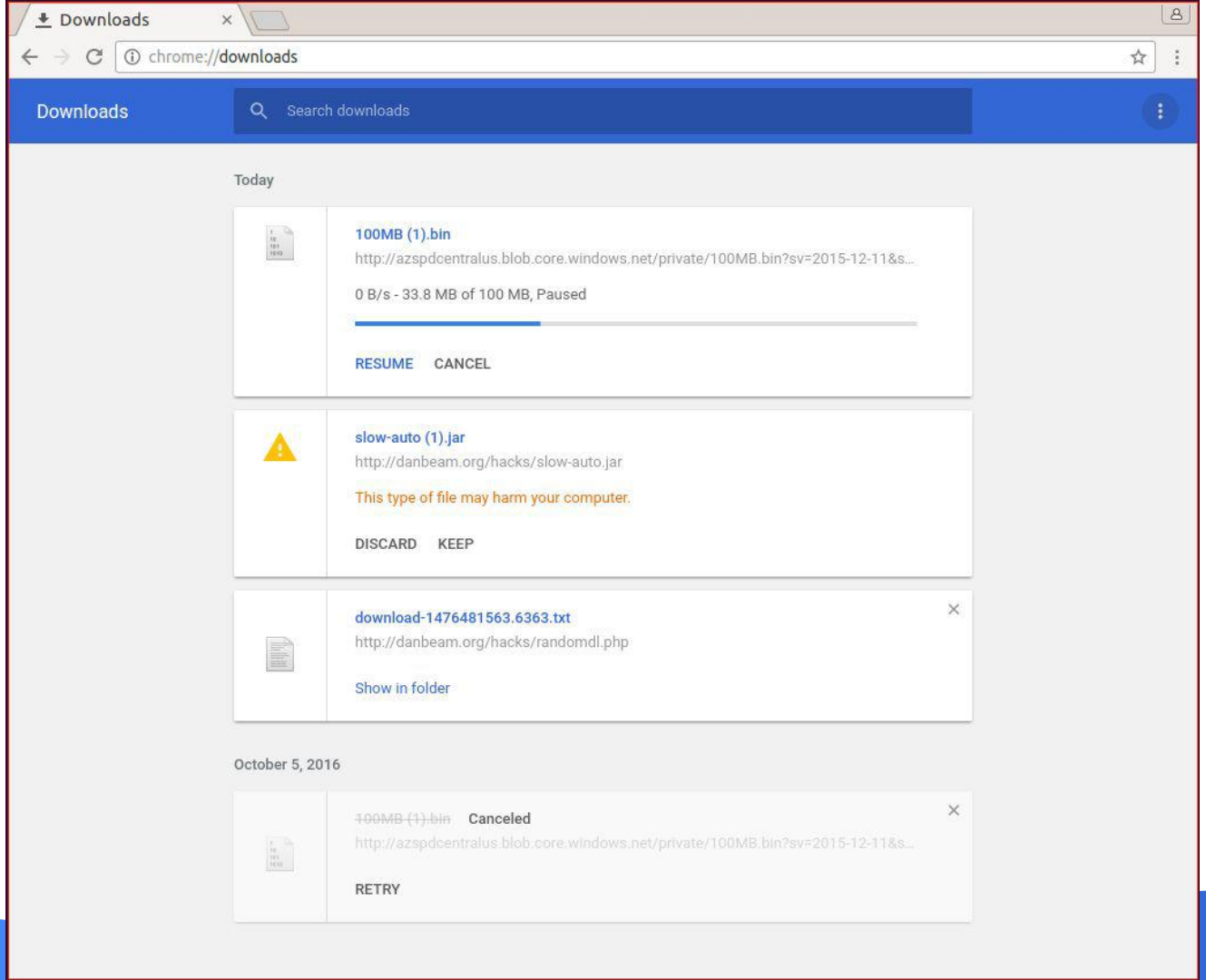
JOIN THE DISCUSSION AT

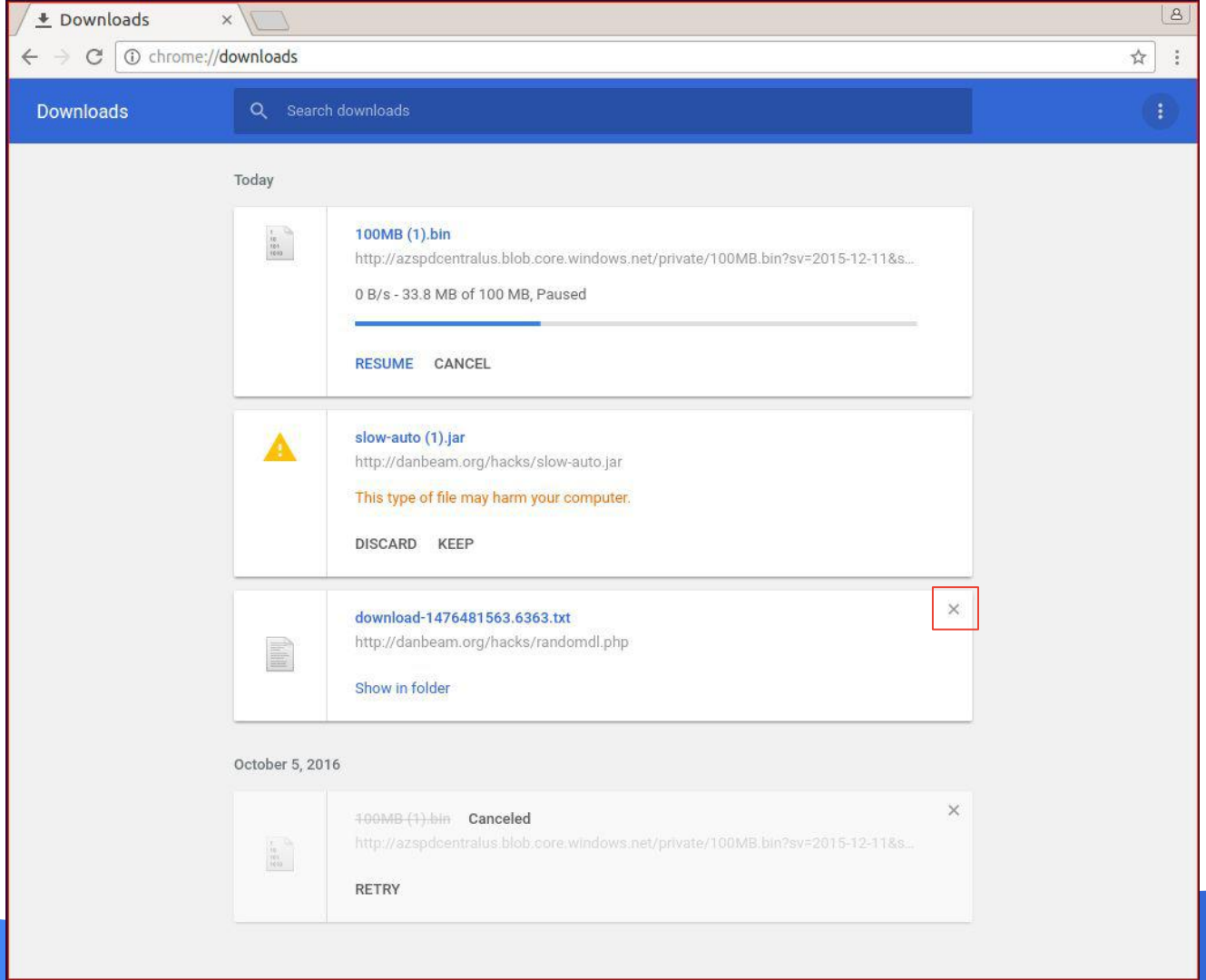

Event (resize)
Function Call (manager.js:76)
Recalculate Style (item.js:281)
Layout (item.js:281)
Recalculate Style (item.js:281)
Layout (item.js:281)
Recalculate Style (item.js:281)
Layout (item.js:281)
Recalculate Style (item.js:281)
Layout (item.js:281)
Recalculate Style (item.js:281)
Layout (item.js:281)
Recalculate Style (item.js:281)
Layout (item.js:281)
Recalculate Style (item.js:281)
Layout (item.js:281)
Recalculate Style (item.js:281)
Layout (item.js:281)

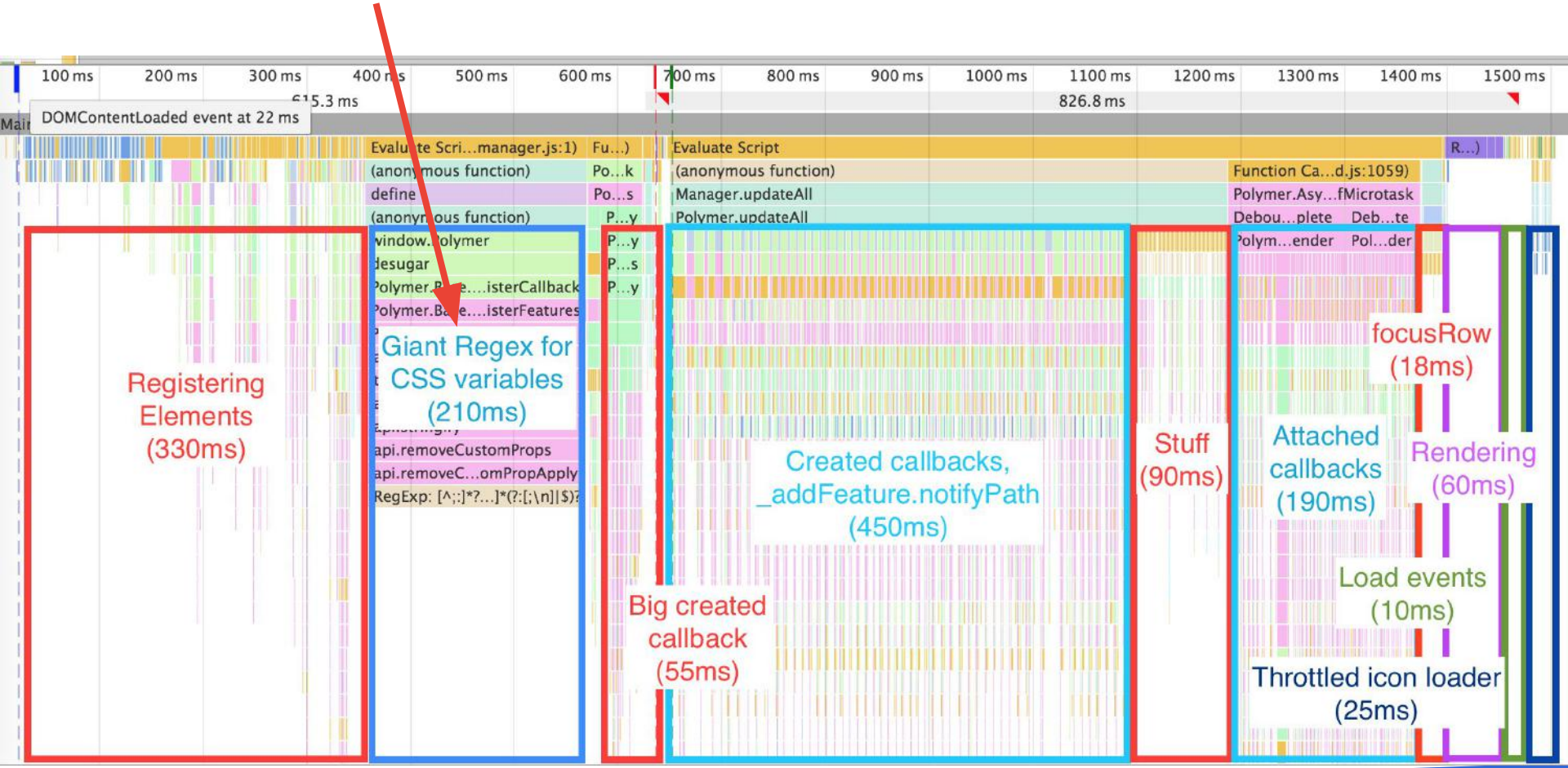


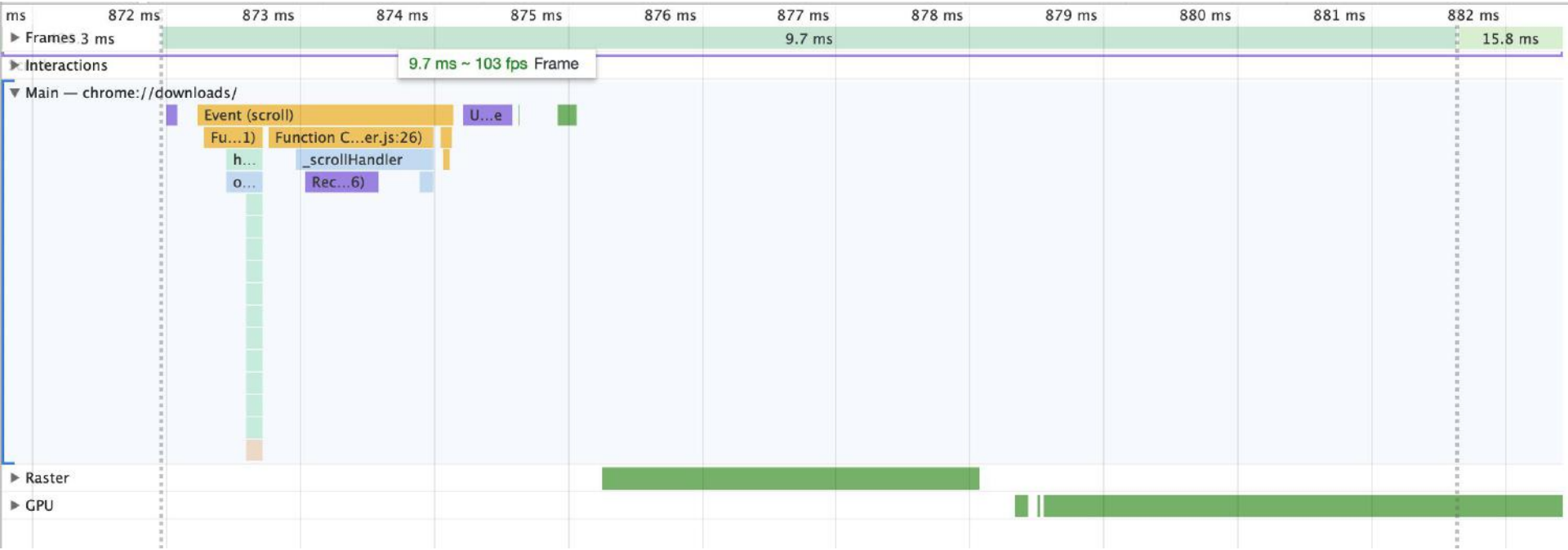
Summary | Aggregated Details

Type	Recalculate Style
Total Time	400.03 ms
Self Time	400.03 ms
Elements affected	1198

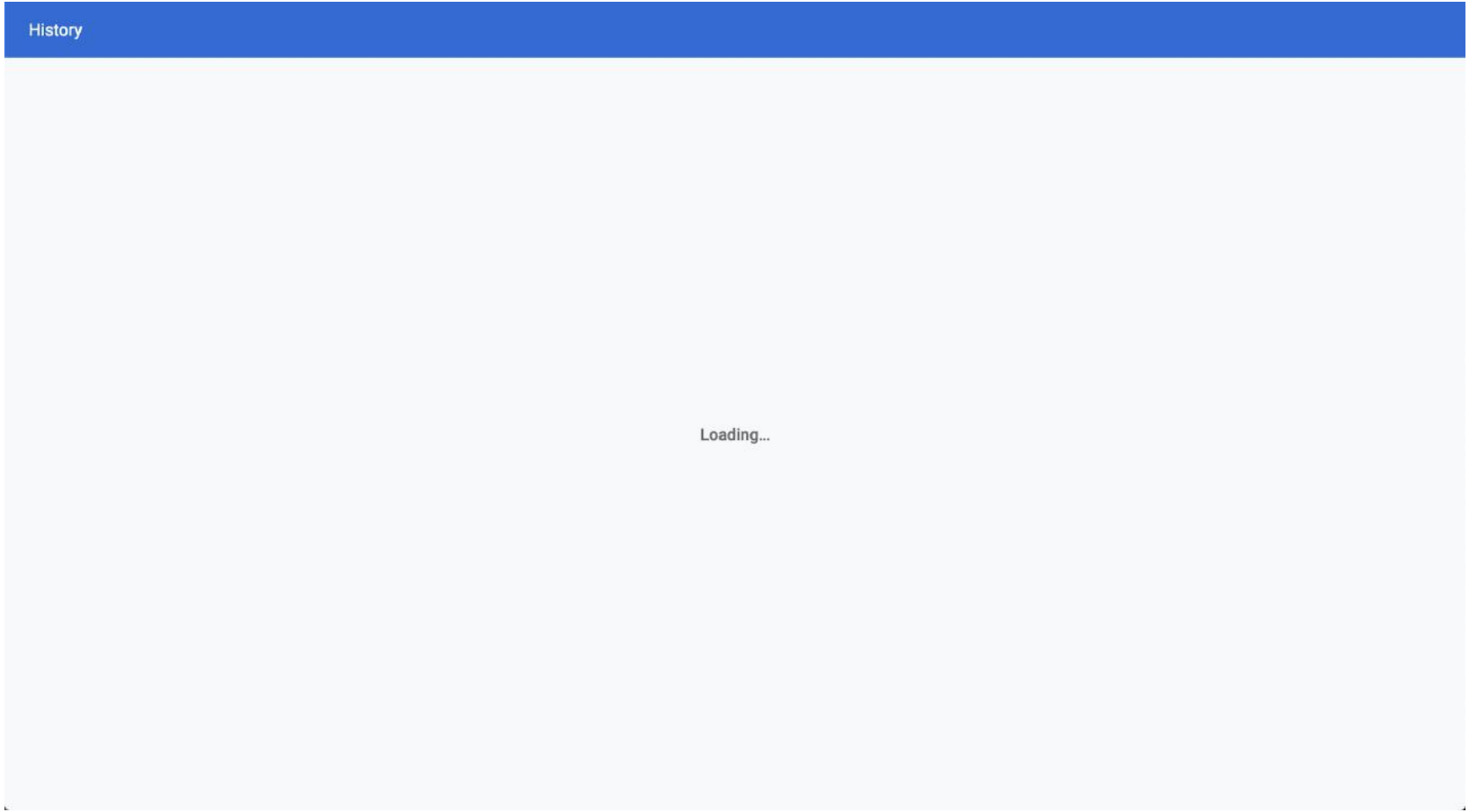

Downloads
chrome://downloads
Downloads
Search downloads
Today
100MB (1).bin
http://azspdcentralus.blob.core.windows.net/private/100MB.bin?sv=2015-12-11&s...
0 B/s - 33.8 MB of 100 MB, Paused
RESUME
CANCEL
slow-auto (1).jar
http://danbeam.org/hacks/slow-auto.jar
This type of file may harm your computer.
DISCARD
KEEP
download-1476481563.6363.txt
http://danbeam.org/hacks/randomdl.php
Show in folder
October 5, 2016
100MB (1).bin
Canceled
http://azspdcentralus.blob.core.windows.net/private/100MB.bin?sv=2015-12-11&s...
RETRY


Downloads
chrome://downloads
Downloads
Search downloads
Today
100MB (1).bin
http://azspdcentralus.blob.core.windows.net/private/100MB.bin?sv=2015-12-11&s...
0 B/s - 33.8 MB of 100 MB, Paused
RESUME
CANCEL
slow-auto (1).jar
http://danbeam.org/hacks/slow-auto.jar
This type of file may harm your computer.
DISCARD
KEEP
download-1476481563.6363.txt
http://danbeam.org/hacks/randomdl.php
Show in folder
October 5, 2016
100MB (1).bin
Canceled
http://azspdcentralus.blob.core.windows.net/private/100MB.bin?sv=2015-12-11&s...
RETRY


100 ms
200 ms
300 ms
400 ms
500 ms
600 ms
700 ms
800 ms
900 ms
1000 ms
1100 ms
1200 ms
1300 ms
1400 ms
1500 ms
615.3 ms
826.8 ms
DOMContentLoaded event at 22 ms
Evaluate Scri...manager.js:1)
Fu...)
Evaluate Script
R...)
(anonymous function)
Po...k
(anonymous function)
Function Ca...d.js:1059)
define
Po...s
Manager.updateAll
Polymer.Asy...fMicrotask
(anonymous function)
P...y
Polymer.updateAll
Debou...plete
Deb...te
vindow.Polymer
P...y
desugar
P...s
Polymer.Base....isterCallback
P...y
Polymer.Base....isterFeatures
Polym...ender
Pol...der
Registering Elements (330ms)
Giant Regex for CSS variables (210ms)
api.removeCustomProps
api.removeC...omPropApply
RegExp: [^;:]*?...]*(?:[;\n]|$)?
Big created callback (55ms)
Created callbacks, _addFeature.notifyPath (450ms)
Stuff (90ms)
Attached callbacks (190ms)
focusRow (18ms)
Rendering (60ms)
Load events (10ms)
Throttled icon loader (25ms)


ms
872 ms
873 ms
874 ms
875 ms
876 ms
877 ms
878 ms
879 ms
880 ms
881 ms
882 ms
Frames 3 ms
9.7 ms
15.8 ms
Interactions
9.7 ms ~ 103 fps Frame
Main — chrome://downloads/
Event (scroll)
U...e
Fu...1)
Function C...er.js:26)
h...
_scrollHandler
o...
Rec...6)
Raster
GPU

Pretend to be fast


History
Loading...

Facebook's loading shims

Capacity Planning

Capacity Planning

You’ve replaced engineering ingenuity with barrels of cash. Now what?

Capacity Planning

Seems simple at first!

- In a lot of cases, your server responds linearly with traffic
  - E.g. You know that if you allocate an additional 2x memory and CPU, your server can handle 2x QPS at the same latency
- However, at the extreme ends, servers start responding non-linearly with load, which can cause all sorts of headaches

Capacity Planning -- Automation

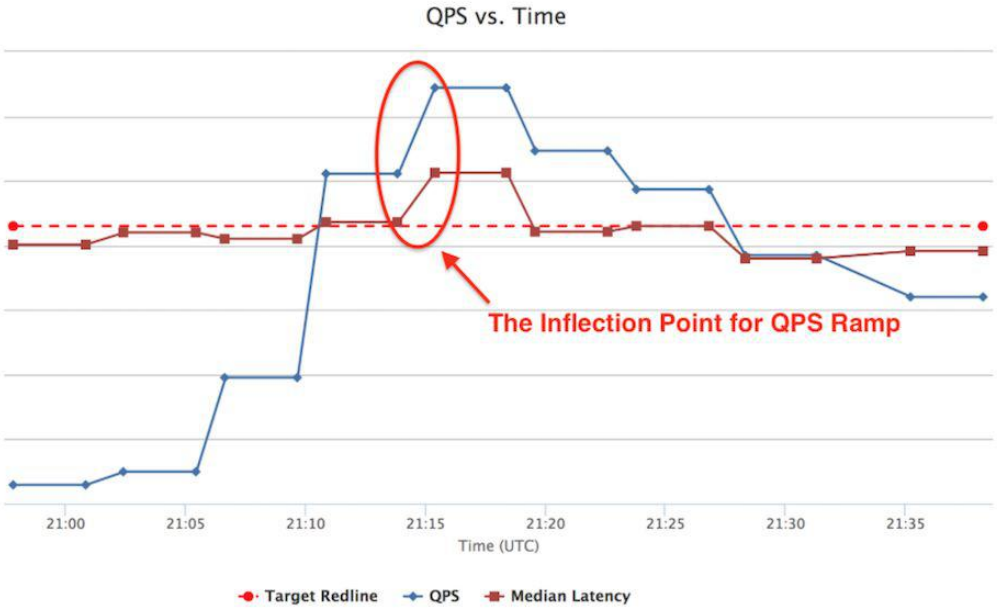
Automation can help us out a lot.

For a lot of cases, an automated system can spin up new instances of our server as traffic increases.

Even more granular, a service like Google Cloud could increase your server's resources (e.g. memory, CPU) as traffic increases.

But how do we deal with those non-linear server responses?

Load testing case study



Common Failure Modes of Distributed Systems

A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable.

-- Leslie Lamport

Common failure modes of distributed systems

Correctness bugs:

- Can often be detected in regression tests
- These systems are often data-parallel, so you can test correctness with limited resources

Performance bugs:

- Hard to catch in synthetic tests
- Typically needs full scale and representative load to properly test

Above: Common error screen around the time you were a baby

Producer-consumer rate mismatch

Even if you have admission control, a rate mismatch can be very problematic

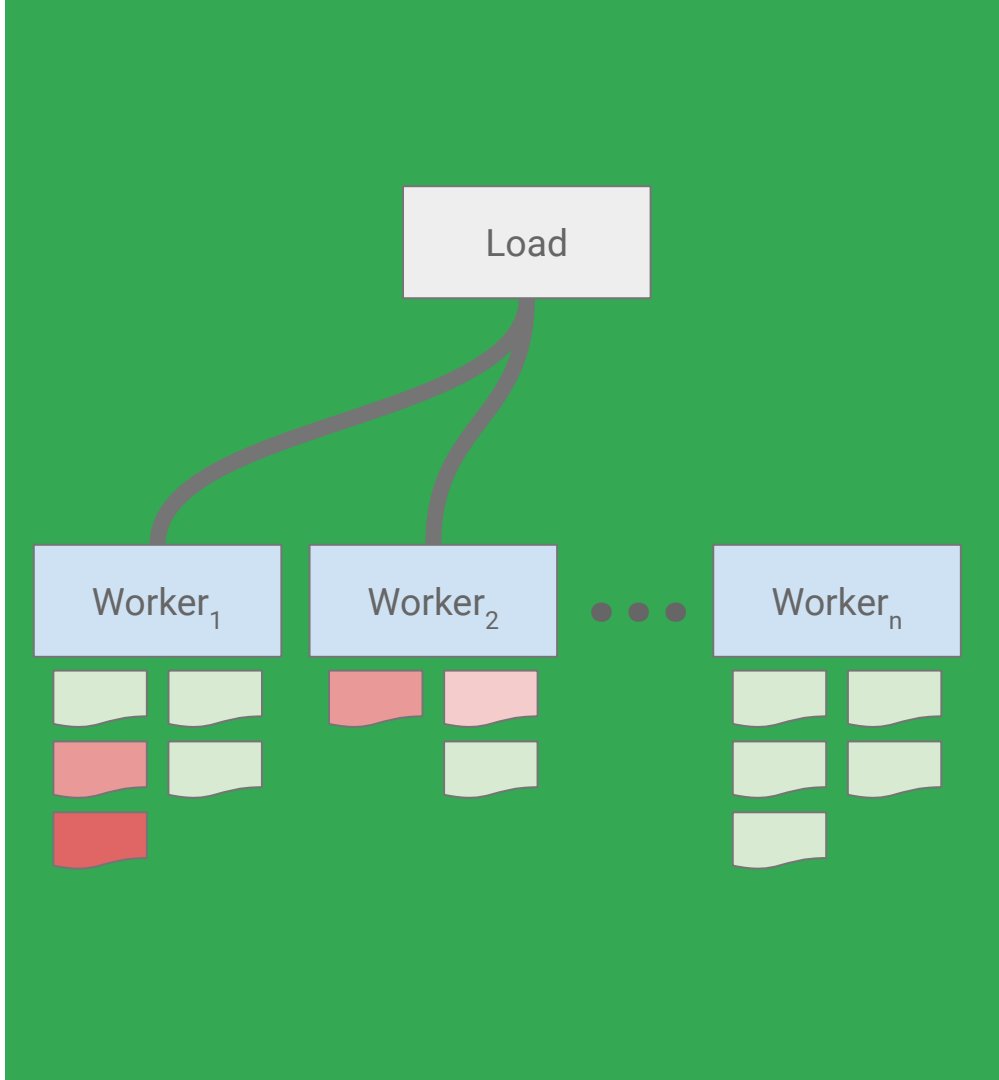
- Best case, the back pressure propagates through your system and the system stalls or the failure is visible to the user
- Worst case, some node in the middle of your computation fails because it is overloaded

Hotspots

The load on your system can be concentrated onto a small working set.

- Results in the machines hosting that fraction of the data getting hotter than average and increasing queuing delay
- Even with load shedding, the node can “heat up” faster than you’d normally rebalance

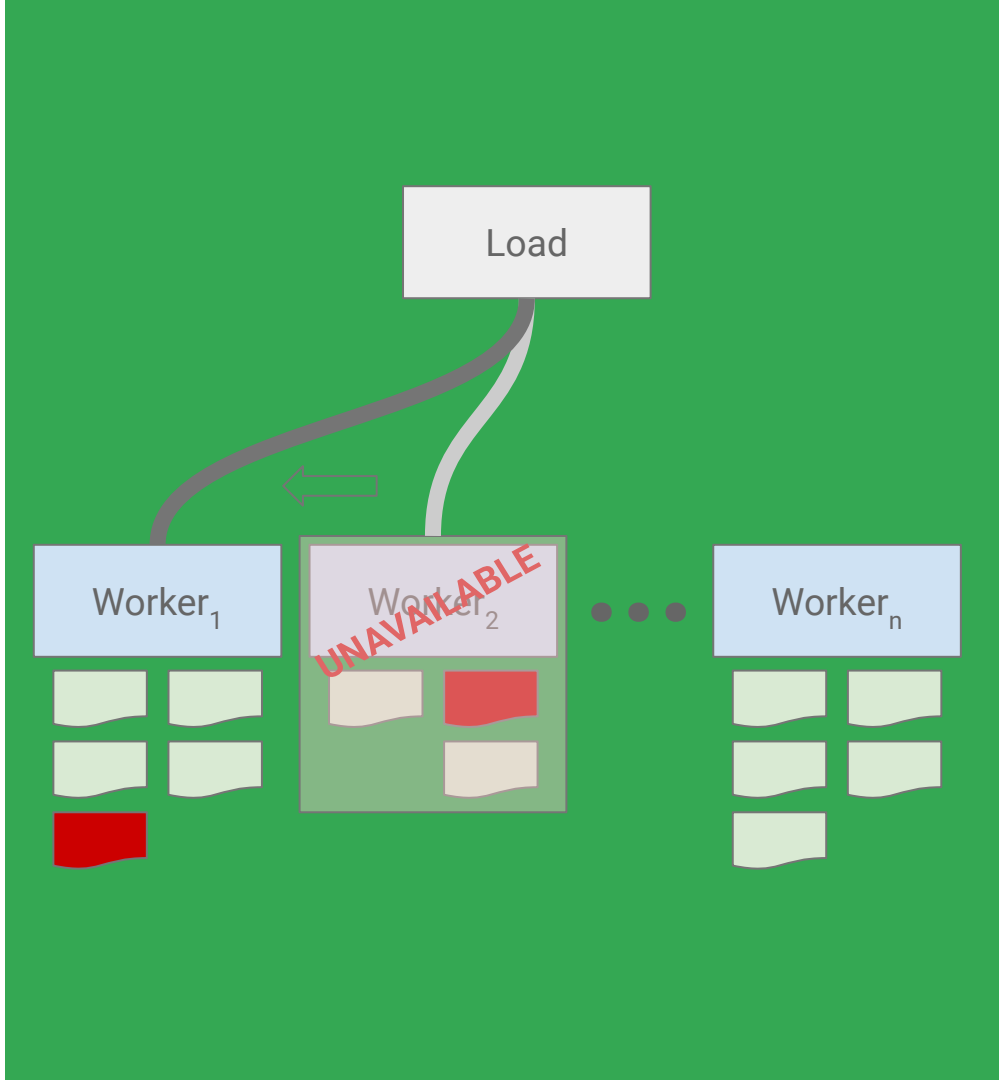
Need to seek ways to make your algorithms hotspot resistant.



Death ray

The evil cousin of the hotspot...

- Typically happens when the load is so overwhelming that admission control can't prevent the process from failing
- However, the load doesn't cease
- When your system recovers and loads those resources onto another node, the death ray follows



Amplified rare events

Even rare things tend to be common at scale:

- Bug triggers 1 in 1M operations, running 10 ops/sec, this bug will trigger daily!

These types of failures abound. A few examples that we've observed:

- Rare checksum computation failure because of processor issue
- Remote cache restart causes tablet server failure

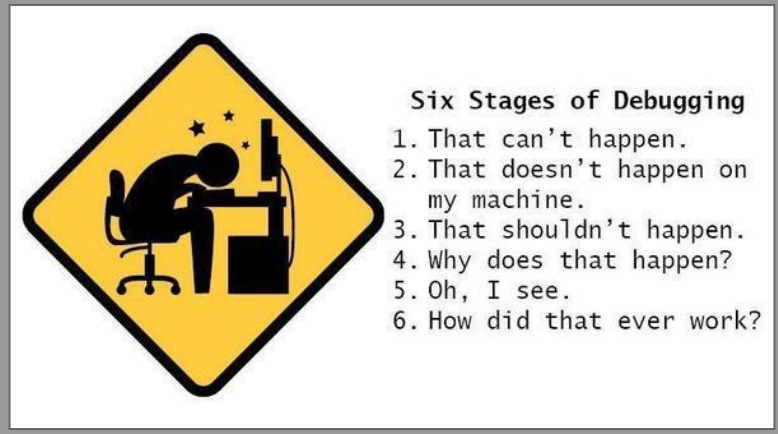
Debugging Distributed Systems

Debugging Distributed Systems

Once you've built a system, you often live with it for a long time.

You need tools to help you understand what the system is doing:

- At a micro scale: per job
- At a macro scale: interactions between jobs



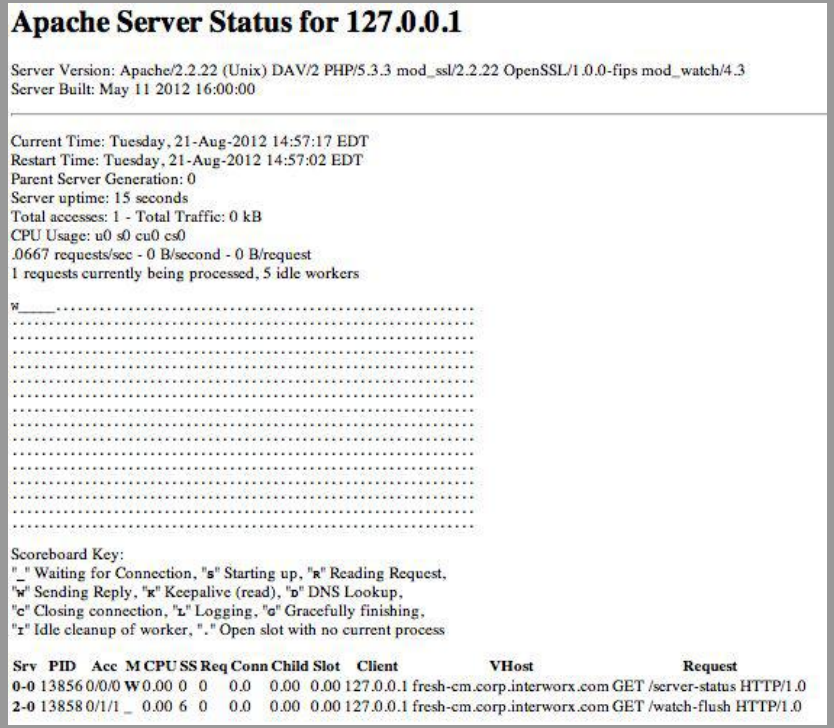
Status Pages

Key features:

- Each process serves over HTTP
- Gives you a local understanding of what is going on in a single process
- Can expose state of internal data or reasons for recent decisions

Especially useful status pages:

- Stacktrace of all currently live threads
- State of all currently pending RPCs



Apache Server Status for 127.0.0.1

Server Version: Apache/2.2.22 (Unix) DAV/2 PHP/5.3.3 mod_ssl/2.2.22 OpenSSL/1.0.0-fips mod_watch/4.3
Server Built: May 11 2012 16:00:00

Current Time: Tuesday, 21-Aug-2012 14:57:17 EDT
Restart Time: Tuesday, 21-Aug-2012 14:57:02 EDT
Parent Server Generation: 0
Server uptime: 15 seconds
Total accesses: 1 - Total Traffic: 0 kB
CPU Usage: u0 s0 cu0 cs0
.0667 requests/sec - 0 B/second - 0 B/request
1 requests currently being processed, 5 idle workers

W____.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

Scoreboard Key:
"_" Waiting for Connection, "S" Starting up, "R" Reading Request,
"W" Sending Reply, "K" Keepalive (read), "D" DNS Lookup,
"C" Closing connection, "L" Logging, "G" Gracefully finishing,
"I" Idle cleanup of worker, "." Open slot with no current process

Srv	PID	Acc	M	CPU	SS	Req	Conn	Child	Slot	Client	VHost	Request
0-0	13856	0/0/0	W	0.00	0	0	0.0	0.00	0.00	127.0.0.1	fresh-cm.corp.interworx.com	GET /server-status HTTP/1.0
2-0	13858	0/1/1	_	0.00	6	0	0.0	0.00	0.00	127.0.0.1	fresh-cm.corp.interworx.com	GET /watch-flush HTTP/1.0

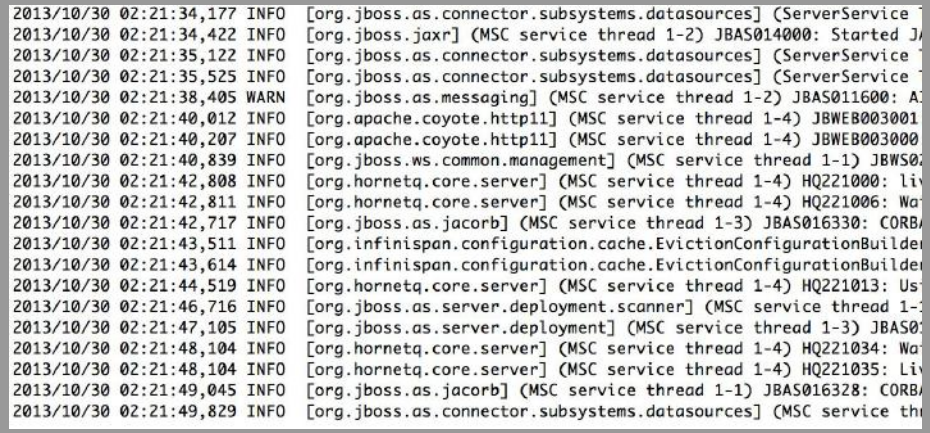
Logging

It is critical to produce useful logs.

- Logs are very useful for debugging failures
- Since you don't know when failures are going to happen, you need to log all the time

What to log:

- When and what requests arrive
- Details about erroneous conditions
- What part of the code is involved



Local Request Tracing

Logs are useful, but they interleave events

- To remedy this, you can maintain request specific logs, often called traces
- Typically only kept for active requests, so they are less useful for retrospective debugging
- Very useful for answering the question: “What is going on right now?”

```
// Annotate a request trace.

const string& request = ...;
if (HitCache()) {
  TRACEPRINTF("cache hit for %s",
              request.c_str());
} else {
  TRACEPRINTF("cache miss for %s",
              request.c_str());
}
```

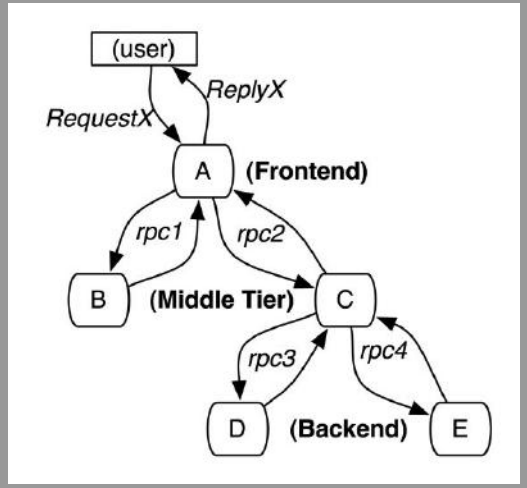
Distributed Tracing

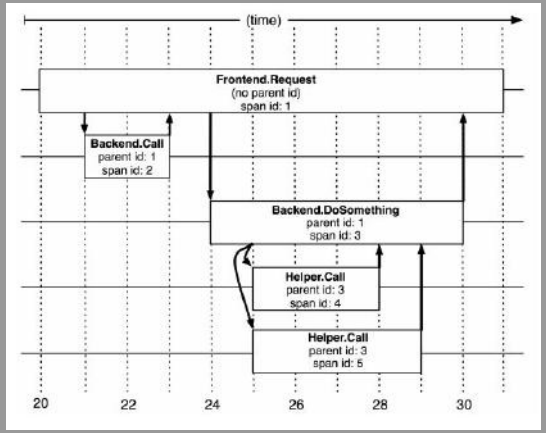
Key features:

- Integrated into each component of the whole system
- A token, representing a logical operation, is passed along each RPC
- Stats are logged by each process and a system aggregates and visualizes the data
- Tend to be verbose, so they are sampled

Types of stats:

- Latency
- Resource Usage (CPU, Disk, etc)





Paper: Dapper, a Large-Scale Distributed Systems Tracing Infrastructure