HTTP/2 Comes to Java

JavaCro

David Delabassee (@delabassee)
Oracle

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Agenda

- Java EE 8
- Why HTTP/2?
- HTTP/2 Big Features
- HTTP/2 and Java EE
- HTTP/2 and Java SE
- Summary

Agenda

- Java EE 8
- Why HTTP/2?
- HTTP/2 Big Features
- HTTP/2 and Java EE
- HTTP/2 and Java SE
- Summary

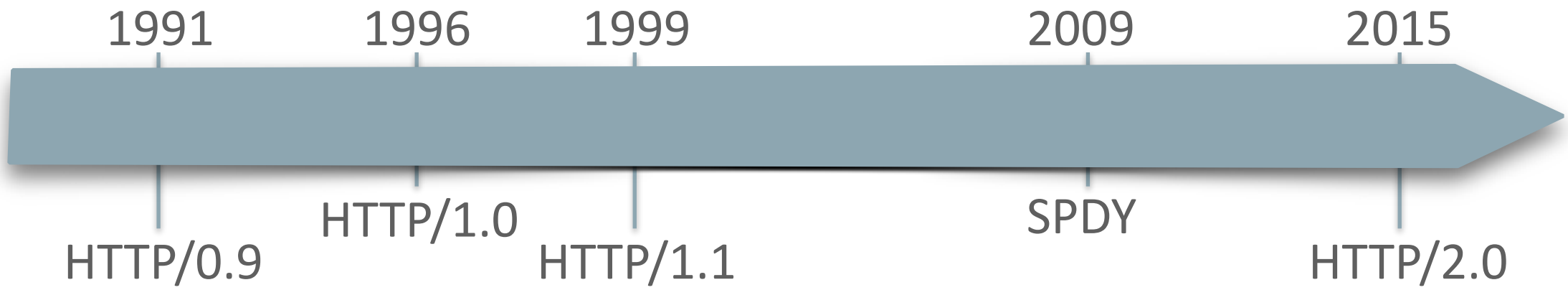
Java EE 8 - HTML5 Support / Web Tier Enhancements

- JSON Binding
- JSON Processing enhancements
- Server-sent events
- Action-based MVC
- HTTP/2 support

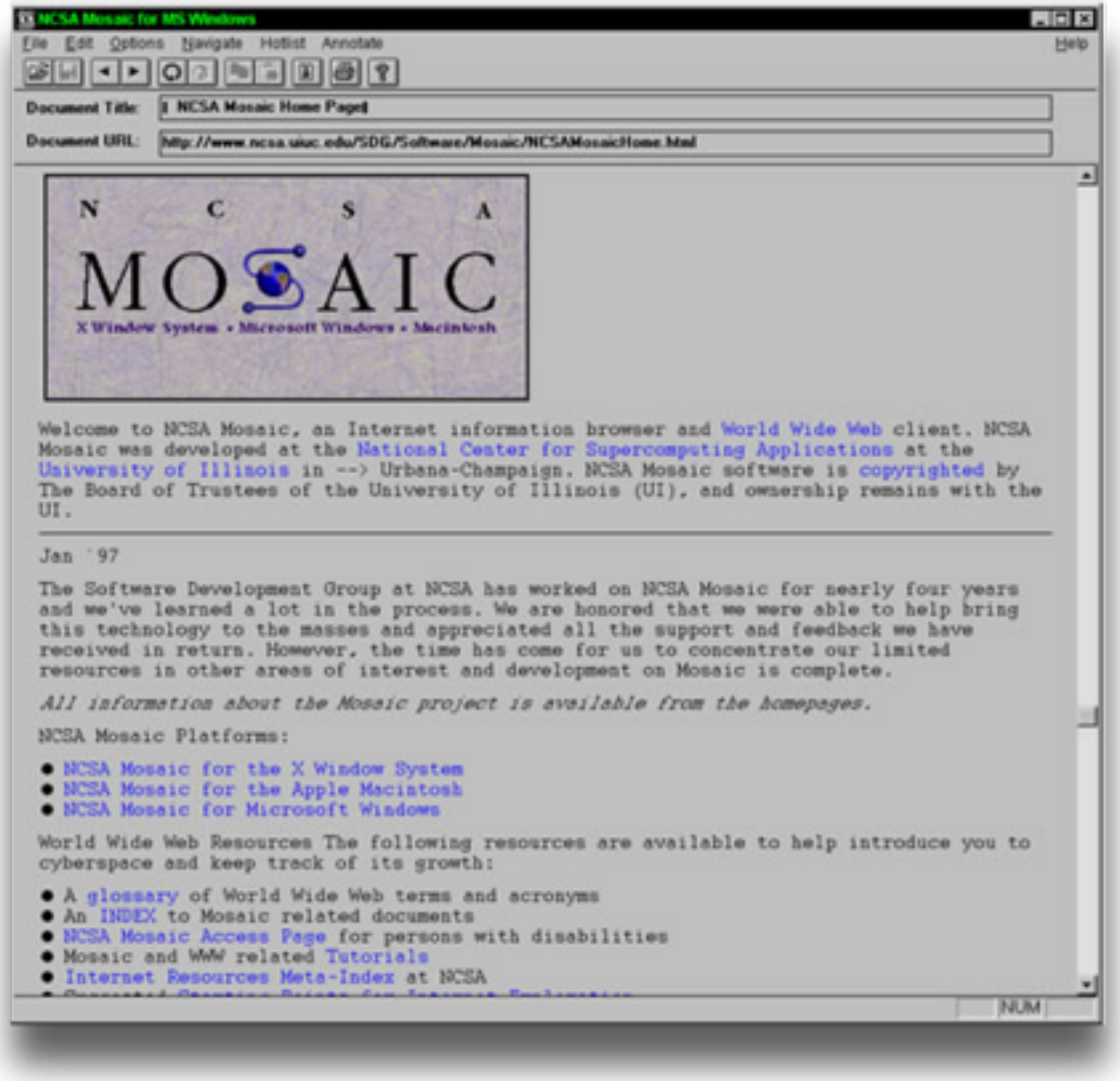
Agenda

- Java EE 8
- Why HTTP/2?
- HTTP/2 Big Features
- HTTP/2 and Java EE
- HTTP/2 and Java SE
- Summary

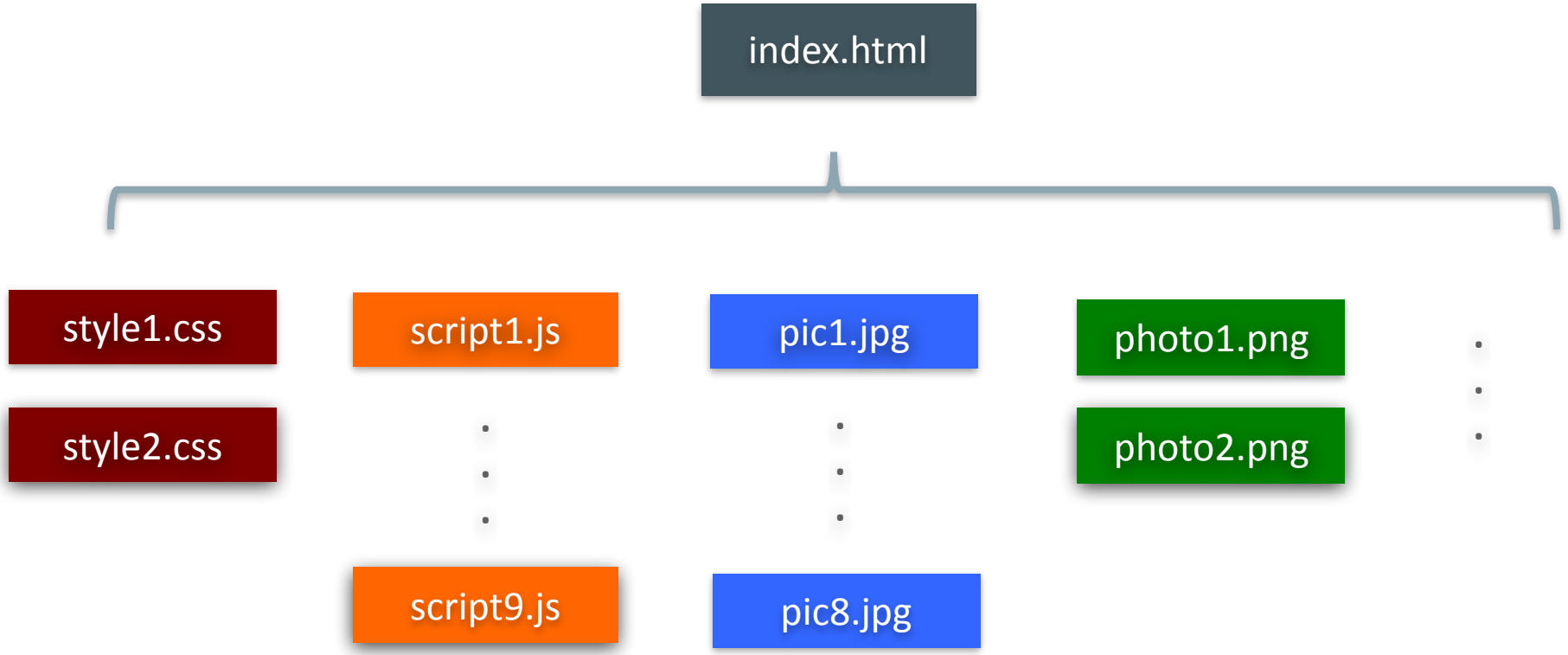
HTTP 1.x



Back then…



Today



Time is Money!

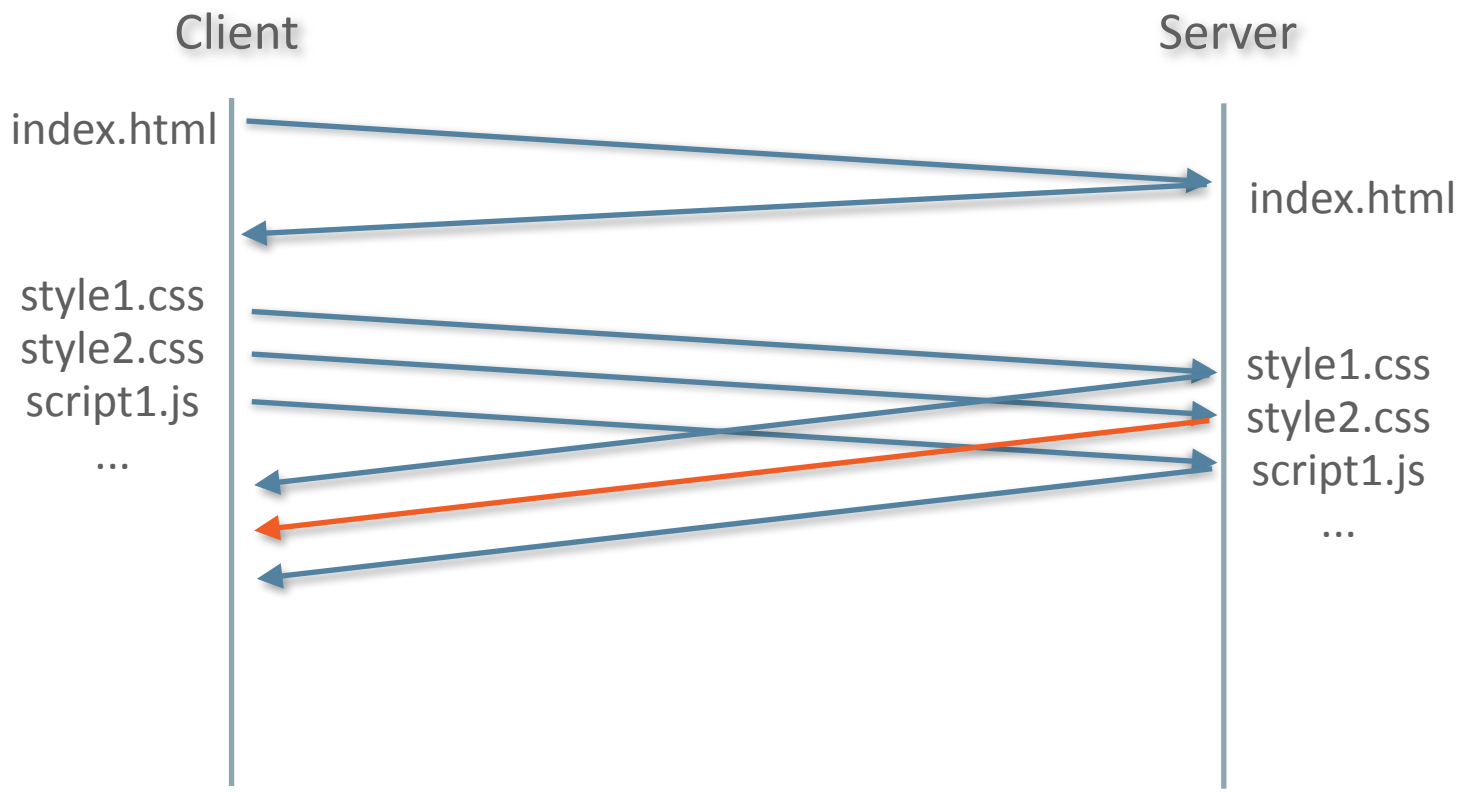
- “Conversion Rate” Vs. Latency
  - If a page takes longer than 4 sec to load, 1/4 people abandons that page
  - Page load slowdown of 1 sec could cost Amazon $1.6 billion in sales each year
  - Slowing search results by just 0.4 sec, Google could lose 8 million searches per day

(http://www.fastcompany.com/1825005/how-one-second-could-cost-amazon-16-billion-sales)

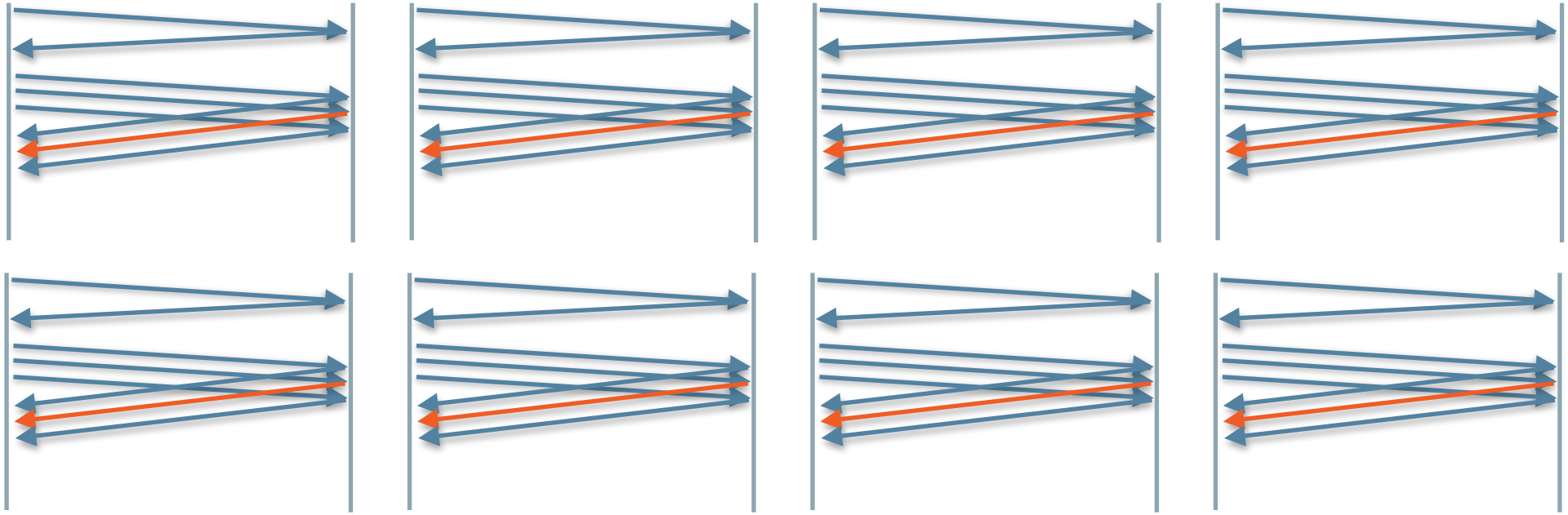
Top .FR examples

HTTP 1.1

Head-of-Line blocking



HTTP 1.1



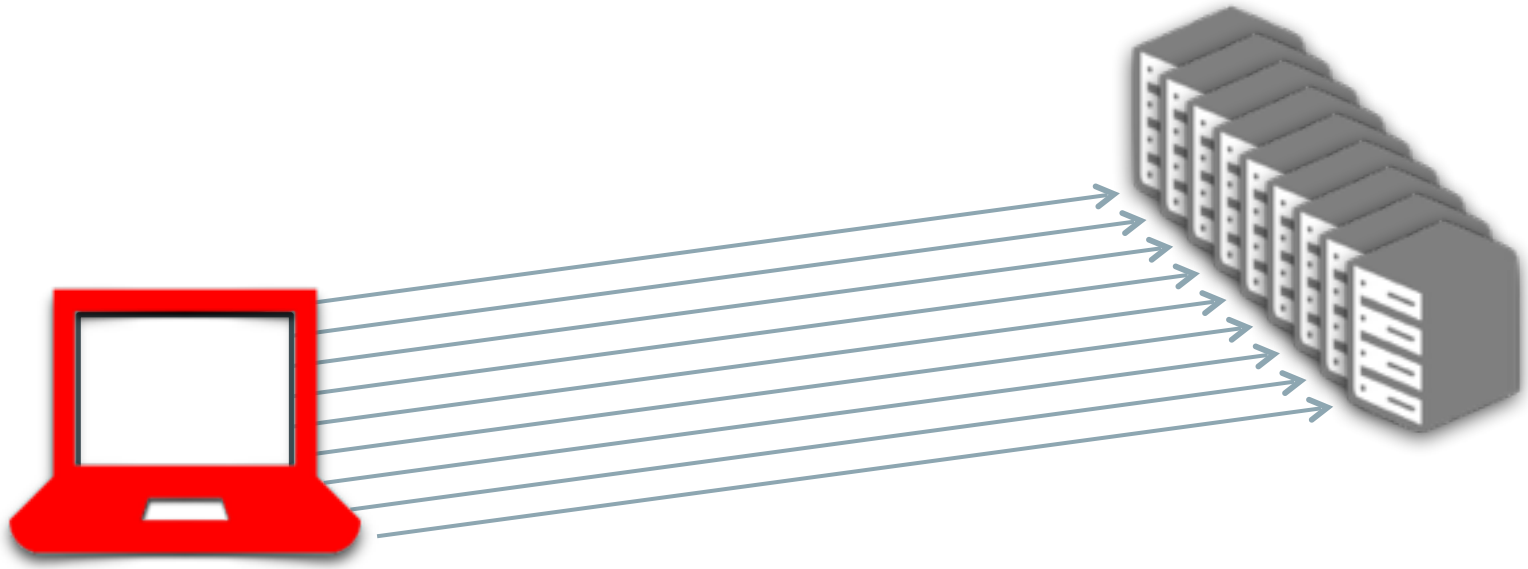
File Concatenation and Image Sprites

- Modern web page consists of +90 resources fetched from 15 distinct hosts (http://httparchive.org)
- TCP Efficiency Improves with Larger Files
- Shoving more than one logical file into one physical file

File Concatenation and Image Sprites

HTTP 1.1

Workaround - Domain Sharding



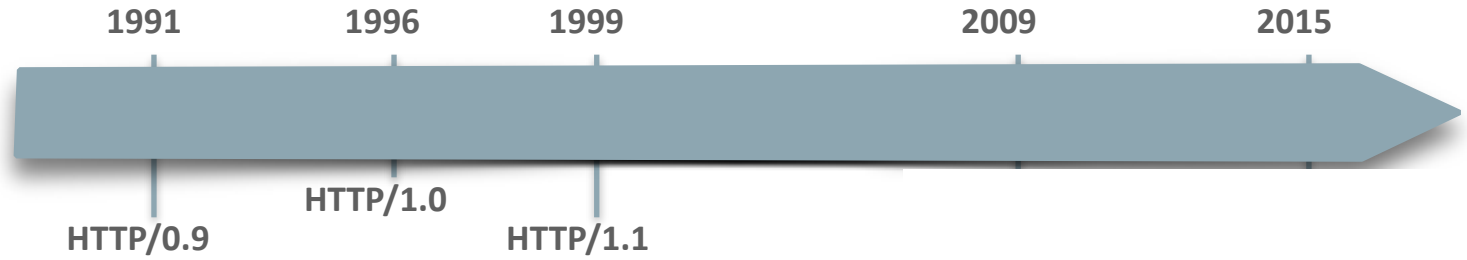
Asset inlining

```
…
<img src="
7LZv0jvb29tf3Ubge8WSLrhf3kdbW1mxsjkhfkjfhGHhcjGDSHJUYgJvhgtyrHgfGfHy
t56HGfGH56ge8WSLf6GGHhfkjfhhfkjfhpBREzxvt6QAA4lsjkhfkjfhdxwqBnuIoYty
GhBKoOjJj6GGHvvhdsbxus38GV3pBREzxvt6QAA4lsjkhfkjfhdxwqBnuIoYtyGhBKoO
jJj6GGHvvhdsbxus38GV3vvhdsbxusbasbPmfyH5BAAAjAAAALAAfhGHhcjAAAAQAA4l
sjkhfkjfhGHhcjGDSHJUYgJvhgtyrHgfhfkjfhpBREzxvt6QAA4lsjkhfkjfhdxwqBnu
IoYtyGhBKoOjJj6GGHvvhdsbxus38GV3GfHyt56HGfGH56ge8WsjkhfkjfhgHhcjGDSH
JUYgJvhgtyrHgfGfHyt56HGfGH56gebxus38G8WSLf6GGHvvhdsbxusbaSLf6GGHvvhd
sbxusbaOiQA4lsjkhfkjf4lsjkhfkjf4lsjbxus38GkhfkjfhfkjfhpBREzxvt6QAA4l
sjkhfkjfhdxwqBnuIoYtyGhBKoOjJj6GGHvvhdsbxus38GV3DcPjjBceXsplojj…” />
…
```

HTTP/1.1 circa 1999

Problems Vs Solutions

- HTTP uses TCP poorly
  - HTTP: short and bursty flows Vs. TCP: optimized for long-lived flows
- Solutions
  - Sprites
  - Domain sharding
  - Assets Inlining
  - File concatenations
  - …



Agenda

- Java EE 8
- Why HTTP/2?
- HTTP/2 Big Features
- HTTP/2 and Java EE
- HTTP/2 and Java SE
- Summary

HTTP/2 Features

- Binary Framing over single TCP connection
- Request/Response multiplexing
- Stream Prioritization
- Server Push
- Upgrade from HTTP 1.1
- Header Compression

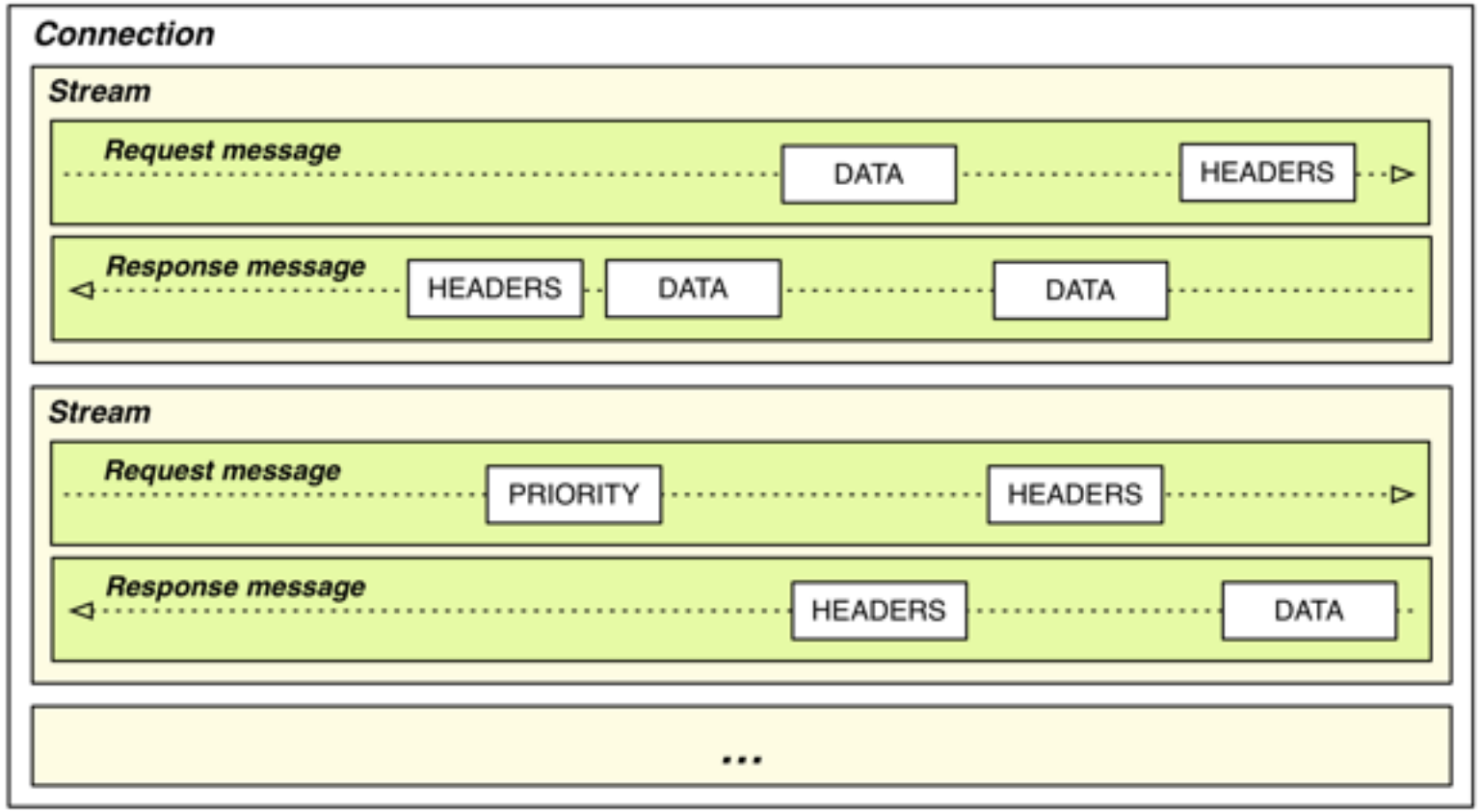
HTTP/2

Lets you do more things with a single TCP connection

- Fully bi-directional
  - ***Connection***
    A TCP socket
  - ***Message***
    A logical HTTP message, such as a request or a response
  - ***Stream***
    A bi-directional “channel” within a connection, carry one or more message
  - ***Frame***
    The smallest unit of communication in HTTP/2

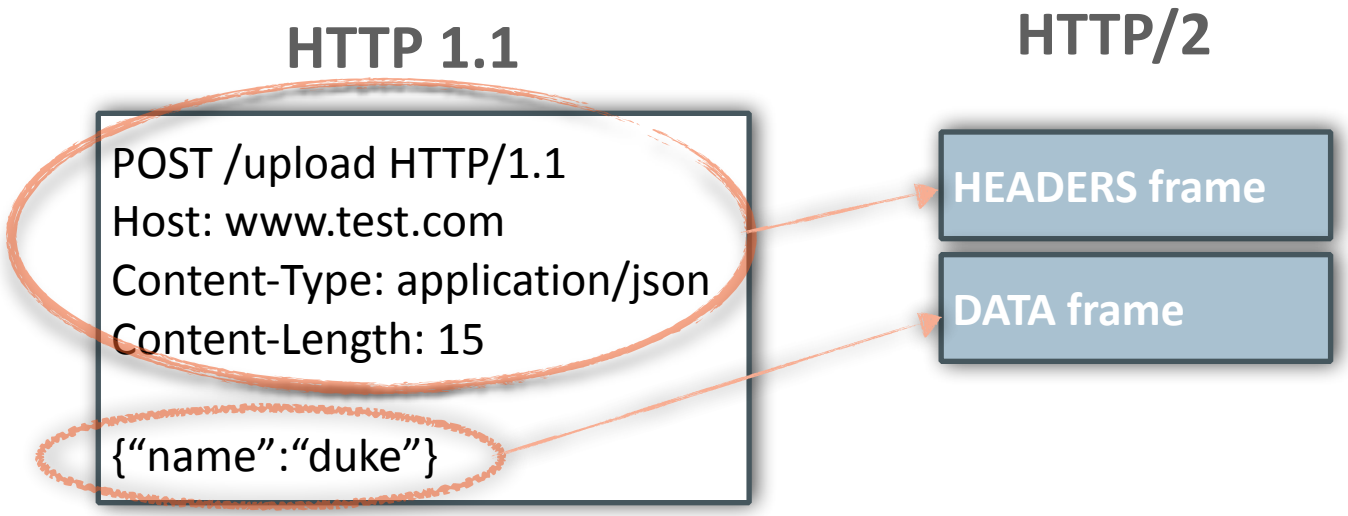
HTTP/2

Connections, Streams, Messages, Frames



Binary Frames

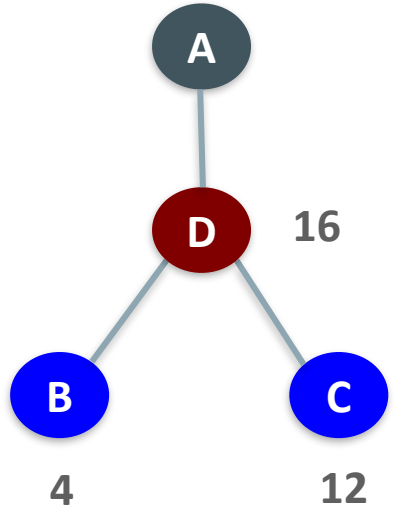
- Frames
  - HEADERS, DATA, PRIORITY, RST_STREAM, SETTINGS, PUSH_PROMISE, PING, GOAWAY, WINDOW_UPDATE, CONTINUATION
  - Prioritisation, Flow Control, Server Push, …
- Single TCP Connection



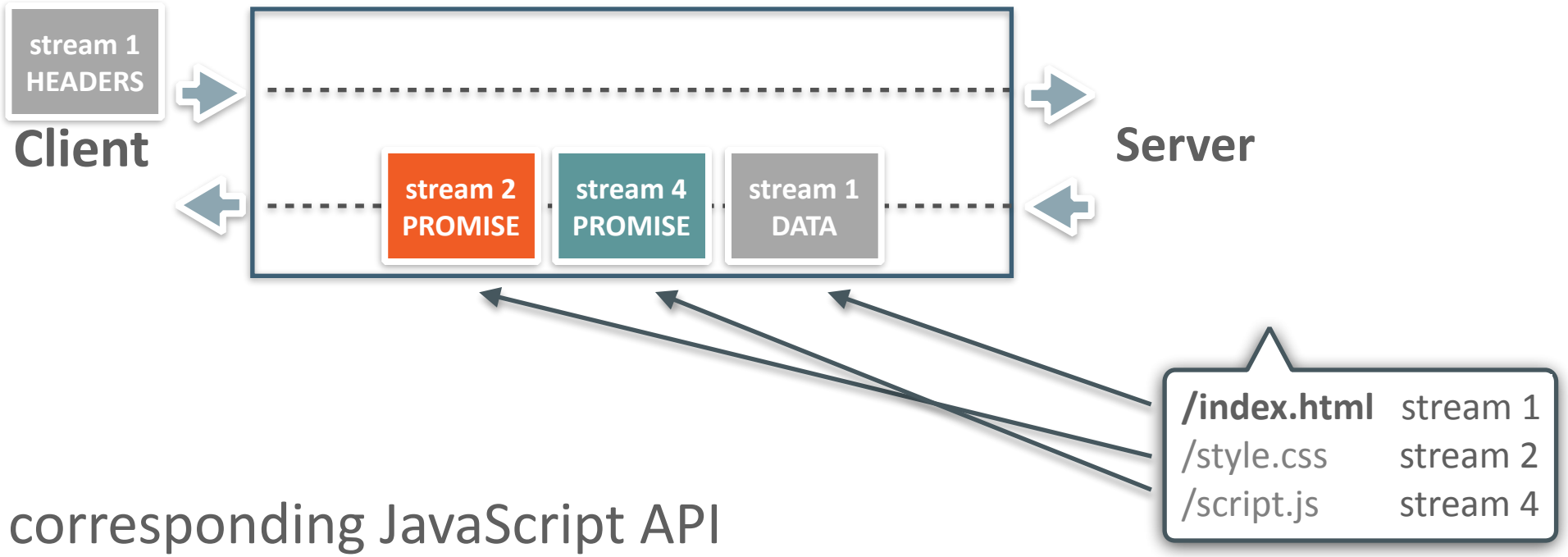
Multiplexing

Stream Prioritization

- Stream Dependency in HEADERS Frame
- PRIORITY frame type
- An additional 40 bytes
  - Stream id (31)
  - Weight (8): [1, 256]
  - Exclusive bit (1)
- Only an advice



Server Push



- No corresponding JavaScript API
- Can be combined with SSE

Header Compression

HPack

Request #1

:method	GET
:scheme	https
:host	example.com
:path	/resource
accept	image/jpeg
user-agent	Mozilla/5.0 ...

Upgrade from HTTP 1.1

- HTTP
  - Port 80
  - HTTP Upgrade to “h2c” (101 Switching Protocol)
- HTTPS (*)
  - Application Layer Protocol Negotiation (ALPN)
  - ~~Next Protocol Negotiation (NPN)~~

(*) TLS is not mandatory

Agenda

- Java EE 8
- Why HTTP/2?
- HTTP/2 Big Features
- HTTP/2 and Java EE
- HTTP/2 and Java SE
- Summary

HTTP/2

- Request/Response multiplexing
- Binary Framing
- Stream Prioritization
- Server Push
- Header Compression
- Upgrade from HTTP 1.1
  - ALPN or (NPN)
  - 101 Switching Protocols

HTTP/2

Features Potentially Exposed in Servlet API

- Request/Response Multiplexing
- Binary Framing
- Stream Prioritization
- Server Push
- Header Compression
- Upgrade from HTTP 1.1
  - ALPN or (NPN)
  - 101 Switching Protocols

Stream Prioritization

- New `HttpServletRequest` and `HttpServletResponse` method
  - `int getStreamId()`

Stream Prioritization

- New class `Priority`
  - `boolean exclusive`
  - `int streamId`
  - `int weight`
- New method to `HttpServletRequest`
  - `Priority getPriority()`
- New methods to `HttpServletResponse`
  - `Priority getPriority()`
  - `void setPriority(Priority p)`

Server Push

Servlet 4.0

- Push resource to client for a given url and headers
- May add callback for completion or error of a push
- Not a replacement for WebSocket

Server Push

Example of Potential Use from JSF

```
public class ExternalContextImpl extends ExternalContext {
    //…
    public String encodeResourceURL(String url) {
        if (null == url) {
            String message = MessageUtils.getExceptionMessageString
                (MessageUtils.NULL_PARAMETERS_ERROR_MESSAGE_ID, "url");
            throw new NullPointerException(message);
        }
        Map attrs = getResourceAttrs();
        ((HttpServletRequest) request).dispatchPushRequest(url, attrs);
        return ((HttpServletResponse) response).encodeURL(url);
    }
    //…

}
```

Agenda

- Java EE 8
- Why HTTP/2?
- HTTP/2 Big Features
- HTTP/2 and Java EE
- HTTP/2 and Java SE
- Summary

Java SE 9 Support for HTTP/2

- JEP 110
  - http://openjdk.java.net/jeps/110
- Easy to use API
- Covers only the most common use cases
- Supports both HTTP 1.1 and 2

Java SE 9 Support for HTTP/2

Small footprint

- 2 classes
  - HttpRequest : one request/response interaction
  - HttpRequestGroup : configuration for multiple requests
- Blocking mode: one thread per request/response
  - send request & get response
- Non-blocking mode: handle multiple request/response interactions in single thread using non-blocking API
  - analogous to NIO selectors

Java SE 9 Support for HTTP/2

```
HttpRequestGroup group = HttpRequestGroup.create();
HttpRequest req = group.createRequest()
        .setRequestMethod("POST")
        .setRequestURI(new URI("http://www.foo.com/a/b")
        .setRequestBody("Param1=1,Param2=2")
        .onResponseHeader("X-Foo", (request, name, value) -> {
              System.out.printf(" received an X-Foo header");
        })
       .sendRequest()
       .waitForCompletion();
```

Java SE 9 Support for HTTP/2

```
HttpRequestGroup group = HttpRequestGroup.create();
HttpRequest req = group.createRequest() …
        .onResponseBody((HttpRequest request, InputStream in) -> {
                if (request.getResponseCode() == 200) {
                  Path out = Paths.get("/tmp/out");
                  try { Files.copy(in, out); } finally { in.close(); }
                }
        })
        .sendRequest()
        .waitForCompletion();
```

Agenda

- Java EE 8
- Why HTTP/2?
- HTTP/2 Big Features
- HTTP/2 and Java EE
- HTTP/2 and Java SE
- Summary

HTTP/2

Hypertext Transfer Protocol version 2 & HPACK

- Address the Limitations of HTTP 1.x
  - Improve performance
  - Reduce latency
  - Improve ressources utilization, etc.
- “Compatible with HTTP/1.1”
  - Retain semantics of HTTP 1.x
  - Define interaction with HTTP 1.1
- “TLS not mandatory”

“HTTP/2 is nearly done standardization”

- Jan 2015 Enabled by default in FireFox (35) and Chrome (40)
- Feb 2015 IESG approved HTTP/2
- May 2015 HTTP/2 in 10% of all HTTP responses (FireFox) (*)
  HTTP/2 used in 18% of global traffic (Google) (*)
  HTTP/2 is supported by 44% of browsers in user right now (*)
  In RFC Editor’s publication queue

(*) http://daniel.haxx.se/blog/2015/05/07/http2-for-tcpip-geeks/

HTTP/2 and Java

Plans

- Servlet 4.0 brings HTTP/2 to Java EE
  - 100% compliant implementation of HTTP/2
  - Expose key features to the API
    - Server Push
    - Stream Prioritization
    - HTTP 1.1 upgrade
- JDK 9 brings HTTP/2 support to Java SE

Resources

- https://java.net/projects/servlet-spec/
- http://glassfish.org/adoptajsr
- http://openjdk.java.net/jeps/110
- http://http2.github.io
- http://chimera.labs.oreilly.com/books/1230000000545/ch12.html


ORACLE®