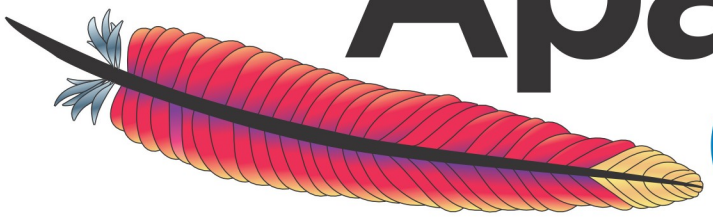


Apache Con North America 2011

Using TCnative with Comet/Asynch

Jean-Frederic Clere, Red Hat
jfclere@gmail.com, November 9th

Presented by

The Apache Software Foundation
http://www.apache.org/

Produced by

STONE CIRCLE PRODUCTIONS, LLC

What I will cover

Who I am

AJAX and Tomcat.

Comet and HTTP/1.1

Asynchronous in 3.0 Specs

NIO (NIO2)

Tomcat native APR/OpenSSL

Performances.

Questions?

My Background

Jean-Frederic Clere

- Red Hat
- Responsible of EWS product.
- Years writing JAVA code and server software

Tomcat committer since 2001

Doing OpenSource since 1999

Cyclist/Runner etc

Lived 15 years in Spain (Barcelona)

Now in Neuchâtel (CH)

Barcelona Remote

Red Hat Office Neuchâtel

AJAX

Still a lot happening:

- Dojo Foundation:
  - CometD (http://cometd.org/)
  - CometD 2: (http://cometd.org/)
- Apache camel: cometd component: (http://camel.apache.org/cometd.html)
- GWT-Comet: (http://code.google.com/p/gwt-comet/)
- WebSockets

What is it:

Asynchronous JavaScript +(and) XML

Uses HTTP:

- HTTP/1.1
- Transfer-Encoding: chunked

Asynchronous data processing

Able to request data from server and update a page in the browser

How does it work: request

Request:

POST /comet/CometServletTest1 HTTP/1.1\n

User-Agent: testclient\n

Host: localhost\n

Transfer-Encoding: chunked\n

\n

How does it work: response

Response:

HTTP/1.1 200 OK\n

Server: Apache-Coyote/1.1\n

Set-Cookie: JSESSIONID=obcoR30qlz7DMJfZmsVTt+Uv; Path=/comet\n

Transfer-Encoding: chunked\n

Date: Mon, 07 Nov 2011 22:09:33 GMT\n

\n

How does work: dialog

Each like:

Length\n

Data\n

Example

Browser to server:

13\n

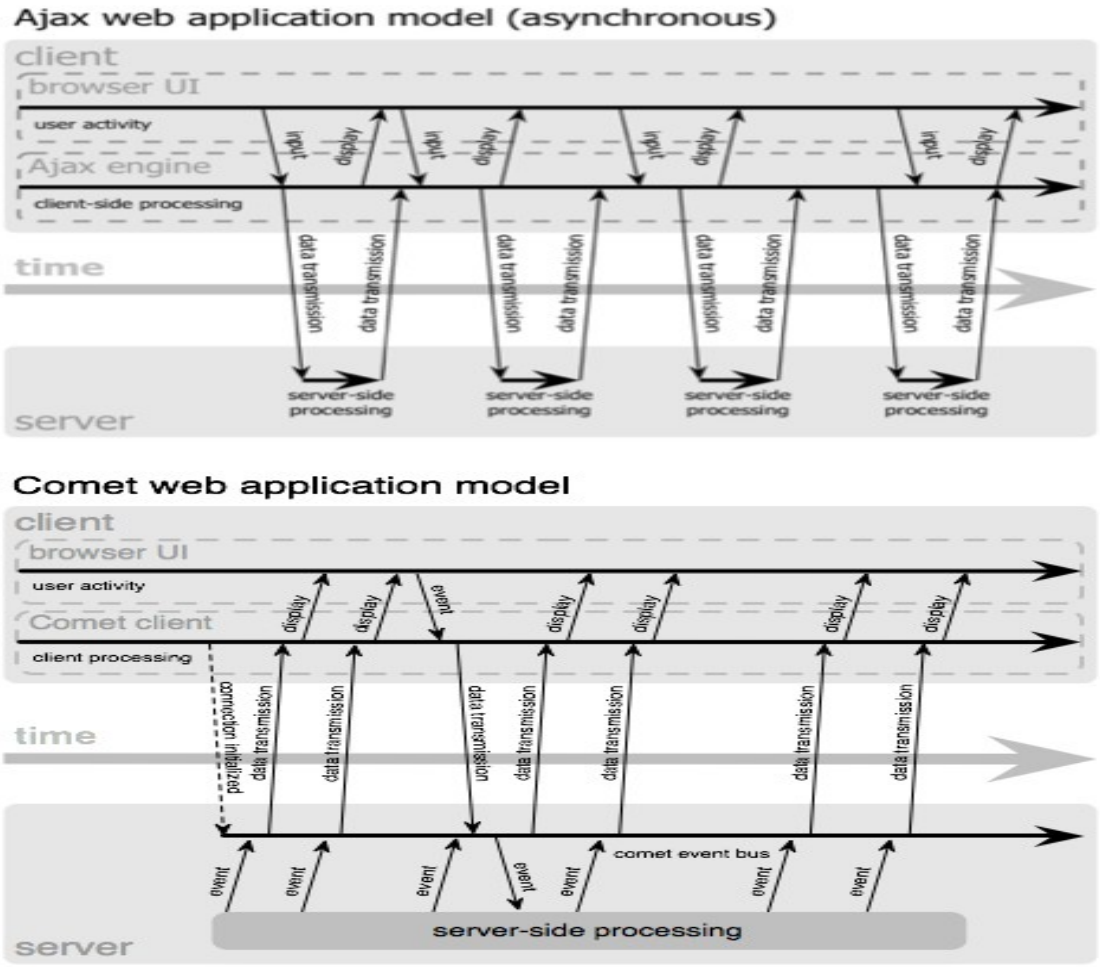
This is the request\n

Server to Browser:

19\n

and this is the response\n

Comet and HTTP/1.1



Asynchronous Servlets: JSR315

Callback in container

Reinvoke the Servlet to finish handling

Start asynchronous processing

Not non blocking IO:

- It works for http/ajp j.io, n.io, apr.

Prevent blocking servlet inside servlet code.

Don't use a thread while the request is inactive.

Code Example:

```
@WebServlet(asyncSupported=true, urlPattern=”/path")
public class TestAsyncServlet extends Httpervlet {
    public void doPost(HttpServletRequest req, HttpServletResponse res) {
        // Servlet Code
        // .....
        // Call startAsync
        AsyncContext context = request.startAsync();
        // Give AsyncContext to the Listener MonListener
       context.addListener(MonListener);
        // .....
    }
}

 public class MonListener implements AsyncListener {

        public  void onStartAsyn(AsyncEvent event) {
        }
        // Same for onError(), onTimeout() and onComplete()
     }
 }
```

Comet/Tomcat

- Open HTTP request, leave it open.
- Instant or delayed response(s), leave it open.
- Tomcat processes just like servlet
  - *CometProcessor extends HttpServlet(or should)*
  - *CometFilter extends Filter*
- Connection remains open and is writable
- Requires NIO or APR (TC-native connector):
  - org.apache.coyote.http11.Http11NioProtocol
  - org.apache.coyote.http11.Http11AprProtocol

Code Example

```
public class ChatServlet
    extends HttpServlet implements CometProcessor { ... }

public void int() { ... }
public void destroy() { ... }

public void event(CometEvent event){
if (event.getEventType() == CometEvent.EventType.BEGIN) {}
else if (event.getEventType() == CometEvent.EventType.ERROR) {}
else if (event.getEventType() == CometEvent.EventType.END) {}
else if ( event.getEventType() == CometEvent.EventType.READ) {}
}
```

NIO / NIO2 Sockets

NIO: JAVA6 protocol=org.apache.coyote.http11.Http11NioProtocol

ServerSocketChannel/SocketChannel etc

NIO2: JAVA7

- AsynchronousServerSocketChannel/Future/AsynchronousSocketChannel/CompletionHandler etc
- more easy non blocking writes
- better performances for new connector? :-(
- CompletionHandler looks to be the problem.
- Required by JSR 340 Servlet 3.1

APR

APR : Apache Portable Runtime:

- software library that provides a predictable and consistent interface to underlying platform-specific implementations.
- API on which software developers may code and be assured of predictable (identical) behavior regardless of the platform on which their software is built.

APR OpenSSL

OpenSSL:

- robust, commercial-grade, full-featured, and Open Source toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols.

- general purpose cryptography library.

Tomcat Native

JNI: Java Native Interface

- Interface C code.

Use APR:

- Portable code.

Use OpenSSL:

- high performances crypto library

Connector: (TCnative is used magically)

```
<Listener className="org.apache.catalina.core.AprLifecycleListener"
    SSLEngine="on" />
```

DEMO

Just a small chat servlet demo.

Performances...

The idea is to check the behavior under load.

Small “servlet” that returns the sessionid

6020 connections

Testing up to 30000 request/s

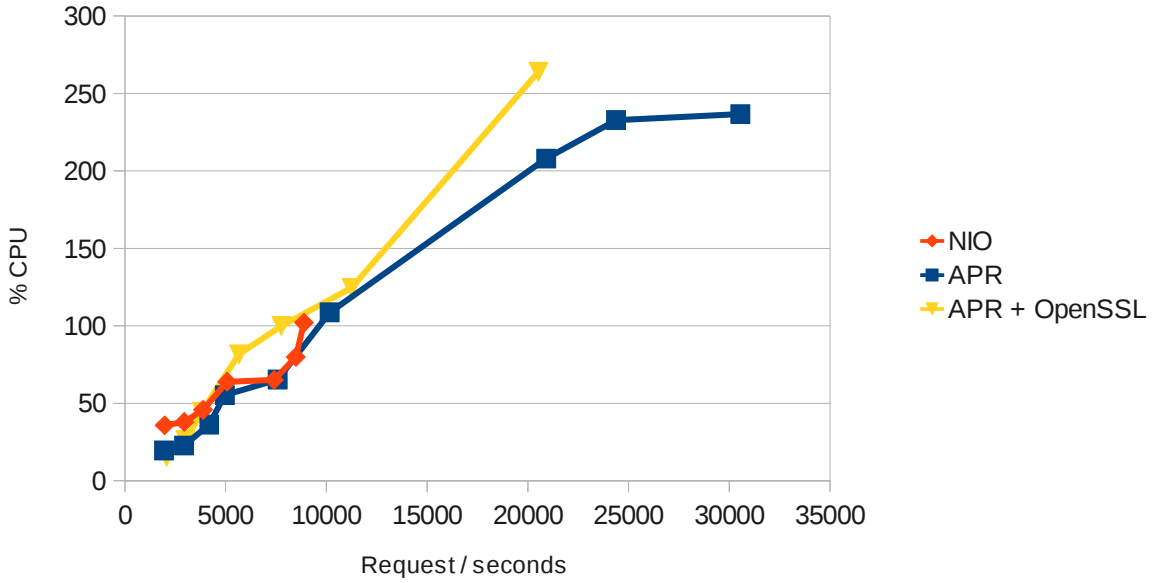
Request ~40 bytes

Response ~20 bytes

4 processors 3 GHz 2Mbytes cache

4037392 kB mem

Performances results



Next: Tomcat 8

Servlet 3.1: JSR 340

WebSocket: JSR ?

HOW:

- JAVA 7
- TC-native?

Contact

- Jean-Frederic Clere
  - jfclere@gmail.com
  - http://jfclere.blogspot.com