Google™ 10 IO

# What's Hot in Java for App Engine

Toby Reyelts
Don Schwarz

May 19, 2010

# Ask questions and take notes

View live notes and ask questions about this session at:

**http://bit.ly/appengine6**

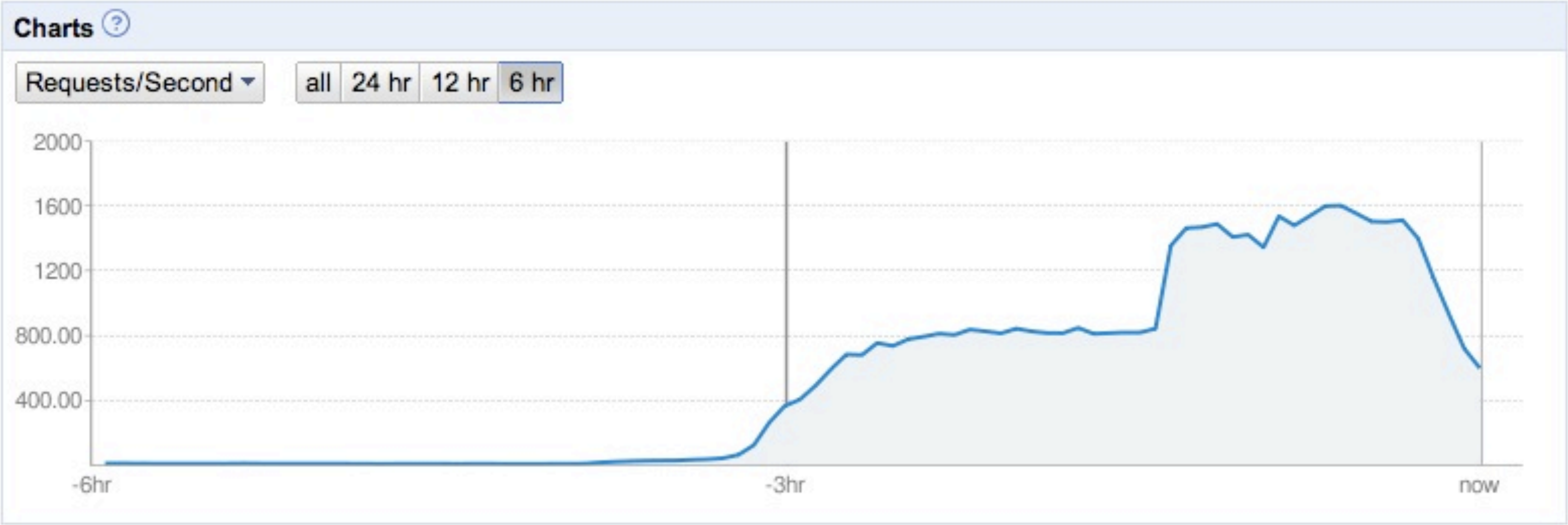
# Agenda

- Java support: one year later
- Dance Dance Robot
  - Demo
  - Code study
- Improvements
  - New functionality
  - Performance optimizations
  - Improved compatibility

# App Engine for Java: one year later

- We host 100,000 Java applications
  – Over 1/3rd of all App Engine apps

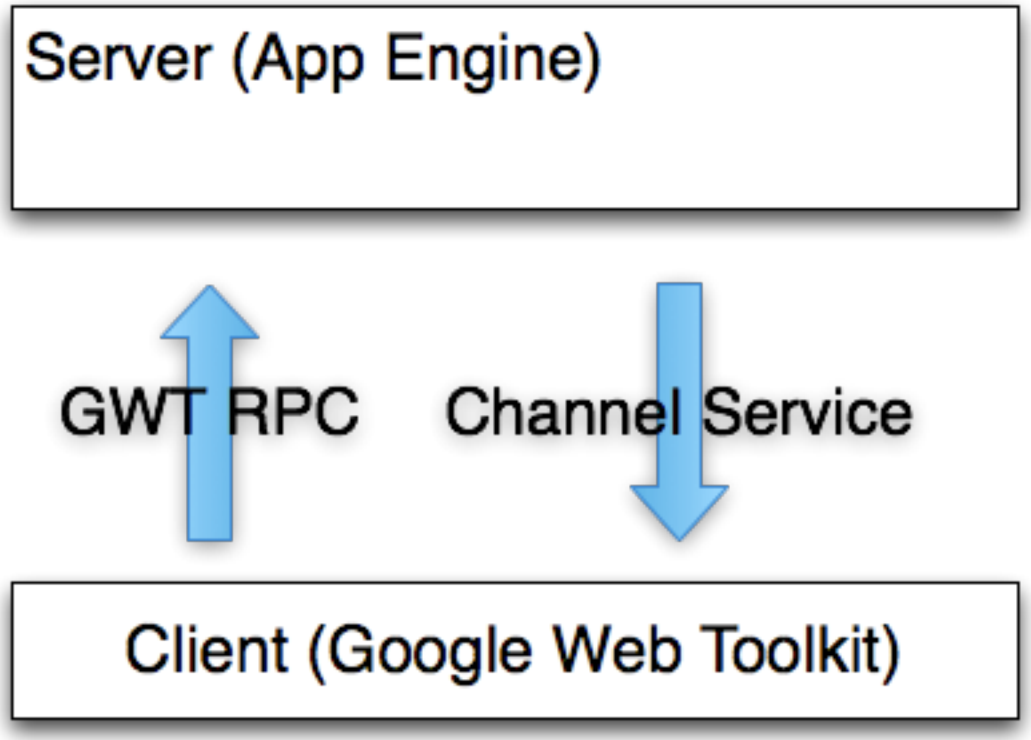
- We serve 1000s of requests per second
  – Also host many applications with large traffic spikes
    - e.g. Gigya Socialize



# Demo

# Game Design

## Transport



Server (App Engine)

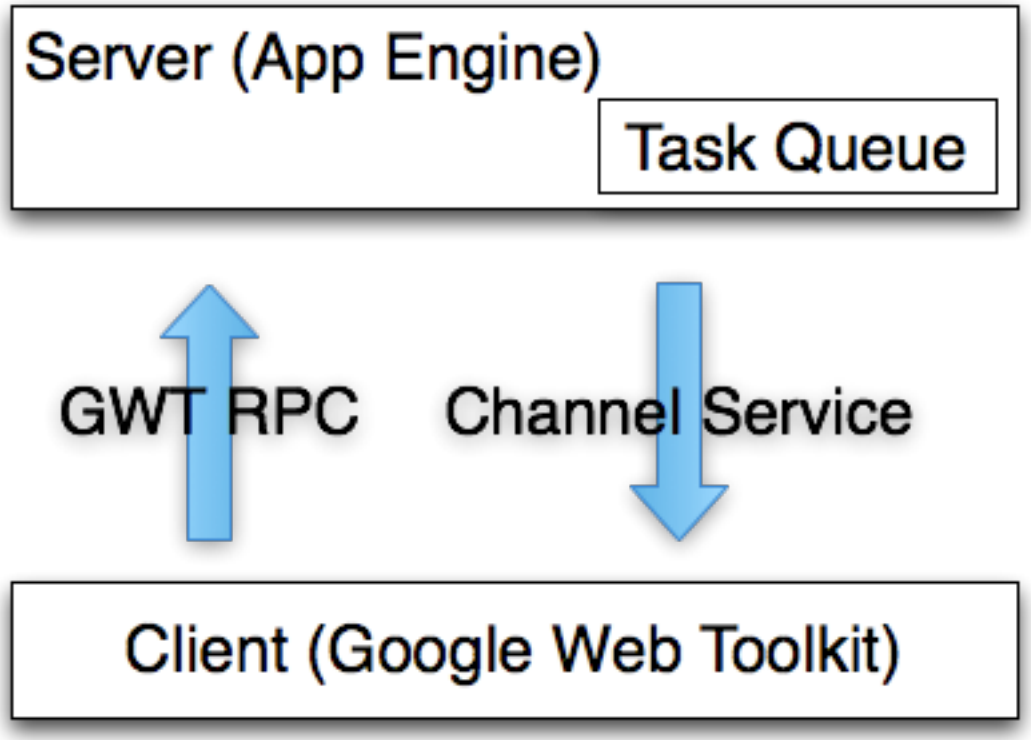
GWT RPC

Channel Service

Client (Google Web Toolkit)

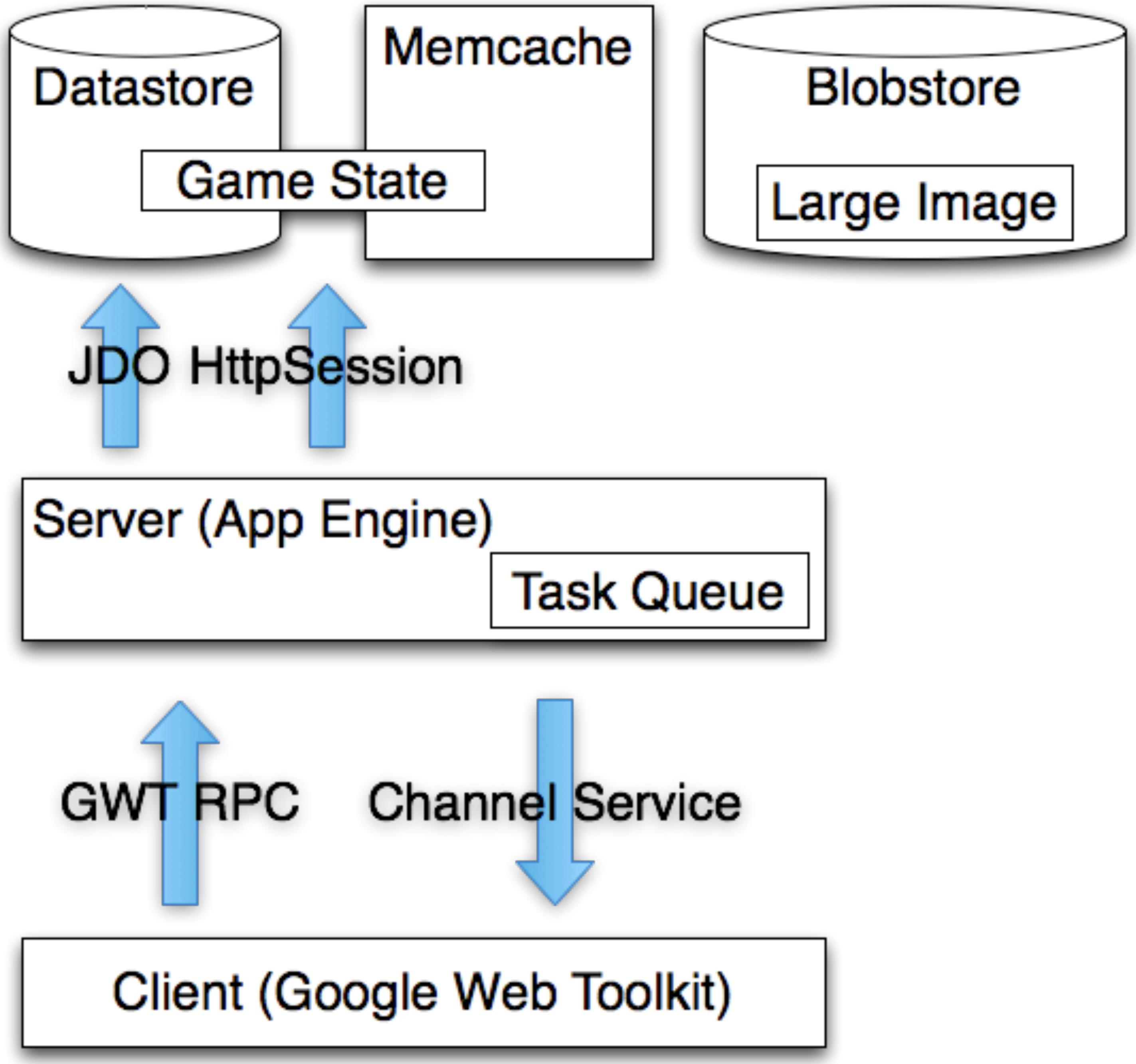
# Game Design

## Task Queues



# Game Design

## Storage



# Channel Service

Asynchronous Server ⟺ Client Communication

- Channel-based
- Bi-directional

- Server
  - Send messages via ChannelService object
  - Receive messages in a web hook
- Client
  - JavaScript library
  - Receive server messages in a callback
- Built on Gmail chat client (Google Talk)

# Channel Service

## Server API

```
   /**
 * ChannelService allows you to manage two-way connections
 * with clients.
 */
public interface ChannelService {

  /**
   * Creates a channel associated with the provided applicationKey
   */
  String createChannel(String applicationKey);

  /**
   * Sends a ChannelMessage to the client.
   */
  void sendMessage(ChannelMessage message);

  /**
   * Parse the incoming message in request. This method
   * should only be called within a channel webhook.
   */
  ChannelMessage parseMessage(HttpServletRequest request);
}
```

# Channel Service

## Client API

### Languages

- JavaScript
- Java (Google Web Toolkit)

### Sample code

```
var channel = new wnd.goog.appengine.Channel(channelId);
var socket = channel.open();

socket.onopen = function(event) {
  // socket is now fully functional
};

socket.onmessage = function(event) {
  // handle string msg (event.data)
};

socket.send(msg);
```

# Task Queue Service

- Allows you to do work in the background
  - Up to 50 requests/sec of offline requests

- Works in DevAppServer
  - Automatic execution at specified rates
  - Can see individual tasks and execute manually

- Tasks can take part in a datastore transaction
  - Enqueue a task only when a commit succeeds

# Blobstore Service

- Allows users to upload large files

- File upload handled by our infrastructure
  - You get a callback with a blob reference
  - Can query, delete existing blobs

- Blobs can be served back to the user (streaming)
  - Or retrieved a chunk at a time programatically

- Images API can take blobs as input source
  - Useful for thumbnailing user-provided images

# Appstats

- Easy to use profiling of API calls for perf tuning
- In Java, uses **`ApiProxy`** wrapper technique
  - Mentioned in last year's Google I/O session
- Stores API call data to memcache
- Built-in servlet renders results and timing stats

# New functionality

# Datastore improvements

- Cursors
  - Can iterate over results across HTTP requests
  - No more 1000 query limit
- Bulk ID allocation
  - Makes bulk upload feasible
- Opt-in to eventual consistency
  - Can speed up queries if stale data is good enough
- Statistics
  - Admin Console graphs of entity counts, sizes
  - Can be queried programmatically also
- Many JDO/JPA improvements

# URLFetch improvements

- Deadlines are now configurable
  - Up to 10 secs per request

- Asynchronous API calls
  - Make up to 10 simultaneous calls from each request
  - Future-based API:
    - `Future<HTTPResponse> fetchAsync(HTTPRequest)`
  - Other APIs will expose asynchronous APIs in the future

# Better integration through web hooks

- XMPP API
  - Can send and receive XMPP messages
- Incoming email
  - Receive email as an HTTP callback
- Google Wave API v2
  - Allows **active** Wave robots as well as passive

# Unit testing support

- Run unit tests using local API implementations

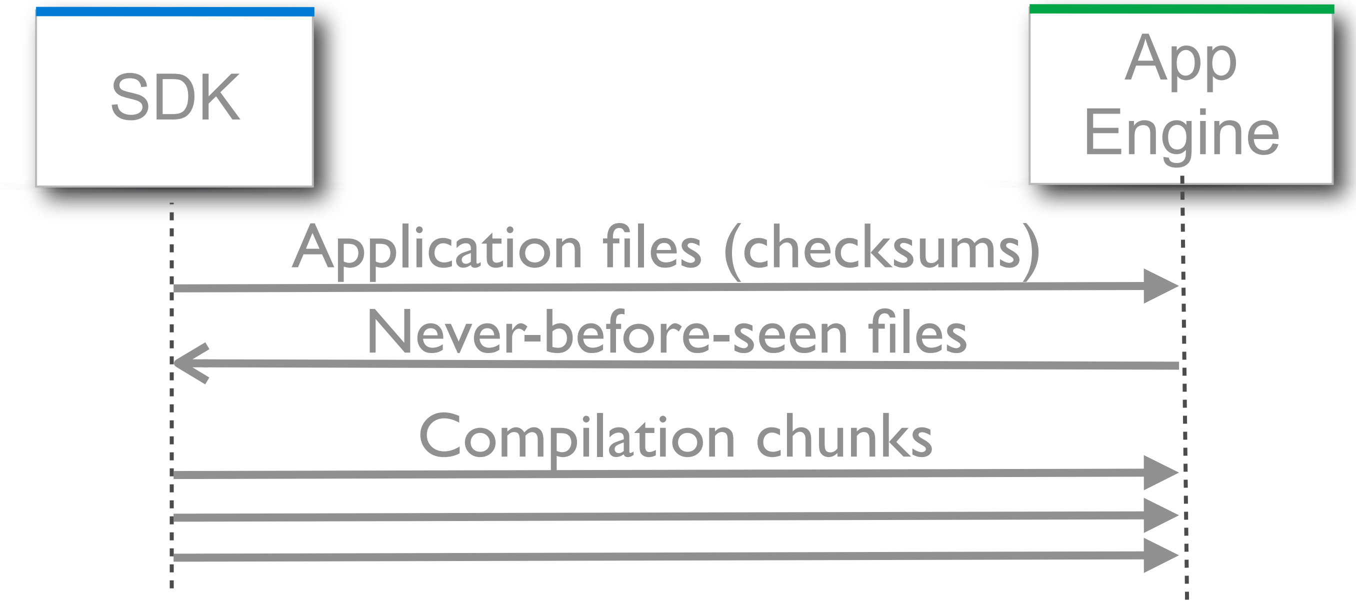
- Unit testing infrastructure in a separate jar
  - **`appengine-testing.jar`**
  - Configures **`ApiProxy`** to discover API impls on classpath

- **`TestConfig`** classes let you configure each API
  - Datastore disk usage disabled by default
  - Easily specify max memcache size

# Performance optimizations

# Precompilation

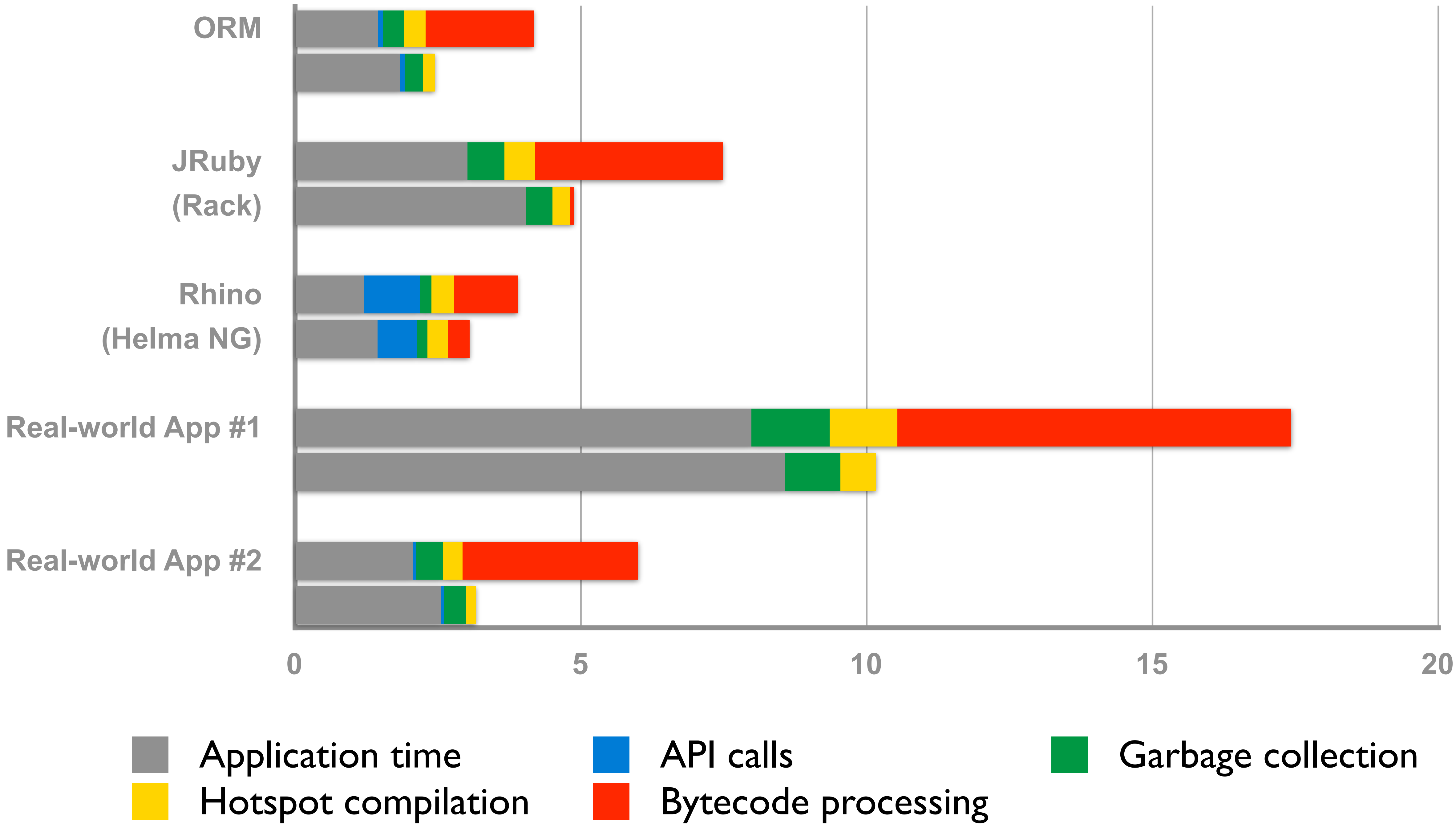
- Process application bytecode at deployment time
  - Allows many Java libraries to work in our sandbox
  - Saves time during loading requests
  - Opportunity for more expensive optimizations



- Occasionally need to re-process bytecode for all apps
  - Process each unique file only once
  - Total bytecode: only ~35 GB

# Precompilation results

## Loading request latency (before and after)



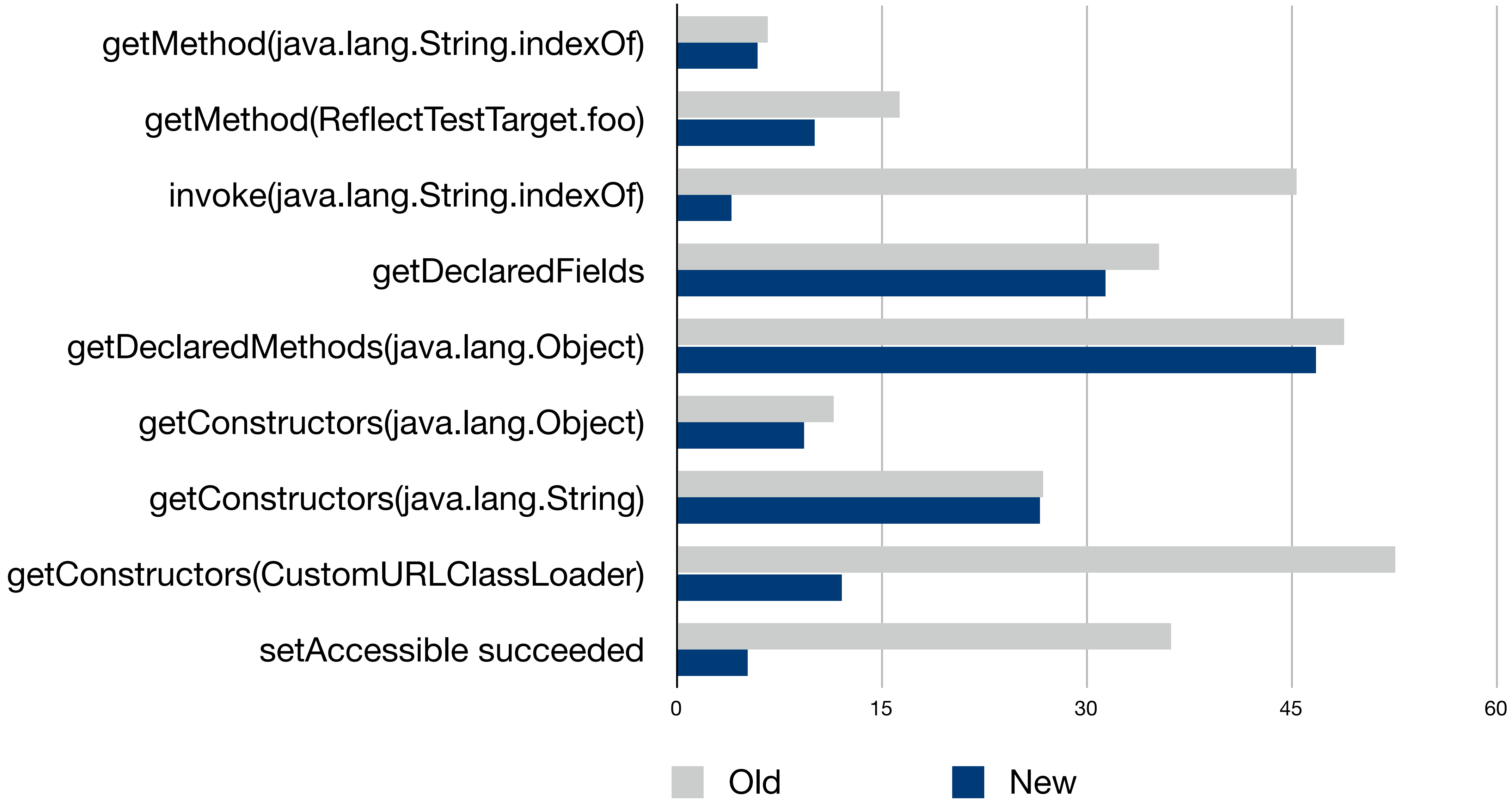
# Reflection Optimizations

- Big improvements
  - Caching reflection access checks
  - Failure can be expensive

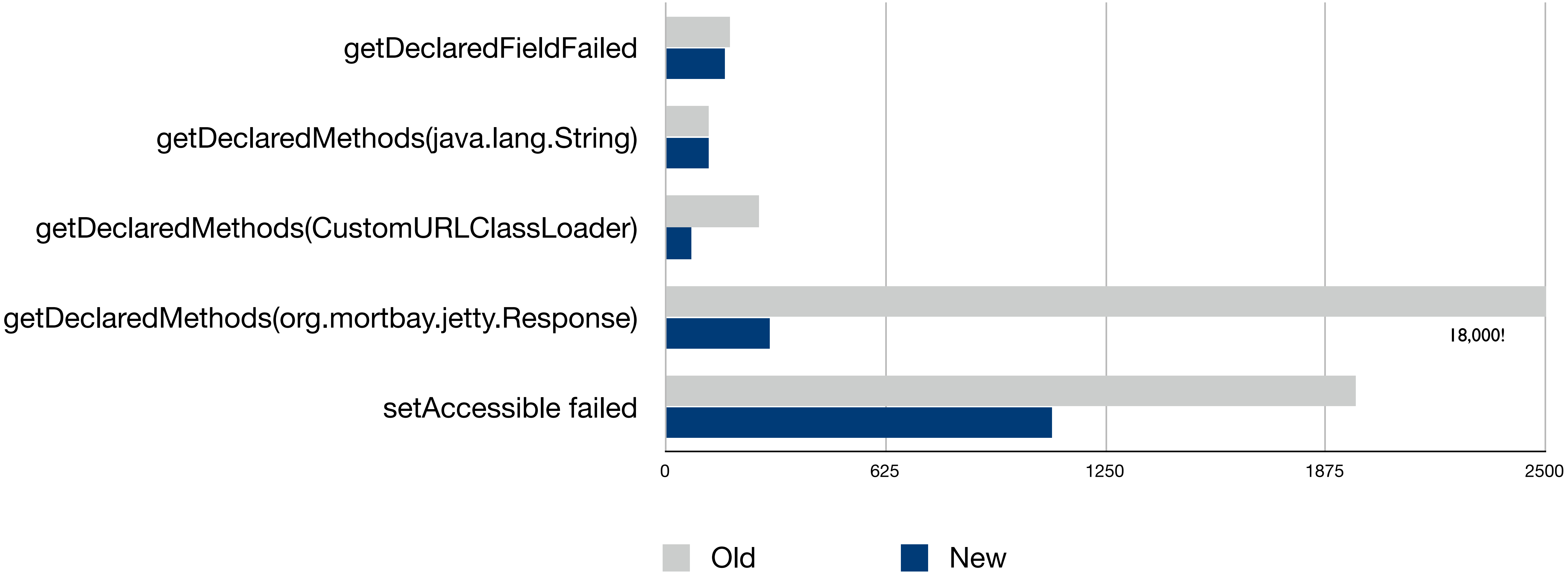
- Sample Grails app
  - 11K (!) reflective methods calls in loading request
  - 50% against three methods

- Conclusion
  - 10% faster startup for Groovy and JRuby

# Reflection Performance Part 1



# Reflection Performance Part 2



# Future performance improvements

- API call and I/O latency
  - Much faster memcache calls

- Reduced JIT and GC time
  - Improved parallelization

- Reserved instances
  - Dedicated JVMs to reduce loading requests
  - Will cost money, details coming soon
  - Grants greater visibility into use of JVMs

# Improved Compatibility

# Opened up more of the JRE

- Libraries
  - JAXB (`javax.xml.bind`)
  - StAX (`javax.xml.stream`)
  - XPath (`javax.xml.xpath`)

- Additional classes
  - `javax.annotation.Resource`
  - `javax.annotation.Resources`
  - `java.util.zip.ZipConstants`

- Many inner classes required for serialization

# More supported libraries

- Just to name a few
  - Hessian (4.0.6)
  - JDOM (1.1)
  - Jersey (1.1.5)
  - MyFaces (2.0.0)
  - OpenAMF
  - PureMVC
  - Restlet (2.0M5)
  - Struts 2
  - Tapestry (5.1)
  - VRaptor (3)
  - Vaadin (6.1)
  - Wicket
  - ...

# DevAppServer Sandbox Emulation

- New
  - WhiteList enforced
    - Works for reflection, too.
  - Reflection permissions enforced
- How?
  - Bytecode instrumented by JVM agent
  - Runtime shim modifies app behavior
- All perfect then?
  - Unfortunately still a few corner cases

# Resources

- Speakers:
  - **Toby Reyelts <tobyr@google.com>**
  - **Don Schwarz <schwardo@google.com>**

- Questions and notes:
  - http://bit.ly/appengine6

- Demo source code:
  - http://code.google.com/p/dance-dance-robot
  - http://dance-dance-robot.appspot.com

Google™ 10 IO