

O'REILLY

Velocity

Web Performance and
Operations Conference



Front-end Performance Improvements at YouTube

Progressive Enhancement and Beyond

Alex Nicksay

Senior Web Developer, YouTube





2 billion views every day

35 hours of uploads every minute

A Performance Goal

**Start Playing Video
As Soon As Possible**

Performance Optimization #1

**JavaScript at the Bottom
and
Embedding Flash**

Embedding Flash (Before)

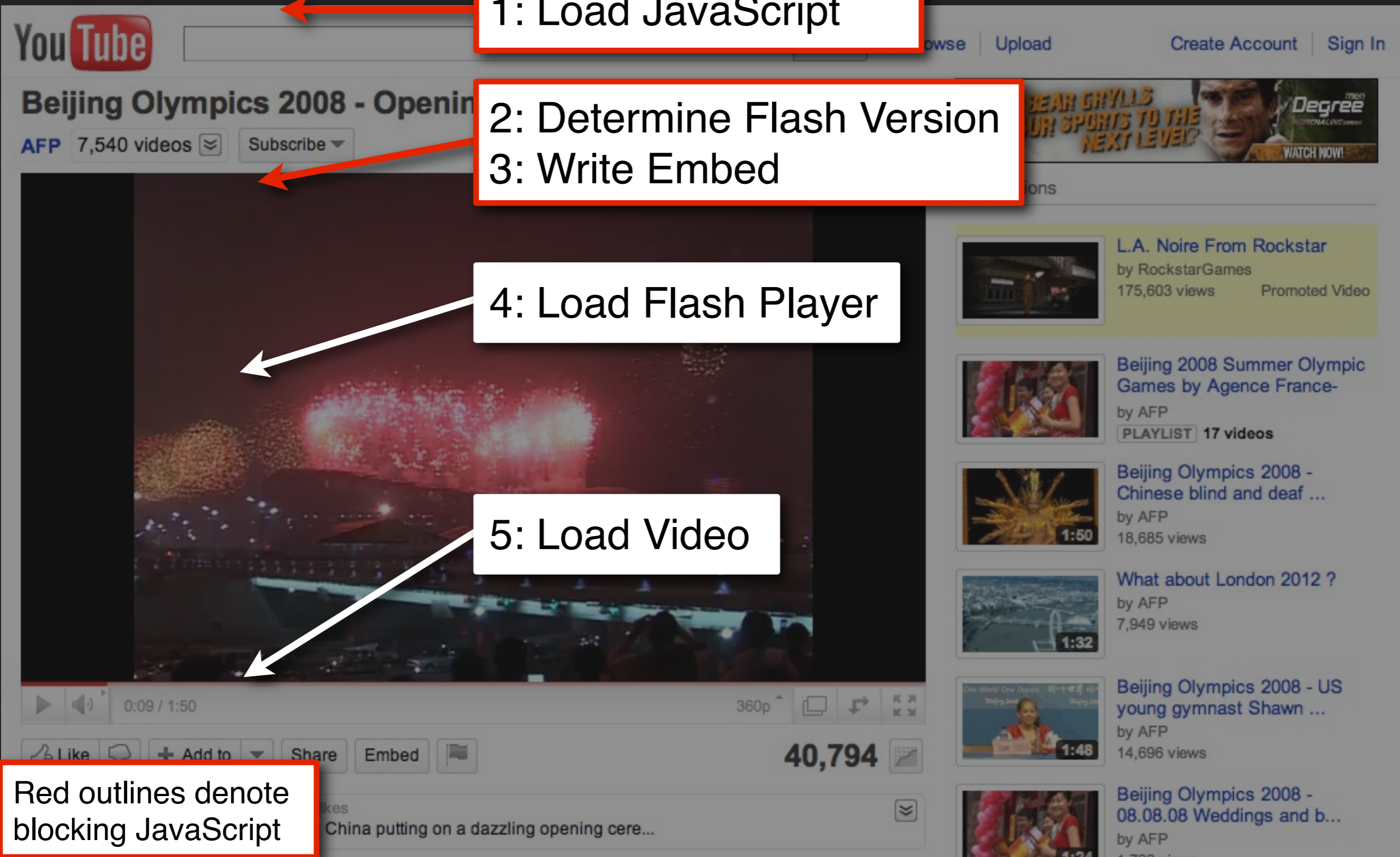
1: Load JavaScript

2: Determine Flash Version
3: Write Embed

4: Load Flash Player

5: Load Video

Red outlines denote
blocking JavaScript



Embedding Flash

- Realization:
 - Most visitors fall into one of two categories:
 - They have a recent version of Flash installed
 - They do not have Flash installed
- Optimization:
 - Optimistically embed the most common case without JS and do version checking and player updating at the bottom of the page when scripts load

Embedding Flash (After)

The image shows a YouTube video player for the video "Beijing Olympics 208 - Opening ceremony". The video is currently playing, showing a night scene with fireworks. The player interface includes a progress bar at 0:09 / 1:50, a volume icon, and a 360p resolution indicator. Below the video are buttons for Like, Add to, Share, and Embed. The page also features a search bar, navigation links (Browse, Upload), and user options (Create Account, Sign In). A sidebar on the right shows video suggestions.

Annotations on the image indicate the loading sequence:

- 1: Load Flash Player**: A white arrow points to the top-left corner of the video player area.
- 2: Load Video**: A white arrow points to the bottom-left corner of the video player area.
- 3: Load JavaScript**: A red arrow points to the bottom-right corner of the video player area. The text "3: Load JavaScript" is enclosed in a red outline.
- 4: Determine Flash Version**: A white arrow points to the bottom-right corner of the video player area.
- 5: Update Embed (if needed)**: A white arrow points to the bottom-right corner of the video player area.

A red outline around the "3: Load JavaScript" text and a red arrow pointing to the bottom-right corner of the video player area indicate that these steps are blocking JavaScript.

Red outlines denote blocking JavaScript

Impact on Performance



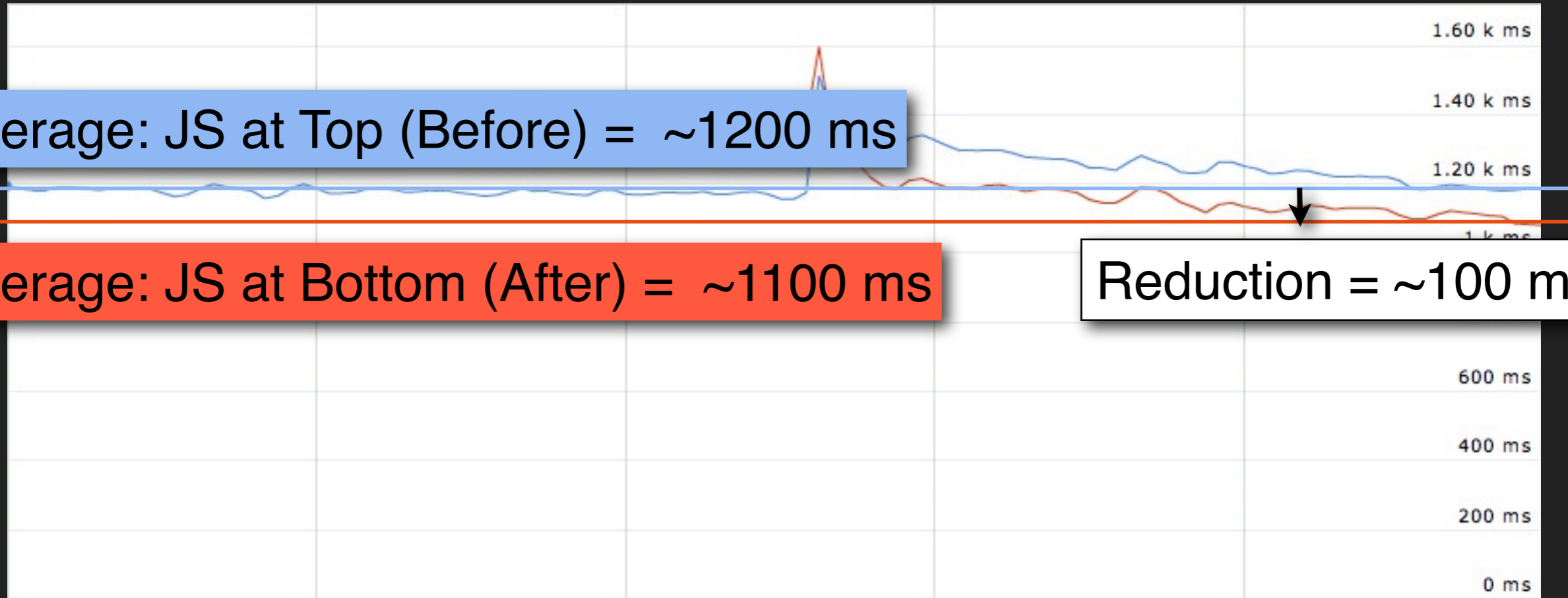
Time to Parse Document Head

Impact on Performance

Average: JS at Top (Before) = ~1200 ms

Average: JS at Bottom (After) = ~1100 ms

Reduction = ~100 ms



Time until Flash Starts

Performance Optimization #2

Preloading the Video Connection

Preloading the Video Connection

- Why do it?
 - Making new connections can be expensive
 - Video download can begin sooner by preloading the connection

Preloading the Video Connection

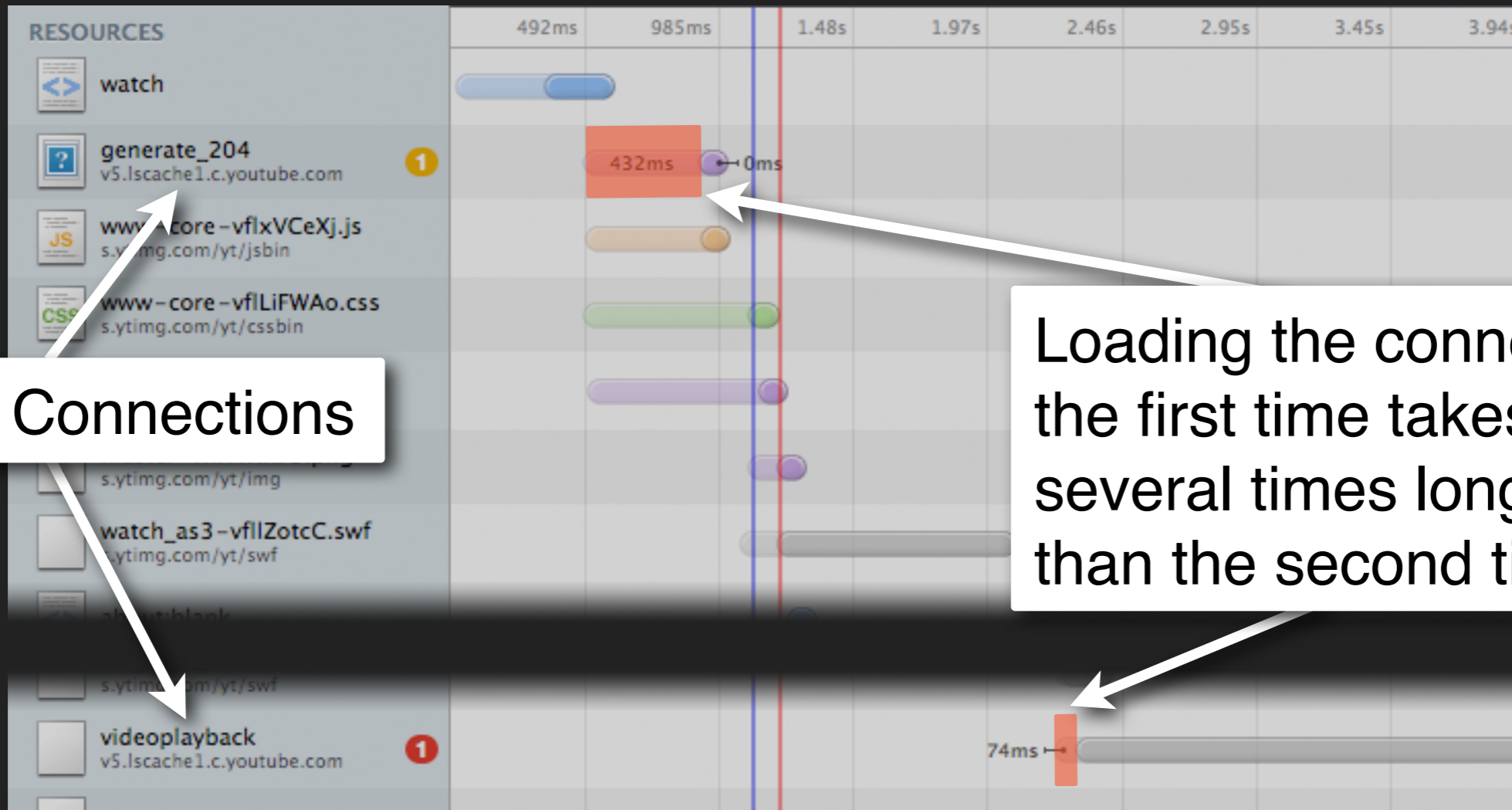
- How do you do it?

```
<head>  
  <script>  
    var img = new Image();  
    img.src = videoConnectionUrl;  
  </script>  
</head>
```

Preloading the Video Connection

- What does it do?
 - Resolves DNS while page is rendering, before it is needed
 - Maintains an open connection for later use

Impact on Page Load



Video Connections

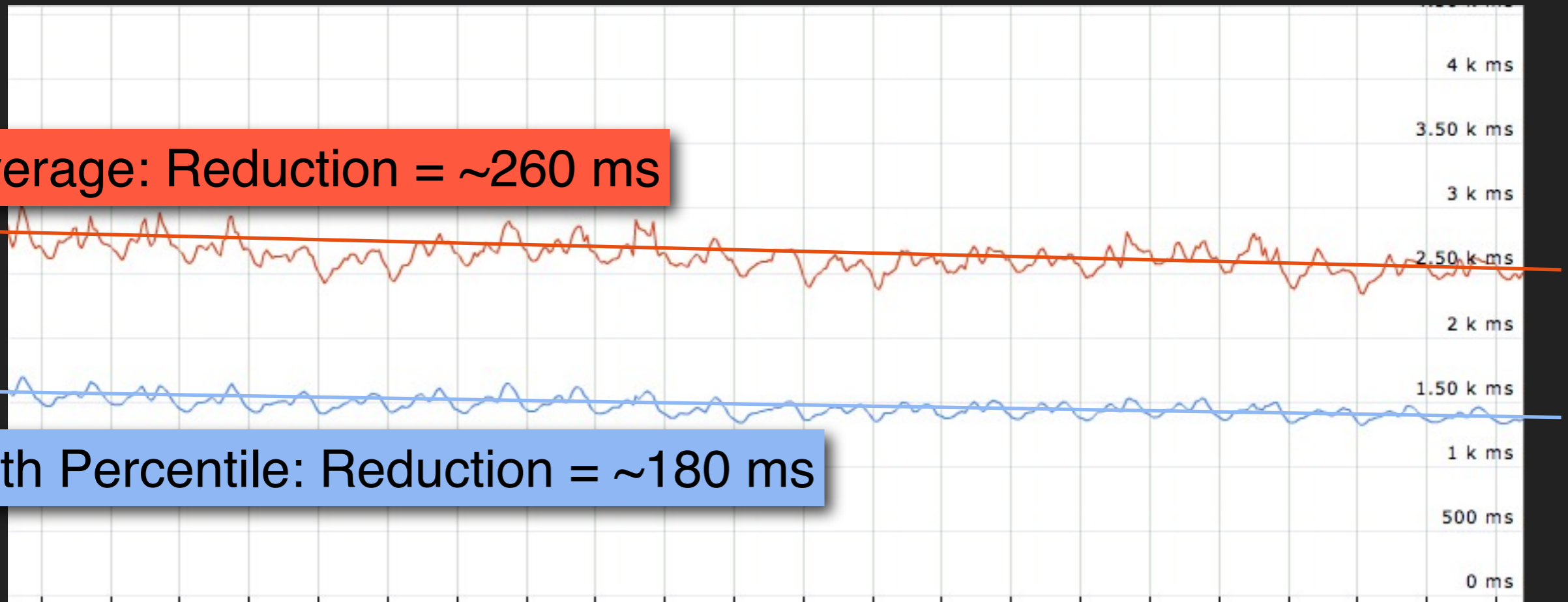
Loading the connection the first time takes several times longer than the second time

Timeline of Resources Loaded

Impact on Performance

Average: Reduction = ~260 ms

50th Percentile: Reduction = ~180 ms

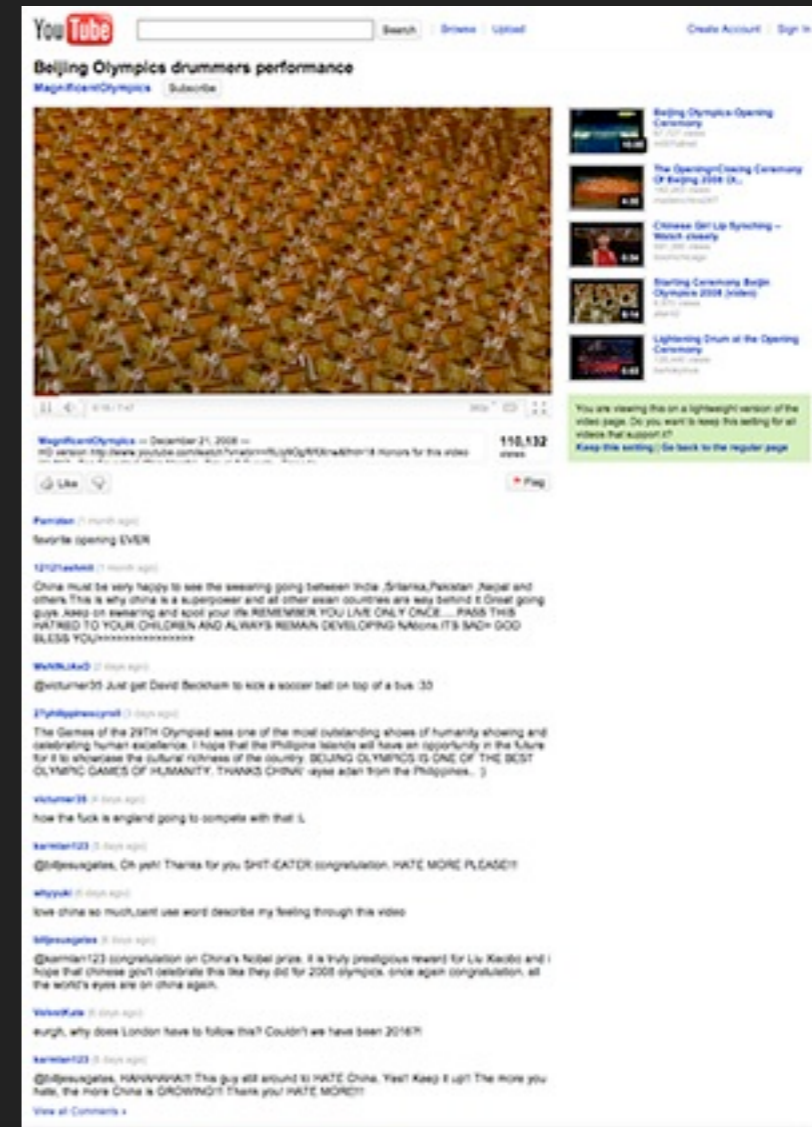
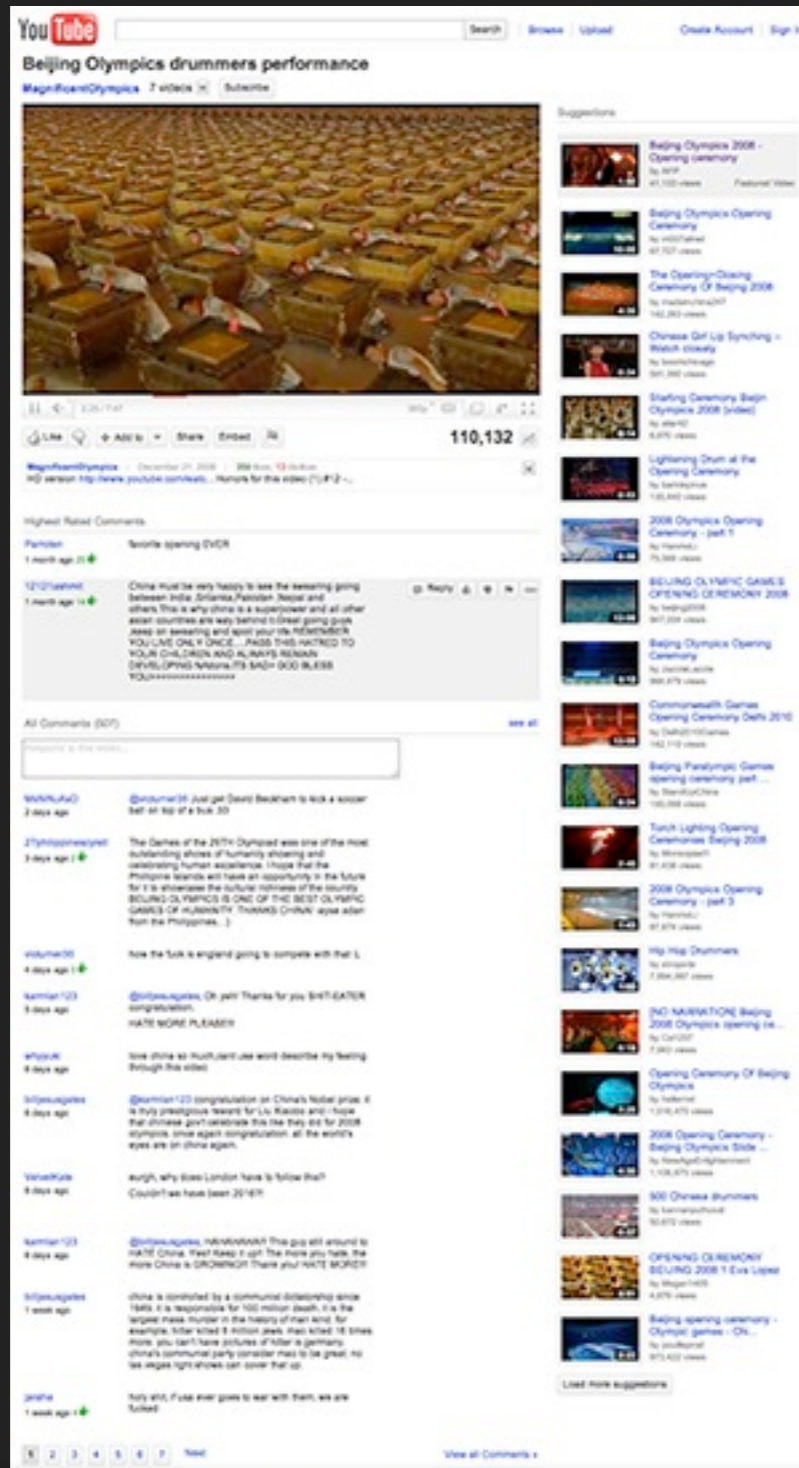


Time until Video Download Begins

Performance Optimization #3

Feather: Lightweight Version

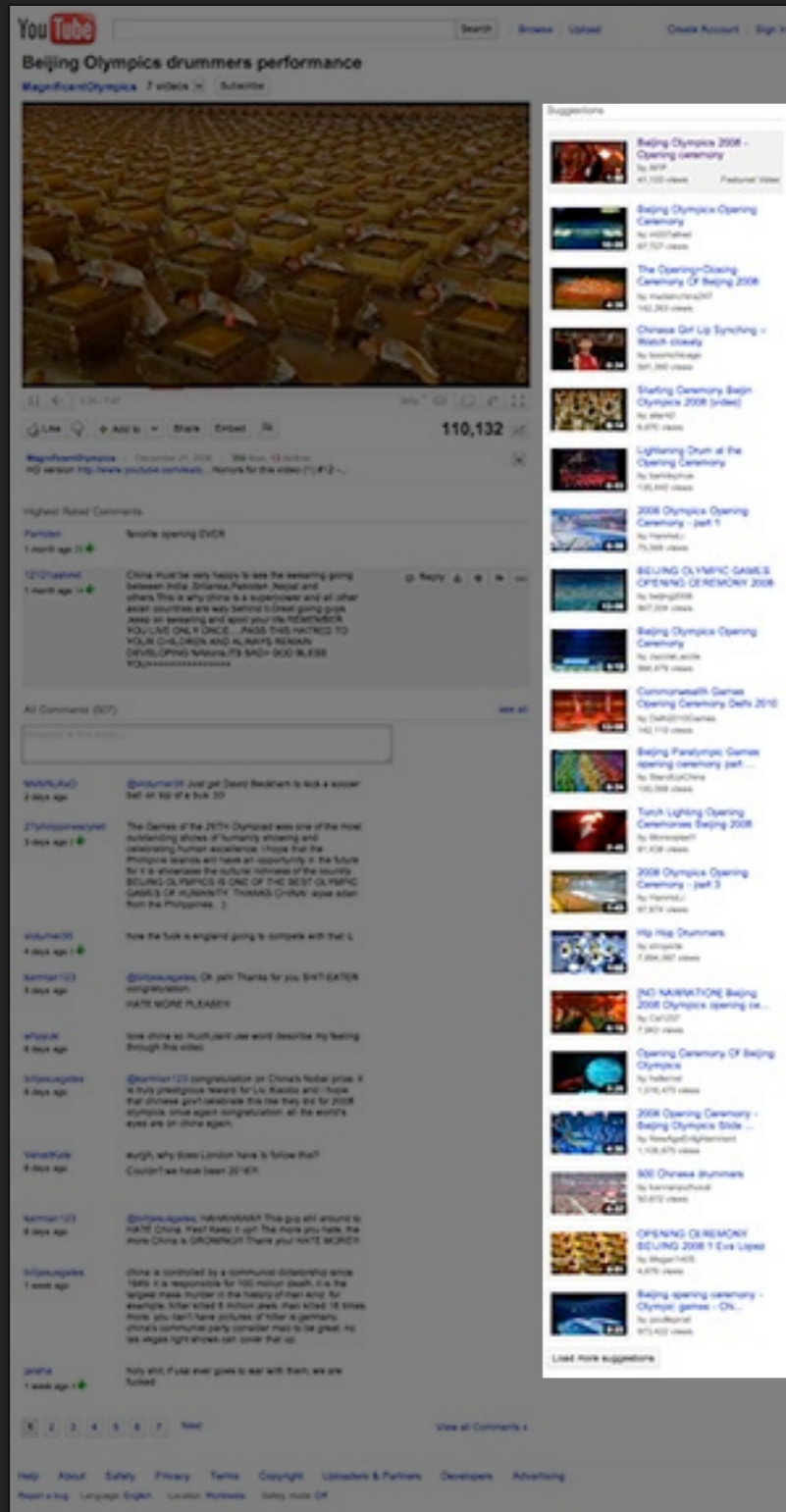
Feather: Lightweight Version



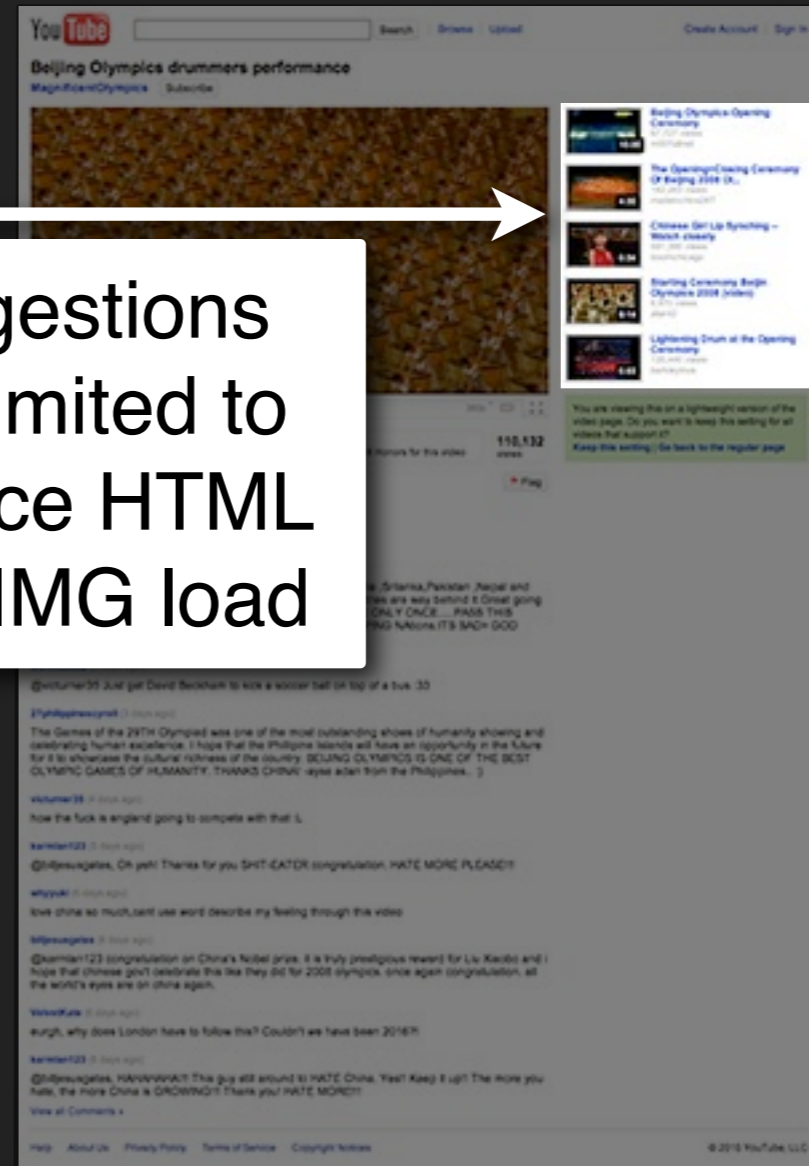
Feather (Lightweight Version)

Watch (Standard Version)

Feather: Lightweight Version

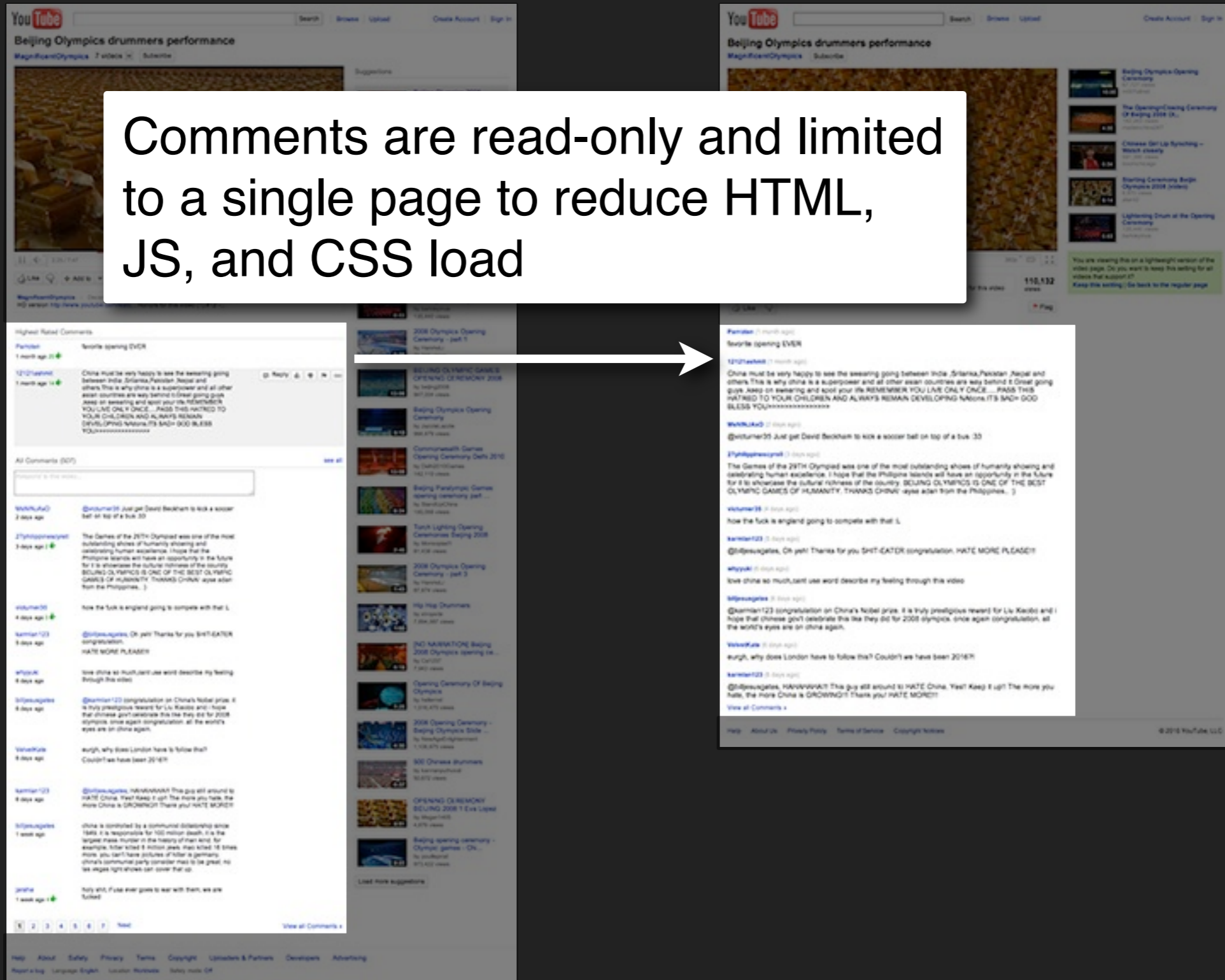


Suggestions
are limited to
reduce HTML
and IMG load

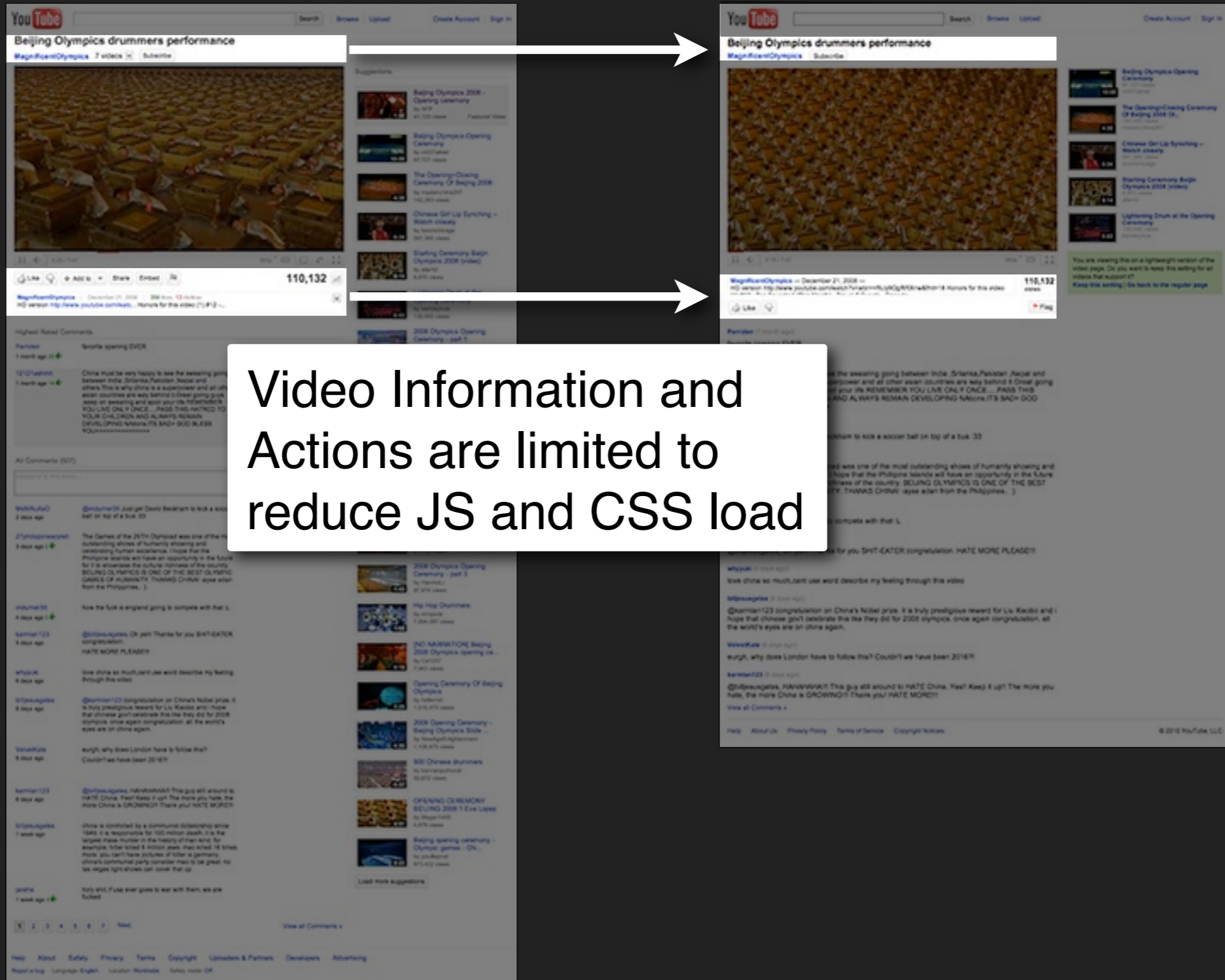


Feather: Lightweight Version

Comments are read-only and limited to a single page to reduce HTML, JS, and CSS load



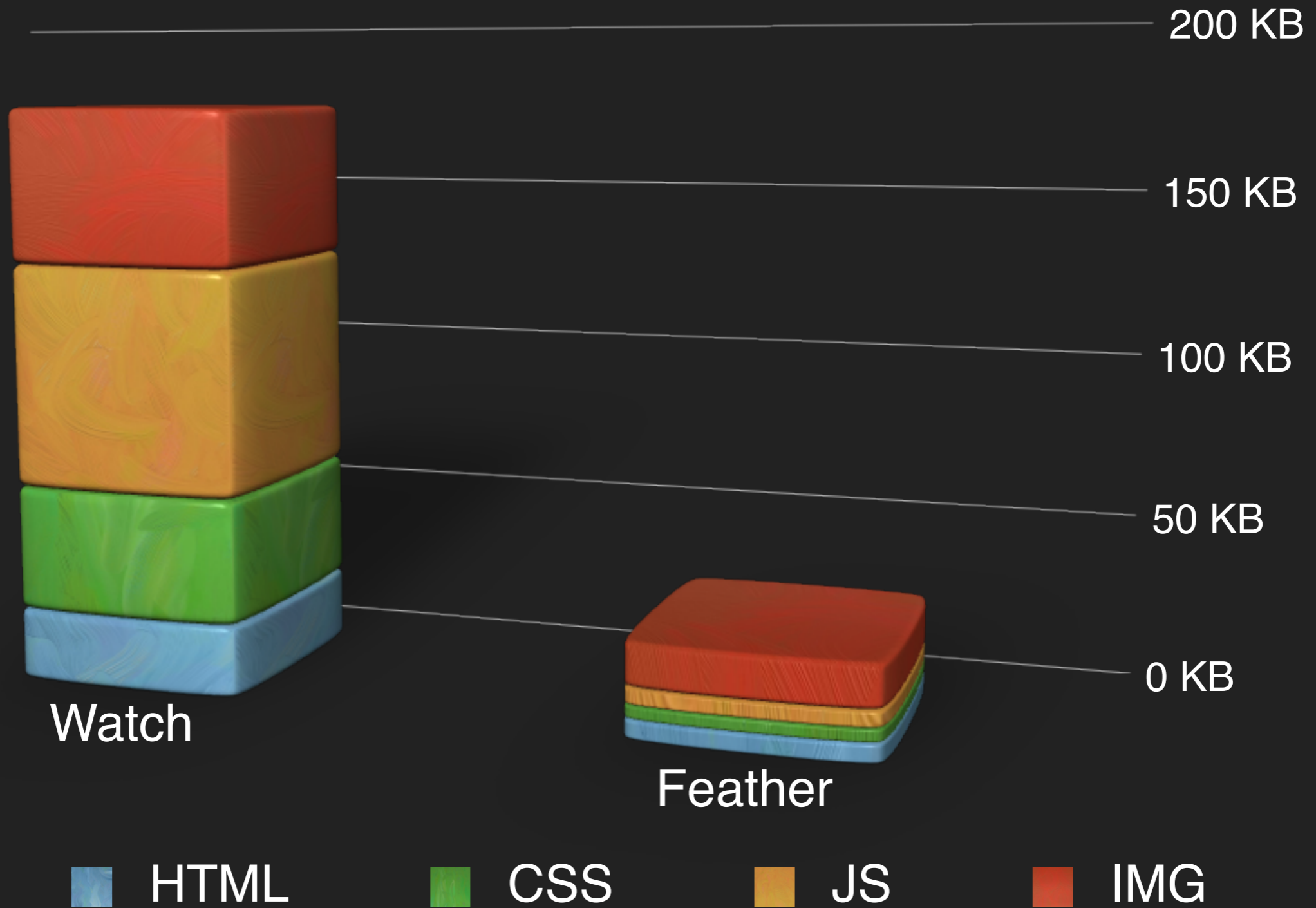
Feather: Lightweight Version



Video Information and
Actions are limited to
reduce JS and CSS load

You are viewing this on a lightweight version of the video page. Do you want to keep this setting for all videos that support it? Keep this setting - Go back to the regular page

Feather: Lightweight Version



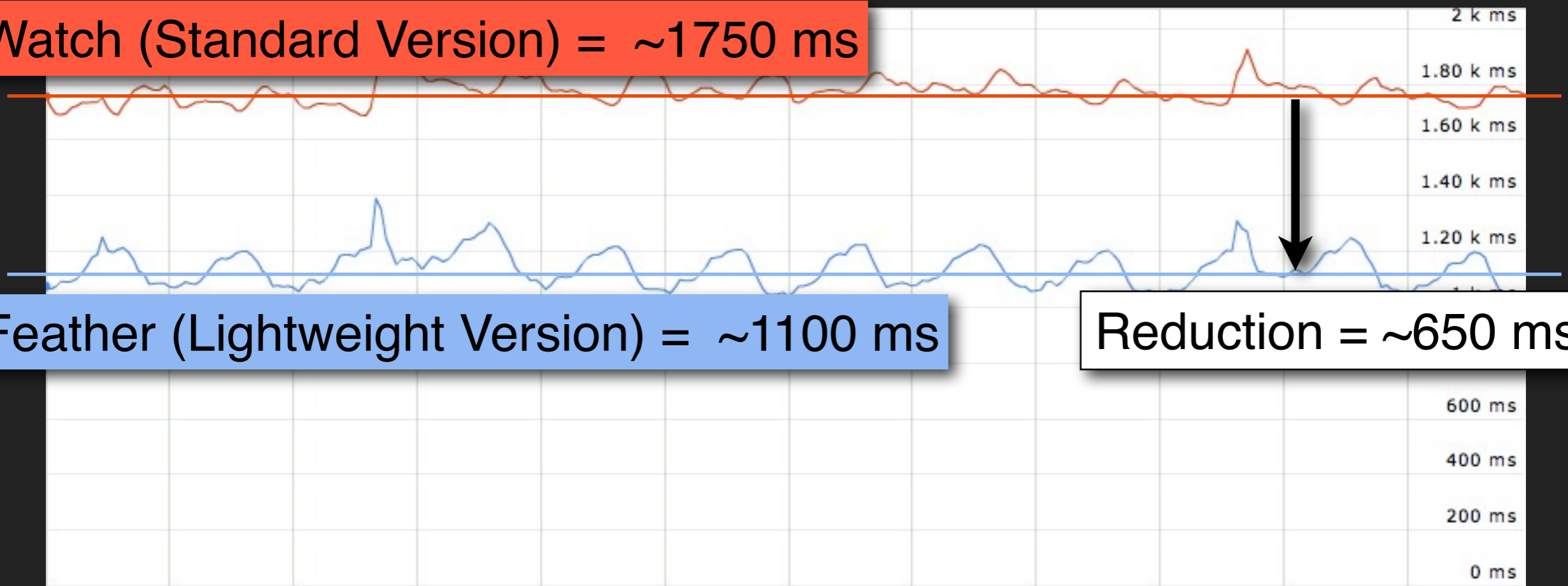
Sizes of Resources Loaded

Impact on Performance

Watch (Standard Version) = ~1750 ms

Feather (Lightweight Version) = ~1100 ms

Reduction = ~650 ms



Time to Load Page

Progressive Enhancement

UIX Widget System

UIX Widget System

- Delay-loading non-essential content and resources increases performance
- Traditionally, interactive widgets are rendered by JS, requiring scripts to be loaded before page rendering
- Traditionally, each new piece of dynamically loaded content needs to have JS initialization
- What do we need?
 - Lightweight framework for fast, easy, dynamic loading of new content (HTML) and new widgets (CSS/JS)

UIX Widget System

- What is it?
 - A centrally-managed, delegated-behavior widget system
 - Separates content (HTML) from interaction (JS)
- What does that mean?
 - JS can be delay-loaded after the page is rendered
 - New widgets can be registered at any time
 - New HTML can be dynamically updated at any time
 - Event handling is automatic
 - Widgets in new content work immediately

Demo

UIX Widget System

UIX Widget System: Architecture

Event
Handling

UIX
Behaviors

Class
Matching

UIX
Widget

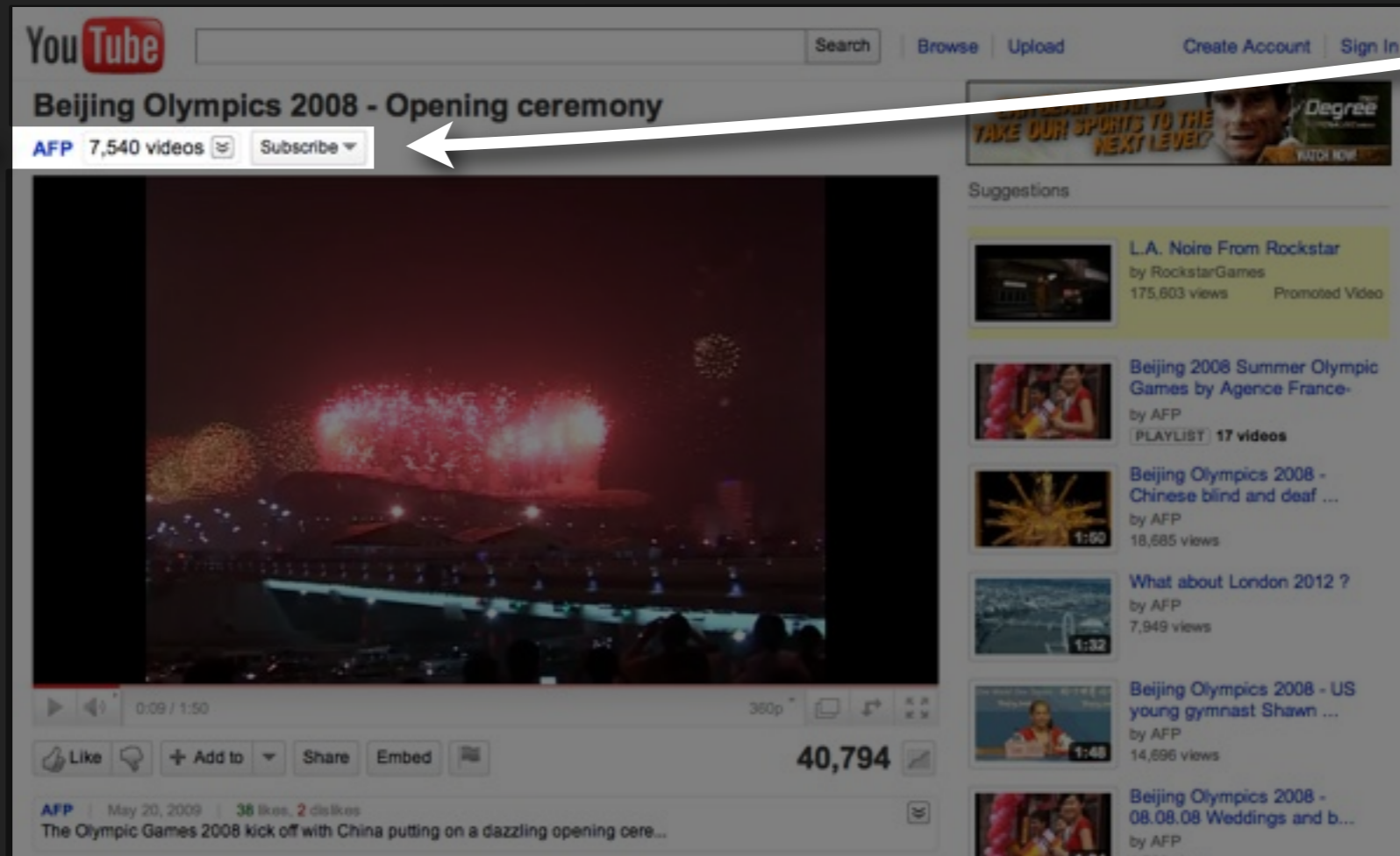
UIX
Widget

UIX
Widget

UIX
Widget

Event
Bubbling

Action
Execution



UIX Widget System: Behaviors

Event
Handling

UIX
Behaviors

Class
Matching

UIX
Widget

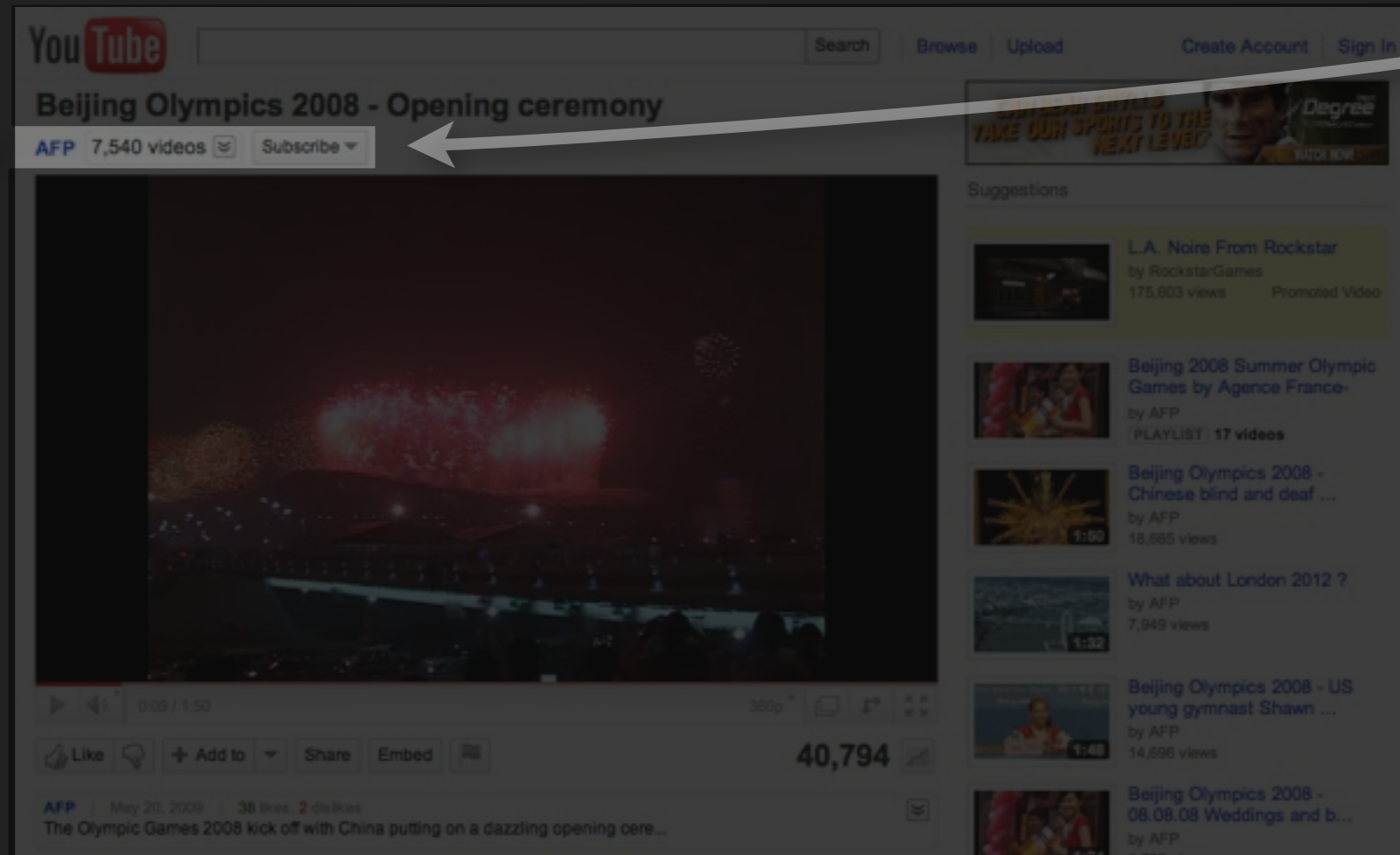
UIX
Widget

UIX
Widget

UIX
Widget

Event
Bubbling

Action
Execution



UIX Widget System: Behaviors

- A registry stores “behaviors”
- A behavior is an action that is executed any time an event happens on a type of element
- A behavior has three components:
 - What: a JS function to execute
 - Where: a CSS class to match
 - When: a JS event to handle



UIX Widget System: Behaviors

```
// Sample registry
```

```
uix.registry = {  
  'click': {  
    'widget1-css': widget1.onClick,  
    'widget2-css': widget2.onClick  
  },  
  'mouseover': {  
    'widget2-css': widget2.onmouseover  
  },  
  'mouseout': {  
    'widget2-css': widget2.onmouseout  
  }  
};
```



UIX
Behaviors

UIX Widget System: Behaviors

Event
Handling

UIX
Behaviors

Class
Matching

UIX
Widget

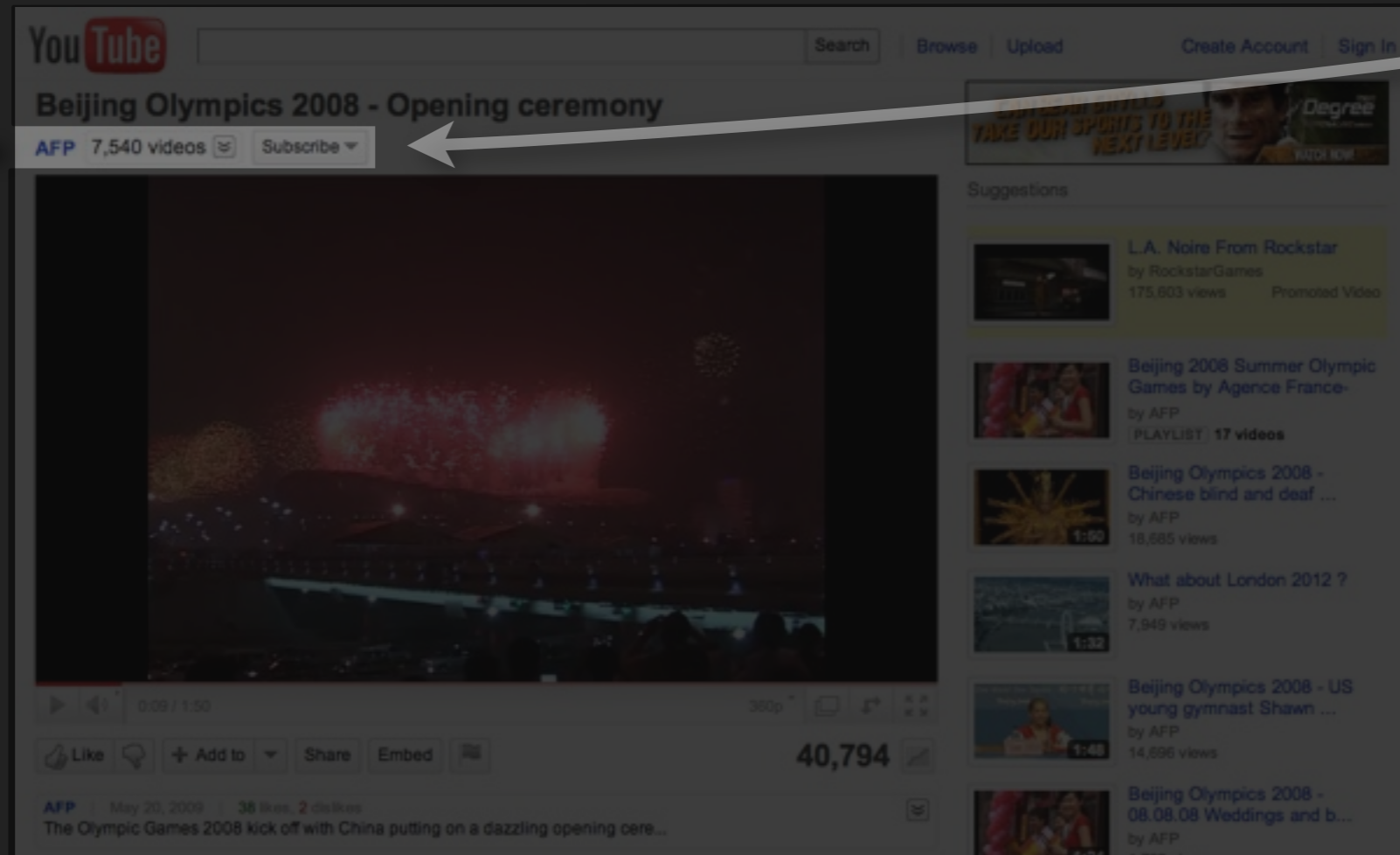
UIX
Widget

UIX
Widget

UIX
Widget

Event
Bubbling

Action
Execution



UIX Widget System: Behaviors



- Events bubble to the top of the document
- A single event handler manages multiple browser events

UIX Widget System: Behaviors

```
// Sample event listeners
```

```
document.addEventListener(  
    'click', uix.handleEvent);  
document.addEventListener(  
    'mouseover', uix.handleEvent);  
document.addEventListener(  
    'mouseout', uix.handleEvent);
```



UIX
Behaviors

UIX Widget System: Behaviors

Event
Handling

UIX
Behaviors

Class
Matching

UIX
Widget

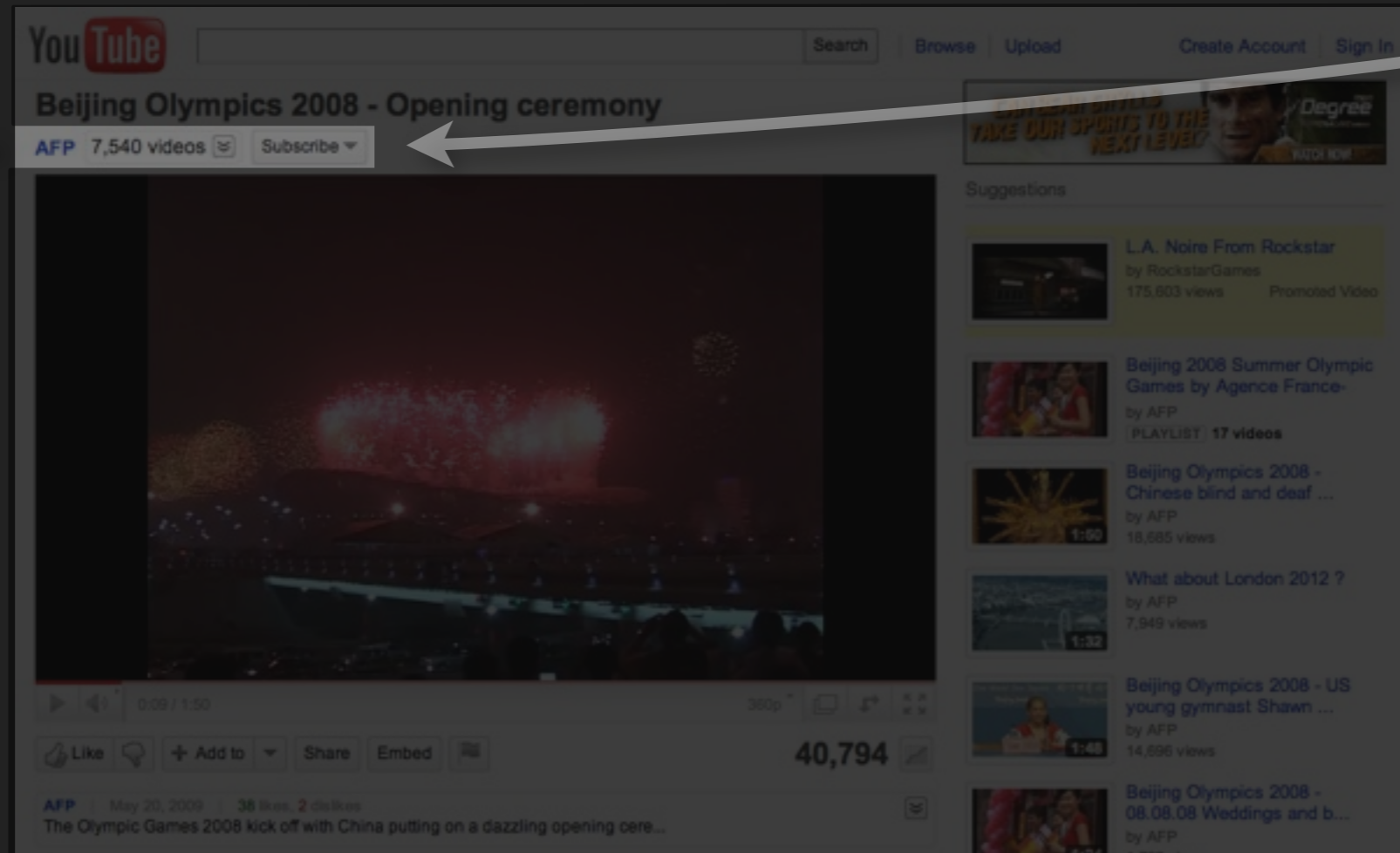
UIX
Widget

UIX
Widget

UIX
Widget

Event
Bubbling

Action
Execution



UIX Widget System: Behaviors



- The event handler matches the JS event with CSS classes in the registry to execute the actions of behaviors

UIX Widget System: Behaviors

```
// Sample event handler
```

```
uix.handleEvent = function(evt) {  
  if (evt.type in uix.registry) {  
    var actions = uix.registry[evt.type];  
    for (var css in actions) {  
      var el = findNodeOrParentWithClass(  
        evt.target, css);  
      if (el) {  
        actions[css](el, evt);  
      }  
    }  
  }  
};
```



UIX
Behaviors

UIX Widget System: Widgets

Event
Handling

UIX
Behaviors

Class
Matching

UIX
Widget

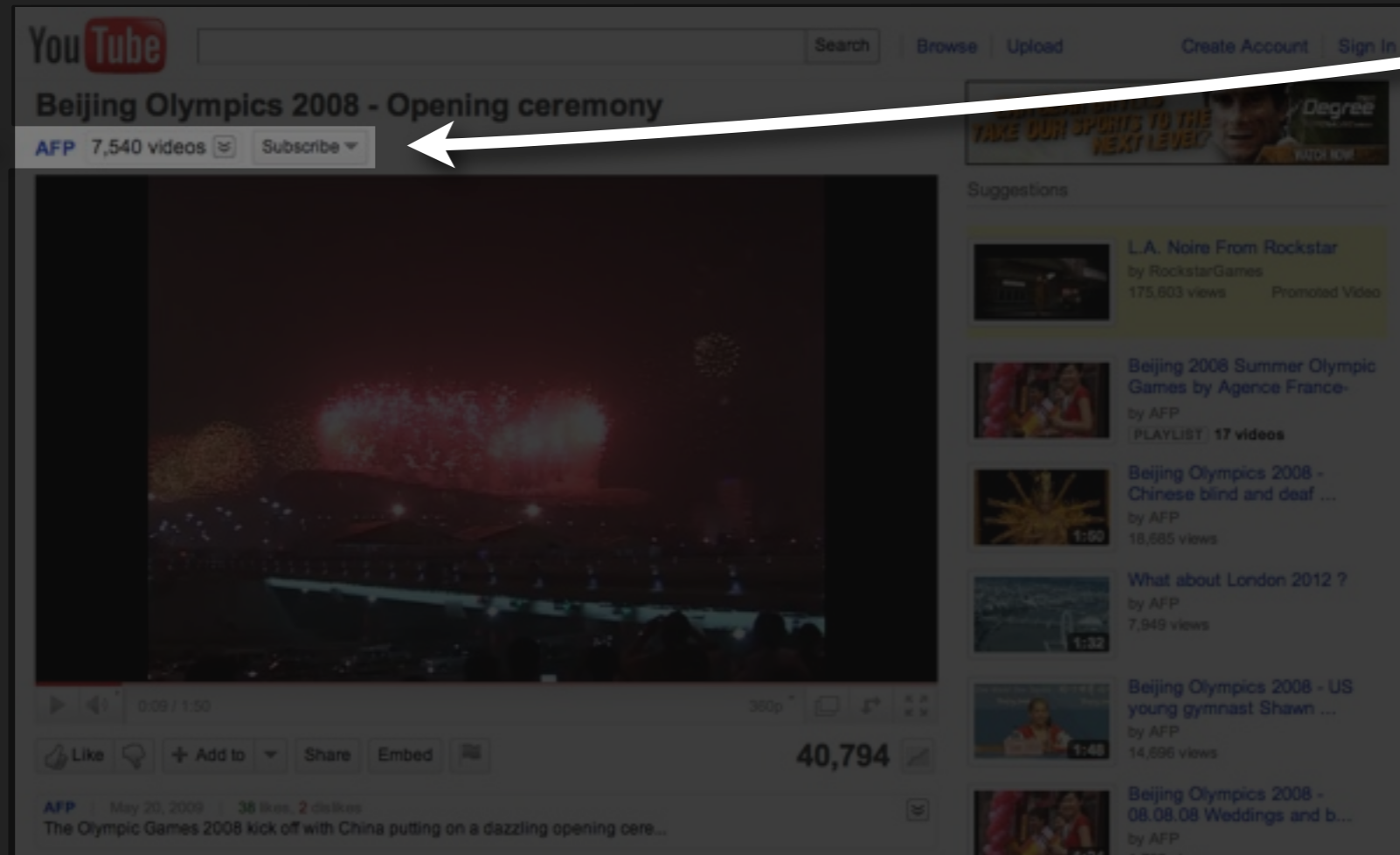
UIX
Widget

UIX
Widget

UIX
Widget

Event
Bubbling

Action
Execution



UIX Widget System: Widgets

UIX
Widget

- The JS for a widget is a collection of related functions that act on an HTML element in response to events
- The HTML for a widget is an element with a specific CSS class

UIX Widget System: Widgets

```
// Sample widget function
```

```
widget1.onClick = function(el, evt) {  
    if (hasClass(el, 'active') {  
        removeClass(el, 'active');  
    } else {  
        addClass(el, 'active');  
    }  
};
```

UIX
Widget

```
// Sample widget structure
```

```
<span class="widget1-css">...</span>
```


UIX Widget System: Delay-Loading

Event
Handling

UIX
Behaviors

Class
Matching

UIX
Widget

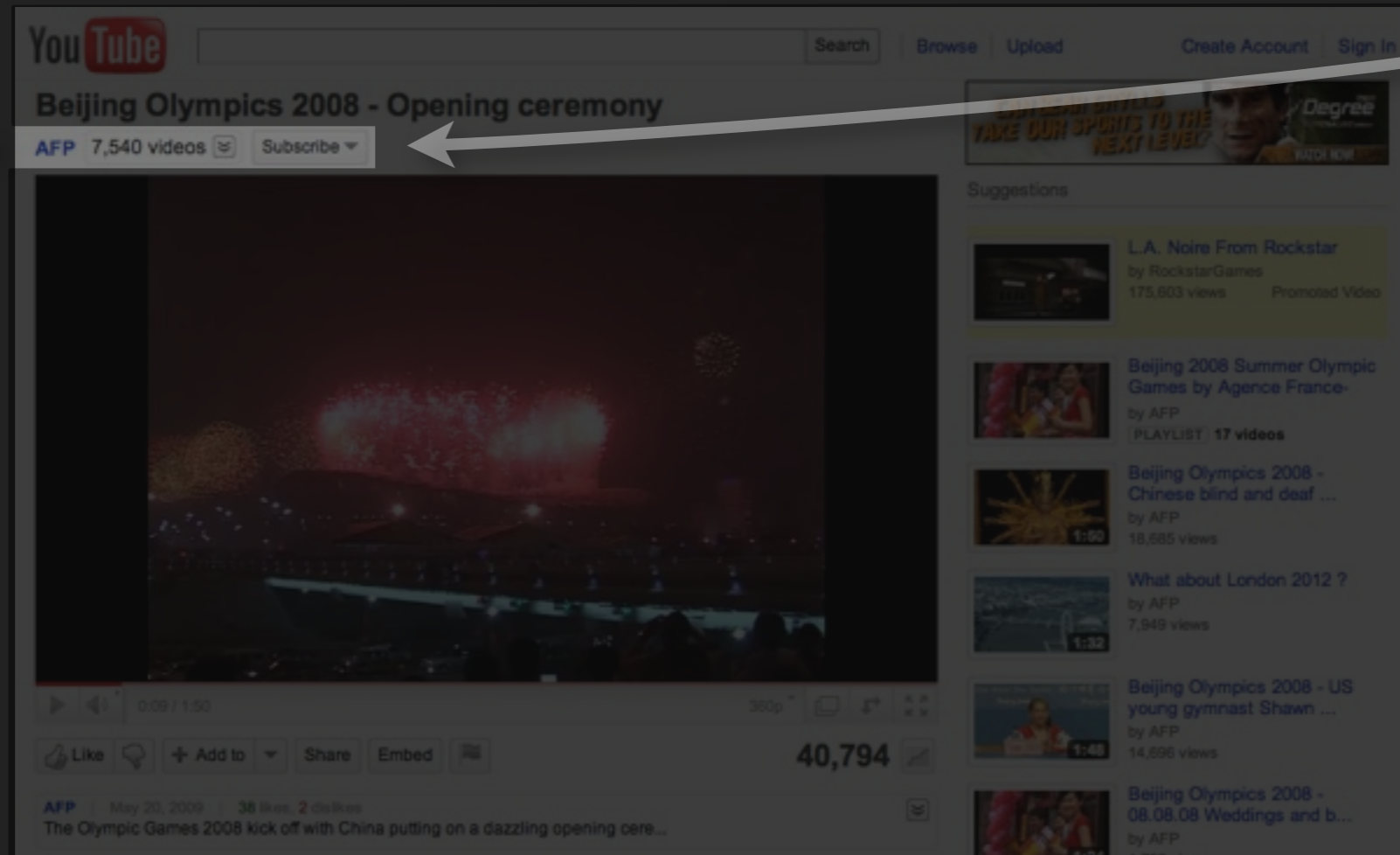
UIX
Widget

UIX
Widget

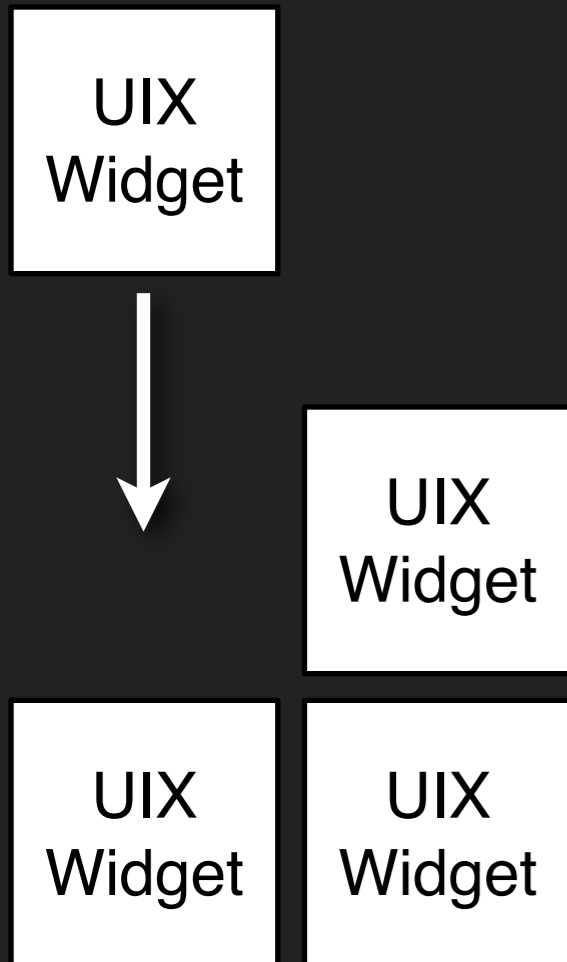
UIX
Widget

Event
Bubbling

Action
Execution

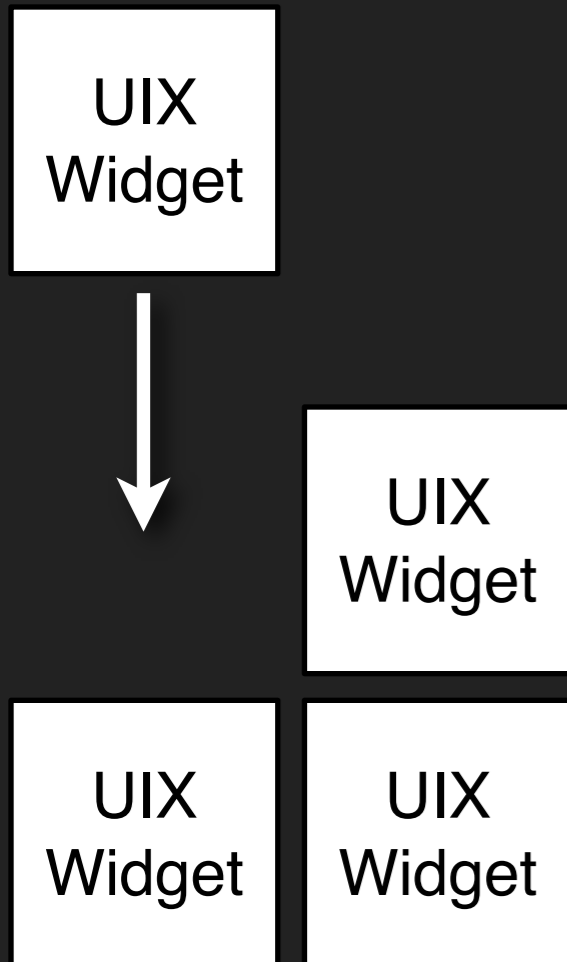


UIX Widget System: Delay-Loading



- Dynamically add new widgets by registering new behaviors

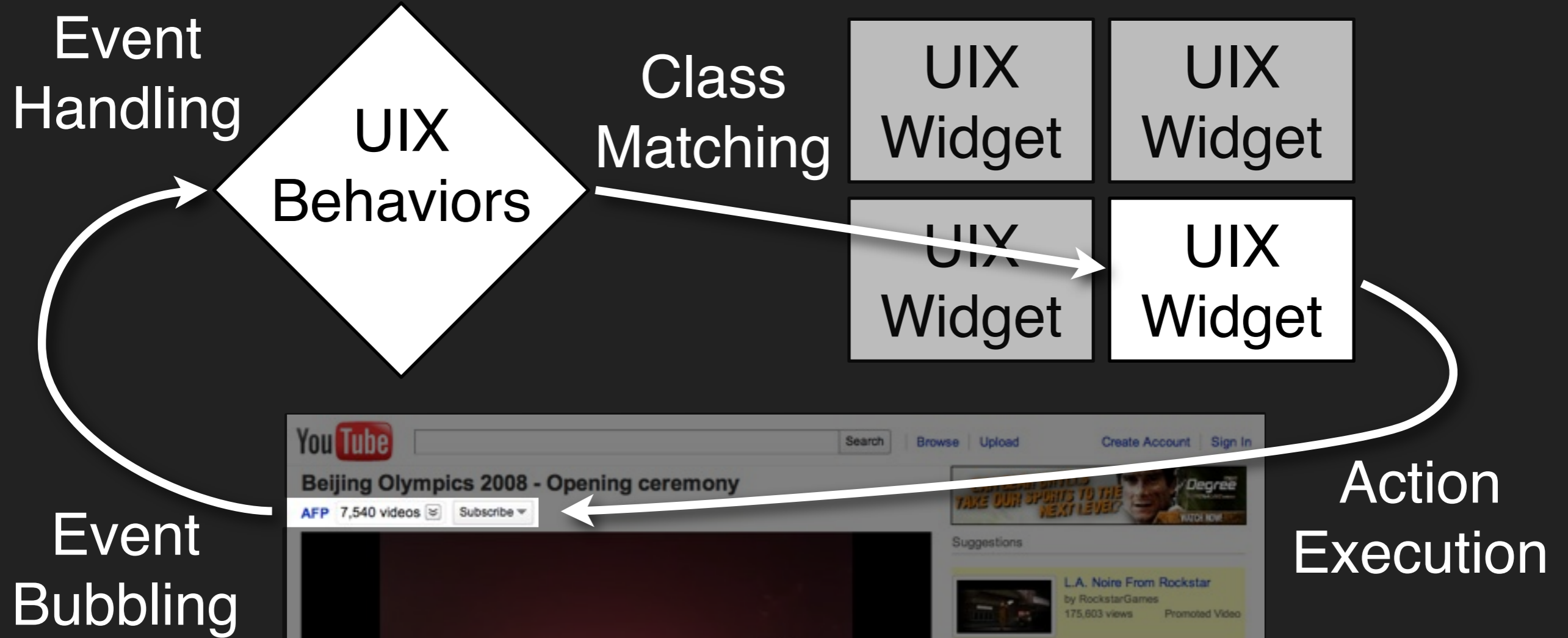
UIX Widget System: Delay-Loading



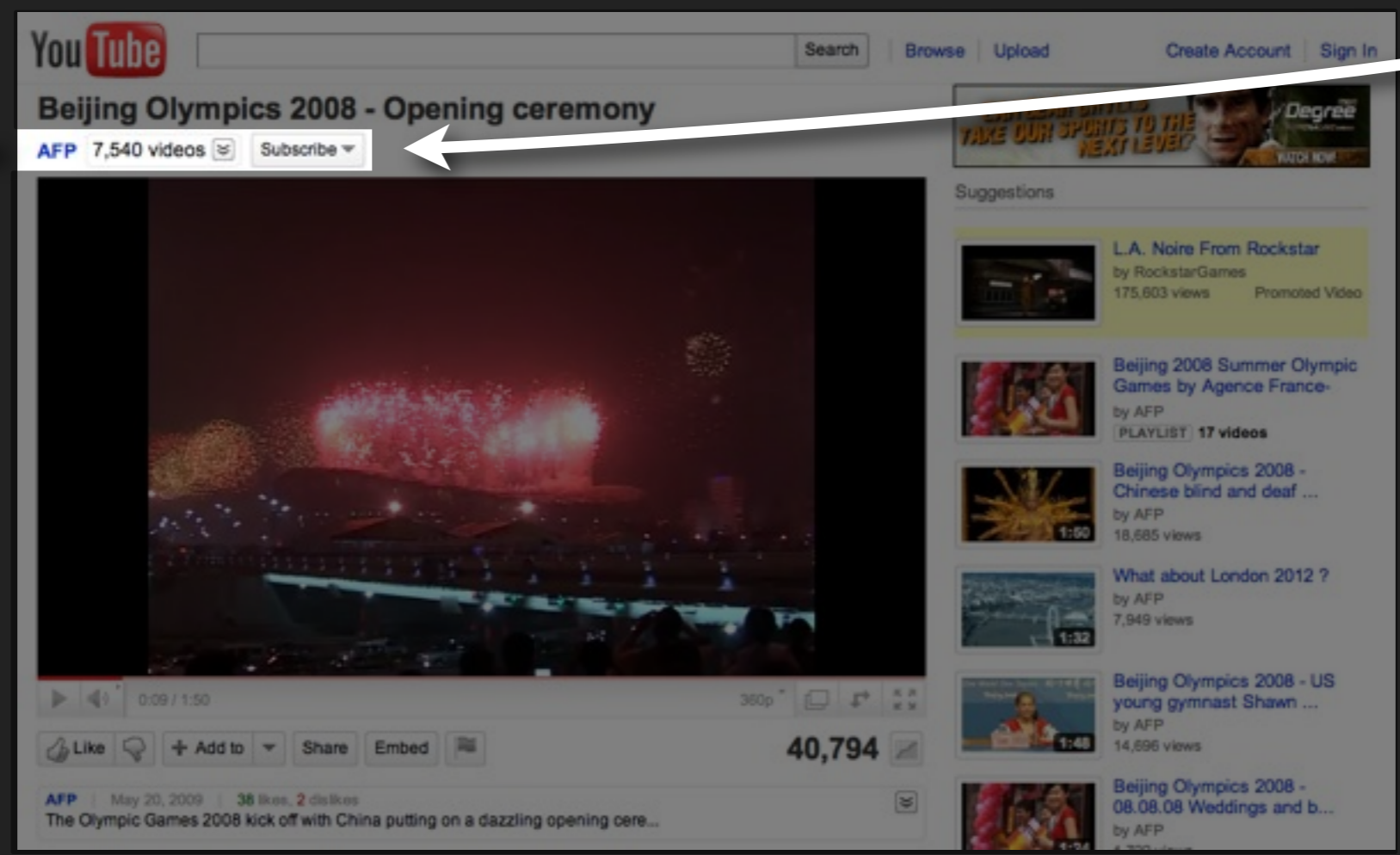
```
// Sample register function
```

```
uix.register = function(css, type, fn) {  
  if (!(type in uix.registry)) {  
    uix.registry[type] = {};  
  }  
  uix.registry[type][css] = fn;  
};
```

UIX Widget System: Architecture



Event Bubbling



Action Execution

Demo

UIX Widget System

Summary

- Performance Optimization #1
JavaScript at the Bottom and Embedding Flash
- Performance Optimization #2
Preloading the Video Connection
- Performance Optimization #3
Feather: Lightweight Version
- Progressive Enhancement
UIX Widget System

Thank You

Slides and Demos

<http://alexnicksay.com/velocitychina/>