

High Performance Web Sites

14 rules for faster pages



Nate Koechley

natek@yahoo-inc.com

nate.koechley.com/blog

developer.yahoo.com/yui

Important Note:

During my session at the @media 2007 conference I presented “12 Rules.” In the original presentation by Steve Souders and Tenni Theurer, and in Steve’s forthcoming O’Reilly book, there are 14 Rules. To keep things consistent I’ve added the two missing rules back into this presentation:

#12: Remove duplicate scripts

#14: Make Ajax cachable and small

By reinstating these two extra rules the numbering now matches what you’ll find in the book. Conveniently, #12 (from my in-person presentation) is the only number impacted by these additions.

Thanks!

Britpack Diamond Geezer Award





LONDON
15th-16th JUNE

1. Explored memory footprint & CPU impact
 - Event & Object management
2. Introduced *event delegation*
3. Shared optimal file placement
 - CSS at top (in `<head>`)
 - JS at bottom (before `</body>`)



1. Share results of our research into what impacts page performance.
2. Offer 12 specific rules to follow that will make your sites immediately and markedly faster.

“A case study in knowledge sharing...Yahoo is firmly committed to openness and to discussing stuff with the international technical community.”

—PPK on quirksmode.org:

Why talk about
performance?



Yahoo! Exceptional Performance Team

Steve Souders

Architect

souders@yahoo-inc.com

Tenni Theurer

Director

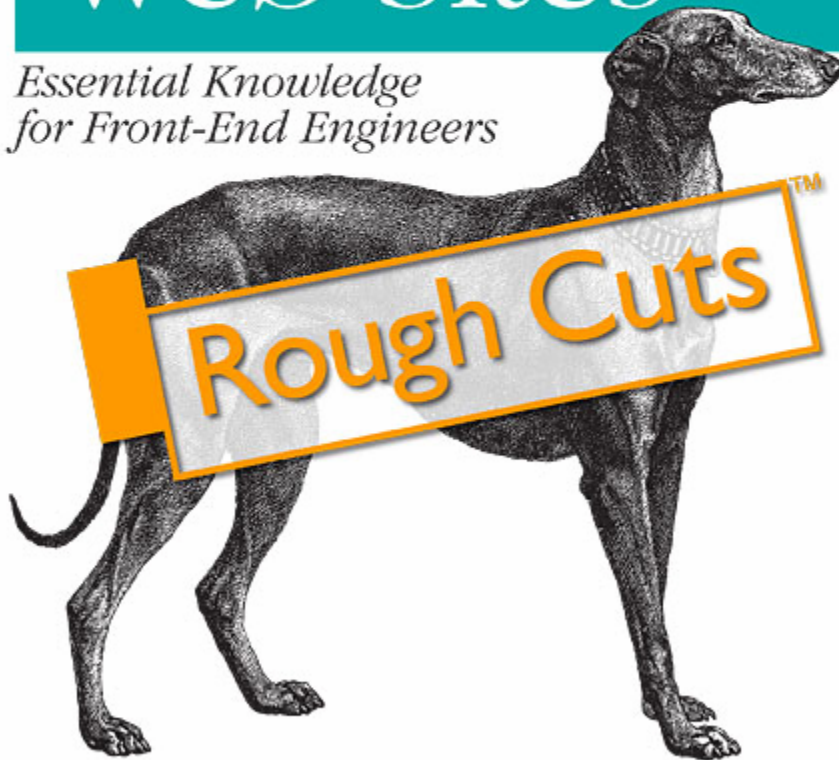
tenni@yahoo-inc.com



14 Steps to Faster-Loading Web Sites

High Performance Web Sites

*Essential Knowledge
for Front-End Engineers*



O'REILLY®

Steve Souders

Rough Cuts: Now
Amazon Pre-order: Now
Hardcopy: Sept 2007

Also, 3 hour workshop at
the upcoming:



<http://www.oreilly.com/catalog/9780596514211/>

Two
Performance
Flavors:

**Response
Time
&
System
Efficiency**



Our focus is
on response
time of web
products



A large, stylized white question mark is positioned on the left side of the image. It is set against a dark red background with a rough, cracked, and textured appearance, resembling aged leather or stone. The question mark has a thick, slightly irregular outline and a small circular hole at its base.

Do we care?

**Does it
matter?**

How much?

Is it worth it?

The Importance of Front-End Performance

Back-end

= 5%

Front-end

= 95%

Even here, front-end = 88%



Back-end vs. Front-end

	Empty Cache	Full Cache
amazon.com	82%	86%
aol.com	94%	86%
cnn.com	81%	92%
ebay.com	98%	92%
google.com	86%	64%
msn.com	97%	95%
myspace.com	96%	86%
wikipedia.org	80%	88%
yahoo.com	95%	88%
youtube.com	97%	95%

**Until now our
optimization efforts
have targeted the
tip of the iceberg.**



Foundational Research:

#1

Perception

perceived response time

show performance speed
crawl boring snail
stagnant urgent instant
accelerate unresponsive
snap achievement
impatient delay
better improve action
moderate than subtle
drag apathetic prolonged
load sluggish swift cope
maneuvering give prompt
advance cofaster heavy rush
unmemorable obscure
satisfying feel exceptional
brisk rapid exciting

what is the end user's experience?

It's in the eye of the beholder

1. Perception and usability are important performance metrics.
2. More relevant than actual *unload-to-onload* time.
3. Definition of "user onload" is undefined or varies from one web page to the next.

#2

“80% of consequences come
from 20% of causes”

—Vilfredo Pareto

```
YAHOO.util.Motion = function(el, attributes, duration, method) {  
  if (el) {  
    this.initMotion(el, attributes, duration, method);  
  }  
}  
  
YAHOO.util.Motion.prototype = new YAHOO.util.Anim();  
  
YAHOO.util.Motion.prototype.initMotion = function(el, attributes, duration, method) {  
  YAHOO.util.Anim.call(this, el, attributes, duration, method);  
}
```

YAHOO! USER INTERFACE BLOG

News and Articles about Designing and Developing with Yahoo! Libraries.

Blog

About

Performance Research, Part 1: What the 80/20 Rule Tells Us about Reducing HTTP Requests

November 28, 2006 at 12:58 pm by Tenni Theurer | [In Development](#) |

This is the first in a series of articles describing experiments conducted to learn more about optimizing web page performance. You may be wondering why you're reading a performance article on the YUI Blog. It turns out that most of web page performance is affected by front-end engineering, that is, the user interface design and development.

It's no secret that users prefer faster web sites. I work in a dedicated team focused on quantifying and improving the performance of Yahoo! product web sites. As part of our work, we conduct experiments related to web page performance. We are sharing our findings so that other front-end engineers join us in accelerating the user experience on the web.

The 80/20 Performance Rule

Vilfredo Pareto, an economist in the early 1900s, made a famous observation where 80% of the nation's wealth belonged to 20% of the population. This was later generalized into what's commonly referred to as the Pareto principle (also known as the 80-20 rule), which states for any phenomenon, 80% of the consequences come from 20% of the causes. We see this phenomenon in software engineering where 80% of the time is spent in only 20% of the code. When we optimize our applications, we know to focus on that 20% of the code. This same technique should also be applied when optimizing web pages. Most performance optimization today are made on the parts that generate the HTML document (apache, C++, databases, etc.), but these parts only

SYNDICATE

All Entries:

[+ MY YAHOO!](#) [RSS](#)

All Comments:

[+ MY YAHOO!](#) [RSS](#)

RECENT POSTS

[Welcoming Mike Lee and Dav Glass to the YUI Team](#)

[YUI Theater — Doug Geoffray: "From the Mouth of a Screenreader"](#)
[YUI Theater — Doug Geoffray: "From the Mouth of a Screenreader"](#)
[YUI Theater — Doug Geoffray: "From the Mouth of a Screenreader"](#)

[Happy Birthday to the Pattern Library, Too!](#)

[YUI Theater — "Browser Wars Episode II: Attack of the DOMs"](#)

[Free Excerpt: Nicholas Zakas on YUI Connection Manager, from Professional Ajax 2nd Edition](#)

RECENT READERS



tenni08

#3

\$\$\$

#3

€€€

#3

£££

#3

Cache

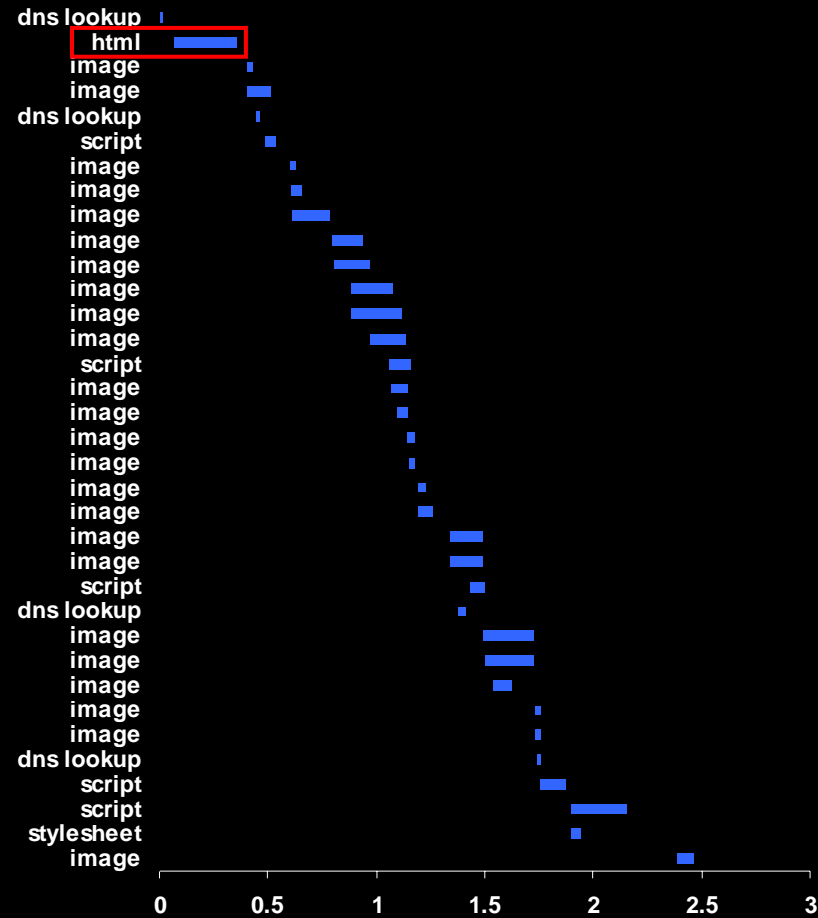
Empty vs. Full Cache



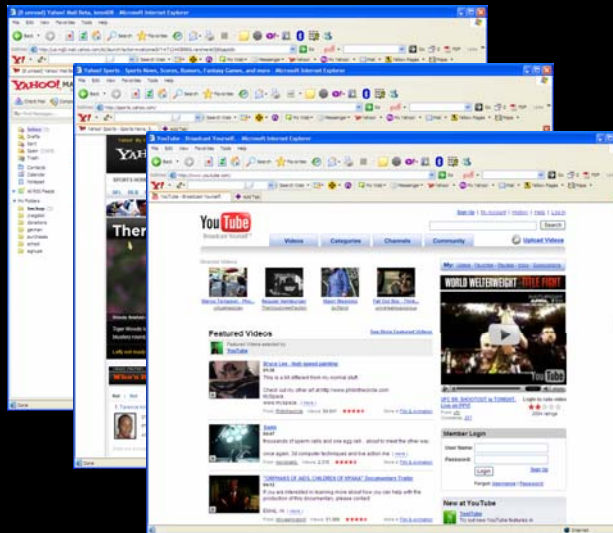
Empty vs. Full Cache



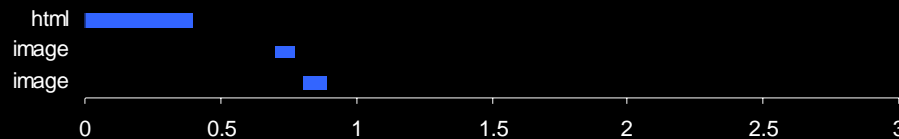
with an empty cache



Empty vs. Full Cache



Empty vs. Full Cache



Expires header



with a full cache

Empty vs. Full Cache

Empty Cache		Full Cache	
28.0K	1 HTML document	28.0K	1 HTML document
1.9K	1 Style Sheet File	0.1K	2 Images
59.5K	4 JavaScript Files		
78.7K	24 Images		
168.1K	Total size	28.1K	Total size
30	HTTP requests	3	HTTP requests
2.4s	Response time	0.9s	Response time

empty cache
2.4 seconds

full cache
0.9 seconds

83% fewer bytes

90% fewer HTTP requests

#3b

(sadly, the cache
doesn't work as well
as we wish it did.)

How much does caching benefit our users?

Q1: What % of users view a page with an empty cache?

Q2: What % of page views are with an empty cache?

Browser Cache Experiment

Add a new image to your page

```

```



with the following response headers:

Expires: Thu, 15 Apr 2004 20:00:00 GMT

Last-Modified: Wed, 28 Sep 2006 23:49:57 GMT

Browser Cache Experiment

Two possible response codes:

200 – The browser does not have the image in its cache.

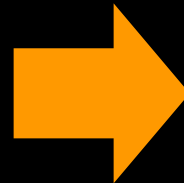
304 – The browser has the image in its cache, but needs to verify the last modified date.

Browser Cache Experiment

Q1: What % of users view with an empty cache?


$$\frac{\text{\# unique users with at least one 200 response}}{\text{total \# unique users}}$$

Q2: What % of page views are with an empty cache?


$$\frac{\text{total \# of 200 responses}}{\text{\# of 200 + \# of 304 responses}}$$

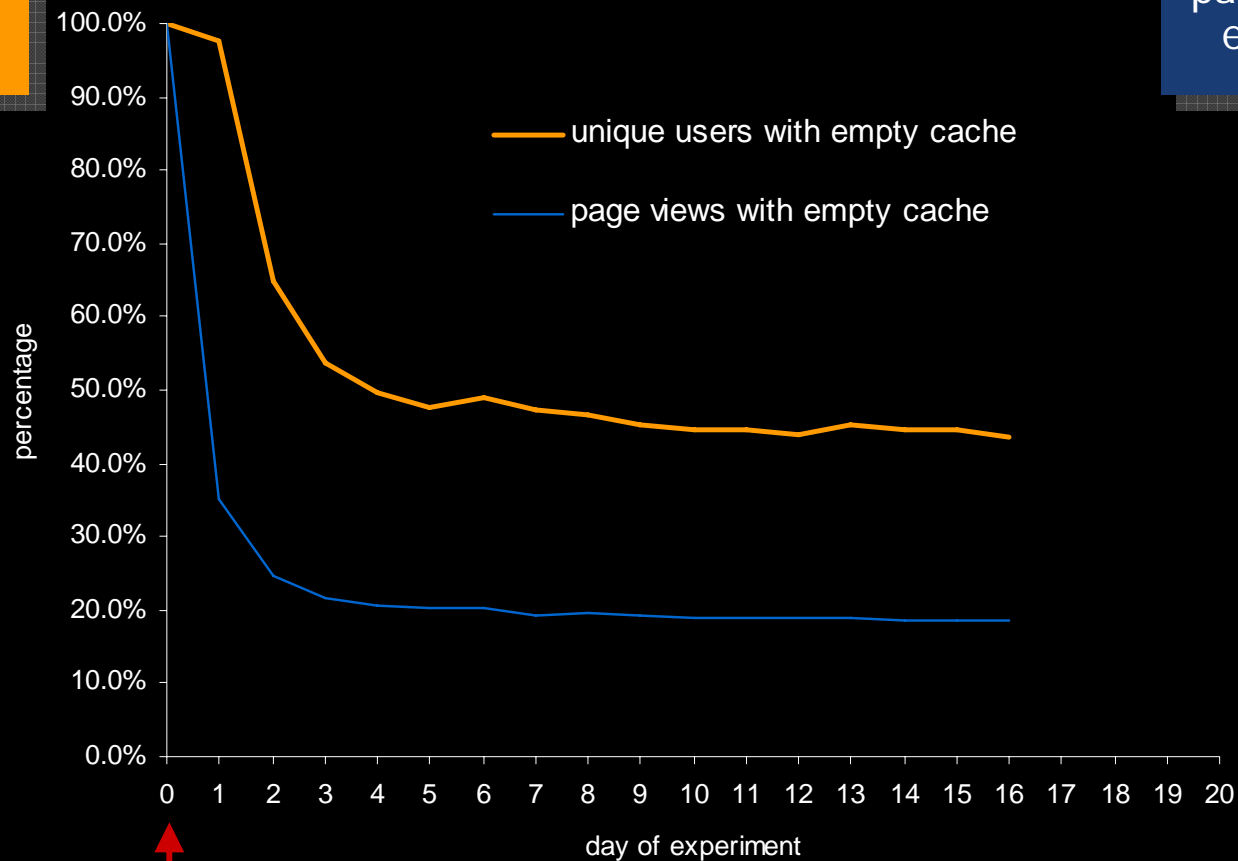

Surprising Results

users with
empty cache

40-60%

page views with
empty cache

~20%



Experiment Takeaways

1. The empty cache user experience is more prevalent than you think!
2. Therefore, optimize for both full cache *and* empty cache experience.


```
YAHOO.util.Motion = function(el, attributes, duration, method) {  
    if (el) {  
        this.initMotion(el, attributes, duration, method);  
    }  
};  
YAHOO.util.Motion.prototype = new YAHOO.util.Anim();  
YAHOO.util.Motion.prototype.initMotion = function(el, attributes, duration, method) {  
    YAHOO.util.Anim.call(this, el, attributes, duration, method);  
};
```

YAHOO! USER INTERFACE BLOG

News and Articles about Designing and Developing with Yahoo! Libraries.

Blog

About

search in blog...

Performance Research, Part 2: Browser Cache Usage - Exposed!

January 4, 2007 at 12:24 pm by Tenni Theurer | In [Development](#) |

This is the second in a series of articles describing experiments conducted to learn more about optimizing web page performance. You may be wondering why you're reading a performance article on the YUI Blog. It turns out that most of web page performance is affected by front-end engineering, that is, the user interface design and development.

In an earlier post, I described [What the 80/20 Rule Tells Us about Reducing HTTP Requests](#). Since browsers spend 80% of the time fetching external components including scripts, stylesheets and images, reducing the number of HTTP requests has the biggest impact on reducing response time. But shouldn't everything be saved in the browser's cache anyway?

Why does cache matter?

It's important to differentiate between end user experiences for an empty versus a full cache page view. An "empty cache" means the browser bypasses the disk cache and has to request all the components to load the page. A "full cache" means all (or at least most) of the components are found in the disk cache and the corresponding HTTP requests are avoided.

The main reason for an empty cache page view is because the user is visiting the page for the first time and the browser has to download all the components to load the page. Other reasons include:

- The user visited the page previously but cleared the browser cache.
- The browser cache was automatically cleared, based on the browser's settings.

SYNDICATE

All Entries:



All Comments:



RECENT POSTS

[Welcoming Mike Lee and Dav Glass to the YUI Team](#)

[YUI Theater – Doug Geoffray: "From the Mouth of a Screenreader"](#)

[YUI Implementation Focus: Dustin Diaz's DED|Chain](#)

[Happy Birthday to the Pattern Library, Too!](#)

[YUI Theater – "Browser Wars Episode II: Attack of the DOMs"](#)

[Free Excerpt: Nicholas Zakas on YUI Connection Manager, from Professional Ajax 2nd Edition](#)

RECENT READERS



tenni08

#4



Cookies

Set Scope Correctly

1



user requests
www.yahoo.com



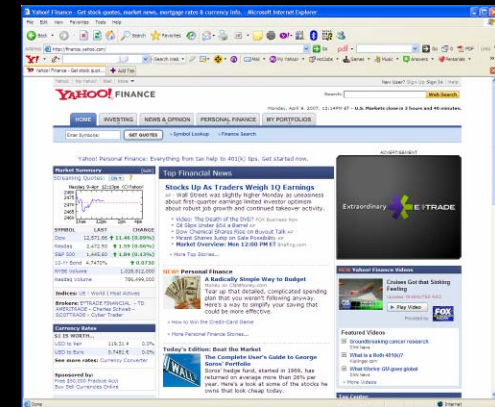
HTTP response header sent by the web server:

HTTP/1.1 200 OK

Content-Type: text/html; charset=utf-8

Set-Cookie: C=abcdefghijklmnopqrstuvxyz; domain=.yahoo.com

Because broad scope adds up



HTTP request header sent by the browser:

GET / HTTP/1.1

Host: **finance.yahoo.com**

User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; ...

Cookie: C=abcdefghijklmnpqrstuvwxyz;

Impact on Response Time

Cookie Size	Time	Delta
0 bytes	78 ms	0 ms
500 bytes	79 ms	+1 ms
1000 bytes	94 ms	+16 ms
1500 bytes	109 ms	+31 ms
2000 bytes	125 ms	+47 ms
2500 bytes	141 ms	+63 ms
3000 bytes	156 ms	+78 ms

keep sizes low

80 ms delay

dialup users

Cookie Sizes across the Web

	Total Cookie Size
Amazon	60 bytes
Google	72 bytes
Yahoo	122 bytes
CNN	184 bytes
YouTube	218 bytes
MSN	268 bytes
eBay	331 bytes
MySpace	500 bytes

Experiment Takeaways

1. eliminate unnecessary cookies
2. keep cookie sizes low
3. set cookies at the appropriate domain (or sub-domain) level
4. set Expires date appropriately

```
YAHOO.util.Motion = function(el, attributes, duration, method) {  
  if (el) {  
    this.initMotion(el, attributes, duration, method);  
  }  
}  
  
YAHOO.util.Motion.prototype = new YAHOO.util.Anim();  
  
YAHOO.util.Motion.prototype.initMotion = function(el, attributes, duration, method) {  
  YAHOO.util.Anim.call(this, el, attributes, duration, method);  
}
```

YAHOO! USER INTERFACE BLOG

News and Articles about Designing and Developing with Yahoo! Libraries.



Blog

About

search in blog...

Performance Research, Part 3: When the Cookie Crumbles

March 1, 2007 at 4:41 pm by Tenni Theurer | In Development |

This article, co-written by Patty Chi, is the third in a series of articles describing experiments conducted to learn more about optimizing web page performance (Part 1, Part 2). You may be wondering why you're reading a performance article on the YUI Blog. It turns out that most of web page performance is affected by front-end engineering — that is, the user interface design and development.

HTTP cookies are used for a variety of reasons such as authentication and personalization. Information about cookies is exchanged in the HTTP headers between web servers and browsers. This article discusses the impact of cookies on the overall user response time.

HTTP Quick Review

Cookies originate from web servers when browsers request a page. Here is a sample HTTP header sent by the web server after a request for `www.yahoo.com`:

```
HTTP/1.1 200 OK  
Content-Type: text/html; charset=utf-8  
Set-Cookie: C=abcde; path=/; domain=.yahoo.com
```

The header includes information about the response such as the protocol version, status code, and content-type. The Set-Cookie is also included in the response and in this example the name of the cookie is "C" and the value of the cookie is "abcde". Note: The maximum size of a cookie is

SYNDICATE

All Entries:



All Comments:



RECENT POSTS

Welcoming Mike Lee and Dav Glass to the YUI Team

YUI Theater — Doug Geoffray: "From the Mouth of a Screenreader"

YUI Implementation Focus: Dustin Diaz's DED|Chain

Happy Birthday to the Pattern Library, Too!

YUI Theater — "Browser Wars Episode II: Attack of the DOMs"

Free Excerpt: Nicholas Zakas on YUI Connection Manager, from Professional Ajax 2nd Edition

RECENT READERS



tenni08

<http://yuiblog.com/blog/2007/03/01/performance-research-part-3>

#15



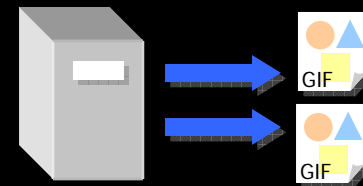
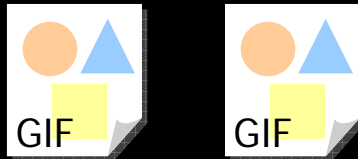
Parallel Downloads

Parallel Downloads

Two components

in parallel

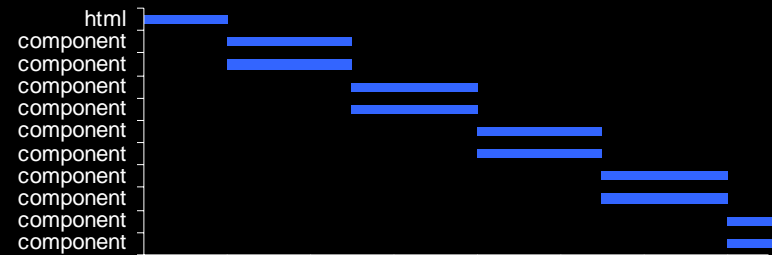
per hostname



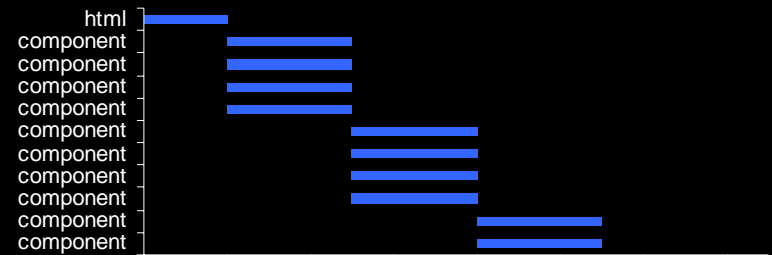
per HTTP/1.1

Parallel Downloads

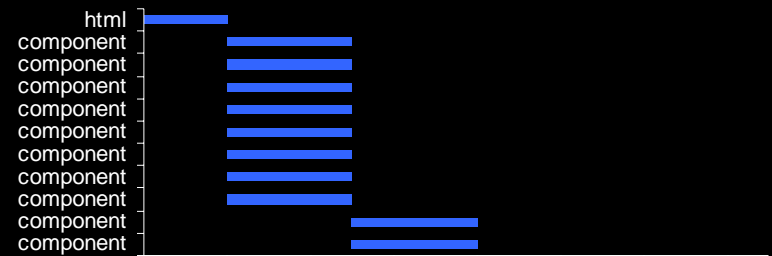
Two in parallel



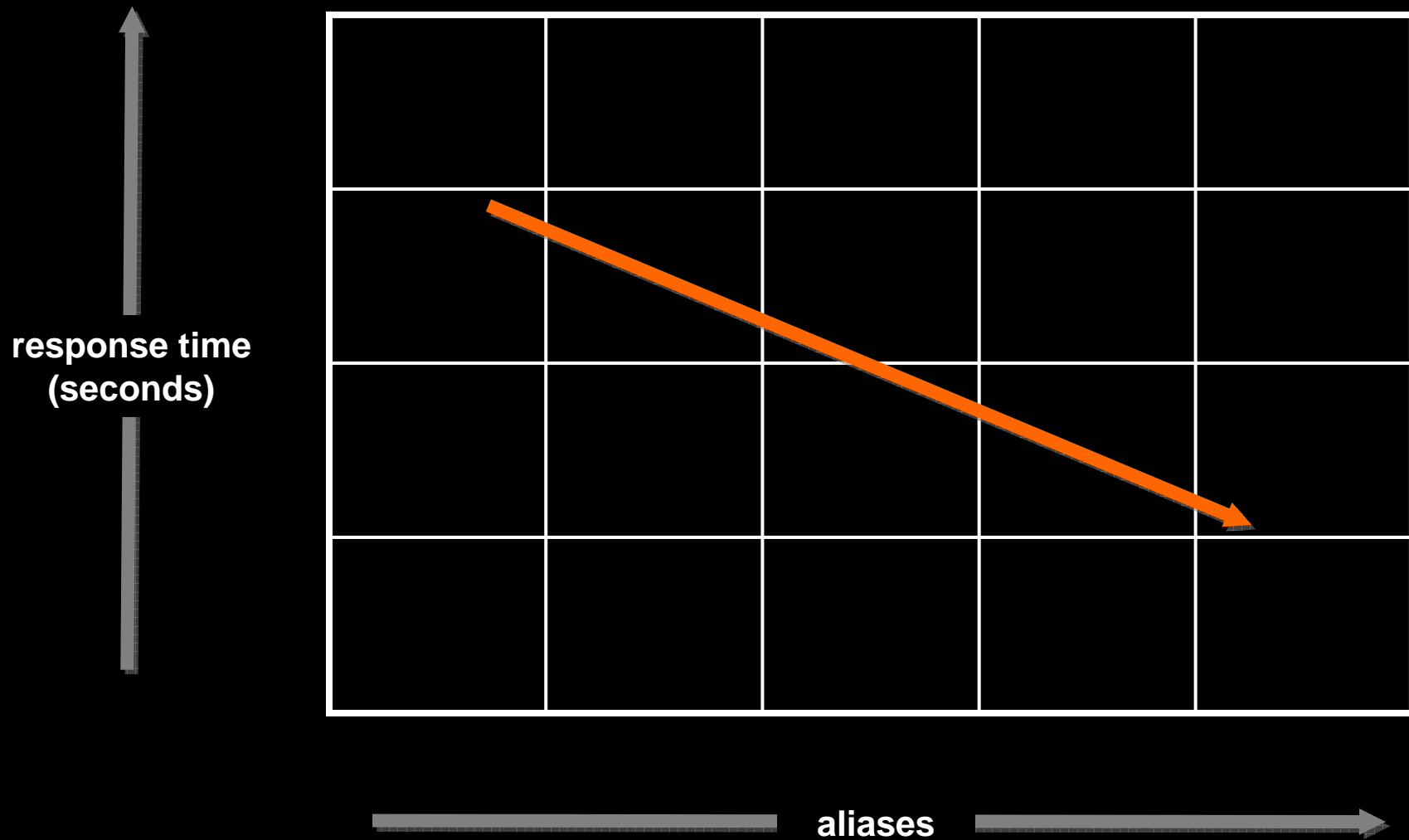
Four in parallel



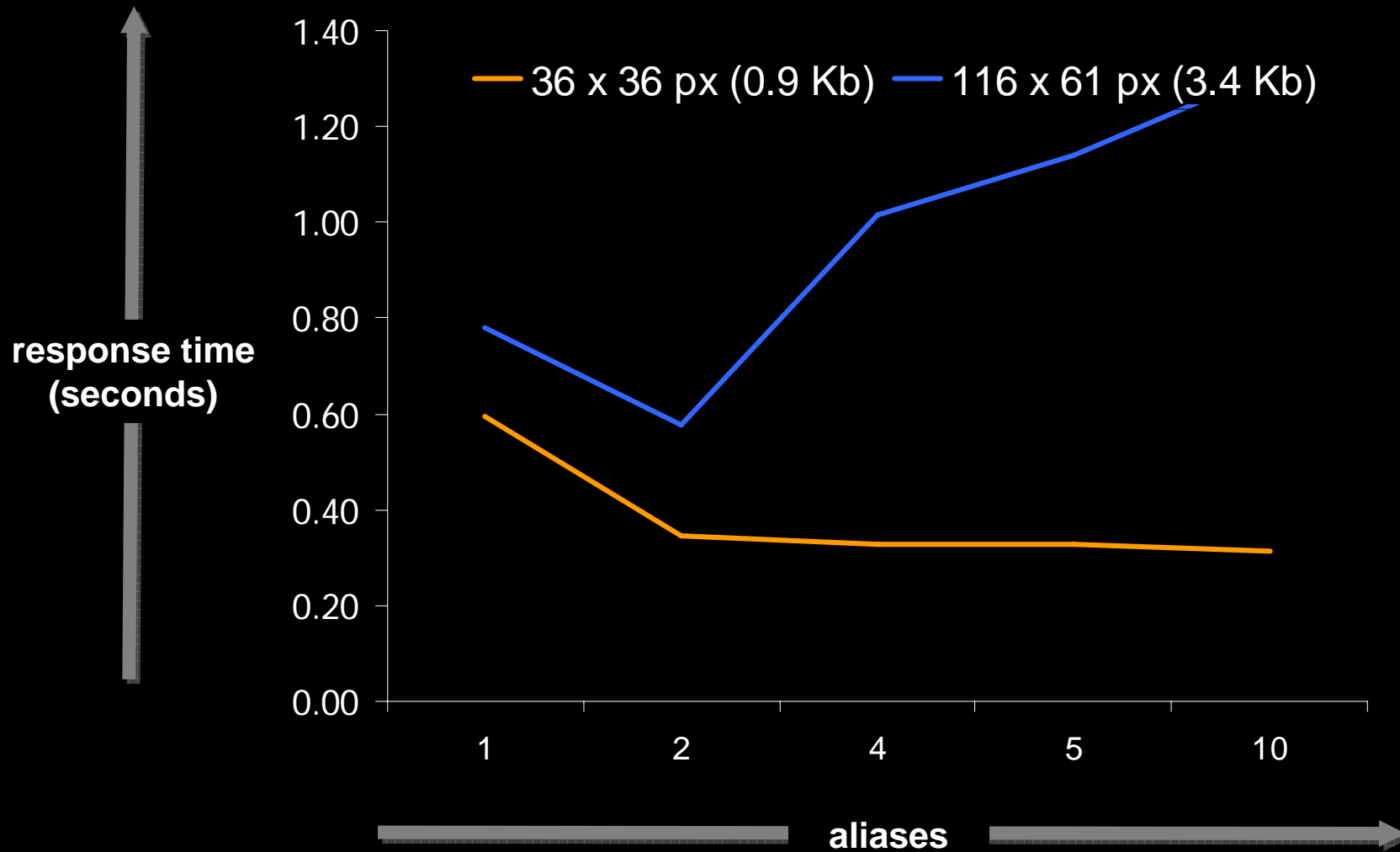
Eight in parallel



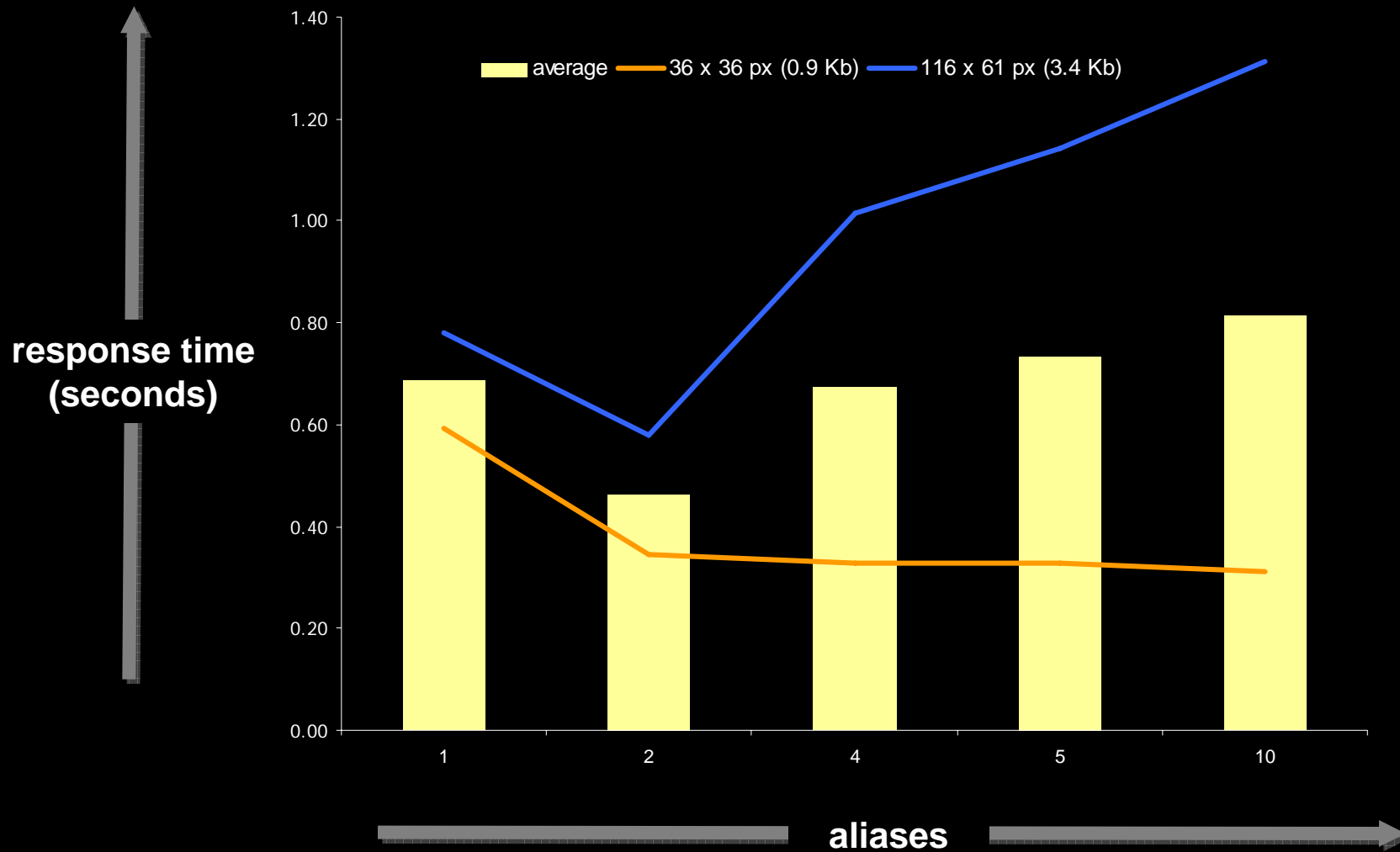
Maximizing Parallel Downloads



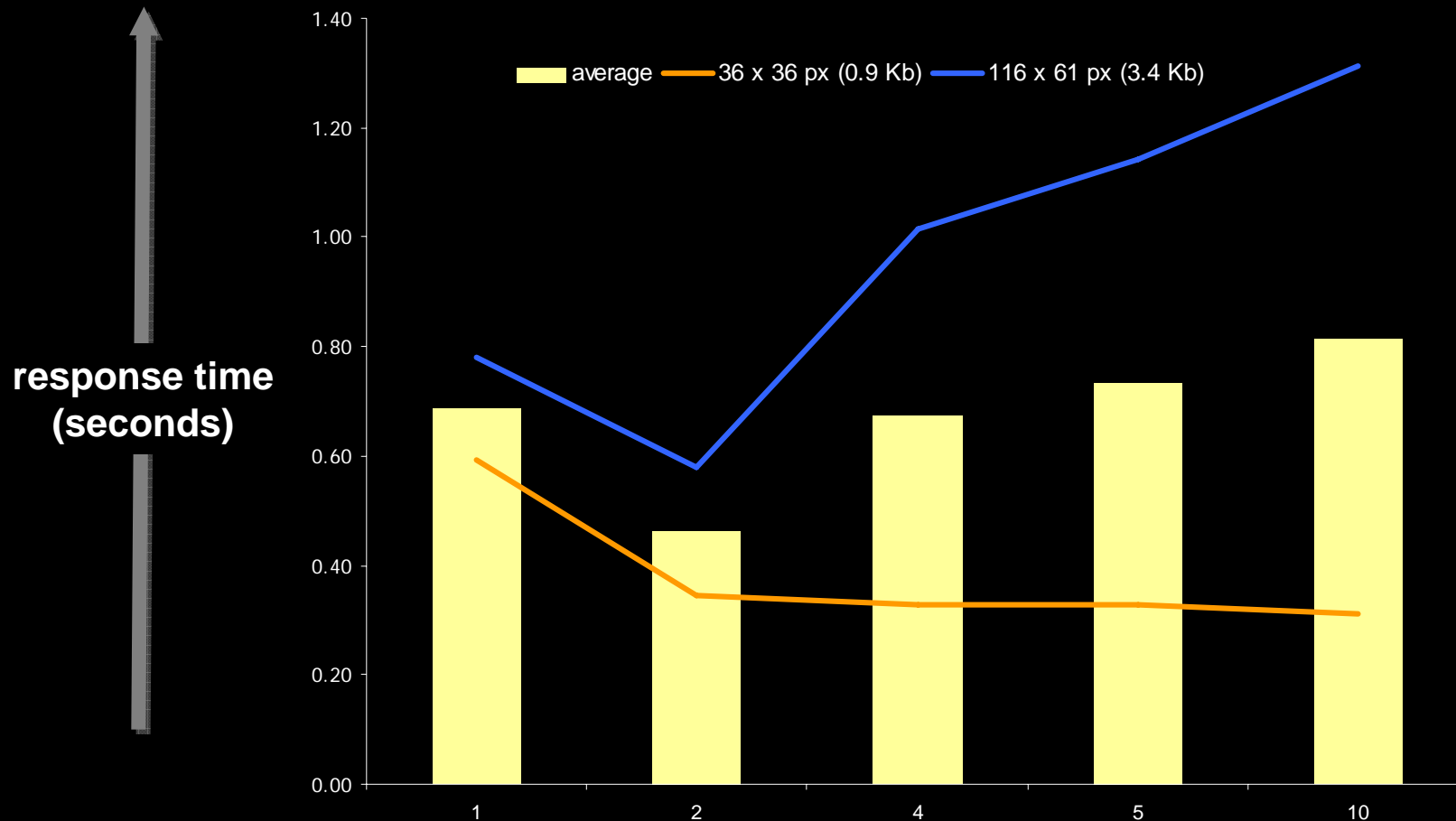
Maximizing Parallel Downloads



Maximizing Parallel Downloads



Maximizing Parallel Downloads



**rule of thumb: use at least two but
no more than four aliases**

Experiment Takeaways

1. consider the effects of CPU thrashing
2. DNS lookup times vary across ISPs and geographic locations
3. domain names may not be cached

```
YAHOO.util.Motion = function(el, attributes, duration, method) {  
    if (el) {  
        this.initMotion(el, attributes, duration, method);  
    }  
};  
YAHOO.util.Motion.prototype = new YAHOO.util.Anim();  
YAHOO.util.Motion.prototype.initMotion = function(el, attributes, duration, method) {  
    YAHOO.util.Anim.call(this, el, attributes, duration, method);  
};
```

YAHOO! USER INTERFACE BLOG

News and Articles about Designing and Developing with Yahoo! Libraries.



Blog

About

Performance Research, Part 4: Maximizing Parallel Downloads in the Carpool Lane

April 11, 2007 at 11:47 am by Tenni Theurer | In [Development](#) |

This article, co-written by [Steve Souders](#), is the fourth in a series of articles describing experiments conducted to learn more about optimizing web page performance ([Part 1](#), [Part 2](#), [Part 3](#)). You may be wondering why you're reading a performance article on the YUI Blog. It turns out that most of web page performance is affected by front-end engineering, that is, the user interface design and development.

Parallel Downloads

The biggest impact on end-user response times is the number of components in the page. Each component requires an extra HTTP request, perhaps not when the cache is full, but definitely when the cache is empty. Knowing that the browser performs HTTP requests in parallel, you may ask why the number of HTTP requests affects response time. Can't the browser download them all at once?

The explanation goes back to the HTTP/1.1 spec, which suggests that browsers download two components in parallel per hostname. Many web pages download all their components from a single hostname. Viewing these HTTP requests reveals a stair-step pattern, as shown

SYNDICATE

All Entries:

[+ MY YAHOO!](#) [RSS](#)

All Comments:

[+ MY YAHOO!](#) [RSS](#)

RECENT POSTS

[Performance Research, Part 4: Maximizing Parallel Downloads in the Carpool Lane](#)

[JSON and Browser Security](#)

[YUI Version 2.2.1 Released, and Graded Browser Support Update](#)

[Welcoming Mike Lee and Dav Glass to the YUI Team](#)

[YUI Theater — Doug Geoffray: "From the Mouth of a Screenreader"](#)

[YUI Implementation Focus: Dustin Diaz's DED|Chain](#)

RECENT READERS

Summary

What the 80/20 Rule Tells Us about
Reducing HTTP Requests

<http://yuiblog.com/blog/2006/11/28/performance-research-part-1/>

Browser Cache Usage – Exposed!

<http://yuiblog.com/blog/2007/01/04/performance-research-part-2/>

When the Cookie Crumbles

<http://yuiblog.com/blog/2007/03/01/performance-research-part-3/>

Maximizing Parallel Downloads in the
Carpool Lane

<http://yuiblog.com/blog/2007/04/11/performance-research-part-4/>

14 Rules

(presented as “one dozen rules”)



1. Make fewer HTTP requests
2. Use a CDN
3. Add an Expires header
4. Gzip components
5. Put CSS at the top
6. Move JS to the bottom
7. Avoid CSS expressions
8. Make JS & CSS external
9. Reduce DNS lookups
10. Minify JS
11. Avoid redirects
12. Remove duplicate scripts
13. Turn off ETags
14. Make Ajax cachable and small

Rule 1: Make fewer HTTP requests

CSS sprites

Combined / concatenated JS and CSS
files

image maps

inline (data) images

CSS Sprites



```
<span style="
  background-image: url('sprites.gif');
  background-position: -260px -90px;">
</span>
```

size of combined image is less

<http://alistapart.com/articles/sprites>

Combined Scripts, Combined Stylesheets

	Scripts	Stylesheets
amazon.com	3	1
aol.com	18	1
cnn.com	11	2
ebay.com	7	2
froogle.google.com	1	1
msn.com	9	1
myspace.com	2	2
wikipedia.org	3	1
yahoo.com	4	1
youtube.com	7	3
Average	6.5	1.5

Inline (data:) Images

data: URL scheme

data:[<mediatype>][;base64],<data>

<IMG ALT="Red Star" 

SRC="data:image/gif;base64,R0lGODlhDAAMALMLAPN8ffBiYvWWlvrKy/FvcPewsO9VVf
ajo+w6O/zl5estLv/8/AAAAAAAAAAAAAAAAACH5BAEAAAsALAAAAAAMAAwAAAQzcElZyryT
EHyTUgknHd9xGV+qKsYirKkwDYiKDBiatt2H1KBLQRFIJAIKywRgmhWAIlEEADs=">

not supported in IE

avoid increasing size of HTML pages:
put inline images in cached stylesheets

<http://tools.ietf.org/html/rfc2397>

The end.

Rule 2: Use a CDN

amazon.com	Akamai
aol.com	Akamai
cnn.com	
ebay.com	Akamai, Mirror Image
google.com	
msn.com	SAVVIS
myspace.com	Akamai, Limelight
wikipedia.org	
yahoo.com	Akamai
youtube.com	

distributed your static content before
distributing your dynamic content

Rule 3: Add an Expires header

not just for images

	Images	Stylesheets	Scripts	%	Median Age
amazon.com	0/62	0/1	0/3	0%	114 days
aol.com	23/43	1/1	6/18	48%	217 days
cnn.com	0/138	0/2	2/11	1%	227 days
ebay.com	16/20	0/2	0/7	55%	140 days
froogle.google.com	1/23	0/1	0/1	4%	454 days
msn.com	32/35	1/1	3/9	80%	34 days
myspace.com	0/18	0/2	0/2	0%	1 day
wikipedia.org	6/8	1/1	2/3	75%	1 day
yahoo.com	23/23	1/1	4/4	100%	n/a
youtube.com	0/32	0/3	0/7	0%	26 days



Rule 4: Gzip components

- you can affect users' download times
- 90%+ of browsers support compression

Gzip vs. Deflate

	Size	Gzip		Deflate	
		Size	Savings	Size	Savings
Script	3.3K	1.1K	67%	1.1K	66%
Script	39.7K	14.5K	64%	16.6K	58%
Stylesheet	1.0K	0.4K	56%	0.5K	52%
Stylesheet	14.1K	3.7K	73%	4.7K	67%

Gzip compresses more

Gzip supported in more browsers

Gzip: not just for HTML

	HTML	Scripts	Stylesheets
amazon.com	x		
aol.com	x	some	some
cnn.com			
ebay.com	x		
froogle.google.com	x	x	x
msn.com	x	deflate	deflate
myspace.com	x	x	x
wikipedia.org	x	x	x
yahoo.com	x	x	x
youtube.com	x	some	some

gzip scripts, stylesheets, XML, JSON
(not images, PDF)



Free YUI Hosting includes:

- Aggregated files
- With Expires headers
- On a CDN
- Gzipped

Free Hosting of YUI Files from Yahoo!

February 22, 2007 at 9:18 pm by Nate Koechley | In Development |

Including with this week's release of YUI 2.2.0, the one-year anniversary of the YUI open-source release, and as a nod to the YUI Party just a few days ago, we're opening up free YUI hosting from the Yahoo! network to all YUI implementers. If you're using YUI for your own project, we'll serve the files for you — gzipped, with good cache-control, using our state-of-the-art network, for free. You count on these files being continuously available because they're the same files served by the same source that we use to host YUI implementations at Yahoo!.

Files served from Yahoo!'s network include version numbers in filepaths, allowing you to reference a specific version in your code. Previous versions are retained even as new versions are released. While we're providing no explicit QA with respect to the availability of legacy code, our intent is to serve as a permanent availability of legacy YUI files.

Why Provide YUI Hosting on Yahoo!'s Network?

We're opening up the benefits of YUI to Yahoo! users for the same reasons we open-source our in-house JavaScript toolkit to other web companies. The progress being made by developers in richness and usability today is healthy for the web and, by extension, good for Yahoo! We want to do everything we can do to enhance that evolution — whether it's opening up YUI, hosting YUI files, or creating best-of-breed APIs like the recently-announced Browser-Based Authentication system.

SYNDICATE

All Posts

[+ MY YAHOO!](#) [RSS](#)

All Comments

[+ MY YAHOO!](#) [RSS](#)

All Development Posts

[+ MY YAHOO!](#) [RSS](#)

All Design Posts

[+ MY YAHOO!](#) [RSS](#)

[+ MY YAHOO!](#) [RSS](#)

[+ MY YAHOO!](#) [RSS](#)

"In the Wild" Posts

[+ MY YAHOO!](#) [RSS](#)

Performance Posts

[+ MY YAHOO!](#) [RSS](#)

RECENT POSTS

An Interview with Ted Husted, Creator of YUI Community Site "Planet Yazaar"

Douglas Crockford To Speak at the Yahoo! Widgets Conference on "JavaScript: The Good Parts"

YUI Theater — Grady Booch: "The Promise, the Limits, the Beauty of Software"

Rule 5: Put CSS at the top

stylesheets block rendering in IE

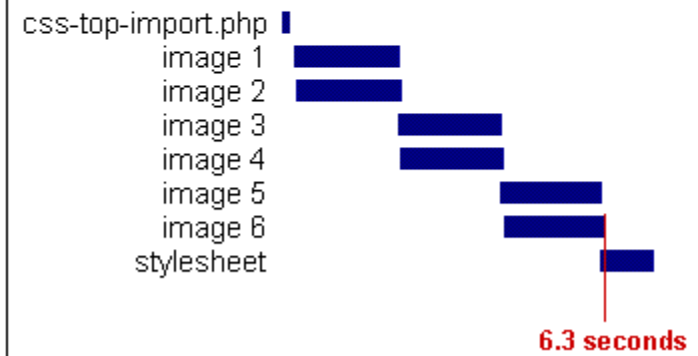
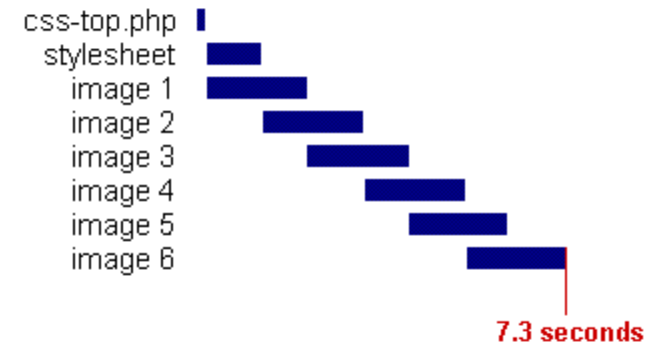
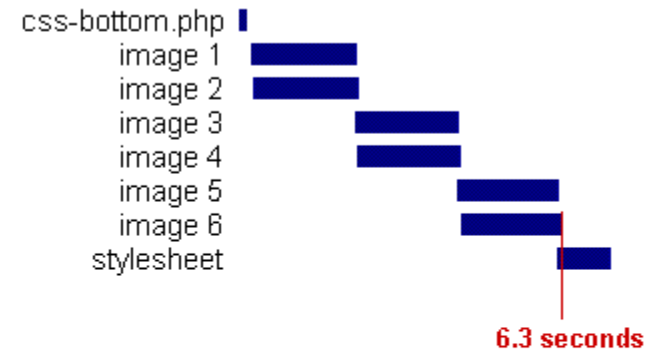
<http://stevesouders.com/examples/css-bottom.php>

solution: put stylesheets in HEAD (per spec)

avoids Flash of Unstyled Content

use `<link>` (not `@import`)

Slowest is actually the Fastest



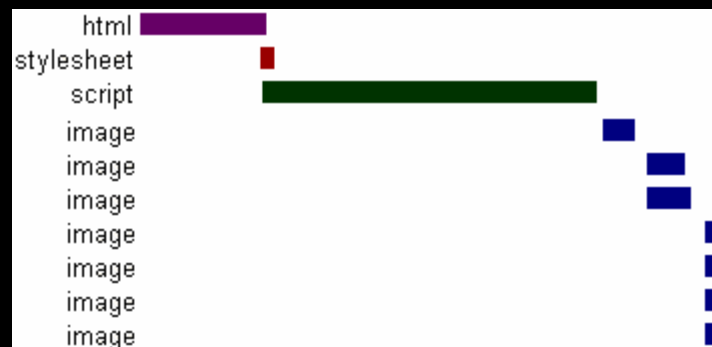
Rule 6: Move scripts to the bottom

scripts block rendering of everything
below them in the page

scripts block parallel downloads across
all hostnames

IE and FF

<http://stevesouders.com/examples/js-middle.php>



What about defer?

- script defer attribute is not a solution
- blocks rendering and downloads in FF
 - slight blocking in IE

Rule 7: Avoid CSS expressions

Can be used to set CSS properties dynamically in IE

```
width: expression(  
    document.body.clientWidth < 600 ?  
    "600px" : "auto" );
```

But problematic because expressions execute many times

- mouse move, key press, resize, scroll, etc.

<http://stevesouders.com/examples/expression-counter.php>

Rule 8: Make JS and CSS external

Inline: bigger HTML but no HTTP request

External: cachable but extra HTTP

Variables:

- page views per user (per session)
- empty vs. full cache stats
- component re-use

External is typically better

- home pages may be an exception due to cache behavior of browser's startpage.

Post-Onload Download

inline in front page

download external files after onload

```
window.onload = downloadComponents;  
function downloadComponents() {  
    var elem = document.createElement("script");  
    elem.src = "http://.../file1.js";  
    document.body.appendChild(elem);  
    ...  
}
```

speeds up secondary pages

Dynamic Inlining

start with post-onload download

set cookie after components
downloaded

server-side:

- if cookie, use external
- else, do inline with post-onload
download

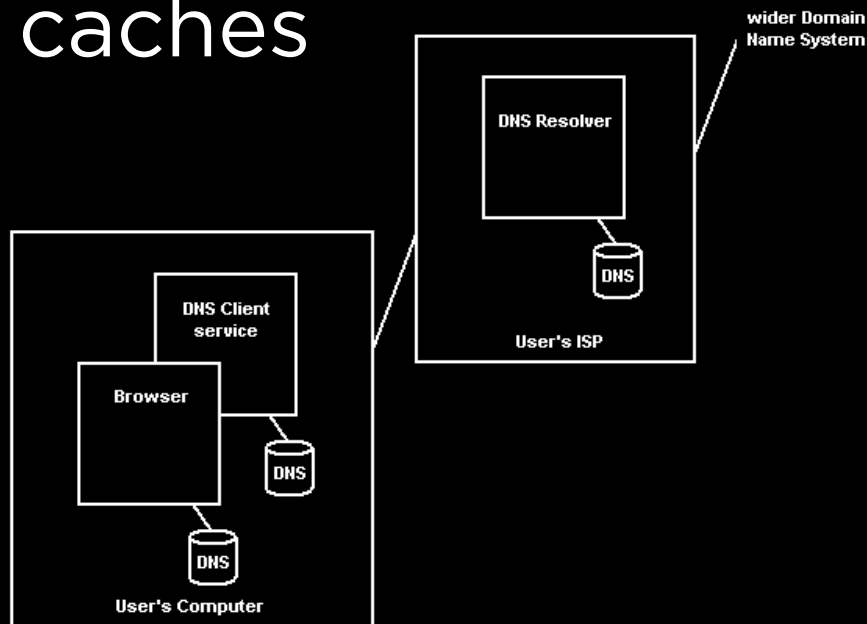
cookie expiration date is key
speeds up all pages

Rule 9: Reduce DNS lookups

typically 20-120 ms

block parallel downloads

OS and browser both have DNS
caches



Best practice:

Max 2-4 hosts

Use keep-alive

TTL (Time To Live)

www.amazon.com	1 minute
www.aol.com	1 minute
www.cnn.com	10 minutes
www.ebay.com	1 hour
www.google.com	5 minutes
www.msn.com	5 minutes
www.myspace.com	1 hour
www.wikipedia.org	1 hour
www.yahoo.com	1 minute
www.youtube.com	5 minutes

TTL – how long record can be cached
Browser settings override TTL

Twaking Browser's DNS Cache

IE

- DnsCacheTimeout: 30 minutes
- KeepAliveTimeout: 1 minute
- ServerInfoTimeout: 2 minutes

Firefox

- network.dnsCacheExpiration: 1 minute
- network.dnsCacheEntries: 20
- network.http.keep-alive.timeout: 5 minutes
- Fasterfox Extension:
 - 1 hour, 512 entries, 30 seconds

Rule 10: Minify JavaScript

	Minify External?	Minify Inline?
www.amazon.com	no	no
www.aol.com	no	no
www.cnn.com	no	no
www.ebay.com	yes	no
froogle.google.com	yes	yes
www.msn.com	yes	yes
www.myspace.com	no	no
www.wikipedia.org	no	no
www.yahoo.com	yes	yes
www.youtube.com	no	no

minify inline scripts, too

Minify vs. Obfuscate

	Original	JSMin Savings	Dojo Savings
www.amazon.com	204K	31K (15%)	48K (24%)
www.aol.com	44K	4K (10%)	4K (10%)
www.cnn.com	98K	19K (20%)	24K (25%)
www.myspace.com	88K	23K (27%)	24K (28%)
www.wikipedia.org	42K	14K (34%)	16K (38%)
www.youtube.com	34K	8K (22%)	10K (29%)
Average	85K	17K (21%)	21K (25%)

minify – it's safer

<http://crockford.com/javascript/jsmin>

<http://dojotoolkit.org/docs/shrinksafe>

Rule 11: Avoid redirects

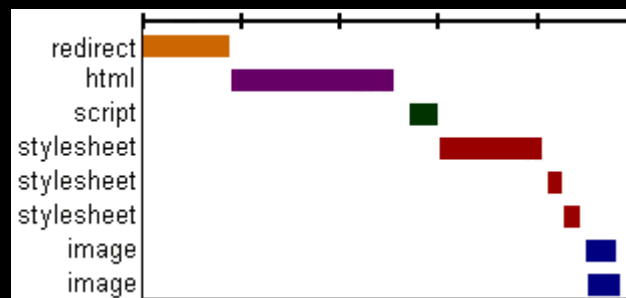
3xx status codes – mostly 301 and 302

HTTP/1.1 301 Moved Permanently

Location: <http://stevesouders.com/newuri>

add Expires headers so redirect
headers are cached

Redirects are worst form of blocking



<http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>

Redirects

Redirects	
www.amazon.com	no
www.aol.com	yes – secondary page
www.cnn.com	yes – initial page
www.ebay.com	yes – secondary page
froogle.google.com	no
www.msn.com	yes – initial page
www.myspace.com	yes – secondary page
www.wikipedia.org	yes – secondary page
www.yahoo.com	yes – secondary page
www.youtube.com	no

Rule 12: Remove Duplicate Scripts

(this rule was not presented live)

hurts performance

- extra HTTP requests (IE only)
- extra executions

atypical?

- 2 of 10 top sites contain duplicate scripts

team size, # of scripts

Script Insertion Functions

```
<?php
function insertScript($jsfile) {
    if ( alreadyInserted($jsfile) ) { return; }

    pushInserted($jsfile);

    if ( hasDependencies($jsfile) ) {
        $dependencies = getDependencies($jsfile);
        for ( $i = 0; $i < count($dependencies); $i++ ) {
            insertScript($dependencies[$i]);
        }
    }

    echo '<script type="text/javascript" src="' .
        getVersion($jsfile) . '"></script>';
}
?>
```


Rule 13: Turn off ETags

(this was #12 when presented live)

unique identifier returned in response

ETag: "c8897e-ae-4165acf0"

Last-Modified: Thu, 07 Oct 2004 20:54:08 GMT

used in conditional GET requests

If-None-Match: "c8897e-ae-4165acf0"

If-Modified-Since: Thu, 07 Oct 2004 20:54:08 GMT

Breaks caching:

if ETag doesn't match, can't send 304

Rule 14: Make AJAX cacheable and small

(this rule was not presented live)

XHR, JSON, iframe, dynamic scripts
can still be cached, minified, and
gzipped

a personalized response should still be
cacheable by that person

AJAX Example: Yahoo! Mail Beta

address book XML request

```
→ GET /yab/[...]&r=0.5289571053069156 HTTP/1.1
  Host: us.xxx.mail.yahoo.com
← HTTP/1.1 200 OK
  Date: Thu, 12 Apr 2007 19:39:09 GMT
  Cache-Control: private,max-age=0
  Last-Modified: Sat, 31 Mar 2007 01:17:17 GMT
  Content-Type: text/xml; charset=utf-8
  Content-Encoding: gzip
```

address book changes infrequently

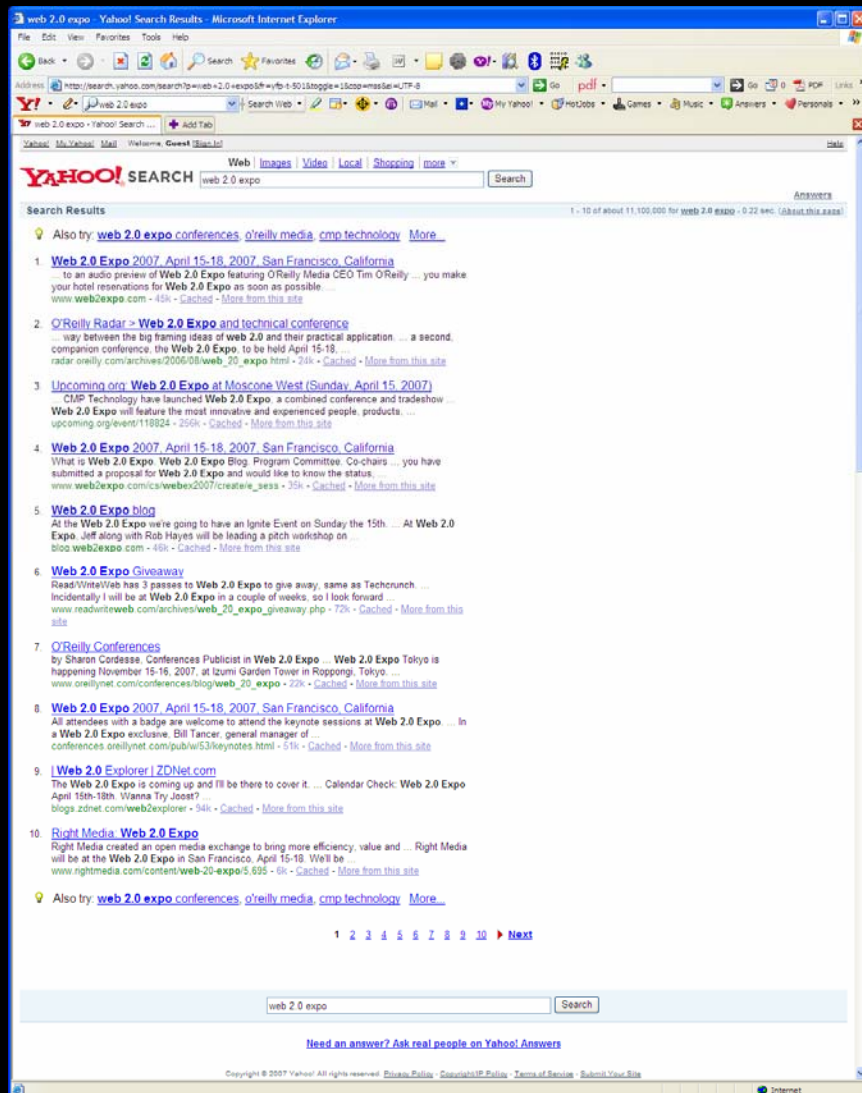
- cache it; add last-modified-time in URL



Case Studies

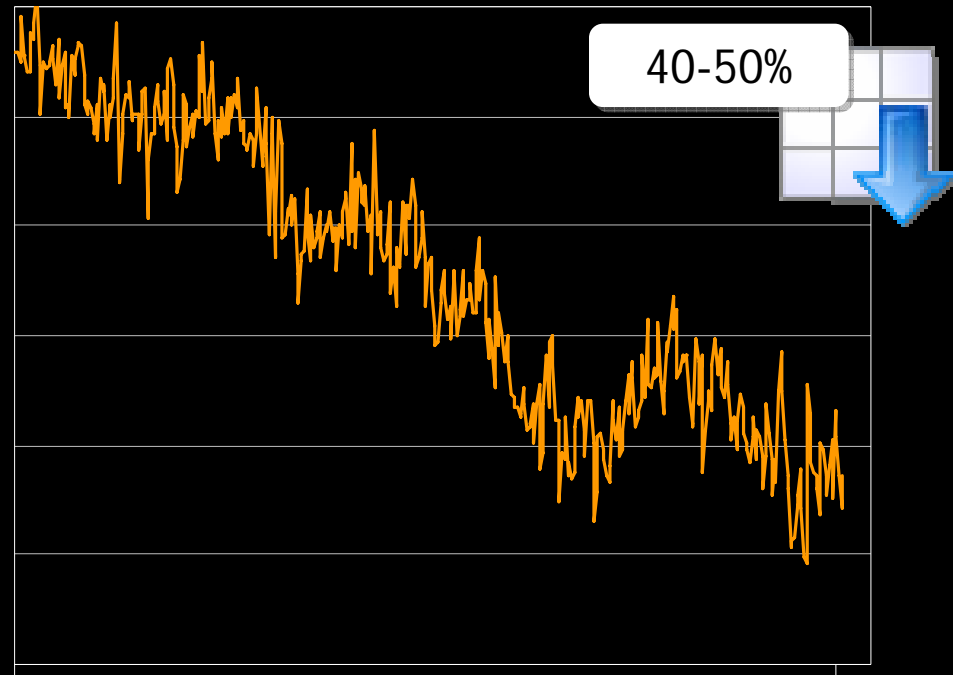
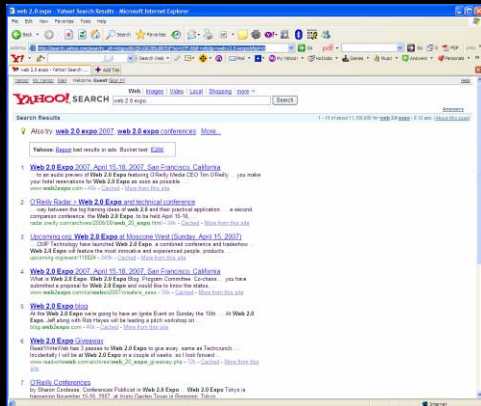
Case Study:

YAHOO! SEARCH



1. moved JS to onload
2. removed redirects
3. used image sprites
4. hosted JS on CDN
5. combined JS files

Case Study: **YAHOO!** SEARCH



1/25/06

3/25/07

What about performance and Web 2.0 apps?

client-side CPU is more of an issue
user expectations are higher
start off on the right foot: care!
measuring is different

Case Study:

YAHOO! MAIL
BETA

open messages in their own tabs

select a message in the list

view message in preview pane

instant
messaging

The screenshot displays the Yahoo! Mail BETA interface. On the left is a sidebar with folders like Inbox (8), Drafts, Sent, Spam (3200), and Trash (4). The main area shows an inbox with a table of messages. The message 'tina site updates' from Katherine Lehr is selected. To the right, a preview pane shows the content of this message. At the top right, an instant messaging window for 'dan theurer' is open. The bottom of the screen features a status bar with system information and a taskbar with various application icons.

	From	Subject	Date	Size
<input type="checkbox"/>	Jennifer Lee	Re: finding tina palmer	Sun, 4/8/07 8:55 AM	5KB
<input type="checkbox"/>	agbeh@yahoo.com	[agbeh] Digest Number 3274	Sun, 4/8/07 6:23 AM	17KB
<input type="checkbox"/>	cchang@mba1999.hbs.edu	Happy Easter!	Sun, 4/8/07 3:48 AM	10KB
<input type="checkbox"/>	John	IPCC Report, global warming, solar power	Sat, 4/7/07 5:28 PM	31KB
<input type="checkbox"/>	Jeanette Evans	FW: need soccer balls...	Sat, 4/7/07 4:41 PM	4KB
<input checked="" type="checkbox"/>	Katherine Lehr	tina site updates	Sat, 4/7/07 6:41 AM	2KB
<input type="checkbox"/>	Katherine Lehr	Fwd: Mountain Girl	Sat, 4/7/07 6:29 AM	5KB
<input type="checkbox"/>	BusinessWeek Online's Insid	Mittal & Son	Fri, 4/6/07 3:52 PM	37KB

tina site updates
Katherine Lehr <kath042@yahoo.com> View To: tenni08 <tenni08@yahoo.com>

the media madness continues...

4/6/07 Todd & Tina's story appears on the Good Day Iowa morning show airing on the local FOX affiliate. (no link)

4/6/07 Todd is interviewed on 100.7 The Fox, a local radio station. (no link)

4/7/07 Todd and Tina make front page news on the Gazette, eastern Iowa's major newspaper. (no link)

the day is here!!!!!!!

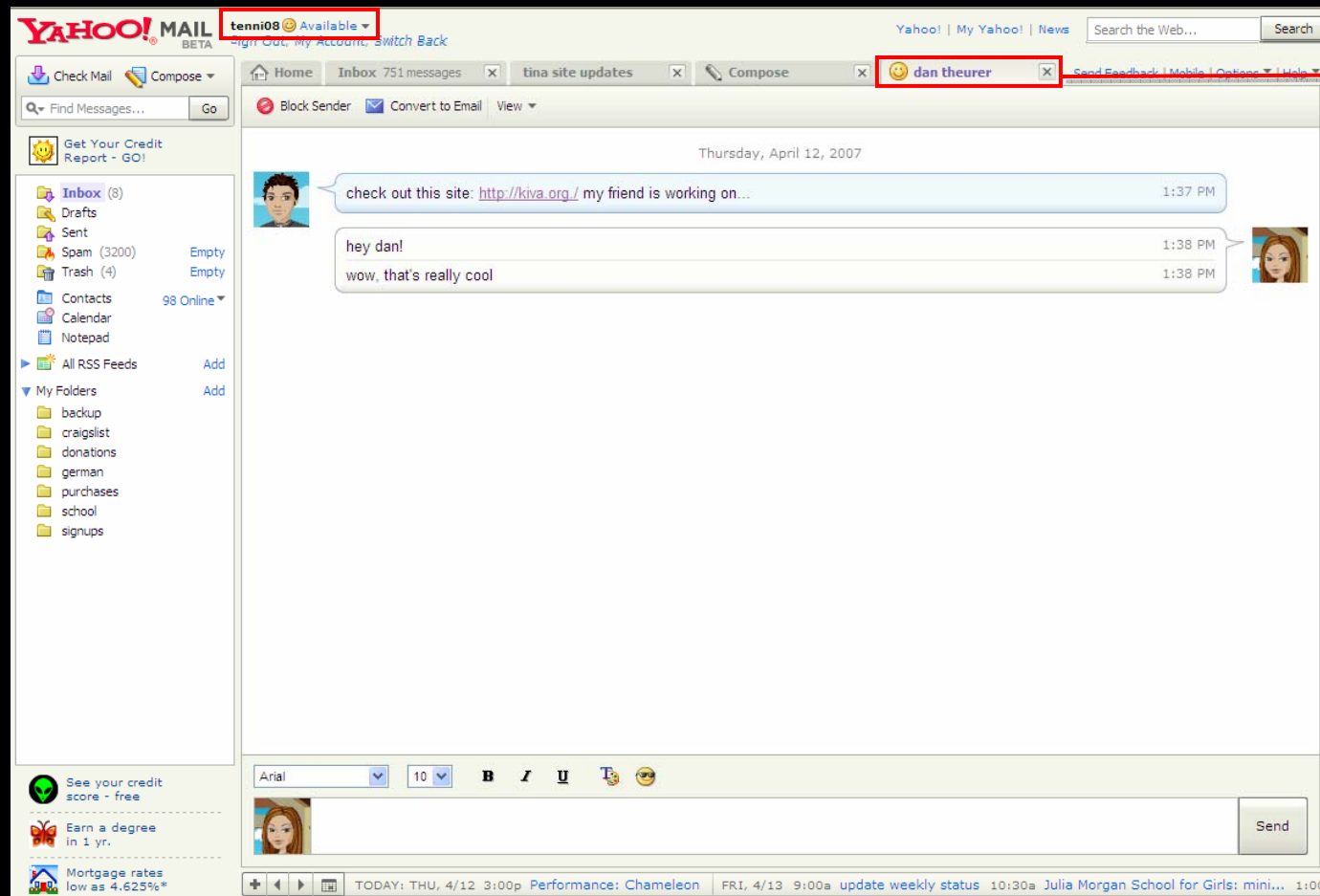
The fish are biting.
Get more visitors on your site using Yahoo! Search Marketing.
http://searchmarketing.yahoo.com/arp/sponsoredsearch_v2.php

calendar

Case Study:

YAHOO! MAIL
BETA

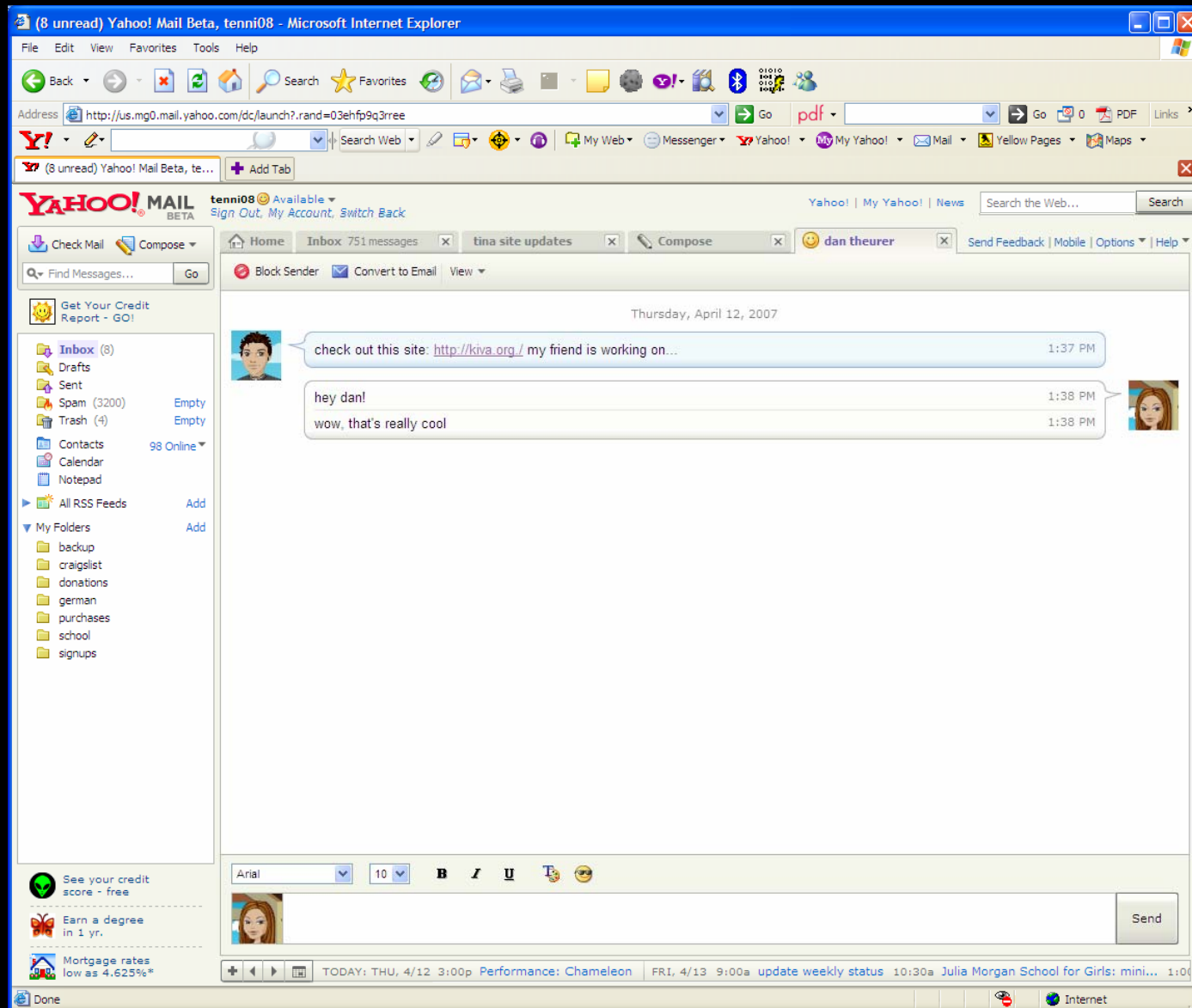
Does it meet user expectations?



instant
messaging

Case Study:

YAHOO! MAIL
BETA



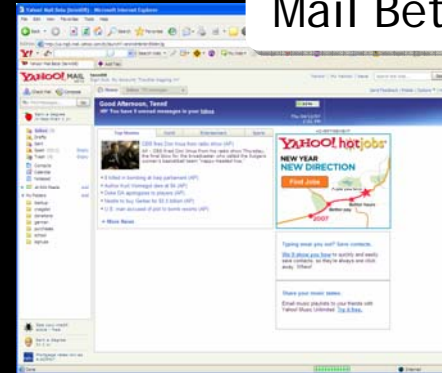
Case Study:

YAHOO! MAIL
BETA

Mail Classic



Mail Beta



Work flow

Time

Time

mail.yahoo.com

2.40 s

12.48 s

view inbox folder

1.66 s x 3 =

4.98 s

1.52 s

read message (x3)

2.13 s x 3 =

6.39 s

0.51 s x 3 =

1.53 s

compose message

2.21 s

0.34 s

confirm send

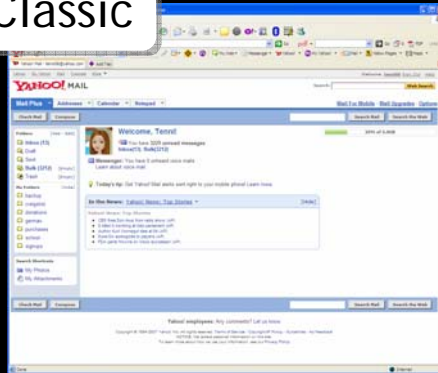
2.10 s

0 s

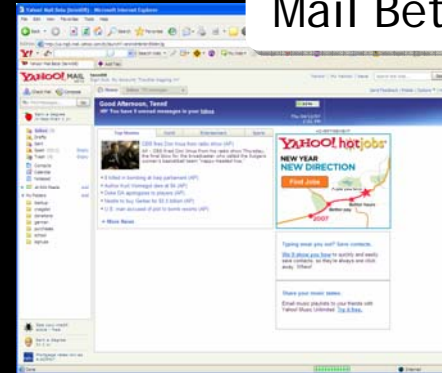
Case Study:



Mail Classic



Mail Beta



	Time	Time	Delta
mail.yahoo.com	2.40 s	12.48 s	+420%
view inbox folder	4.98 s	1.52 s	-70%
read message (x3)	6.39 s	1.53 s	-76%
compose message	2.21 s	0.34 s	-85%
confirm send	2.10 s	0 s	-100%
total time:	18.08 s	15.87 s	-12%

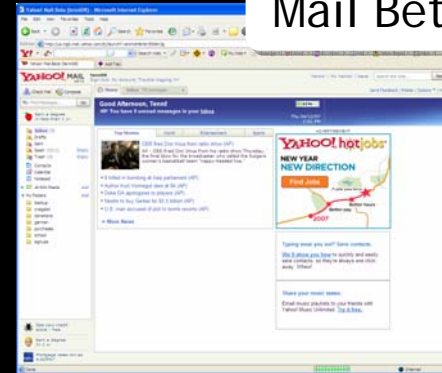
Case Study:



Mail Classic



Mail Beta



	Time	Time	Delta
mail.yahoo.com	2.40 s	12.48 s	+420%
view inbox folder	4.98 s	1.52 s	-70%
read message (x3)	6.39 s	1.53 s	-76%
compose message	2.21 s	0.34 s	-85%
confirm send	2.10 s	0 s	-100%
total time:	18.08 s	15.87 s	-12%

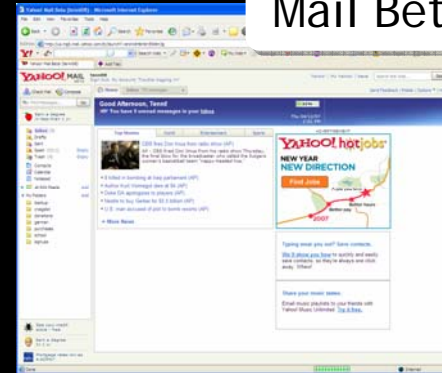
Case Study:



Mail Classic



Mail Beta



	Time	Time	Delta
mail.yahoo.com	2.40 s	12.48 s	+420%
view inbox folder	4.98 s	1.52 s	-70%
read message (x3)	6.39 s	1.53 s	-76%
compose message	2.21 s	0.34 s	-85%
confirm send	2.10 s	0 s	-100%
total time:	18.08 s	15.87 s	-12%

Live Analysis

IBM Page Detailer

packet sniffer

Windows only

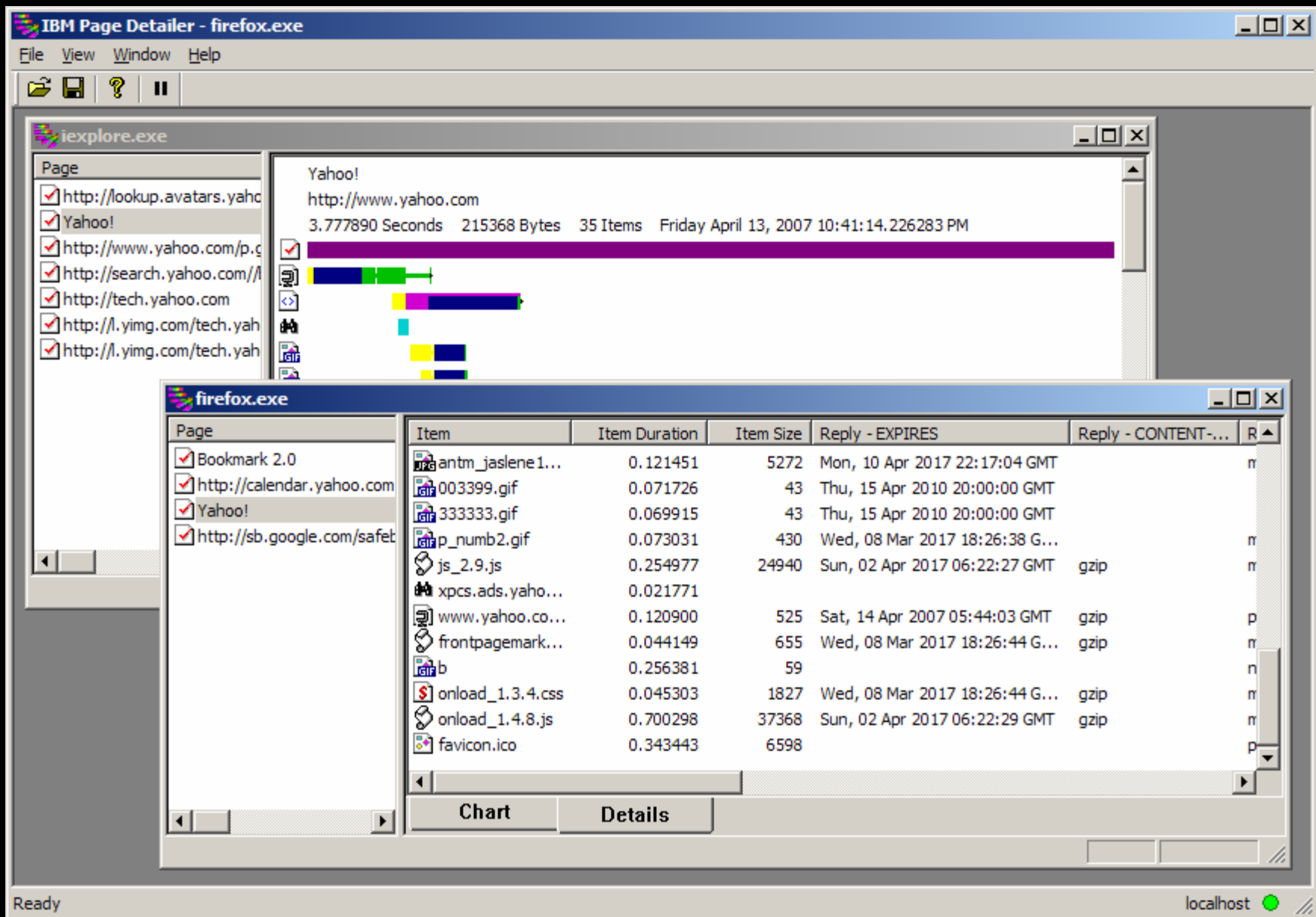
IE, FF, any .exe

c:\windows\wd_WS2s.ini

Executable=(NETSCAPE.EXE),(NETSCP6.EXE),(firefox.exe)

free trial, \$300 license

<http://alphaworks.ibm.com/tech/pagedetailer>



<http://alphaworks.ibm.com/tech/pagedetailer>

Fasterfox

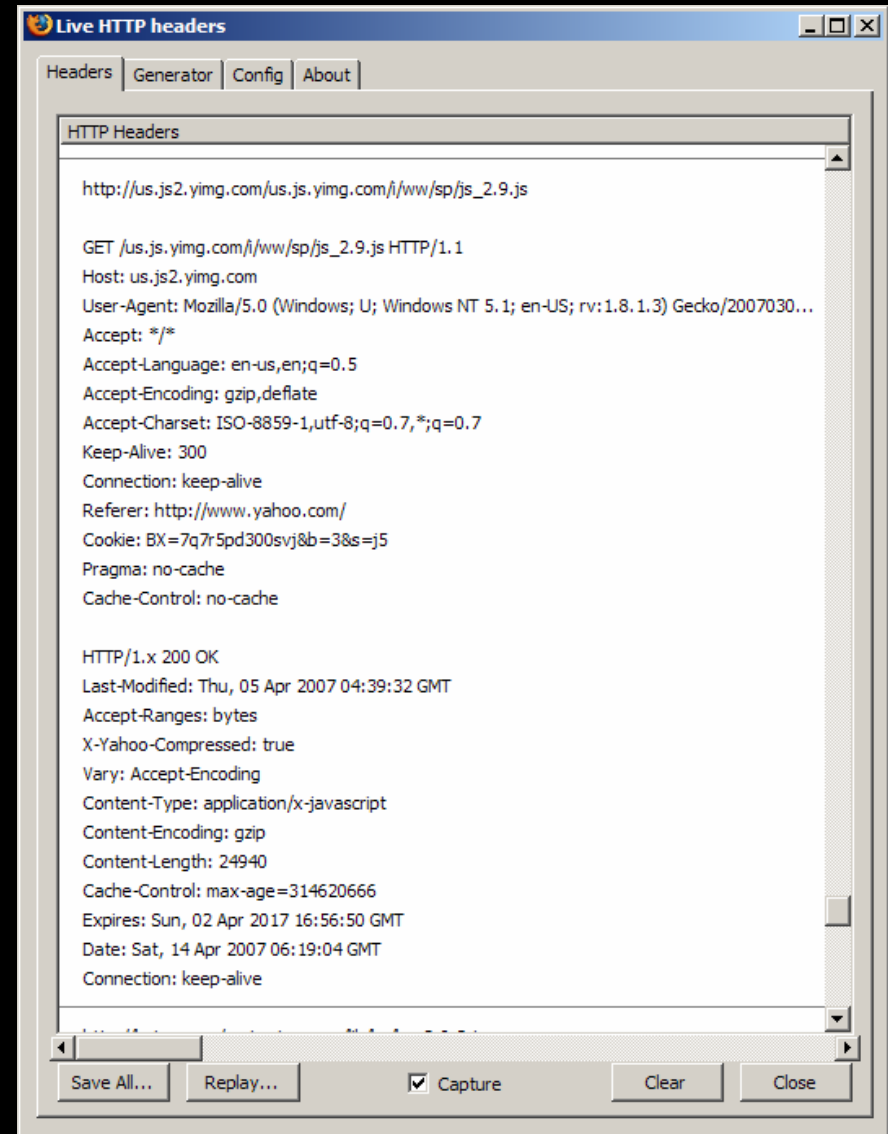
measures load time of pages
alters config settings for faster
loading
Firefox extension
free



<http://fasterfox.mozdev.org/>

LiveHTTPHeaders

view HTTP headers
Firefox extension
free



<http://livehttpheaders.mozdev.org/>

Firebug

web development evolved

inspect and edit HTML

tweak and visualize CSS

debug and profile JavaScript

monitor network activity (caveat)

Firefox extension

free

<http://getfirebug.com/>

Yahoo! - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.yahoo.com/

Make Y! your home page

Yahoo! Voice: Make international calls for as low as 1¢/min.

at&t YAHOO! HIGH SPEED INTERNET

My Yahoo! My Mail

Search: Web Search

Web Images Video Local Shopping more

Y! Answers: Ask a question Answer questions

Page Options

Autos Calendar Finance Games GeoCities Groups HotJobs Maps Movies Music

Featured Entertainment Sports Life

Apr 14, 2007

Disturbing thoughts

Find out what Yahoo! users think of "Disturbia" – a modern twist on the classic "Rear Window." » [Reviews](#)

• Showtimes • Trailer • Photos

• LeBeouf to star in 'Indiana Jones 4'

Users review the new thriller 'Disturbia'

Scientists: Scrap the Internet and start over

Hi, Steven Sign Out

Mail Messenger Radio

Weather 81°F Local Horoscopes

Get the BIGGEST REFUND and do your taxes for FREE. **TurboTax** Free Edition

DON'T BE LATE! GET IT DONE NOW!

Inspect Edit swf1() <div#ad.ad <div.colpadding <div#right <div#rightcx <div#colcx <div#page <body.ywide <html

Console HTML CSS Script DOM Net YSlow Coder

Options

Style Layout DOM Options

```

<div class="colpadding">
  <div id="pa" class="md">
    <script type="text/javascript">
    <div id="ad" class="ad">
      <script language="javascript">
      <script src="http://us.js2.yimg.com/us.yimg.com/a/1-/java/pro...
      <script language="javascript">
      <embed width="350" height="100" wmode="opaque" pluginspage="h...
      <div id="lnkdiv" style="margin-top: 2px;">
      <script language="javascript">
      <noscript>
  
```

body * { www.yahoo.com (line 174)

```

line-height: 1.22em;
}

```

Inherited from div#ad.ad

.ad { www.yahoo.com (line 2125)

```

text-align: center;
}

```

body * { www.yahoo.com (line 174)

```

line-height: 1.22em;
}

```

Inherited from div.colpadding

body * { www.yahoo.com (line 174)

```


```

Done 6.820s YSlow

YSlow

performance lint tool

grades web pages for each rule

Firefox extension

Yahoo! internal tool

Yahoo! - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.yahoo.com/

Make Y! your home page

Yahoo! Voice: Make international calls for as low as 1¢/min.

at&t YAHOO! HIGH SPEED INTERNET

My Yahoo! My Mail

Search: Web Search

Y! Answers: Ask a question | Answer questions

Page Options

Autos Calendar Finance Games GeoCities Groups HotJobs Maps Movies Music

Featured Entertainment Sports Life

Apr 14, 2007

Disturbing thoughts

Find out what Yahoo! users think of "Disturbia" – a modern twist on the classic "Rear Window." » [Reviews](#)

- Showtimes Trailer Photos
- LeBeouf to star in 'Indiana Jones 4'

Users review the new thriller 'Disturbia'

Scientists: Scrap the Internet and start over

Hi, Steven Sign Out

Mail Messenger Radio

Weather 81°F Local Horoscopes

Get the BIGGEST REFUND and do your taxes for FREE. **TurboTax** Free Edition

DON'T BE LATE! GET IT DONE NOW!

Inspect Performance Stats Components Tools

Console HTML CSS Script DOM Net YSlow Coder

Options

Performance Grade: A (94) Expand All Collapse All

B 1. Minimize HTTP requests

This page has 4 external JavaScript files.

A 2. Use edge computing

A 3. Add an Expires header

B 4. Gzip components

These components are not gzipped:

(1.9K) http://l.yimg.com/us.js.yimg.com/lib/bc/bc_2.0.3.js

B 5. Move CSS to the top

Done 5.185s A 179.9K

Conclusion

Takeaways

focus on the front-end

harvest the low-hanging fruit

reduce HTTP requests

enable caching

you do control user response times

LOFNO – be an advocate for your
users

Links

book: <http://www.oreilly.com/catalog/9780596514211/>
examples: <http://stevesouders.com/examples/>
image maps: <http://www.w3.org/TR/html401/struct/objects.html#h-13.6>
CSS sprites: <http://alistapart.com/articles/sprites>
inline images: <http://tools.ietf.org/html/rfc2397>
jsmin: <http://crockford.com/javascript/jsmin>
dojo compressor: <http://dojotoolkit.org/docs/shrinksafe>
HTTP status codes: <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>
IBM Page Detailer: <http://alphaworks.ibm.com/tech/pagedetailer>
Fasterfox: <http://fasterfox.mozdev.org/>
LiveHTTPHeader: <http://livehttpheaders.mozdev.org/>
Firebug: <http://getfirebug.com/>
YUIBlog: <http://yuiblog.com/blog/2006/11/28/performance-research-part-1/>
<http://yuiblog.com/blog/2007/01/04/performance-research-part-2/>
<http://yuiblog.com/blog/2007/03/01/performance-research-part-3/>
<http://yuiblog.com/blog/2007/04/11/performance-research-part-4/>
YDN: http://developer.yahoo.net/blog/archives/2007/03/high_performanc.html
http://developer.yahoo.net/blog/archives/2007/04/rule_1_make_few.html

CC Images Used

“Zipper Pocket” by [jogales](http://www.flickr.com/photos/jogales/11519576/): <http://www.flickr.com/photos/jogales/11519576/>

“Need for Speed” by [Amnemonia](http://www.flickr.com/photos/marinacvinhal/379111290/):
<http://www.flickr.com/photos/marinacvinhal/379111290/>

“I wonder what flavour it is?” by [blather](http://www.flickr.com/photos/deadlyphoto/411770353/):
<http://www.flickr.com/photos/deadlyphoto/411770353/>

“takeout boxes from Grand Shanghai” by [massdistraction](http://www.flickr.com/photos/sharynmorrow/11263821/):
<http://www.flickr.com/photos/sharynmorrow/11263821/>

“takeout” by [dotpolka](http://www.flickr.com/photos/dotpolka/249129144/) : <http://www.flickr.com/photos/dotpolka/249129144/>

“ice cream cone melting/rome” by [Megandavid](http://www.flickr.com/photos/megandavid/189332042/) :
<http://www.flickr.com/photos/megandavid/189332042/>

“nikon em bokeh” by [dsevilla](http://www.flickr.com/photos/dsevilla/249202834/): <http://www.flickr.com/photos/dsevilla/249202834/>

“maybe” by [Tal Bright](http://www.flickr.com/photos/bright/118197469/): <http://www.flickr.com/photos/bright/118197469/>

“how do they do that” by [Fort Photo](http://www.flickr.com/photos/fortphoto/388825145/):
<http://www.flickr.com/photos/fortphoto/388825145/>

“Gorgeous iceberg 7 [Le Toit du Monde]” by [Adventure Addict](http://www.flickr.com/photos/adventureaddict/35290307/)
<http://www.flickr.com/photos/adventureaddict/35290307/>

“molasses-spice cookies” [ilmungo](http://www.flickr.com/photos/ilmungo/65345233/): <http://www.flickr.com/photos/ilmungo/65345233/>

“Driving is fun” by [Ben McLeod](http://www.flickr.com/photos/benmcleod/59948935/): <http://www.flickr.com/photos/benmcleod/59948935/>

“Dozen eggs” by [aeA](http://www.flickr.com/photos/raeallen/96238870/): <http://www.flickr.com/photos/raeallen/96238870/>

“Max speed 15kmh” by [xxxtoff](http://www.flickr.com/photos/xxxtoff/219781763/): <http://www.flickr.com/photos/xxxtoff/219781763/>

“Stairway to heaven” [ognita](http://www.flickr.com/photos/ognita/503915547/): <http://www.flickr.com/photos/ognita/503915547/>

Thanks!

nate@koechley.com

nate.koechley.com/blog

nate.koechley.com/talks/2007/atmedia-london

Thanks again to Steve Souders & Tenni Theurer