Project Atmosphere: An Ajax Push/Comet Framework

Jeanfrancois Arcand
Senior Staff Engineer
Sun Microsystems
Welcome to Atmosphere

This session provides a detailed introduction to Comet, explaining the underlying protocols and APIs, the challenges for application servers, and then describe the Atmosphere goals with examples.
Speaker’s qualifications

Jeanfrancois is a senior staff engineer working for Sun Microsystems since 2000

Jeanfrancois writes technical articles for CometDaily and JavaLobby and do the monkey @ JavaOne, FISL, JavaPolis, AjaxWorld, ApacheCon, etc.

Agenda

- Introduction to AjaxPush/Comet
- Demos
  - Introduction to Atmosphereå
  - Goals
  - Overall architecture
  - Examples
- Conclusion
Poll, Long Poll and Streaming

Ajax (Polling)

Browser -> Server
request  response
request  response

Ajax Push (Long Poll)

Browser -> Server
request  response
request  event
response

Ajax Push (Streaming)

Browser -> Server
request  event
request  response part
response part
response part
Poll vs Push

• Poll:
  • Send a request to the server every X seconds.
  • The response is “empty” if there is no update.

• Long Poll:
  • Send a request to the server, wait for an event to happen, then send the response.
  • The response is never empty.
  • HTTP specification satisfied: indistinguishable from “slow” server

• Http Streaming:
  • Send a request, wait for events, stream multi-part/chunked response, and then wait for the events.
  • The response is continually appended to.
How Push works

Keep an open connection.

- Deliver data over a previously opened connection
- Always “keep a connection open”
  - do not respond to the initiating request until event occurs
- Keep sending response forever
  - send response in multiple parts without closing the connection in between
Grizzly Comet:

- Grizzlet
- Bayeux
- Continuation Messages Bus
- Grizzly Comet Framework
- Grizzly HTTP (Sync/Async)
- Grizzly NIO Framework
Twitter running on Grizzly Comet
Chat Demo running on Tomcat
What are we looking for

- Asynchronous I/O
  - Asynchronous read and write

- Suspendible Requests
  - Suspend/resume requests/responses

- Easy to use Comet based API
  - push data from one connection to another
  - ability to aggregate/filter/transform data before the push operation

[www.devoxx.com](http://www.devoxx.com)
## Who supports What

<table>
<thead>
<tr>
<th>Container</th>
<th>Asynchronous IO</th>
<th>Suspensible Request/Response</th>
<th>Aggregating/Filtering/Pushing API</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jetty</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Tomcat</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>GlassFish</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Resin</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>WebLogic</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Servlet 3.0</td>
<td></td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
Chat Demo running on Jetty
Grizzly Comet is great, but...

- Since July 2006, the Grizzly Comet Framework and its extensions' popularity keep increasing.
  - Among the most popular framework right now.
- Comet techniques aren't standardized among Web Container
  - Grizzly Comet applications aren't portable (cannot run on Tomcat nor Jetty)
    - Tomcat AIO doesn’t work with GlassFish
- Servlet 3.0 will supports only a subset of Comet
  - No async I/O, No pushing functionalities
  - Support will takes time (Spec to be finalized)
Grizzlet Reloaded: Atmosphere

- Atmosphere is a POJO based framework using Inversion of Control (IoC) to bring Comet to the masses.

- A framework which can run on any Java based Web Server.....without having to wait for Servlet 3.0 or without the needs to learn how Comet support has been differently implemented by current Web Server supporting Comet.

- Brings all Grizzly Comet functionalities to all Web Server

- Run on top of Servlet 2.5 compliant Web Server
Unleashing the Grizzlet 2.0

- Grizzlet is a POJO based approach for writing Comet application.

- Grizzlet only runs on top of the Grizzly Comet Framework right now

- Grizzly community asked many times to have the concept ported to other Web Server

- Evolve the Grizzlet...and make it the core of Project Atmosphere.
Reuse experiences and code...from Grizzly Comet to Jersey!

- Grizzlet (Atmosphere POJO)
- Atmosphere re-usable Components library
- Jersey's Injection Provider, IoC Support, and more....
- Portable Comet Runtime (PCR) API
- Tomcat Comet
- Grizzly Comet
- Jetty Comet
- Blocking WS

User defined
Atmosphere Component
Twitsphere - Initialization

@Grizzlet ("myTwitter")

@Path("twitter")

public class TwitterGrizzlet{

    @Broadcaster(scope="session")
    Broadcaster bc;

    ....

}
@Suspend

@ResumeRule(name="org.atmosphere.streaming")

@POST

@Path("start")

public String onStart(@FormParam("message") String message){

    return("<script id='comet_" + counter++ + "'>" + "window.parent.(" + message + ");</script>"
@POST

@Path("signin")

public void onSignIn(@FormParam("signInName") String
       .signInName ){

        bc.publish(BEGIN_SCRIPT_TAG

            + toJsonp("Welcome back", name)

            + END_SCRIPT_TAG);

    }


@POST
@Path("broadcast")

public void onPost(@FormParam("update") String update){
    bc.publish(update);
}

@POST

@Path("follow")

public void onFollow(@FormParam("follower") String follower,
                     @FormParam("me") String me){

    Broacaster fbc = bc.getByName(follower);
    bc.subscribe(fbc, Broadcaster.PRIVATE);
    fbc.publish(follower + " is now following " + me,
                Broadcaster.PRIVATE);
    bc.publish(me + " is now followed by " + follower,
               Broadcaster.PRIVATE);
}

www.devoxx.com
Annotations

@Path: The path that map the request to the servlet

@Grizzlet: Some information about the Grizzlet

@POST/@GET: The http method used

@FormParam: Lookup a form parameters from the request.
Annotations

@Suspend: Suspend the response (do not commit the response)

@ResumeRule: The class that decide when the response needs to be resumed and committed. NOTE: We are thinking defining @Streaming or @StreamingGet or @LongPollGet

@Interrupted: Get notified when the remote client close a remote connection.

@Timedout: Get notified when the response was suspended without any activities

www.devoxx.com
Annotations

@Context: Get access to Servlet Specific classes like HttpServletRequest/Response

@Broadcaster: publish/subscribe injected object that can be used to push/aggregate/filter information efficiently.

bc.publish(...) // Publish Content

bc.subscribe(...) // Subscribe to another broadcaster

bc.publish(new URL(url)); // Push URL
The Portable Comet Runtime (PCR) currently support Tomcat and Grizzly Comet. Jetty pretty soon.

Exploring/Extending Jersey to support new annotations.

Project available on java.net
Call for Participation

Project page: http://atmosphere.dev.java.net

Twitter: http://twitter.com/atmosphere_java

User: user@atmosphere.dev.java.net

Dev: dev@atmosphere.dev.java.net

Blog: http://weblogs.java.net/jfarcand

We need help an contributors!!!
Conclusion

• Writing portable Comet based application is impossible right now…Servlet 3.0 will only solve part of the issues.

• Atmosphere’s goal is to bring portability, scalability and easy of use to all Comet developers, independent of the platform they are running on.
Q&A
Thanks for your attention!

http://twitter.com/atmosphere_java