Writing Big Applications With Google Web Toolkit

Bruce Johnson
Google, Inc.
bruce@google.com

Topics

Google

A Simpler-Than-Possible Explanation of GWT

Why Ajax Matters

GWT is Software Engineering for Ajax

Big Applications

Summary

Q & A

What is Google Web Toolkit (GWT)?

Google

What is GWT?

A set of tools for building Ajax apps in the Java language

What makes GWT interesting?

Write, run, test, and debug everything in Java

Isn't that called an applet?

GWT converts your working Java source into equivalent JavaScript

GWT is a Java-to-JavaScript compiler?

GWT has a compiler, but the full story is even more interesting…

Topics

Google

A Simpler-Than-Possible Explanation of GWT

Why Ajax Matters

GWT is Software Engineering for Ajax

Big Applications

Summary

Q & A

What I Mean by "Ajax"

Google

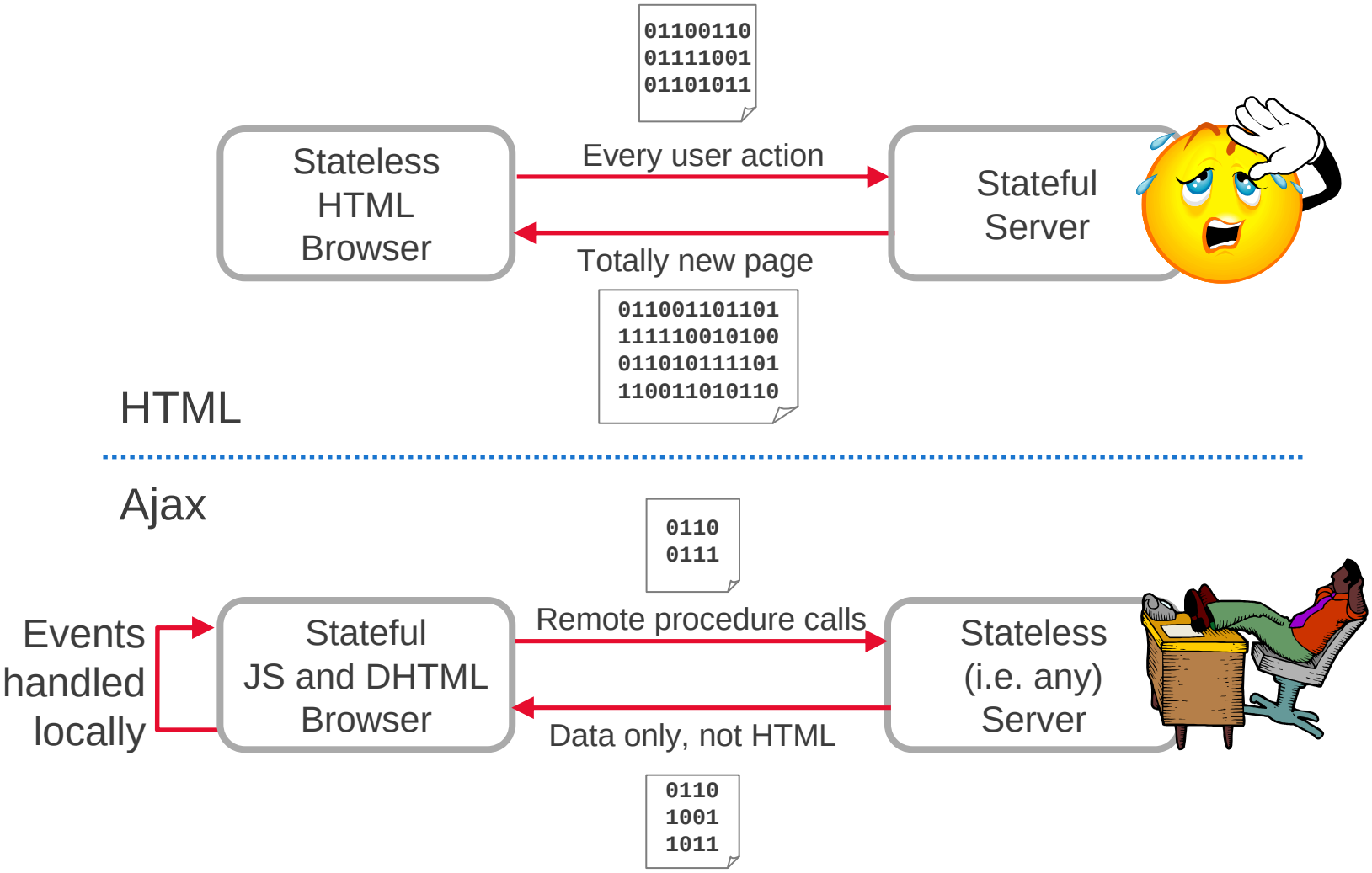
Updating the browser UI without switching pages

Fetching data without switching pages (XHR)

Viewing browsers as smart clients

Basically, re-creating client/server :-)

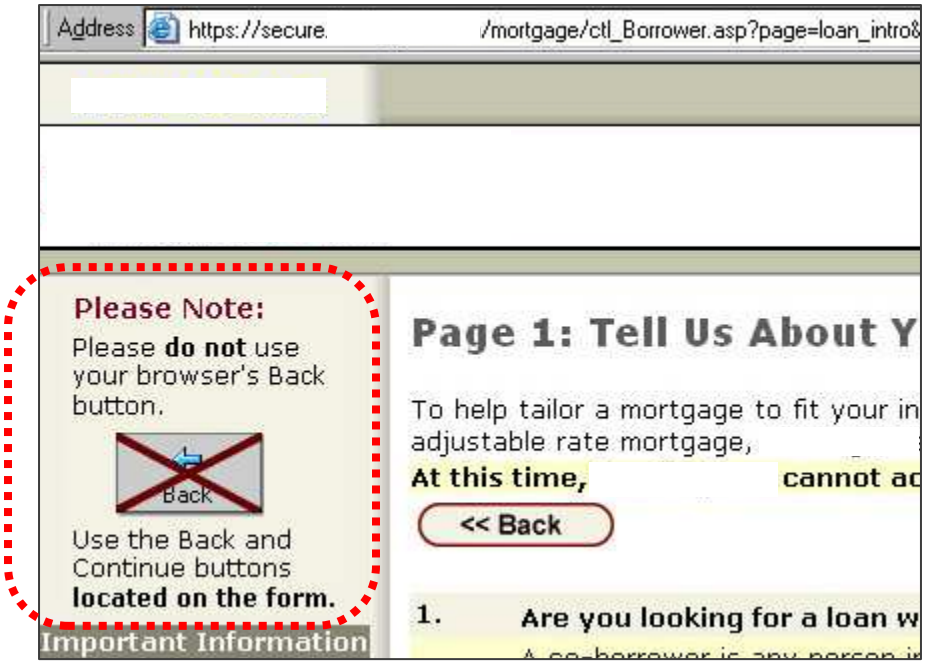
Why Ajax? Infrastructure Benefits



Why Ajax? Usability Benefits

Google

"Do not use your browser's Back button"



What if I do click Back?
Ajax can (in theory) solve this

Why Ajax? More Usability Benefits

Google

"Don't hit reload or we'll charge you twice!"

What if the network hangs? What should I do?
Ajax can (in theory) solve this

To Be Crystal Clear

Google

Ajax is more than a fad because it genuinely benefits all* stakeholders

Ajax will be important for many years to come

***not counting developers**

Topics

Google

A Simpler-Than-Possible Explanation of GWT

Why Ajax Matters

GWT is Software Engineering for Ajax

Big Applications

Summary

Q & A

GWT Requirements Laundry List

Google

Great Ajax apps that are still very webby

History, Bookmarks, a working Back button...

Leverage the Java language, developers, and technologies

IDEs, debugging, JUnit, findbugs, and profiling

Cross-browser with low overhead

Reuse via jars

Fast, simple all-Java remote procedure calls (RPCs)

Scalability (server-side session state not mandatory)

Basically: the impossible…

Unless you translate Java into JavaScript :-)

Code Sample – Hello, Ajax

Google

Demo time...

```
public class Hello implements EntryPoint {

  public void onModuleLoad() {
    Button b = new Button("Click me", new ClickListener() {
      public void onClick(Widget sender) {
        Window.alert("Hello, Ajax");
      }
    });

    RootPanel.get().add(b);
  }

}
```

Demo

Hello, Ajax

Wow, That's So Much Easier

Google

Redefining the problem has been fruitful

Session state? All client, not a server issue

Avoids round trips for UI event handling

Deployment? No fancy server, just compiled JS

Leverage for the biggest Ajax headaches

- Our Mantra: Solve the problem once & wrap it in a class
- History? Create a History class
- Cross-browser? Create an abstract DOM class
- RPC? Create an all-Java RPC mechanism

Rich UI: Widgets and Layout

Google

Build (or better, reuse) widgets

- Written in straight Java
- Code without worrying about browser portability

Separate UI style from logic

- Widgets are styled with CSS
- Automatically load the right CSS for your widgets

Demo

- "Mail" is a desktop-style application

Usability: History and Bookmarks

Google

History is the first thing to go in most Ajax apps

With GWT, it's easy and works well with MVC

```
History.addHistoryListener(myController);
```

History support leads to bookmark support

http://google.com/gulp.html#beta_carroty

Demo

"KitchenSink" shows history, bookmarking, and widgets

Adding GWT Incrementally

Google

GWT does not force you to start over

Attach a GWT module to any page

GWT layout does not expect full control

Provides a gradual transition path to Ajax

Even debug your existing web app in hosted mode

Demo

- "I18N" treats the GWT module as a controller against that knows nothing about the HTML layout

Simple & Powerful Client/Server: RPC

Google

Many solutions out there (JSON, XML-RPC, …)

But a pure Java RPC interface sure is nice!

```
interface SpellerService extends RemoteService {
  /**
   * Checks spelling and suggests alternatives.
   * @param the word to check
   * @return the list of alternatives, if any
   */
  String[] suggest(String word)
}
```

Client and server speak the same language (Java)

Demo

"DynaTable" loads records dynamically

Topics

Google

A Simpler-Than-Possible Explanation of GWT

Why Ajax Matters

GWT is Software Engineering for Ajax

Big Applications

Summary

Q & A

How Big is "Big"?

Google

The obvious question that rarely gets asked

What exactly are we trying to optimize for?

Download speed?

Are we supporting dial-up users?

Startup time?

First run? Subsequent runs? How fast, exactly?

Some particular size cutoff?

Size-on-wire? Size-in-cache?
Is the cutoff arbitrary or based on measured effects?
Funny: compare script size to the size of your images

Low-Hanging Fruit: Compression

Google

Ahead-of-time script compression

- **`C6BD1564339FC70220.cache.html` (119 KB)**
- **`C6BD1564339FC70220.cache.html.gz` (39 KB)**

Our "big" app instantly became 3 times smaller

- The last step of your build should be to zip GWT output

Classic HTML can't use compression so well

- Data changes frequently
- HTML changes rarely
- Mixing them forces compression into the critical path

Low-Hanging Fruit: Caching

Google

GWT supports aggressive script caching

Combine a small "selection script"…
→ `KitchenSink.nocache.html`

```
Expires: <pretty soon>
```

With a larger compiled script…
→ `md5.cache.html`

```
Expires: <when the sun explodes>
```

Viola! Perfect caching! (For image bundles, too!)

Never re-fetch the big script *unless* it has changed

Never *fail* to re-fetch the big script when it *has* changed

GWT Unlocks Client-Side MVC

Google

If you're confident that it's going to be a big app…

The default choice should be client-side MVC

Only tricky part is making your model async

Then again, not so bad…

```
myModel.requestNthItem(14);
…
class MyView implements MyModelListener {
   void onNthItemReceived(int n, Item item) {
…
```

MVC also fits perfectly with GWT history

An Algorithm to Evolve Your Architecture

Google

1. **Start by assuming you have a single page and you're building a traditional client-side MVC app (remember client/server? :-)**
2. **Add code as if you'll never hit a brick wall**
3. **Make sure your app implements history well**
4. **Evaluate the size and speed of your app**
   A. If you're happy, goto 2
   B. If you're unhappy, do all the stuff on the previous slides
   C. If you're still unhappy, see the next slide

Composition Techniques

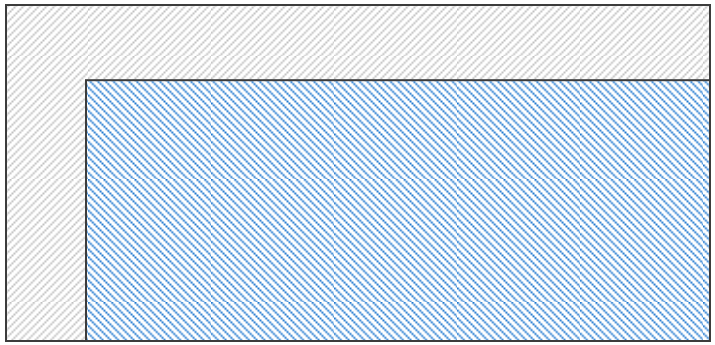
Google

Never feel obligated to keep your GWT app to a single page

- History smoothes this over
- Fast GWT startup makes page switching affordable

When wrangled by GWT, IFRAMEs aren't so evil

- Divide big chunks into IFRAMEs that your controller shows/hides them as necessary



Additional Techniques

Google

Consolidate multiple small RPCs

- Build composite structures and large-grained APIs
- Good rule of thumb: minimize HTTP round-trips
- Server replies with more data than was requested

Create UI lazily

- Fits naturally with history and MVC
- Spread the cost of widget creation across user-time
- See KitchenSink for an example

Topics

Google

A Simpler-Than-Possible Explanation of GWT

Why Ajax Matters

GWT is Software Engineering for Ajax

Big Applications

Summary

Actual Q & A

Open Source – w00t!

Google

Licensed under Apache 2.0

- Source available on Google Code

Making GWT Better (see session this afternoon)

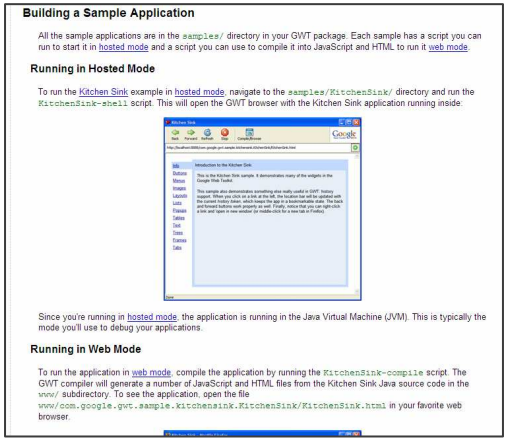
- The spirit of GWT
- Mission statement and design axioms

Great community

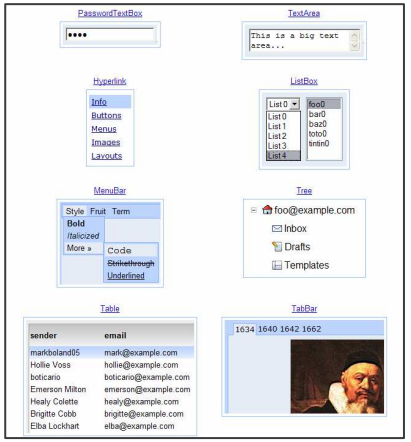
- 8,500+ members in the GWT Developer Forum
- 350+ members in the GWT Contributors Forum
- Many external patches included in GWT 1.4

Documentation Included

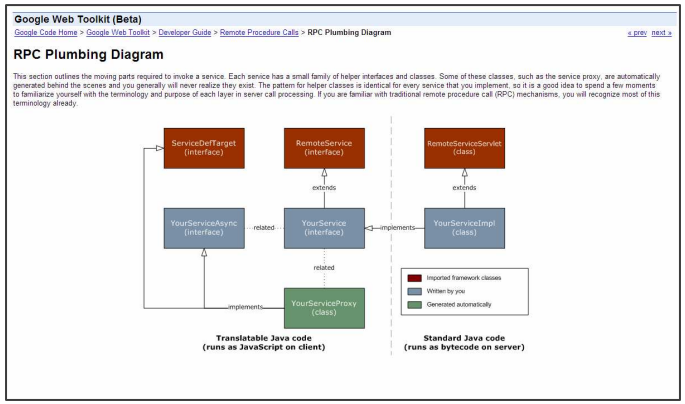
Getting Started Guide



Widget Gallery



Developer Guide



Class Reference

Large and Growing GWT Community

Google

Comprehensive IDE support for GWT (WYSIWYG, too)

Eclipse, IntelliJ IDEA,NetBeans, VistaFei, …

Major applications in production and in development

Google Checkout, Google Base, Google Mashup Editor, …

QuePlix, eTripBuilder, Whirled, DoubleCheck, MyHippocampus, …

Add-on libraries and sample code

100+ projects on Google Code alone

Books and articles

http://www.amazon.com for books

http://www.google.com for articles

Over 1 million downloads of GWT since launch

What's Coming in GWT 1.4?

Google

200+ features and fixes

Major size and speed optimizations

ImageBundle!

New widgets

- RichTextArea, SuggestBox, Splitters, … (several more)

Library enhancements

- NumberFormat, DateTimeFormat
- Benchmarking subsystem
- RPC now supports non-servlet Java back ends

Summary

Google

A PhD in browser quirks is no longer a hiring prereq

Turn Ajax development into software engineering

GWT rewards using good engineering practices

We will share our best work and ideas with you, and we hope you will return the favor

Much more to come…see you online!

Topics

Google

A Simpler-Than-Possible Explanation of GWT

Why Ajax Matters

GWT is Software Engineering for Ajax

Big Applications

Summary

Q & A

Common Questions

Google

Which browsers are supported?

Firefox 1.0, 1.5, 2.0

Internet Explorer 6, 7

Safari 2.0 (3.0 is looking good so far, too)

Opera 9.0

Common Questions

Google

What happens when a new browser comes out?
Do I have to wait for the GWT compiler to be updated?

Definitely no!

- All browser-specific code is in user-level libraries

The JavaScript language itself has very consistent support across browsers

- The DOM API is the real culprit

For backwards-compatible browsers, it's a no-brainer

For other situations, it's straightforward to change the user-level libraries

- Implement a version of DOMImpl for the desired browser

Main point: GWT was designed to never be a roadblock

Common Questions

Google

Isn't it hard to debug the script that GWT produces?

If you need to (or just want to) debug the compiled output, the GWT compiler gives you multiple output options:

-style OBFUSCATED (small, efficient, and fast)
-style DETAILED (nothing is left to the imagination)
-style PRETTY (perfect if you want to actually follow the code)

The output is normal JS

Debug as you would with handwritten JavaScript

Plus…you just won't have to

Common Questions

Google

How big are GWT apps?
Doesn't the compiler produce bloated script?

100 lines or less? Handwritten JS is reasonable

More than 100 lines? Use GWT

Compiler size and speed optimizations will ultimately win

Examples of compiler optimizations

Dead code removal
Type tightening
Devirtualization
Inlining
Aggressive obfuscation

(See next slide for experimental data)

Aggressive Size Optimizations

Google

Tough decision not to support reflection and class loading

Worth it! Three words: Whole program optimization

For example, type tightening to eliminate polymorphism

```
Shape s = new Square(2); // side length of 2
int a = s.getArea();
```

can become

```
Square s = new Square(2);
int a = s.length * s.length;
```

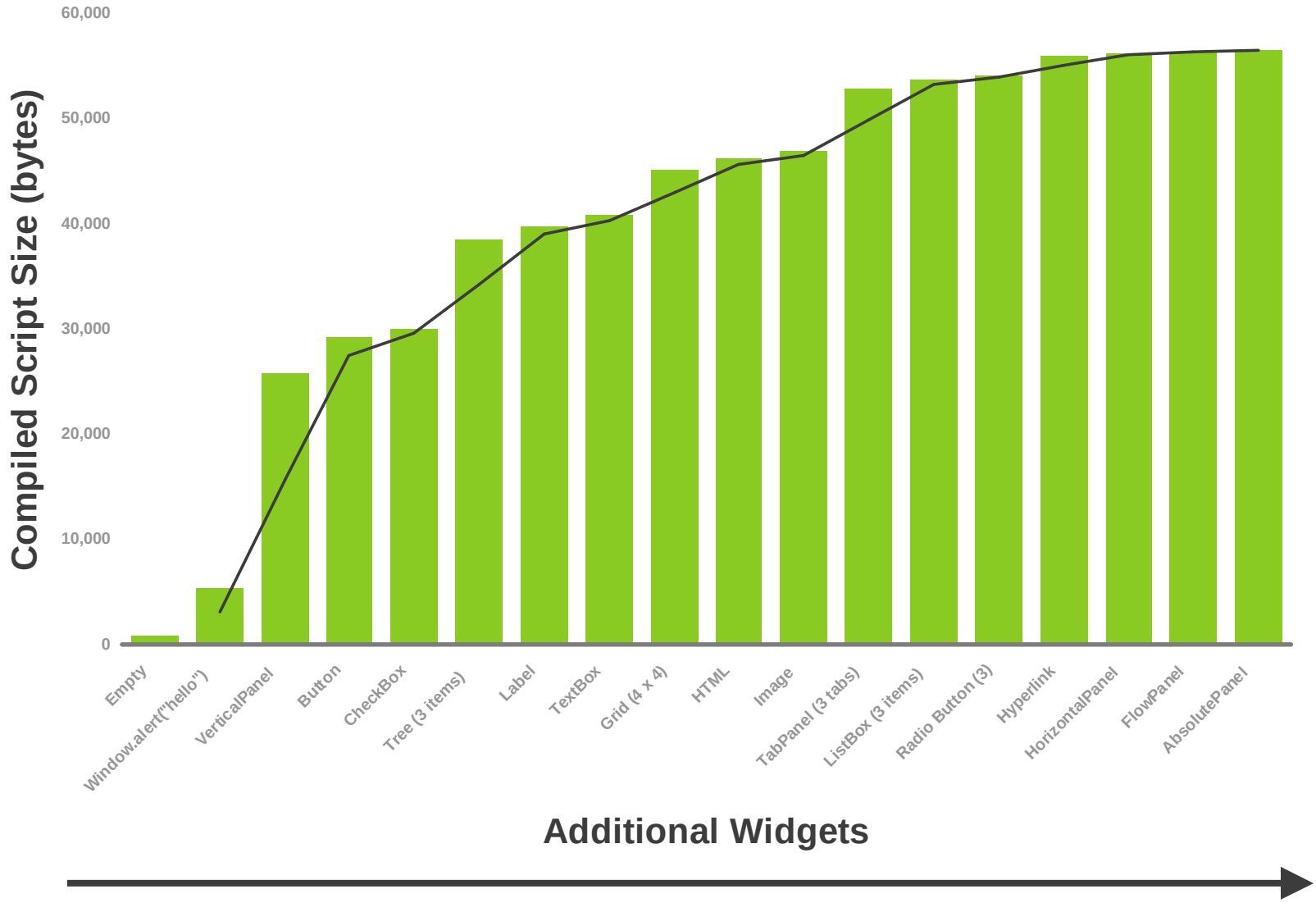
which, if Square's ctor has no side effects, can become

```
int a = 4;
```

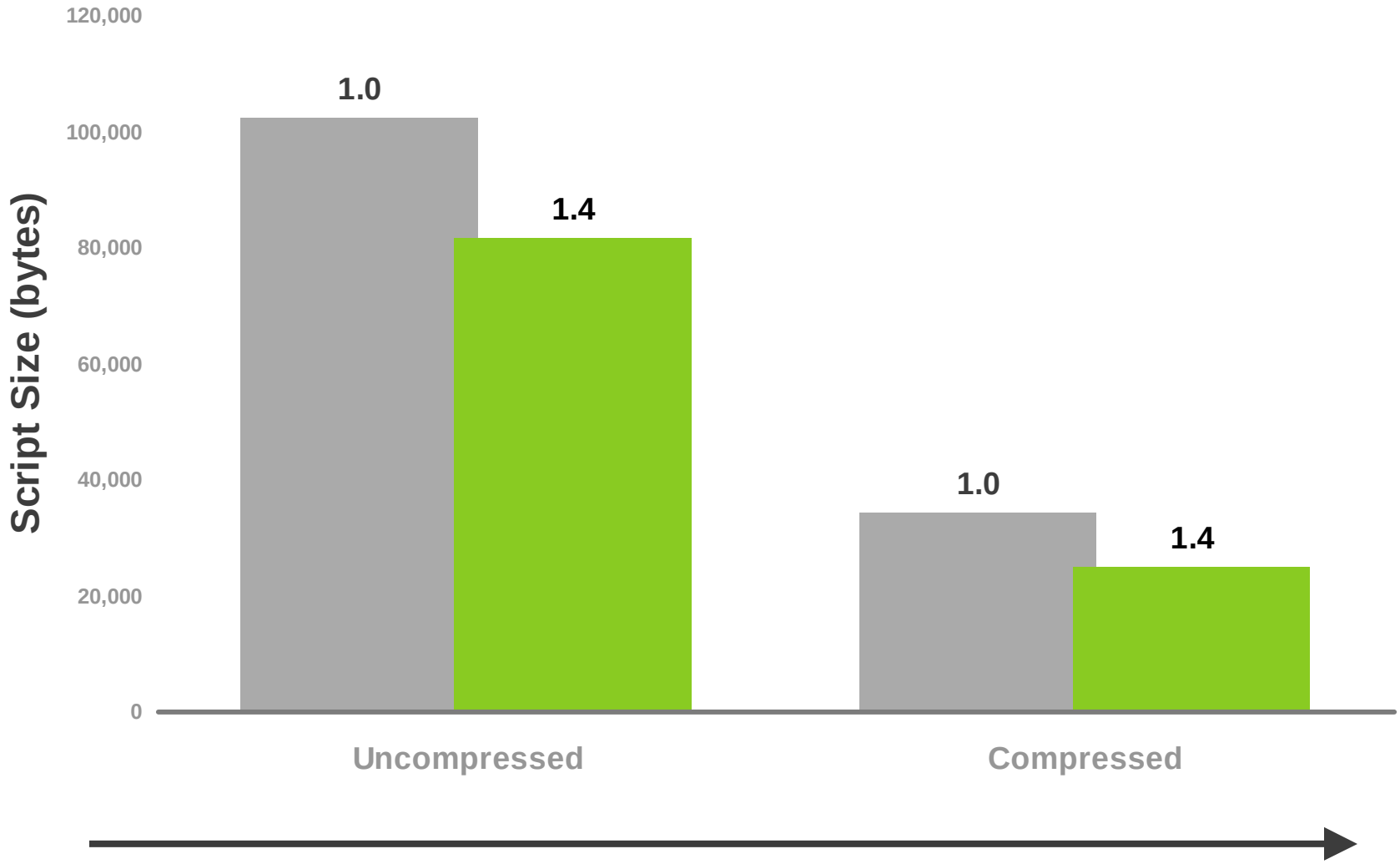
Imagine those sorts of optimizations across your entire app

In JavaScript, reducing size and increasing speed are complementary goals, which makes optimizations *really* fun

Compilation: Only Pay for What You Use



Compilation: Getting Better All The Time



Common Questions

Google

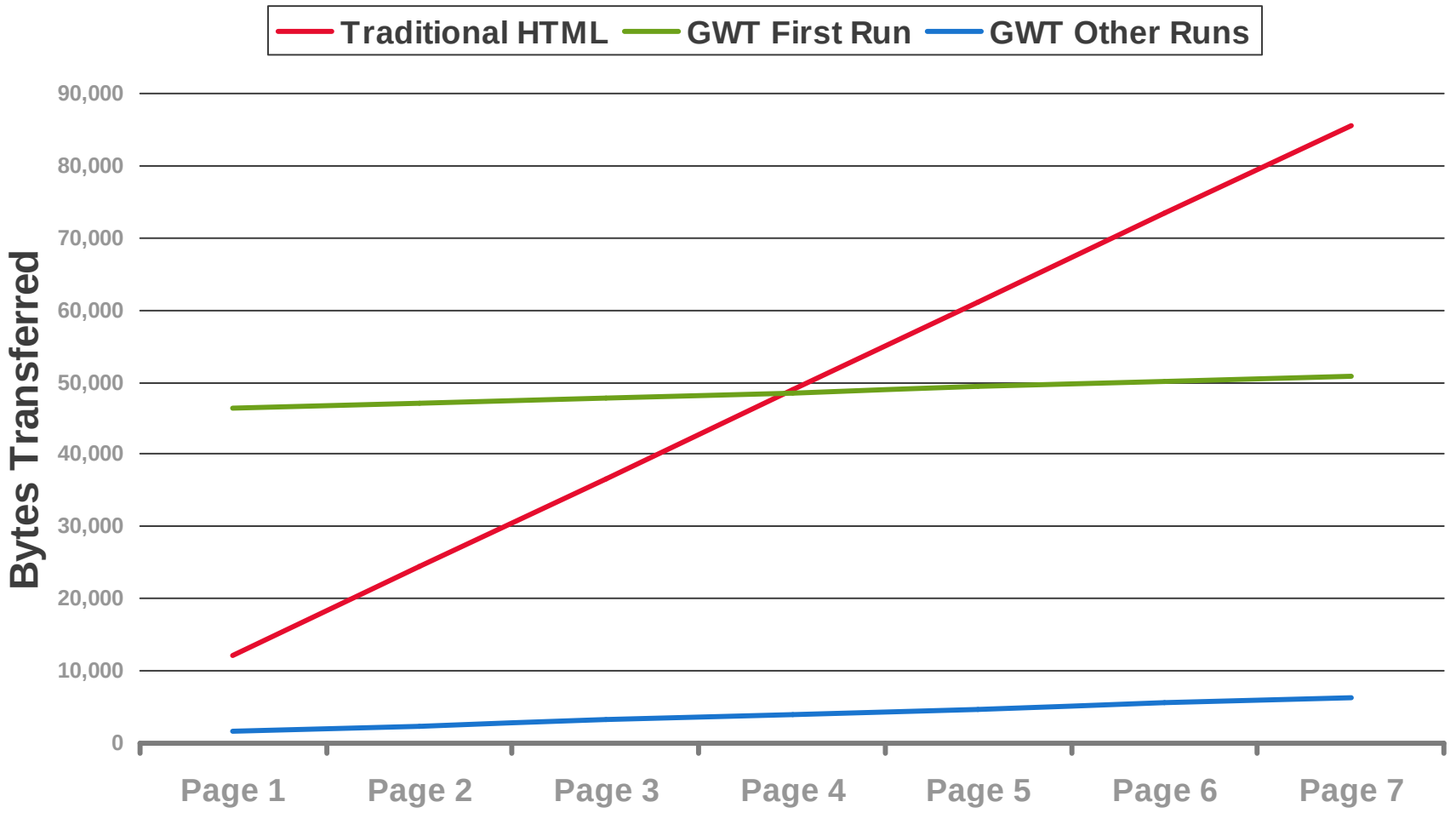
How fast are GWT apps?
Surely I could write faster apps by hand!

Likely to be true for very small apps

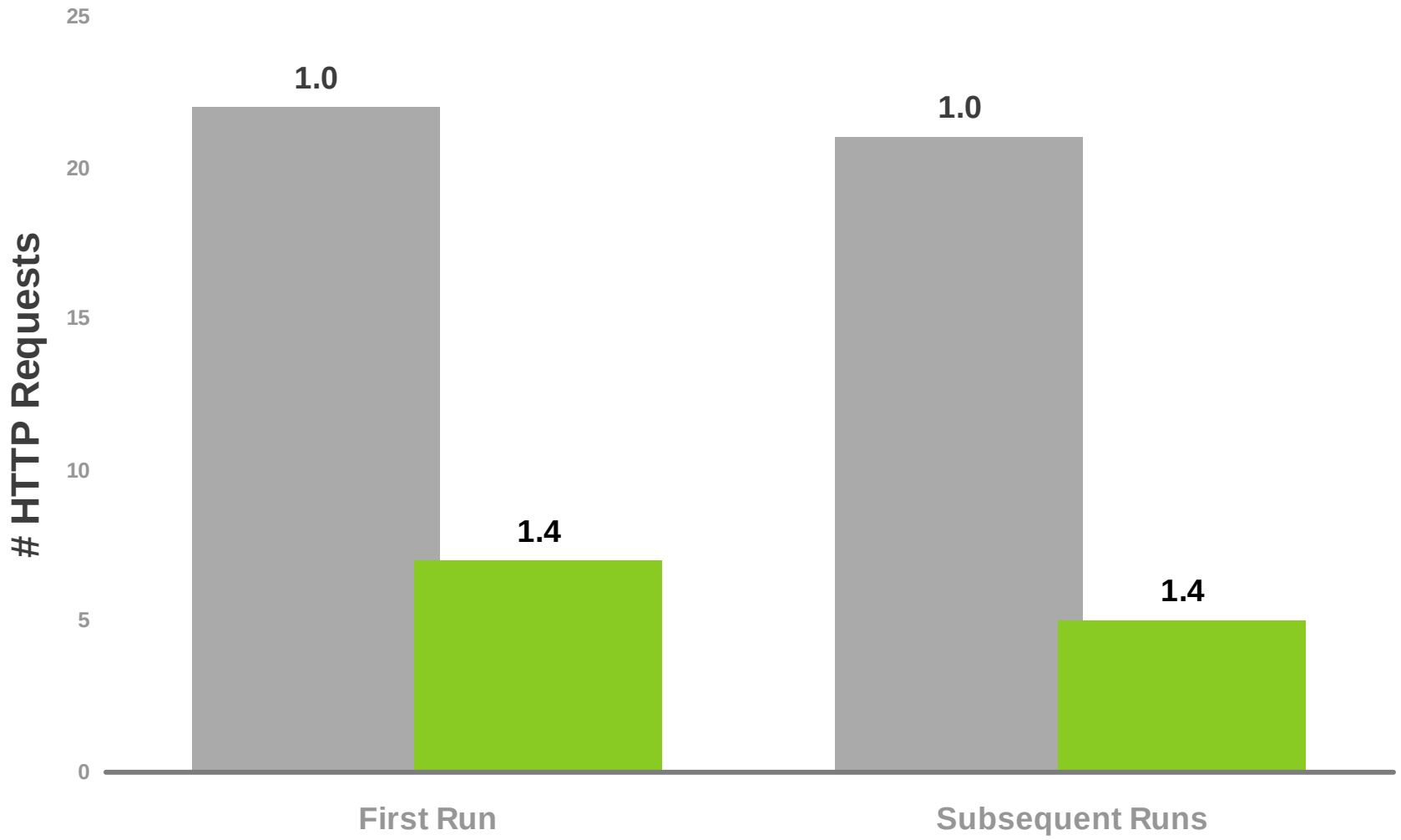
Unlikely to be true for bigger apps due to compiler and class library optimizations

(See following slides for experimental data)

Efficiency: Bandwidth and Startup Time



Latency: Minimizing HTTP Requests



Q&A

(come to “Making GWT Better” for more)