



HTTP Push Technologies

Twin Cities JUG

17 June 2008



Copyright 2008 LearnThinkCode, Inc.

This work is licensed under a Creative Commons Attribution-NonCommercial-Share Alike 3.0 Unported License



Overview

- Scenario (or “what I did in England”)
- HTTP Push (aka Reverse Ajax/Comet)
 - What it is and How it works
- Examples
 - Dojo/Jetty
 - DWR/Grizzly
 - LightStreamer
 - Scala lift
- Issues / Solutions



Scenario



- William Hill – premier UK bookmaker
- Site shows coupons and an AJAX bet slip
 - Coupons are outcomes to bet on (with odds)
 - Slip shows single, accumulator and cast bets
 - Similar to <http://www.sportingbet.com>
- Business promoting “in-running” betting
 - Event is in progress...odds are changing!
- Need to push updates to browser
 - Particularly for outcomes and odds on slip!



Scenario /2



- Technologies
 - Web Framework was Spring MVC
 - AJAX Framework was DWR
 - Comet Framework was LightStreamer
 - Spring for service layer and DAO layer
 - Services updated data in Coherence datagrid
 - Updates to server were “write-through” caches
 - Updates to cache came when prices change
 - Messaging was Oracle Advanced Queues
- Never went live...plug pulled in Feb 2008





HTTP Push

- Lots of internet “push” technologies
 - IM, UUCP, FTP, SMTP, P-IMAP, etc... but not HTTP
 - But push exists on some web-sites
 - Google Apps, JotSpot Live, Meebo, Renkoo, cgi:irc, chess.com
- AJAX is still request-response
 - not a HTML page refresh
 - multiple requests at a time
 - asynchronous response
 - typically updates the DOM for us
- AJAX Push get response without user interaction
 - uses XHR with polling or streaming to get messages
 - messages may be user-initiated or server-initiated
 - the server is an event bus
 - typically updates the DOM for us
- Why do it?
 - Collaboration, live data (grids!), chat, games, etc.!!
 - For social websites - the server is other users
 - Updates to context/data affect what actions I can take
 - Makes the web a two-way communication system



“Reverse AJAX” to Bayeux



- Getting Messages – Polling vs Streaming
 - Polling closes the connection with the response (may be short or long)
 - Streaming continues to push data via the connection
- “Reverse AJAX” coined by Jonathon Downes (DWR)
 - 3 ways to get updates to the browser
 - Polling – new AJAX request every so often
 - Piggyback – send updates with next AJAX request
 - Comet – long lived HTTP (slow load)
- “Comet” coined by Alex Russell (Dojo)
 - named after another kitchen cleaner
 - Umbrella term for long polling and streaming mechanisms
 - Javascript toolkit + Bayeux messaging + Event Server
 - Bayeux allows for interoperability of Javascript toolkits
 - requires **non-blocking app servers!**
- Bayeux is standard protocol for Comet
 - Handshake/Negotiation
 - Pub-Sub / Client-to-Client
 - Channels (like filesystem paths)
 - Can batch messages
 - JSON-based messages

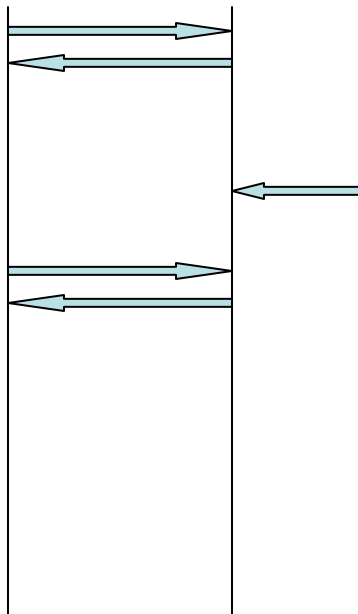


AJAX Poll vs AJAX Push



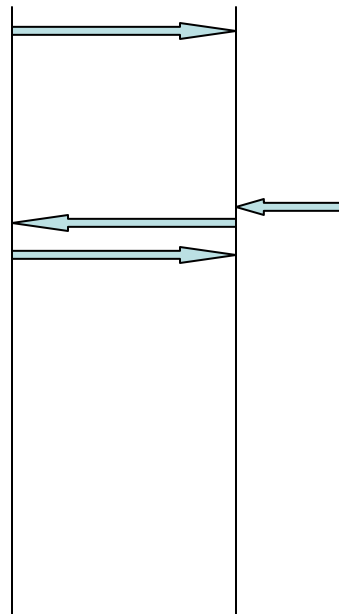
AJAX (Polling)

Browser Server



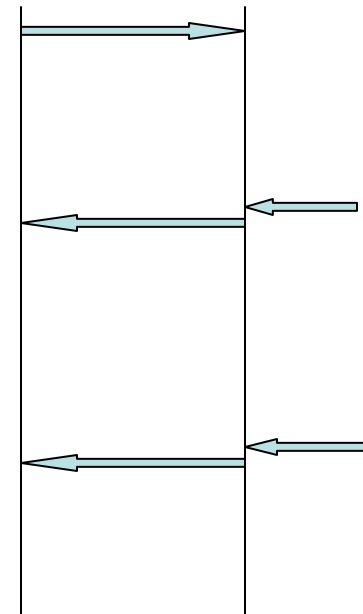
AJAX Push (Long Poll)

Browser Server



AJAX Push (Streaming)

Browser Server





HTTP Push Techniques

- Polling and Piggyback
- Long-Polling
 - waits for data, but closes/reopens connection
- Callback-Polling
 - GET that returns javascript that returns JSON
- iframe Streaming (forever frame)
- htmlfile (ActiveX) Streaming
- XHR Streaming
- Multipart Streaming
 - multipart/x-mixed-replace (non-IE)
- Flash





Demos!!

- Will do...
 - Java/Bayeux
 - Dojo/Jetty
 - DWR/Jetty
 - Dojo/Grizzly
 - DWR/Grizzly
 - Lightstreamer (Java+various/LNP)
 - lift (scala/?)



Dojo



- Dojo
 - Widely-used Javascript framework/toolkit
 - Core (dojo) and Extended (dojox)
 - use what you need
 - interesting download capabilities
 - “raw” API for Bayeux in dojox.cometd
 - Typically ‘object’ methods call bayeux API
 - need to handle some low-level stuff yourself
 - Integrated with Dijit framework (widgets)



Jetty



- Jetty
 - Continuations (hidden behind Bayeux impl)
 - Bayeux API
 - Channels (get/new, has, remove, get-all)
 - Clients (get, has, remove, new)
 - Data filters
 - Bayeux Implementation
 - Add ability to call method for event
 - Used in chat to keep track of members
 - Servlet creates and saves Bayeux object





DWR

- DWR
 - Objects in Javascript and Java are “in synch”
 - Knows of all users of a page
 - `context.getScriptSessionsByPage(url);`
 - Can remotely
 - update values
 - call remote javascript function
 - call remote `script.aculo.us` Effect
 - call remote TIBCO GI
 - Hides the server implementation
 - works with Jetty/Grizzly



Grizzly



- Grizzly
 - NIO/Asynch framework
 - Can be used in Glassfish, Jetty, etc.
 - Has its own Comet (sub)framework
 - Bayeux on the wire
 - Does not use Bayeux API from dojo (AFAIK)
 - Glassfish v3 has Grizzly/Comet installed





Lightstreamer

- Event Bus or Outboard Server
- Closed-source but free (as in beer) version
- Works as “database synchronization” engine
 - tables, rows and columns = tables, items and fields
 - various merge mechanisms (mode)
 - a Lightstreamer table is a group of items and fields and mode
 - Event API
- Client APIs for anything
 - Javascript, Flex/Flash, Java (SE/ME), .NET
- Data Adapter APIs
 - Java, .NET, Remote
- Bandwidth and Frequency Control
 - Auto-throttles
 - Won't send stale data





lift (Scala)

- Scala is a new language
 - based on Java, runs on JVM
 - both functional and object-oriented
 - can call Java code / Java can call Scala
 - Scala features (competing with Java 7)
 - XML (XHTML) in the code
 - Closures
 - Actors
- Lift is web framework for Scala
 - closures kept around between requests
 - classes generate HTML, S.* methods generate xhtml
 - CometActor as built-in class and `<lift:comet>` tag





Other Potential Protocols

- Not everyone likes Bayeux...
 - Lightstreamer has its own (LNP)
 - Caplin Liberator has its own (RTTP)
 - Orbited has its own (Revolved - open)
- XMPP/Jabber
 - Pure Javascript implementation!
 - BOSH (based on XMPP)
- Flash/Applet ServerSocket?



Issues / Solutions



- 2 connections per domain
 - IE8/FF3 up the limit regardless of HTTP spec
 - can put comet servers on another domain
- Stream Stoppers
 - old proxy servers
 - Apache mod_jk
 - buggy firewalls
- Its a 'hack' (not a standard)
 - NIO.2 and Servlet 3.0
 - WHATWG and W3C - HTML 5 Event Sources
- Web 3.0?
 - Truly interactive web applications
 - XHTML2/HTML5 + OO Javascript (widget libraries?)
 - Combine with Mozilla Prism/Apple Fluid for zero-deploy "apps"





References

- **Comet**
 - Comet Daily (<http://cometdaily.com>)
 - Bayeux Spec (<http://svn.xantus.org/shortbus/trunk/bayeux/bayeux.html>)
 - Ajaxian (<http://ajaxian.com/by/topic/comet/>)
- **Dojo**
 - Home Page (<http://dojotoolkit.org/>)
 - Alex Russell's blog (<http://alex.dojotoolkit.org>)
- **DWR**
 - Home Page (<http://getahead.org/dwr>)
 - Demo (<http://www.ibm.com/developerworks/web/library/j-jettydwr/>)
- **Jetty**
 - Home Page (<http://www.mortbay.org/jetty-6/>)
 - Demo bundled in Jetty6
- **Grizzly**
 - Home Page (<https://grizzly.dev.java.net/>)
 - Glassfish Home Page (<https://glassfish.dev.java.net/>)
 - Jean-Francois Arcand's blog (<http://weblogs.java.net/blog/jfarcand/>)
 - Demo was from Jean-Francois
- **Lightstreamer**
 - Home Page (<http://www.lightstreamer.com/>)
 - Demo (<http://cometdaily.com/2008/02/12/%E2%80%9Chello-world%E2%80%9D-with-lightstreamer/>)
- **Scala**
 - Home Page (<http://www.scala-lang.org/>)
 - Lift Home Page (http://liftweb.net/index.php/Main_Page)
 - Demo (<http://scala-blogs.org/2007/12/dynamic-web-applications-with-lift-and.html>)



Q&A / Contact Info



- Brian Repko
 - brian.repko@learnthinkcode.com
 - +1 612 229 6779



Copyright 2008 LearnThinkCode, Inc.

This work is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported License