Agenda

- Challenges of HTTP/1.1
- SPDY
- Some data, and results
- How we tune today
- What’s next
rfc2616: HTTP/1.1

drafted: 1998
released June 1999
What else was there in 1998
What else was there in 1998
What else was there in 1998
What else was there in 1998
What else was there in 1998
What else was there in 1998

IEEE Computer Society Awards

1998 Computer Entrepreneur Joint Award

“For their entrepreneurial leadership in the creation of the personal computing industry and the profound changes it has fostered in both business and personal life”

Steven P. Jobs

Steven P. Jobs co-founded Apple Computer with Steve Wozniak and served as Chairman of the Board from 1975 to 1985. He was President of NeXT Inc. from 1985 until his recent return to Apple. Prior to Apple’s founding, he was with Hewlett-Packard and designed video games for Atari in 1974. (Assembled From Who’s Who in America, 1997).

Steven G. Wozniak

Steven G. Wozniak created the Apple I computer and founded the Apple Computer company with Steve Jobs. He was presented the National Medal of Technology in 1995 by President Reagan. Wozniak left Apple in 1985, devoting himself to other ventures, family matters, and teaching. (Assembled from sources, including Kendall 1994).

William H. Gates III

William H. Gates was a founder of Microsoft and has been its Chief Executive Officer and Chairman of the Board since the company’s predecessor partnership was incorporated in 1981. From 1975 to 1981, Mr. Gates was a partner with Paul Allen, Microsoft’s other founder, in the predecessor partnership. Mr. Gates is also a director of ICOS Corporation. (Edited excerpt from: Microsoft Proxy Statement, 1997).

Paul G. Allen

Paul G. Allen has been a director of Microsoft since 1990 and served on its Board from 1983 to 1984. He was a founder of the company and worked at Microsoft from 1975 to 1984. Mr. Allen owns and invests in a suite of companies exploring the potential of multimedia digital communications. His wholly-owned companies include Asymmetric Corporation, Interval Research Corp., Vulcan Ventures Inc. He is also the owner of the Portland Trail Blazers basketball team and the Seattle Seahawks football team, a partner in the entertainment studio DreamWorks SKG, and holds investments in more than 35 technology companies. Mr. Allen is also a director of both HSN Inc. and Ticketmaster Corp. (Edited excerpt from: Microsoft Proxy Statement, 1997).
What else was there in 1998

IEEE Computer Society Awards

1998 Computer Entrepreneur Joint Award

“For their entrepreneurial leadership in the creation of the personal computer and profound changes it has fostered in both business and personal life”

Steven P. Jobs

Steven P. Jobs co-founded Apple Computer with Steve Wozniak and served as President of the Board from 1975 to 1985. He was President of NeXT Inc. from 1985 to 1986 and returned to Apple in 1997 as Chief Executive Officer. (Assembled from Who's Who in Computer Science.

Steven G. Wozniak

Steven G. Wozniak created the Apple I computer and founded Apple Computer with Steve Jobs. He was an inventor of Apple II and received the National Medal of Technology in 1985. (Assembled from Who's Who in Computer Science.

William H. Gates III

William H. Gates was a founder of Microsoft and has been its Chief Executive Officer since 1975. He is President of the Board since the Company's predecessor partnership in 1981. From 1975 to 1982, Mr. Gates was a partner with Paul Allen in the predecessor partnership, Micro-soft. (Assembled from Who's Who in Computer Science.

Paul G. Allen

Paul G. Allen has been a director of Microsoft since 1990 and served as President of the Company from 1960 to 1964. He is a co-founder of the company and worked at IBM from 1964 to 1977. Mr. Allen now works in the video game industry as the CEO of Microsoft Game Studios and is a part owner of the Seattle Seahawks and Seattle Storm. (Assembled from Who's Who in Computer Science.

What else was there in 1998

IEEE Computer Society Awards

1998 Computer Entrepreneur Joint Award

“For their entrepreneurial leadership in the creation of the personal computing industry and profound changes it has fostered in both business and personal life.”

Steven P. Jobs

Steven P. Jobs co-founded Apple Computer with Steve Wozniak and served as a member of the Board from 1976 to 1985. He was President of NeXT Inc. and returned to Apple in 1996. (Assembled from Who’s Who in America, 1998)

Steven G. Wozniak

Steven G. Wozniak co-founded Apple Computer with Steve Jobs and served as a member of the Board from 1976 to 1985. He was President of NeXT Inc. and returned to Apple in 1996. (Assembled from Who’s Who in America, 1998)

William H. Gates III

William H. Gates is the Chief Executive Officer of Microsoft and has been its Chairman of the Board since 1986. He is also a co-founder of the company. weitits.com (Edited excerpt from: Microsoft Proxy Statement, 1998)

Paul G. Allen

Paul G. Allen was one of the founders of Microsoft and has been its Chairman of the Board since 1986. He is also a co-founder of the company. weitits.com (Edited excerpt from: Microsoft Proxy Statement, 1998)

Technical Awards • Computer Entrepreneur • Past Recipients • 1998 Entrepreneur List

20 images, ~65 KB entire page

Did They Plan for That!?
HTTP/1.1

- Compression
- Headers
- Poor connection management
  - Only one request can be done at a time (Half duplex)
  - Particularly important on high latency links (mobile)
- short lived connections
- security - optional
Compression

* [http://code.google.com/speed/articles/web-metrics.html](http://code.google.com/speed/articles/web-metrics.html) + Cotendo measurements
Compression

☐ Optional
☐ Only for response body
☑ ~99% of browsers support

* [http://code.google.com/speed/articles/web-metrics.html](http://code.google.com/speed/articles/web-metrics.html) + Cotendo measurements
Compression

- Optional
- Only for response body
- ~99% of browsers support
- Particularly important on small pipes (mobile)

* [http://code.google.com/speed/articles/web-metrics.html](http://code.google.com/speed/articles/web-metrics.html) + Cotendo measurements
Compression

- Optional
- Only for response body
- ~99% of browsers support
- Particularly important on small pipes (mobile)

- 15%-30% of responses not compressed for no good reason!*  
  - for SSL even worse

* [http://code.google.com/speed/articles/web-metrics.html](http://code.google.com/speed/articles/web-metrics.html) + Cotendo measurements
Compression

☐ Optional

☐ Only for response body

☑~99% of browsers support

☑ Particularly important on small pipes (mobile)

☐ 15%-30% of responses not compressed for no good reason!*

  ☐ for SSL even worse

☐ no request compression

* http://code.google.com/speed/articles/web-metrics.html + Cotendo measurements
Compression

- Optional
- Only for response body
- ~99% of browsers support
- Particularly important on small pipes (mobile)

- 15%-30% of responses not compressed for no good reason!*
  - for SSL even worse
- no request compression
- no header compression

* [http://code.google.com/speed/articles/web-metrics.html](http://code.google.com/speed/articles/web-metrics.html) + Cotendo measurements
Headers

- Redundant
- Repeat on EVERY request/response
- Can’t compress

- User-agent
- Cookies
HTTP Connection - under the hood

Client

Server
HTTP Connection - under the hood

TCP Connect + Request = 1.5 RTT
TCP Connect + SSL Connect + Request = 3.5 RTT
HTTP Connection - under the hood

Client → Server

TCP Connect + Request = 1.5 RTT
TCP Connect + SSL Connect + Request = 3.5 RTT
HTTP Connection - under the hood

![Diagram showing the flow between Client and Server with RTTs]

- TCP Connect + Request = 1.5 RTT
- TCP Connect + SSL Connect + Request = 3.5 RTT
- Request/Response = 0.5 RTT
HTTP Connection - under the hood

- TCP Connect + Request = 1.5 RTT
- TCP Connect + SSL Connect + Request = 3.5 RTT
- Request/Response = 0.5 RTT
HTTP Connection - under the hood

TCP Connect + Request = 1.5 RTT
TCP Connect + SSL Connect + Request = 3.5 RTT
Request/Response = 0.5 RTT
HTTP Connection - under the hood

- TCP Connect + Request = 1.5 RTT
- TCP Connect + SSL Connect + Request = 3.5 RTT
- Request/Response = 0.5 RTT

Spun additional connections
HTTP Connection - under the hood

- TCP Connect + Request = 1.5 RTT
- TCP Connect + SSL Connect + Request = 3.5 RTT
- Request/Response = 0.5 RTT
- Slow Start - 30KB ~ 4 RTT (vs. 1 RTT warm)

- Spun additional connections
### User Data: average bandwidth

<table>
<thead>
<tr>
<th>Region</th>
<th>BW Mbps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>1.9</td>
</tr>
<tr>
<td>US</td>
<td>5.1</td>
</tr>
<tr>
<td>Mobile - US</td>
<td>1.1</td>
</tr>
<tr>
<td>Mobile - EU</td>
<td>~2</td>
</tr>
</tbody>
</table>

Source: State of the Internet, Q4 2010
http://www.akamai.com/stateoftheinternet/

### User Data: average RTT

<table>
<thead>
<tr>
<th>RTT ms</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Broadband</td>
<td>30</td>
</tr>
<tr>
<td>Coast-to-Coast</td>
<td>100</td>
</tr>
<tr>
<td>3G device-GW</td>
<td>80-200</td>
</tr>
</tbody>
</table>

### Lost Capacity on each RTT

- **US Broadband:** 63.7KB
- **US Mobile:** 27.5KB
- **Average obj size:** 8.1KB

Source: HTTP Archive, 6/1/2011 data for Top 1,000
http://httparchive.org
Security is optional

Enters SPDY!
Requirements

- **Avoid requiring** the website author to change content
  - Allow for incremental changes
  - Performing "better" with content changes is okay
  - Performing "worse" without content changes is unacceptable

- Perform always better, never worse than HTTP

- Drop-in replacement from webapp's point of view
  - Changing the web server/application server is inevitable and therefore acceptable
What is SPDY?

- **Goal:** Reduce Web Page Load time.
  - Multiplexing
    - Better utilize client connections
  - Compression
    - HTTP headers are excessive
    - Uplink bandwidth is limited
  - Prioritization
    - Today the browser holds back
    - Priorities enable multiplexing
  - Encrypted & Authenticated
    - Eavesdropping at the Cafe must be stopped
  - Server Push
    - Websites do some of this today with data URLs
Deployment Status
Deployment Status

- Google
  - Enabled for all Google SSL traffic
  - On by default since Chrome 6
Deployment Status

- **Google**
  - Enabled for all Google SSL traffic
  - On by default since Chrome 6

- **CDNs/Service providers:**
  - Cotendo
Deployment Status

- **Google**
  - Enabled for all Google SSL traffic
  - On by default since Chrome 6

- **CDNs/Service providers:**
  - Cotendo

- **Server implementations:**
  - Strangeloop
  - SPDY Proxy available
  - Others: C++, Python, Ruby, Go
Deployment Status

- **Google**
  - Enabled for all Google SSL traffic
  - On by default since Chrome 6

- **CDNs/Service providers:**
  - Cotendo

- **Server implementations:**
  - Strangeloop
  - SPDY Proxy available
  - Others: C++, Python, Ruby, Go

- **Client implementation**
  - Chrome
  - Android coming...
Results
Yahoo! Homepage, over broadband network
Yahoo! homepage

HTTP

Broadband

RTT = 20ms

Page Load: 1.23 sec
<table>
<thead>
<tr>
<th>Request Path</th>
<th>Status Code</th>
<th>Response Size</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.yahoo.com">www.yahoo.com</a></td>
<td>200 OK</td>
<td>206.9 KB</td>
</tr>
<tr>
<td>combo7metro/g/core_yul_3.1</td>
<td>200 OK</td>
<td>170.9 KB</td>
</tr>
<tr>
<td>sp.png</td>
<td>200 OK</td>
<td>43 B</td>
</tr>
<tr>
<td>combo7metro/g/sicContrib/y</td>
<td>200 OK</td>
<td>21.6 B</td>
</tr>
<tr>
<td>spSprite_pg_slate_20100521.png</td>
<td>200 OK</td>
<td>2.4 K</td>
</tr>
<tr>
<td>yahoo_logo_us_061509.png</td>
<td>200 OK</td>
<td>1.7 K</td>
</tr>
<tr>
<td>sprite_pg_slate_20110516_it</td>
<td>200 OK</td>
<td>9.8 K</td>
</tr>
<tr>
<td>spR_masthd_slate_20100628</td>
<td>200 OK</td>
<td>1006 B</td>
</tr>
<tr>
<td>localNew_20101013.gif</td>
<td>200 OK</td>
<td>213 B</td>
</tr>
<tr>
<td>spAppsUs.png</td>
<td>200 OK</td>
<td>4.1 K</td>
</tr>
<tr>
<td>sprite_pg_monochrome_201012</td>
<td>200 OK</td>
<td>4.9 K</td>
</tr>
<tr>
<td>health_20100602.gif</td>
<td>200 OK</td>
<td>617 B</td>
</tr>
<tr>
<td>monster_20110207.png</td>
<td>200 OK</td>
<td>391 B</td>
</tr>
<tr>
<td>mess_20100602.gif</td>
<td>200 OK</td>
<td>1.1 K</td>
</tr>
<tr>
<td>t9afut8021kC04004tvRDeCE</td>
<td>200 OK</td>
<td>1.1 K</td>
</tr>
<tr>
<td>shine_20100602.png</td>
<td>200 OK</td>
<td>650 B</td>
</tr>
<tr>
<td>froyo.png</td>
<td>200 OK</td>
<td>11.8 K</td>
</tr>
<tr>
<td>animLoading_sm_082208.png</td>
<td>200 OK</td>
<td>1.4 K</td>
</tr>
<tr>
<td>froyo-sm.png</td>
<td>200 OK</td>
<td>1.5 K</td>
</tr>
<tr>
<td>bep-sm.png</td>
<td>200 OK</td>
<td>5.3 K</td>
</tr>
<tr>
<td>gaida-sm.png</td>
<td>200 OK</td>
<td>2.2 K</td>
</tr>
<tr>
<td>jellyfish-emails5-sm.png</td>
<td>200 OK</td>
<td>2.2 K</td>
</tr>
<tr>
<td>scotttrade_purplegif.gif</td>
<td>200 OK</td>
<td>2.5 K</td>
</tr>
<tr>
<td>92x55Swr1019fc.jpg</td>
<td>200 OK</td>
<td>1.7 K</td>
</tr>
<tr>
<td>55x30Wrl818Qj3.gif</td>
<td>200 OK</td>
<td>3.5 K</td>
</tr>
<tr>
<td>300x250Wrl818Qj3.gif</td>
<td>200 OK</td>
<td>9.8 K</td>
</tr>
<tr>
<td>yoga-vn.png</td>
<td>200 OK</td>
<td>9.8 K</td>
</tr>
<tr>
<td>lincolnpark-vnsm.png</td>
<td>200 OK</td>
<td>1.8 K</td>
</tr>
<tr>
<td>babydog2-vnsm.png</td>
<td>200 OK</td>
<td>1.5 K</td>
</tr>
<tr>
<td>img_e802b974.png</td>
<td>200 OK</td>
<td>2.4 K</td>
</tr>
<tr>
<td>img_ba2e2192.png</td>
<td>200 OK</td>
<td>1.9 K</td>
</tr>
<tr>
<td>yahoo-logo-sm-png8.png</td>
<td>200 OK</td>
<td>743 B</td>
</tr>
<tr>
<td>addchoice_1.png</td>
<td>200 OK</td>
<td>447 B</td>
</tr>
<tr>
<td>sprite_videocon_20100201.png</td>
<td>200 OK</td>
<td>2 K</td>
</tr>
<tr>
<td>ybank_22_111908.png</td>
<td>200 OK</td>
<td>469 B</td>
</tr>
<tr>
<td>combo7metro/g/core_yul_3.1</td>
<td>200 OK</td>
<td>323 K</td>
</tr>
<tr>
<td>bc_205js</td>
<td>200 OK</td>
<td>1.9 K</td>
</tr>
<tr>
<td>83708199821914S</td>
<td>200 OK</td>
<td>42 B</td>
</tr>
<tr>
<td>S522971123.sz=1x1&amp;ord=13</td>
<td>302 Moved Temp</td>
<td>ad.doubleclick.net</td>
</tr>
<tr>
<td>yahoo.png</td>
<td>200 OK</td>
<td>1.8 K</td>
</tr>
<tr>
<td>b7=3e1VWKJ2yeVZQ5m</td>
<td>200 OK</td>
<td>43 B</td>
</tr>
<tr>
<td>49-1x1.gif</td>
<td>200 OK</td>
<td>43 B</td>
</tr>
<tr>
<td>eye-test.png</td>
<td>200 OK</td>
<td>2 K</td>
</tr>
<tr>
<td>eye-test.png</td>
<td>200 OK</td>
<td>0</td>
</tr>
</tbody>
</table>

Page Load: **1.21 sec**

-2%
Amazon.com Homepage, over 3G network
Amazon.com Homepage

HTTP

3G, AT&T
RTT = 200+ms

Page Load: 12.50 sec
Amazon.com Homepage

SPDY (via Cotendo)

3G AT&T
RTT = 200+ms

Page Load: 6.26 sec

-49%
HTTPS vs SPDY (Production)

SPDY vs HTTPS Chrome 12, Mar 22-April 5

SPDY

HTTPS

15.4% improvement

PLT (ms)
Increasing Parallelism

SPDY vs HTTP Time to First Byte for a Request

- HTTP
- SPDY

Percentile vs Time-to-First-Byte (ms)
Less is More - Conns, Bytes, Packets

Connections Per Web Page
- HTTP
- SPDY

SPDY vs HTTP Total Packets (Top-45 pages)
- HTTP
- SPDY

SPDY vs HTTP Upload KB Sent (Top-45 pages)
- HTTP
- SPDY

SPDY vs HTTP Download KB (Top-45 pages)
- HTTP
- SPDY

51% reduction
19% reduction
4% reduction
Not Too Shabby WebSockets

- docs.google.com has a "hanging get" for every doc open
- how to scale beyond 6 connections per domain?
  - docs[1-N].google.com
- gets expensive, complicated, and is horribly inefficient
- SPDY is easy, works great, efficient
- Header compression mitigates the inefficiency of a hanging GET
Can't We Address Latency & Security Separately?
Can't We Address Latency & Security Separately?

No.
Can't We Address Latency & Security Separately?

- If eavesdropping in the cafe is still possible in 2020 with trivial tools, we have failed our users.

- Security is not just for banks!
  - Social/Mail/Content is a major target
  - example: Comodo attack

- Firesheep tools make sniffing easy

- Major content providers want privacy
  - Facebook opt-in
  - Twitter opt-in
  - Google working toward 100%
How we tune today
High Performance Web Sites

1. Make fewer HTTP requests
2. Use CDN
3. Add expires header
4. Gzip Components
5. Put stylesheets at the top
6. Put scripts at the bottom
7. Avoid CSS expressions
8. Make JS and CSS external
9. Reduce DNS lookups
10. Minify JS
11. Avoid redirects
12. Remove duplicate scripts
13. Configure Etags
14. Make Ajax cacheable
15. Sharding domains
High Performance Web Sites

1. Make fewer HTTP requests
2. Use CDN
3. Add expires header
4. Gzip Components
5. Put stylesheets at the top
6. Put scripts at the bottom
7. Avoid CSS expressions
8. Make JS and CSS external
9. Reduce DNS lookups
10. Minify JS
11. Avoid redirects
12. Remove duplicate scripts
13. Configure Etags
14. Make Ajax cacheable
15. Sharding domains

Due To HTTP Limitations!
Firetruck
Domain Sharding
Domain Sharding

- Pros:
  - Parallelism in HTTP
  - Cookieless domains for static content
Domain Sharding

- Pros:
  - Parallelism in HTTP
  - Cookieless domains for static content
Domain Sharding

- **Pros:**
  - Parallelism in HTTP
  - Cookieless domains for static content

- **Cons:**
  - DNS time
  - Connection establishment overhead
    - TCP slowstart / warmup
    - SSL
  - maintaining many connections
  - user cache
  - Prevents browser optimizations
Reduce number of requests

- Inlining
- Spriting
- Data URLs
- Combining javascripts and CSS files

Cons:
- reduced cache efficiency
- Complicated
New Class of FEO
Rethink your optimizations

- As rules change, your need to re-evaluate your best practices
  - 13 years is a long time! start at the beginning...

- But also presents many new opportunities
  - Delivery priorities
  - Smart Push
  - Simple web code
  - Automate!
Image Compression

Number of Requests

- Images: 55.0
- Scripts: 13.0
- Stylesheets: 3.5
- Other: 7.0

Transfer Size in KB

- Images: 415
- Scripts: 119
- Stylesheets: 25
- Other: 99
- HTML: 32

Source: http://httparchive.org/trends.php?s=Top1000
In mobile even stronger!

**Number of Requests**

- Images: 31
- HTML: 4
- Scripts: 7
- Stilesheets: 2
- Other: 2

**Transfer Size in KB**

- Images: 228
- HTML: 42
- Scripts: 80
- Stilesheets: 20
- Other: 6

Lossy compression is the way to go!

Billy Hoffman / Zoompf

AP Photo/John Bazemore
Lossy-Compression is Not for Everything
Progressive image compression

Baseline

Start download → End download
Partial image → Final Image

Progressive

Start download → End download
Less quality → Final Image

http://images.sixrevisions.com/2010/12/01-02_baseline_vs_progressive.jpg
Get HTML
Get all objects

Client

Server
Summary
- SPDY actually delivers!
SPDY actually delivers!
- but we are still in the early phases
- SPDY actually delivers!
  - but we are still in the early phases
  - participate and be part of the solution
- SPDY actually delivers!
  - but we are still in the early phases
  - participate and be part of the solution
• SPDY actually delivers!
  ▪ but we are still in the early phases
  ▪ participate and be part of the solution

• Rules CAN be broken. Changes are possible!
• SPDY actually delivers!
  • but we are still in the early phases
  • participate and be part of the solution

• Rules CAN be broken. Changes are possible!
  • HTTP is built into the internet architecture, it can’t be changed.
• SPDY actually delivers!
  • but we are still in the early phases
  • participate and be part of the solution

• Rules CAN be broken. Changes are possible!
  • HTTP is built into the internet architecture, it can’t be changed.
  • What’s next?
- SPDY actually delivers!
  - but we are still in the early phases
  - participate and be part of the solution

- Rules CAN be broken. Changes are possible!
  - HTTP is built into the internet architecture, it can’t be changed.
  - What’s next?
- SPDY actually delivers!
  - but we are still in the early phases
  - participate and be part of the solution

- Rules CAN be broken. Changes are possible!
  - HTTP is built into the internet architecture, it can’t be changed.
  - What’s next?

- Opens the door to implement new class of optimizations
- SPDY actually delivers!
  - but we are still in the early phases
  - participate and be part of the solution

- Rules CAN be broken. Changes are possible!
  - HTTP is built into the internet architecture, it can’t be changed.
  - What’s next?

- Opens the door to implement new class of optimizations
- SPDY actually delivers!
  - but we are still in the early phases
  - participate and be part of the solution

- Rules CAN be broken. Changes are possible!
  - HTTP is built into the internet architecture, it can’t be changed.
  - What’s next?

- Opens the door to implement new class of optimizations

- Get engaged NOW!
Measurement - Proving us Right

- No 3rd party tools (yet)!

- Chrome developer tools
  - not good for large scale data...

- w3c Web Performance tools - Navigation Timing
  - Supported on IE9, and Chrome 11+

- Site metrics:
  - Google Analytics