Ajax Push

For Revolutionary Enterprise Applications

Micha Kiener
mimacom ag

Ted Goddard, Ph.D.
ICEsoft Technologies

5420

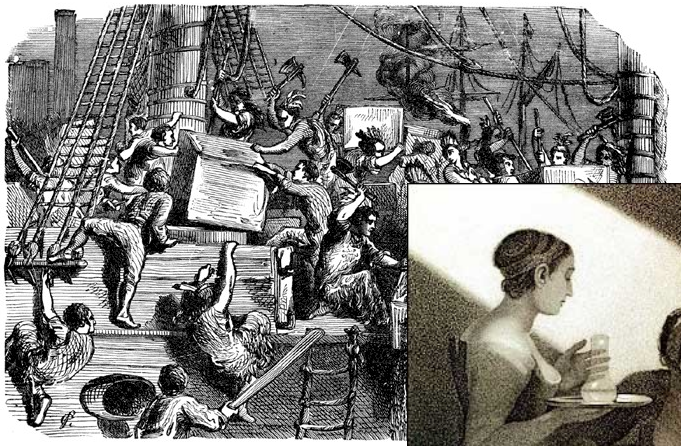
JAZOON08
THE INTERNATIONAL CONFERENCE ON JAVA TECHNOLOGY
JUNE 23 - 26, 2008 ZURICH

Agenda

- Web2.0™
- Multi-user Ajax Demo
- Push for enterprise collaboration
- Asynchronous HTTP on the Wire
- Asynchronous HTTP and the Server
- Developing Asynchronous Applications
- Conclusion

What sort of revolution?

“And yet it moves.”



American Re

Dump everythin

evolution

e Bastille

Scientific Revolution

Experimentation and Reason

Web 2.0

A Web by the people, for the people.

- Documents on the web increasingly generated by users

- Out of the Information Age, into the Participation Age
- As a whole, the World Wide Web is a collaborative environment, but individual pages are only weakly so
- Are web user interfaces becoming more powerful?
- Is the user an HTTP client?

Ajax

Ajax is a state of mind.

- It was AJAX (Asynchronous JavaScript™ Technology with XML)
  - or Asynchronous JavaScript technology with XMLHttpRequest
  - now it's Ajax (not an acronym) because many different techniques satisfied the same goals
  - coined by Jesse James Garrett in 2005 to sell an insurance company on re-writing all their software
- Is the web defined by the W3C or by browser implementers? (Ajax does not exist in W3C universe yet.)
- Ajax decouples user interface from network protocol
- Ajax is the leading edge of the user interface possible with current popular browsers
- The user experience is important

The Asynchronous Web Revolution

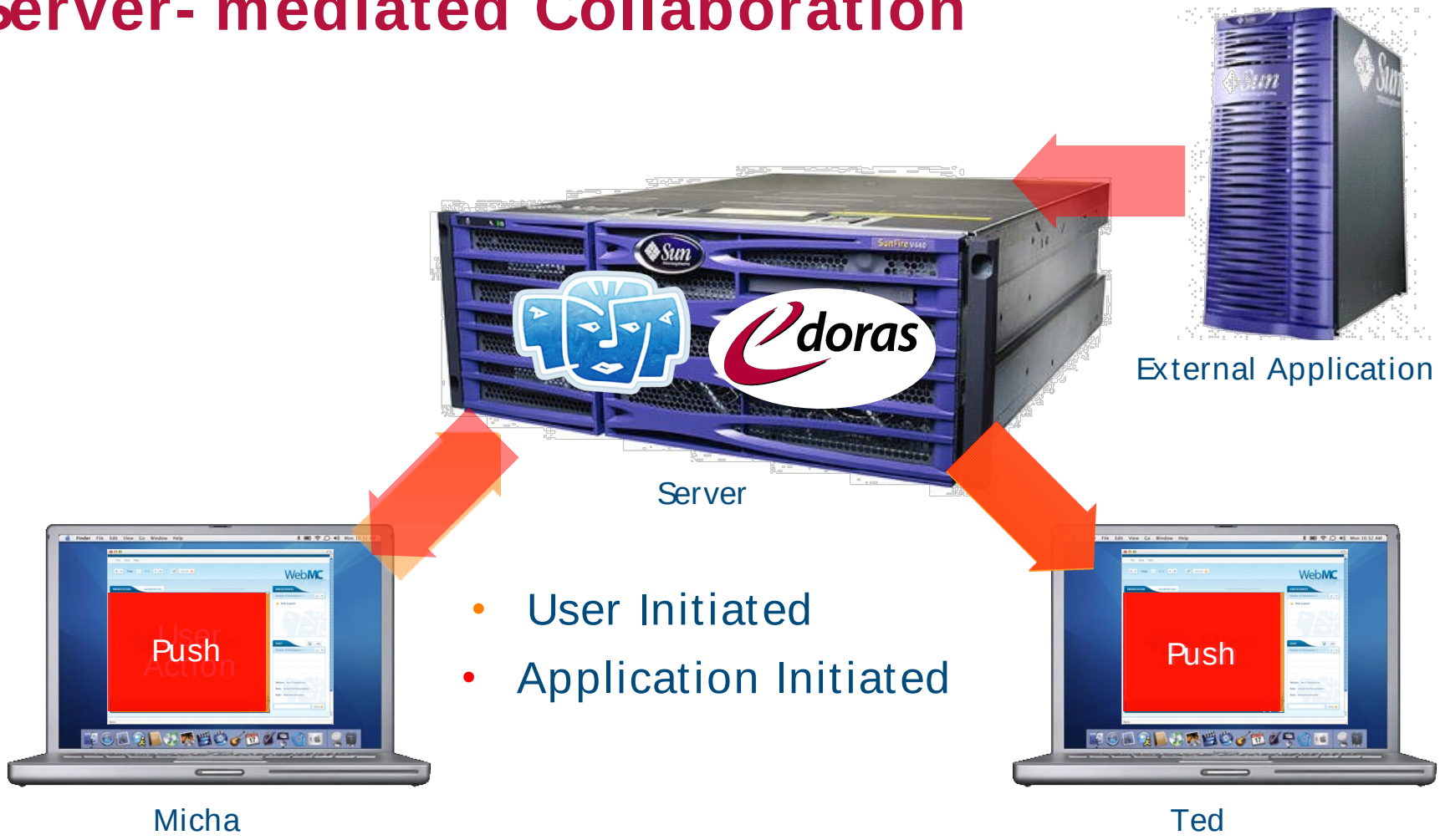
The Web enters the Participation Age.

- Ajax is still typically synchronous with user events
- Full asynchrony has updates pushed from server any time

- Update pages after they load
- Send users notifications
- Allow users to communicate and collaborate within the web application

- Called “Ajax Push”, “Comet”, or “Reverse Ajax”
  - This is the full realization of Ajax, now fully asynchronous

Server- mediated Collaboration



- User Initiated
- Application Initiated

Applications in the Participation Age

Application- mediated communication.

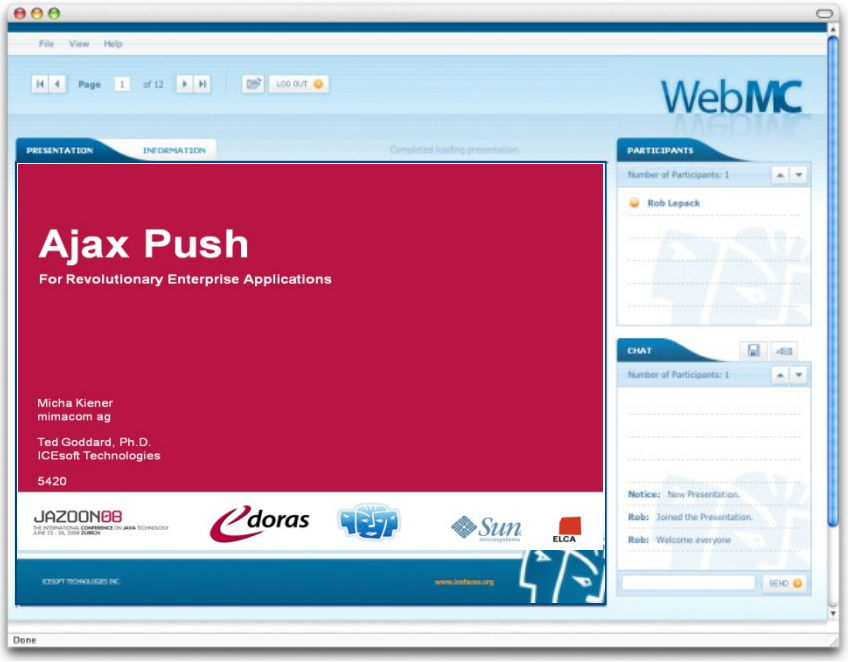
- Distance learning
- Collaborative authoring
- Auctions
- Shared WebDAV filesystem
- Blogging and reader comments
- SIP- coordinated mobile applications
- Hybrid chat/ email/ discussion forums
- Customer assistance on sales/ support pages
- Multi- step business process made collaborative
- Shared trip planner or restaurant selector with maps
- Shared calendar, “to do” list, project plan
- Games
- Enterprise shared record locking and negotiation

Agenda

- Web 2.0
- Multi-user Ajax Demo
- Push for enterprise collaboration
- Asynchronous HTTP on the Wire
- Asynchronous HTTP and the Server
- Developing Asynchronous Applications
- Conclusion

Asynchronous Ajax Demo

GlassFish/ Grizzly with ICEfaces WebMC



http://webmc.icefaces.org/

Agenda

- Web 2.0
- Multi-user Ajax Demo
- Push for enterprise collaboration
- Asynchronous HTTP on the Wire
- Asynchronous HTTP and the Server
- Developing Asynchronous Applications
- Conclusion

Didn't we have that already?

Push- mechanisms in Rich- Clients

- Rich- Clients connected to the server in a keep- alive manner
- Full Java- API is available within the client for networking and event- handling
- Server can push an event to the client any time
  - Either by having the client polling for events (optionally combined with a heart- beat, ping- like request)
  - Or by callback from the server
- Since the technology behind is well- known and transparent, its easy to use push for collaborative features and updating mechanisms
- Rich- Clients were always claimed to support push- features

Showcase for Push

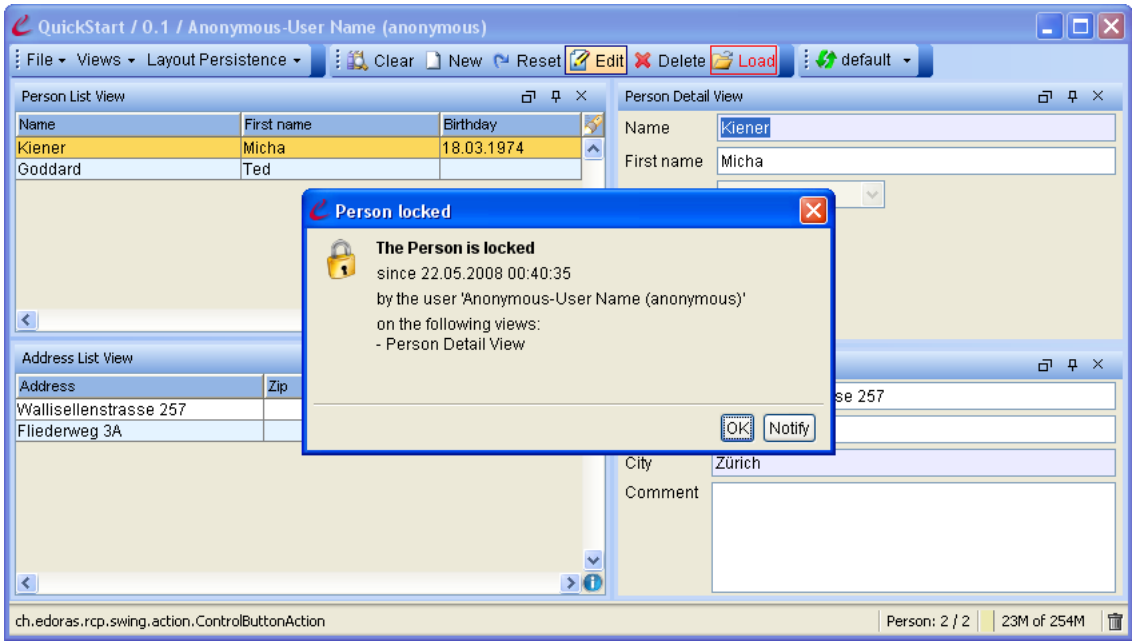
Rich- Client based demo

- Collaboration through editing and pessimistic locking
- A list of Person objects which may be edited and created
- With pessimistic locking, a lock- object must be obtained before the object is editable
- If the lock is held by another user, it should be possible to notify him so he can release the lock
- When data is changed, all views should be automatically updated

Showcase for Push

Features using Rich- Client Push or Ajax Push

- Push in a Rich- Client (Demo)
  - Automatically updating changed / added / removed data sets
  - Collaborative notifications in the context of pessimistic locking



Showcase for Push

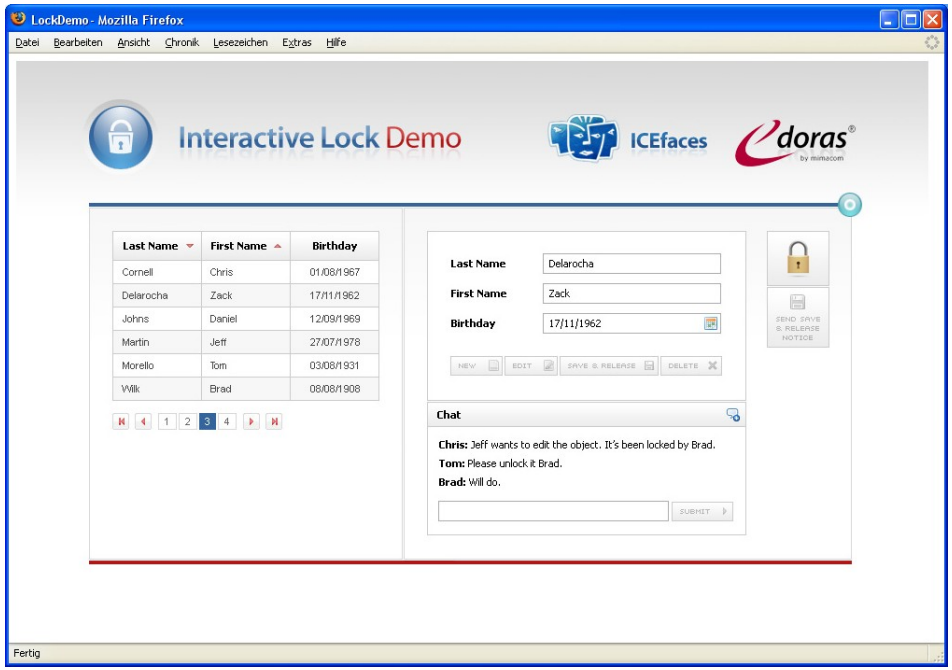
Features using Rich- Client Push or Ajax Push

- With Ajax, people stay longer on the same page, hence automatic page update is needed
- As the Web gets more and more social, collaborative tasks come in place
- Updating and collaboration are inherently asynchronous and need some push- mechanism to be fulfilled
- The next demo shows the collaborative features of the rich-client in an Ajax Push web- environment

Showcase for Push

Features using Rich- Client Push or Ajax Push

- Push in a Web- Client (Demo)
  - Automatically updating changed / added / removed data sets
  - Collaborative notifications in the context of pessimistic locking

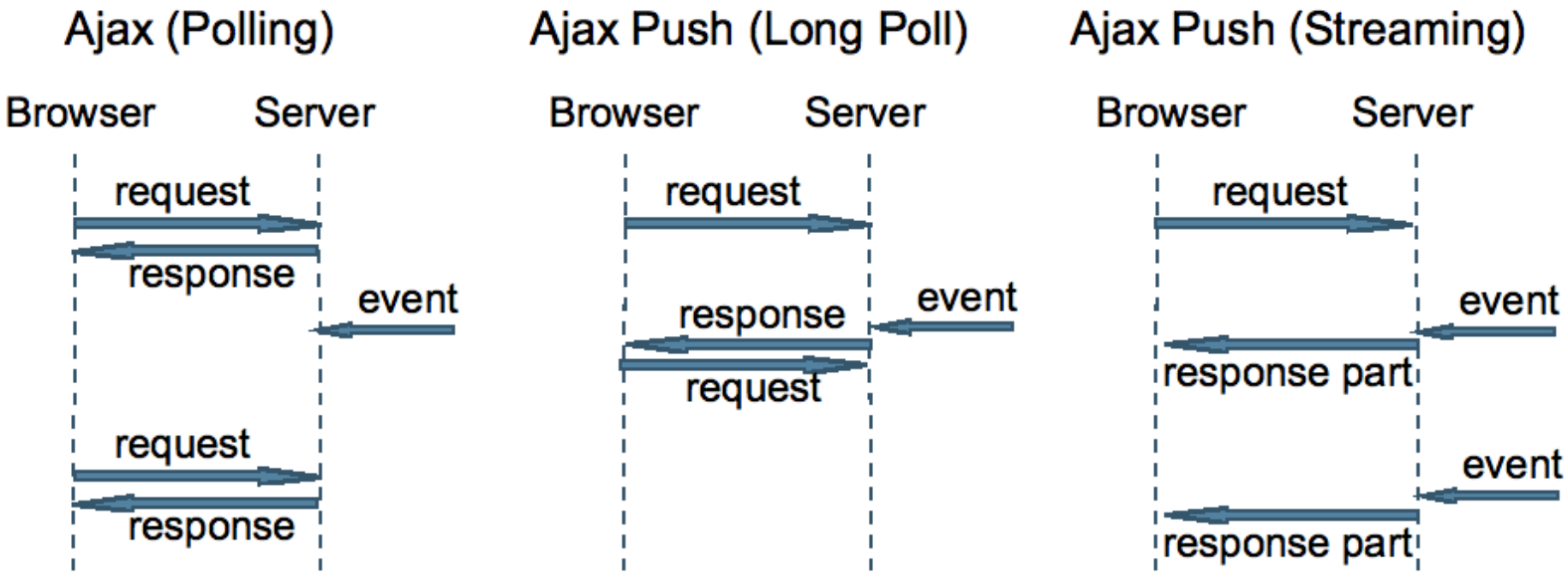


Agenda

- Web 2.0
- Multi-user Ajax Demo
- Push for enterprise collaboration
- Asynchronous HTTP on the Wire
- Asynchronous HTTP and the Server
- Developing Asynchronous Applications
- Conclusion

Ajax Poll vs Ajax Push

Bending the rules of HTTP.



Bayeux/ Cometd

JSON Pub/ Sub.

```
[
  {
    "channel": "/some/name",
    "clientId": "83js73jsh29sjd92",
    "data": { "myapp" : "specific data", value: 100 }
  }
]
```

- JSON Messages are published on specified channels
- Channel operations: connect, subscribe, unsubscribe, etc.
- Multiple transports: polling, long-polling, iframe, flash
- Server implementations: Perl, Python, Java™ programming language
- Server-side reflector with no server-side application possible

Ajax Push

HTTP message flow inversion.

```
GET /auctionMonitor/block/receive-updates?icefacesID=1209765435
HTTP/1.1
Accept: */*
Cookie: JSESSIONID=75CF2BF3E03F0F9C6D2E8EFE1A6884F4
Connection: keep-alive
Host: vorlon.ice:18080
HTTP/1.1 200 OK
Content-Type: text/xml;charset=UTF-8
Content-Length: 180
Date: Thu, 27 Apr 2006 16:45:25 GMT
Server: Apache-Coyote/1.1

<updates>
  <update address="_id0:_id5:0:chatText">
    <span id="_id0:_id5:0:chatText">Howdy</span>
  </update>
</updates>
```

Agenda

- Web 2.0
- Multi-user Ajax Demo
- Push for enterprise collaboration
- Asynchronous HTTP on the Wire
- Asynchronous HTTP and the Server
- Developing Asynchronous Applications
- Conclusion

Servlet Thread Catastrophe

Strangled by a thread for every client.

GET / updates HTTP/ 1.1

Connection: keep- alive

GET / updates HTTP/ 1.1

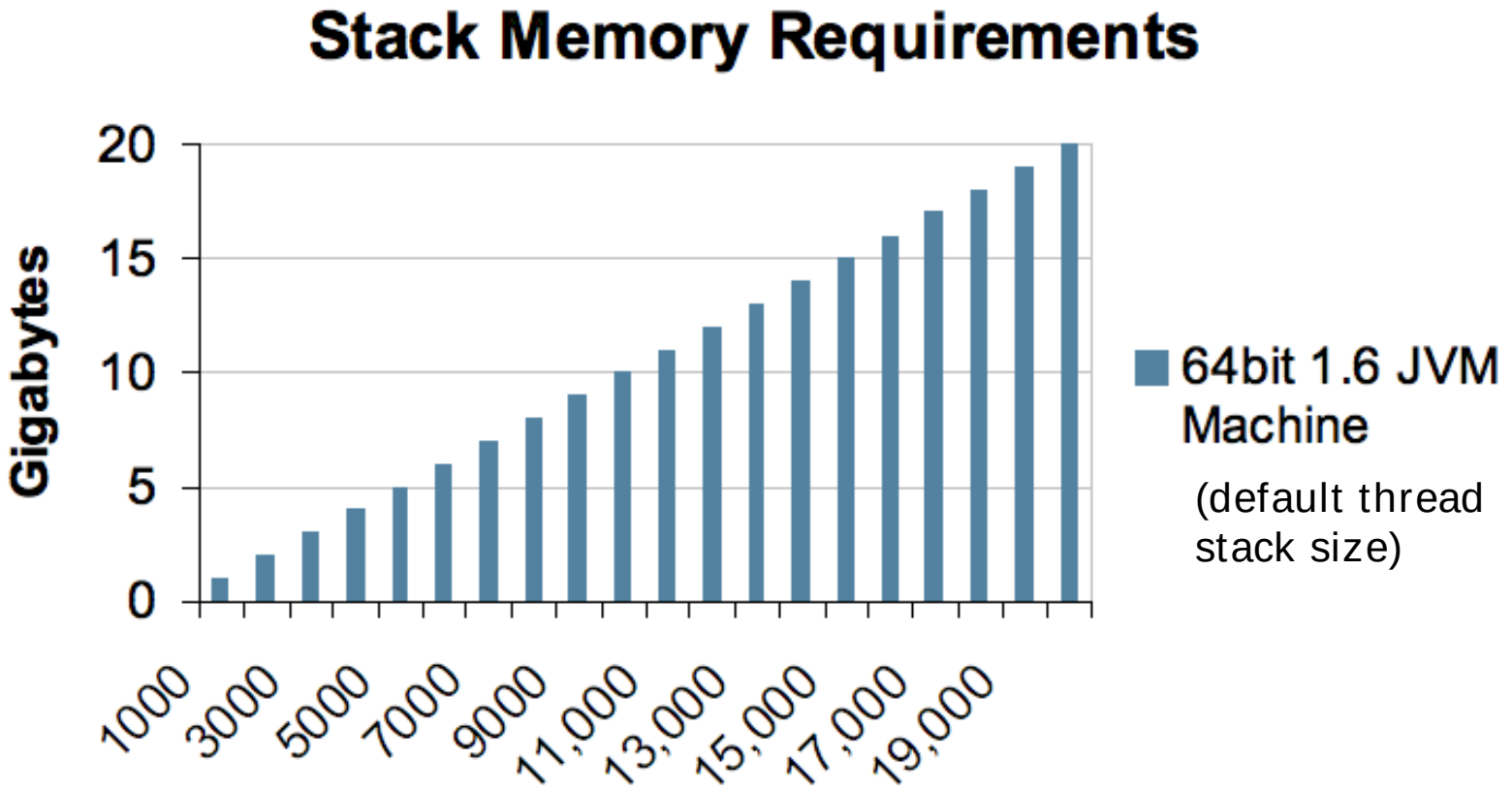
Connection: keep- alive

GET / updates HTTP/ 1.1

Connection: keep- alive

Architecture Challenges

The serious effect of blocking threads.



Server- side Ajax Push: Who supports what

The asynchronicity matrix.

Container	Asynchronous IO	Suspendible Request/ Response	Delivery Guarantee
Jetty		X	
Tomcat	X	X	
GlassFish	X	X	X
Resin		X	
WebLogic		X	

Jetty

service() will resume shortly.

```
import org.mortbay.util.ajax.Continuation;

service(request, response) {
    Continuation continuation = ContinuationSupport
        .getContinuation(request, this);
    ...
    continuation.suspend();
    response.getWriter().write(message);
}
```

Asynchronously and elsewhere in the application ...

```
message.setValue("Howdy");
continuation.resume();
```


Tomcat 6

Eventful Comet.

```
import org.apache.catalina.CometProcessor;

public class Processor implements CometProcessor {

public void event(CometEvent event)  {
    request = event.getHttpServletRequest();
    response = event.getHttpServletResponse();

    if (event.getEventType() == EventType.BEGIN) { ...
    if (event.getEventType() == EventType.READ)  { ...
    if (event.getEventType() == EventType.END)   { ...
    if (event.getEventType() == EventType.ERROR) { ...
}
```

Asynchronously and elsewhere in the application ...

```
message.setValue("Howdy");
response.getWriter().write(message);
event.close();
```

GlassFish

Suspend with Grizzly.

```
CometContext context =
    CometEngine.getEngine().register(contextPath);
context.setExpirationDelay(20 * 1000);

SuspendableHandler handler = new SuspendableHandler();
handler.attach(response);
cometContext.addCometHandler(handler);

class SuspendableHandler implements CometHandler  {

    public void onEvent(CometEvent event) {
       response.getWriter().println(event.attachment());
       cometContext.resumeCometHandler(this);
    }
```

Asynchronously and elsewhere in the application ...

```
message.setValue("Howdy");
cometContext.notify(message);
```


Servlet 3.0

Future Asynchronous Standard.

- Defined by JSR-315 Expert Group
- DWR, Jetty, Tomcat, GlassFish project, and ICEfaces participants
- Standard asynchronous processing API being defined
  - Asynchronous I/O
  - Suspendible requests
  - Delivery guarantee not included

- Will improve portability of DWR, Cometd, and ICEfaces
- (But unless you write Servlets today, this API will be hidden by your chosen Ajax framework.)

Agenda

- Web 2.0
- Multi-user Ajax Demo
- Push for enterprise collaboration
- Asynchronous HTTP on the Wire
- Asynchronous HTTP and the Server
- Developing Asynchronous Applications
- Conclusion

JavaScript Polling

Are we there yet? Are we there yet? Are we there yet? …

```
function poll()  {
  setTimeout('poll()', 10000);
  req = new XMLHttpRequest();
  req.onreadystatechange = update();
  req.open("POST", "http://server/getMessage.jsp");
}

function update() {
  chatLog.innerHTML = req.responseText;
}

poll();
```

Cometd

Distributed, loosely coupled, scripting

JavaScript

```
function update(message) {
  chatLog.innerHTML = message.data.value;
}
...
  cometd.subscribe("chat", remoteTopics, "update")
  cometd.publish("chat", message)
```

Java

```
import dojox.cometd.*;

Channel channel = Bayeux.getChannel("chat", create);
channel.subscribe(client);
```

Asynchronously and elsewhere in the application ...

```
message.setValue("Howdy");
channel.publish(client, message, "chat text");
```

DWR

JavaScript RPC

```
import org.directwebremoting.proxy.dwr.Util;

scriptSessions =
    webContext.getScriptSessionsByPage(currentPage);
    util = new Util(scriptSessions);
```

To “Reverse Ajax” and invoke arbitrary JavaScript:

```
util.addScript(ScriptBuffer script);
```

Asynchronously and elsewhere in the application ...

```
util.setValue(“form:chat:_id3”, “Howdy”);
```


ICEfaces

Preserve MVC with Transparent Ajax.

PageBean.java

```
public class PageBean {
   String message;

   public String getMessage() {
      return message;
   }

   public void setMessage(String
message) {
      this.message = message;
   }

}
```

Presentation Model

Page.xhtml

```
<f:view
   xmlns:f="http://java.sun.com/jsf/core"
   xmlns:h="http://java.sun.com/jsf/html" >

  <html>
    <body>
      <h:form>
        <h:inputText
           value="#{pageBean.message}" />
      </h:form>
    </body>
  </html>

</f:view>
```

Declarative User Interface

A language for Ajax Push that preserves Designer and Developer roles

ICEfaces

High level push.

```
import org.icefaces.application.SessionRenderer;
```

To update all users in the application:

```
SessionRenderer.render(SessionRenderer.ALL_SESSIONS);
```

Or to keep track of groups of users:

```
SessionRenderer.addCurrentSession("chat");
```

Asynchronously and elsewhere in the application ...

```
message.setValue("Howdy");
SessionRenderer.render("chat");
```

The JSF lifecycle runs and each user's page is updated from the component tree and current model state.

Agenda

- Web 2.0
- Multi-user Ajax Demo
- Push for enterprise collaboration
- Asynchronous HTTP on the Wire
- Asynchronous HTTP and the Server
- Conclusion

Summary

The Asynchronous Web Revolution is Now

- The Asynchronous Web will revolutionize human interaction
- Ajax Push is the key to enterprise collaboration for the Web
- Push can scale with Asynchronous Request Processing
- ICEfaces and edoras provide the high-level capabilities for enterprise collaboration features in your application
- Visit us at the mimacom stand in the exhibition to talk with the experts and see more demos

Thank you!

Questions?

Micha Kiener
mimacom ag
http://www.mimacom.ch
micha.kiener@mimacom.ch

Ted Goddard
ICEsoft Technologies
http://www.icefaces.org
ted.goddard@icesoft.com