Deploying Ajax Comet Applications

war stories from the trenches of

Presented at The Ajax Experience Boston 2008

by Greg Wilkins

CTO Webtide.com

Developer Jetty & Cometd

webtide

So! you have a web-2.0 app!
It has round corners!

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Curabitur mi est, cursus sit amet, pellentesque et, ultricies a, ipsum. Nulla facilisi. Sed quis lacus. Aenean ut risus et lectus blandit gravida. Nam sed nunc. Aliquam non felis non diam aliquam gravida. Phasellus quis sem. Curabitur at velit. Vivamus libero velit, condimentum sit amet, tempus ut, aliquam sit amet, velit. Nunc hendrerit ante. Quisque egestas feugiat erat. Morbi tellus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Nulla ac ante sit amet metus hendrerit euismod. Aenean vestibulum, lectus in eleifend tempor, quam libero iaculis dolor, pellentesque pellentesque lorem nibh ut urna. Nulla rhoncus, ante sit amet tristique interdum, eros nulla nonummy justo, eu dapibus risus quam sit amet metus. Maecenas tristique augue vel enim. Duis blandit euismod pede. Pellentesque facilisis. Fusce dapibus sapien tristique massa. Nunc accumsan. Integer pretium risus et odio. Phasellus tincidunt rhoncus velit. Donec eu neque at massa mollis iaculis. Aliquam pellentesque auctor mi. Cras ante justo, ultricies quis, iaculis eu, tincidunt ac, velit. Etiam nunc erat, tincidunt sit amet, luctus quis, interdum a, nunc. Suspendisse elit.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Nulla ac ante sit amet metus hendrerit euismod. Aenean vestibulum, lectus in eleifend tempor, quam libero iaculis dolor, pellentesque pellentesque lorem nibh ut urna. Nulla rhoncus, ante sit amet tristique interdum, eros nulla nonummy justo, eu dapibus risus quam sit amet metus. Maecenas tristique augue vel enim. Duis blandit euismod pede. Pellentesque facilisis. Fusce dapibus sapien tristique massa. Nunc accumsan. Integer pretium risus et odio. Phasellus tincidunt rhoncus velit. Donec eu neque at massa mollis iaculis. Aliquam pellentesque auctor mi. Cras ante justo, ultricies quis, iaculis eu, tincidunt ac, velit. Etiam nunc erat, tincidunt sit amet, luctus quis, interdum a, nunc. Suspendisse elit.

and DHTML effects

Useful ...

... or otherwise!

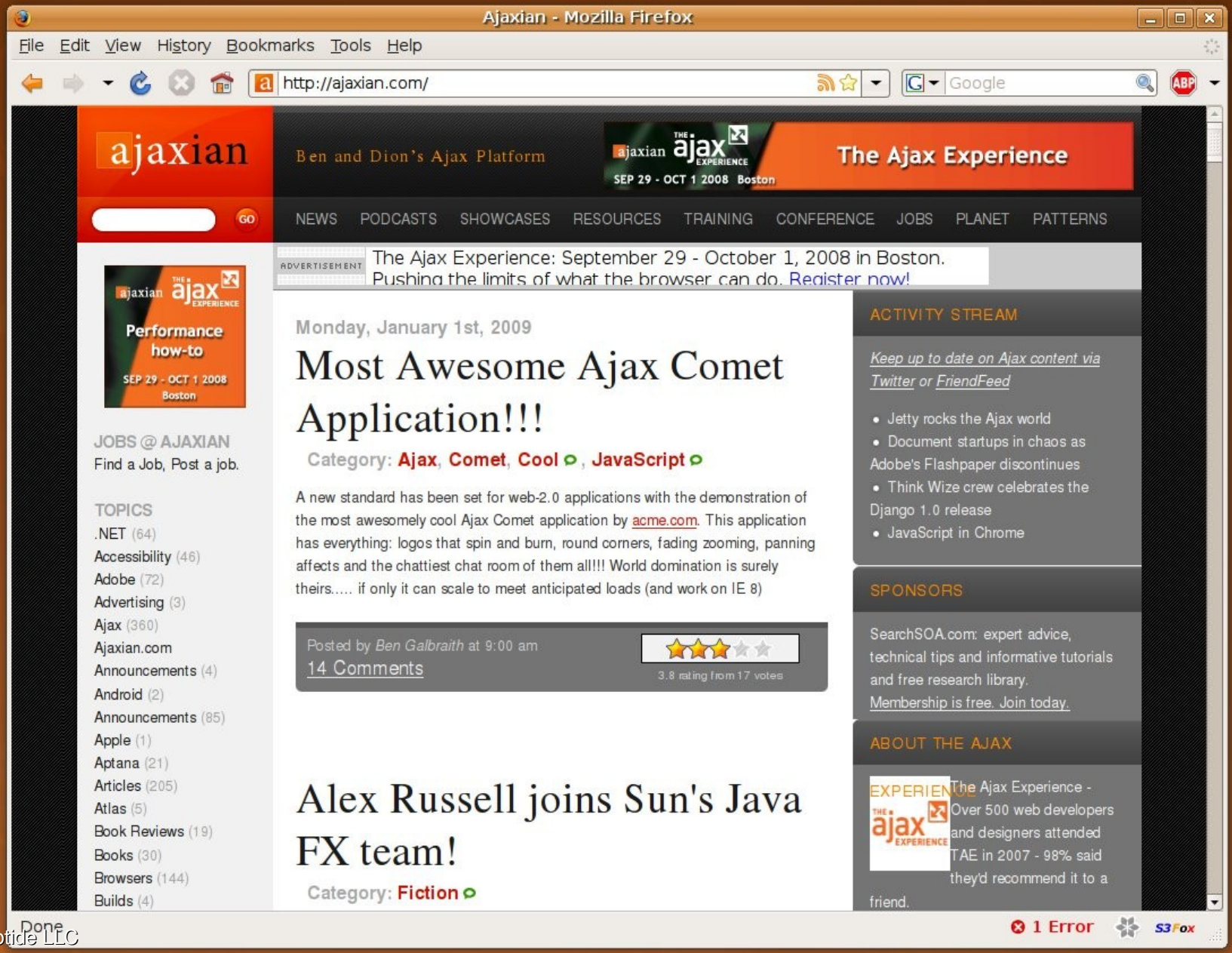

and Comet social networking (chat)!
ARMY
BIG
BELLS
copyright 2008 Webtide LLC

Customers are demanding it!

Management is keen!

Image from Flikr: Donna Cymek

Ajaxian are pimping it!



But it's only a prototype!!!

Will it work at Commercial Scale?

Image from Flikr: morbin

Anecdotes from:

- Completing the Application

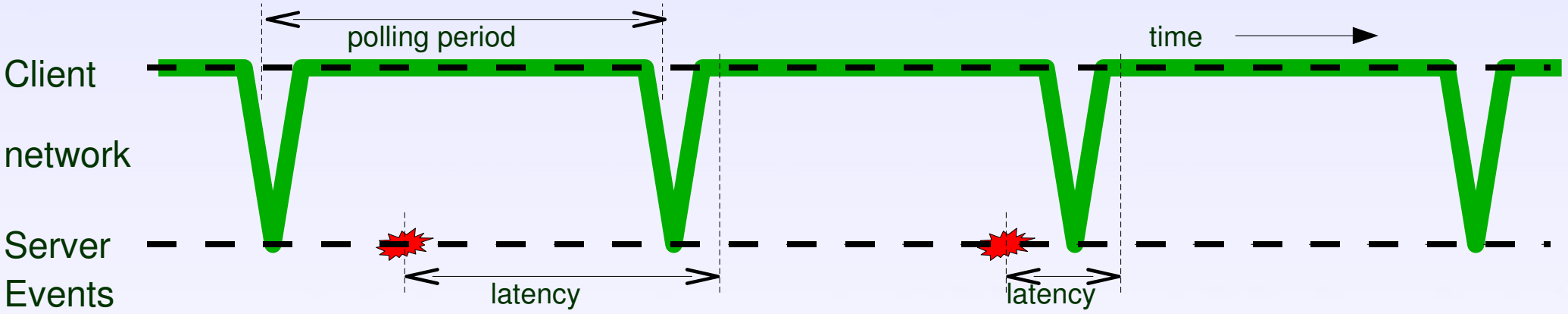
- Deploying the Application

- Production Issues!!

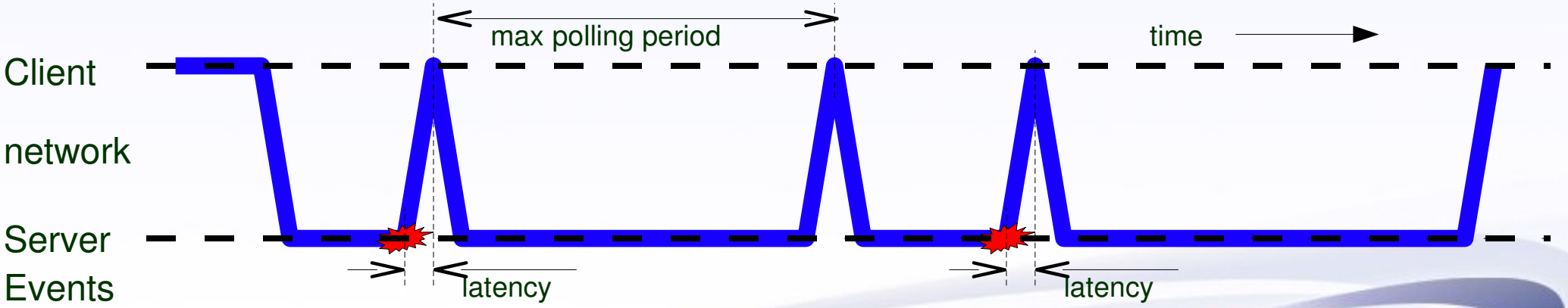
webtide

Comet Refresher

- **Polling**



- **Long Polling**



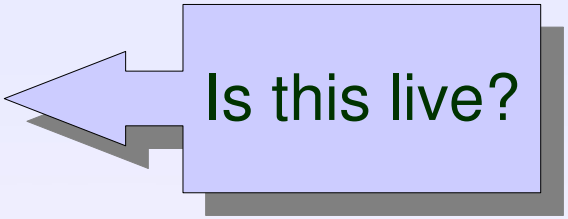
Completing the Application

From prototype to production

Thing to do to complete your comet application

Application Design: Power to the People

Exchange Rates		
Currency	**Value**	**Change**
AUD / USD	0.79215	0.08
AUD / NZD	1.19205	-0.47
AUD / JPY (yen)	82.715	-0.06
AUD / GBP	0.43475	-0.1
AUD / EUR	0.5515	-0.18



- **Users need to know its on**
  - Don't want users waiting for an event that never comes
  - Don't want users clicking reload due to loss of confidence
- **User must be able to turn it off**
  - May wish to view portfolio, but not have it update
  - Specially important for mobile

Toolkit Capabilities & Compatibilities

- **Toolkits selected for 1 capability (eg cometd)**
  - **May not have all the capabilities required**
- **Fx Trading Application**
  - Ext.js selected for visual appeal
  - Dojox cometd selected for cometd
  - Load times and complexity unacceptable
  - **Ported UI to Dojo dijit**
  - **Combination works (almost) for chess.com**
- **System Monitoring Application**
  - GWT selected for developer environment (other projects)
  - Dojo selected for cometd
  - Emprize charting
  - **Integrated OK, but complex**

Work Well & Fail Nice

- Making your comet application work is easy
  - Plenty of examples
  - Focus of frameworks
- Making your application fail nice is hard
  - Handle network failures
  - Browser incompatibilities

Handling Comet Failure

- Comet Libraries can handle some failures
  - Retries after transient comms failures
  - Cometd does retry with configurable backoff
- Application must handle some transients
  - Restore state after server (or load balancer) restart
- Application must handle persistent failures
  - Alert user that connection to server has been lost
- Comet provides new failure handling options
  - State can be preserved in the client over server migration!
  - Jetty has a comet based distributed session manager!!!

webtide

Dojox Cometd Error Handling

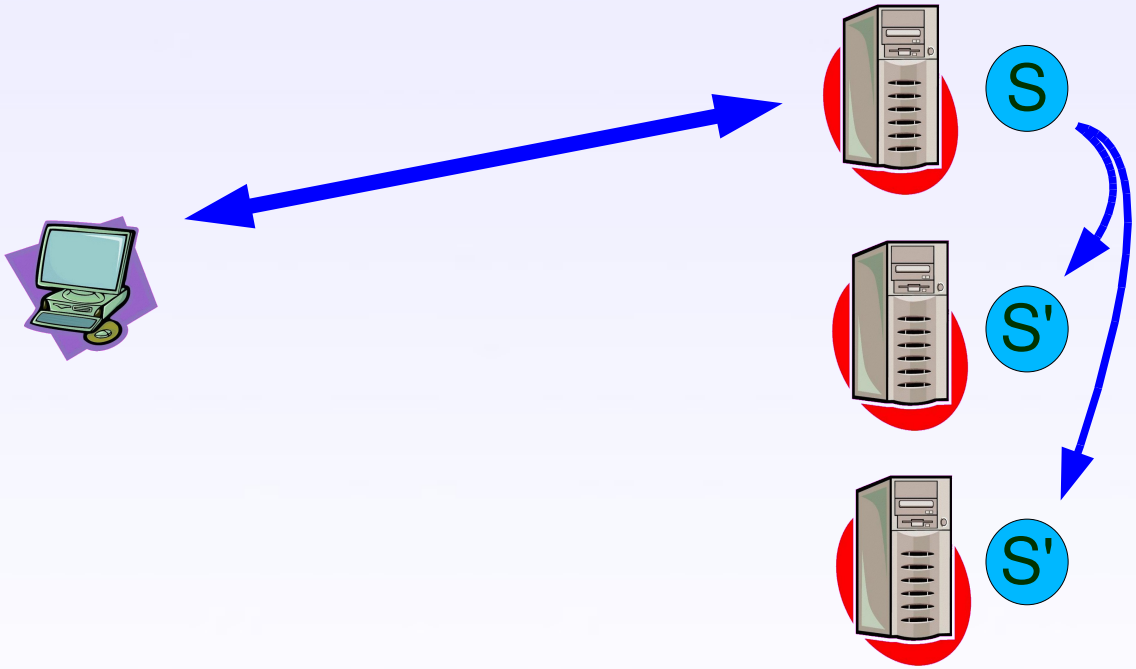
```
room._meta = dojo.subscribe("/cometd/meta", this, function(e){
  if (e.action == "handshake") {
    if (e.reestablish) {
      if (e.successful) {
        dojox.cometd.subscribe("/chat/demo", room, "_chat");
        dojox.cometd.publish("/chat/demo", {
          user: room._username, join: true,
          chat: room._username + " has re-joined"
        });
      }
    }
  }
  else if (e.action == "connect") {
    if (e.successful && !this._connected) {
       room._chat({ data: {
          join: true, user: "SERVER",
          chat: "reconnected!"
        }
      });
    }
    else if (!e.successful && this._connected) {
      room._chat({ data: {
          leave: true, user: "SERVER",
          chat: "disconnected!"
          }
        });
    }
    this._connected = e.successful;
  }
});
```


Comet Session Manager (off topic!)

- **Sessions are clusters tough**
  - **Sessions must be replicated**
    - Node to node
    - Common repository
- **Need somewhere to store session that**
  - **Is not a central point of failure**
  - **Is distributed and scalable**
  - **Low communication overheads**
  - **Wont fail and lose client session**
- **What about the Browser?**
  - **Use comet to store opaque session bundle on browser!**

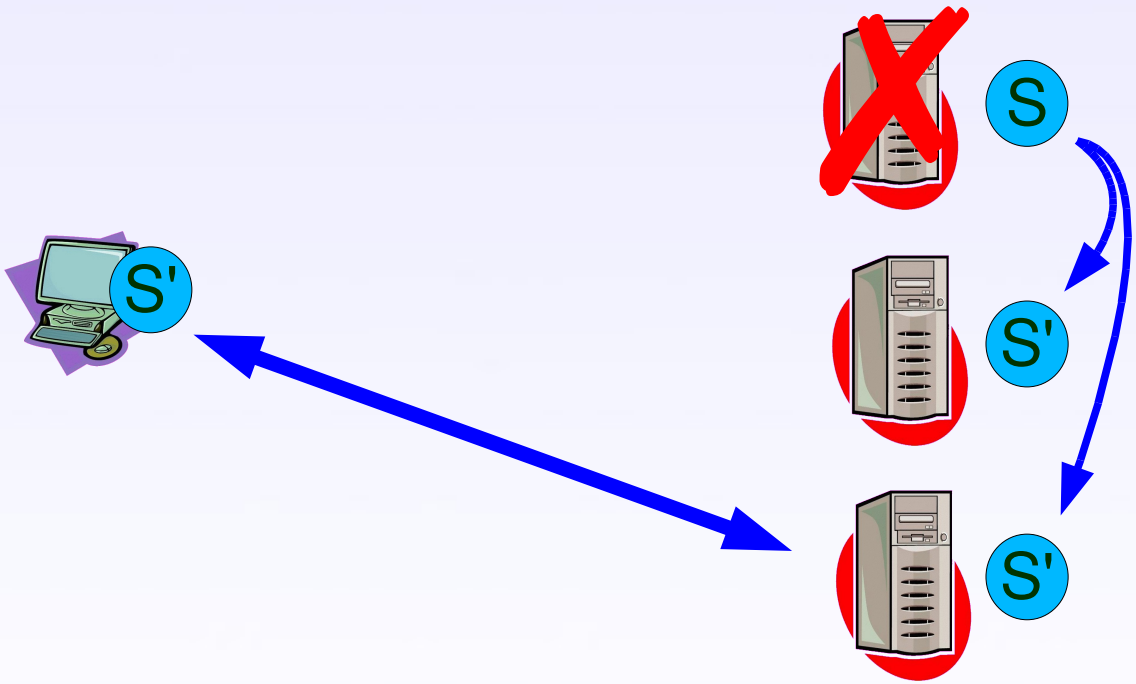
Comet Session Manager (off topic!)

- **Clustered Session Replication**
  - Server replicates state to other nodes



Comet Session Manager (off topic!)

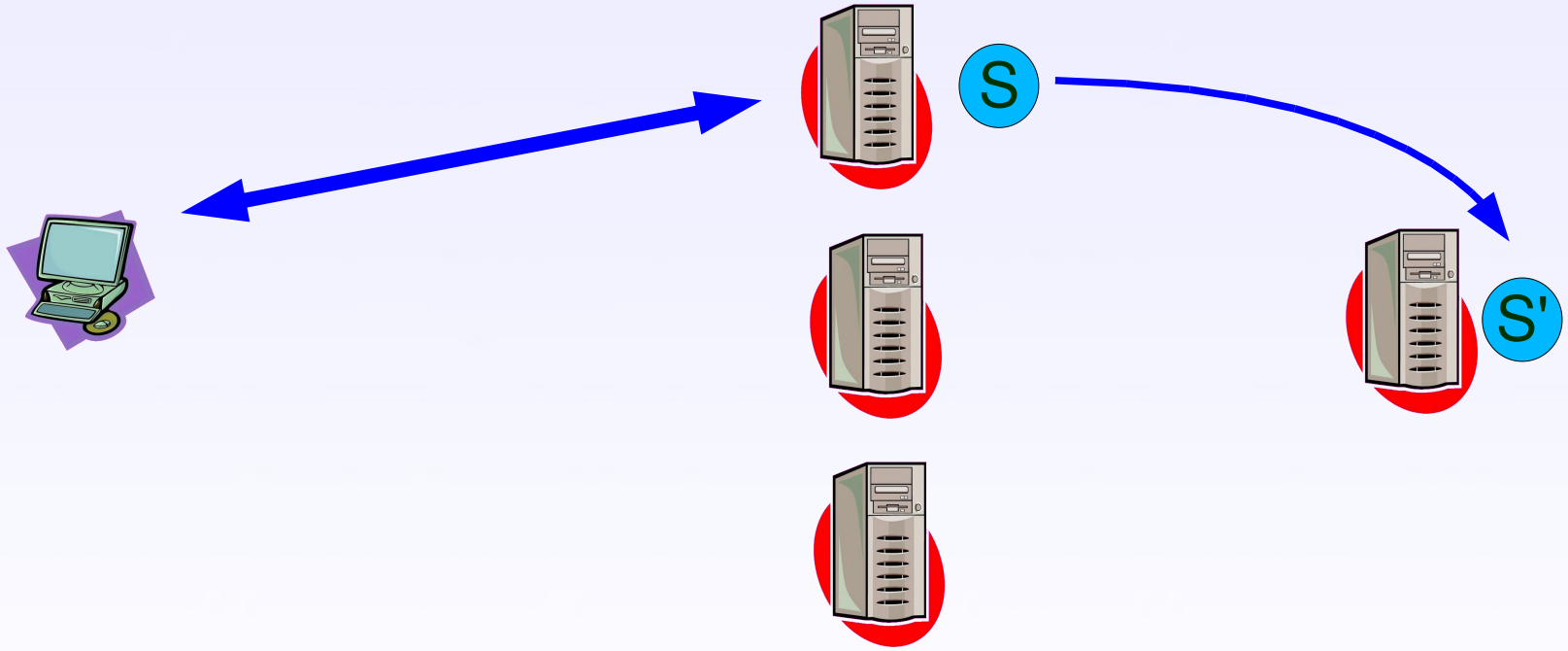
- **Clustered Session Replication**
  - Other nodes have copy of state



  - Poor scalability

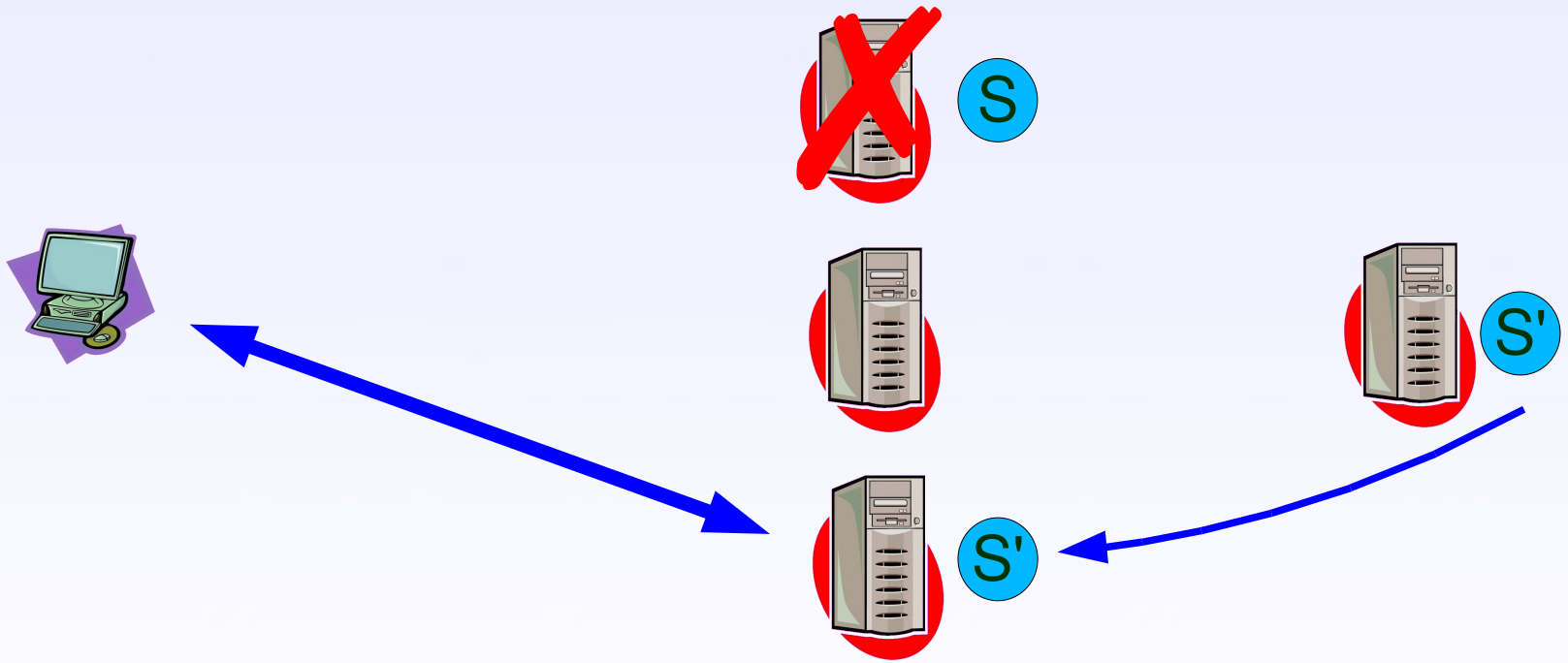
Comet Session Manager (off topic!)

- **Clustered Session Store (eg Terracotta)**
  - Server replicates state to central service



Comet Session Manager (off topic!)

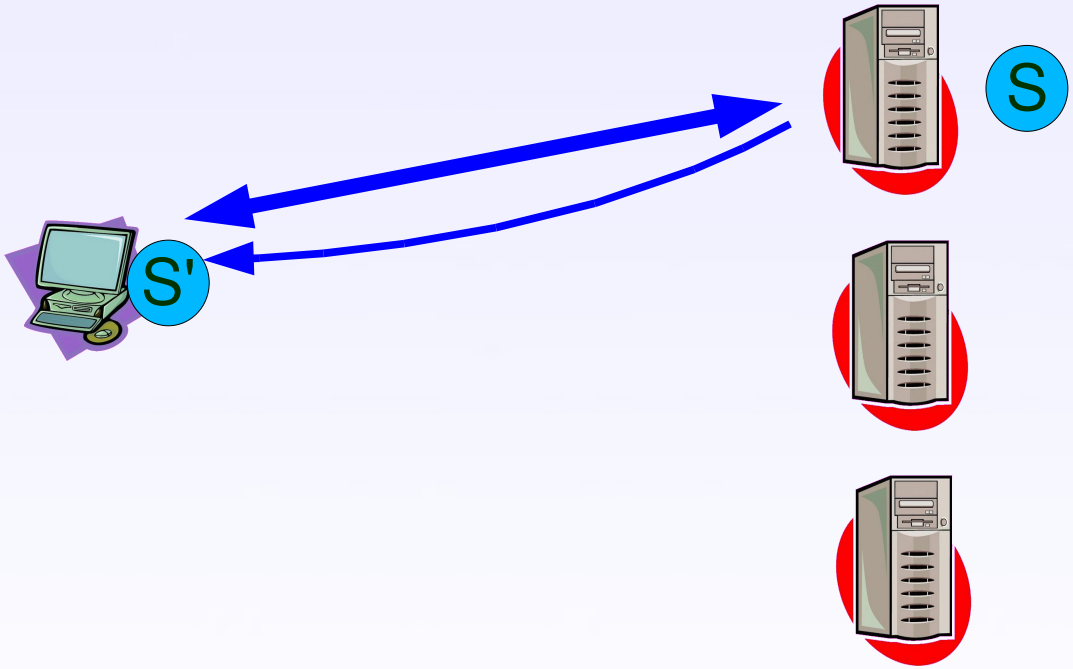
- **Clustered Session Store (eg Terracotta)**
  - Server can recover state from central service



  - Central service can be point of failure and difficult to scale

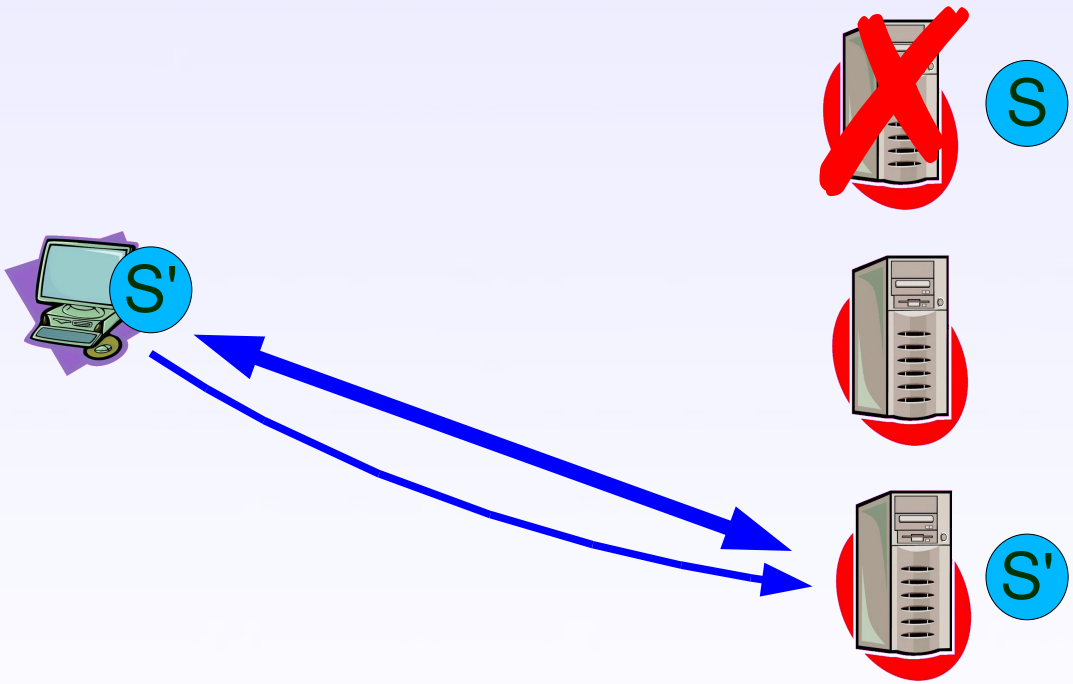
Comet Session Manager (off topic!)

- **Cometd session store. ON CLIENT!**
  - **Server can replicate state to client using comet**



Comet Session Manager (off topic!)

- **Cometd session store. ON CLIENT!**
  - Server can recover state from client using comet



  - As scalable as comet!

Security – Authentication Policy

- **Comet frameworks often are permissive**
  - **Vulnerable to denial of service:**
    - Eg. bad client creating too many channels
    - Eg. bad client sending too many messages
  - **Vulnerable to data injection**
    - Eg. don't want users publishing prices to a stock ticker

- **Cometd has a pluggable security policy**

```
package org.cometd;
public interface SecurityPolicy
{
    boolean canHandshake(Message message);
    boolean canCreate(Client client,String channel,Message message);
    boolean canSubscribe(Client client,String channel,Message messsage);
    boolean canPublish(Client client,String channel,Message messsage);
}
```


Security – Authentication Implementation

- **Read Only IBM Tivoli Policy**

```
public class ROTivoliPolicy implements SecurityPolicy {
  AbstractBayeux _bayeux;
  SimpleTivoliPolicy(AbstractBayeux bayeux) {_bayeux=bayeux}

  public boolean canCreate(Client client, String channel, Message message) {
    // only server side can create channels
    return client.isLocal();
  }
  public boolean canHandshake(Message message) {
     // only authenticated users can handshake
     return isTAMUser(); }
  }
  public boolean canPublish(Client client, String channel, Message messsage){
    // only server side can publish
    return client.isLocal();
  }
  public boolean canSubscribe(Client client, String channel, Message messsage{
    // server or authenticated user can subscribe
    return client.isLocal() || isTAMUser();
  }

  private boolean isTAMUser() {
    HttpServletRequest request = _bayeux.getCurrentRequest();
    return request.getUserPrincipal()!=null &&
           request.getUserPrincipal().toString()
           .equals(request.getHeader("iv-user"))
  }
}
```


Security - Validation

- **Ajax increases the attack surface**
  - **More application code exposed to user data**
    - Any POJO can subscribe to comet messages!
- **Frameworks and Applications must:**
  - **Validate data**
  - **Scan for injected scripts, sql, html etc.**
- **Cometd-jetty provides DataFilters**

```
package org.cometd;
public interface DataFilter
{
    Object filter(Client from, Channel to, Object data)
       throws IllegalStateException;
}
```

  - **Validation may be applied after development!!!**
  - **Easy to audit!!!**

Security - Validation

- Example filter.json configuration

```
[
   {
      "channels": "/**",
      "filter"  : "org.mortbay.cometd.filter.NoMarkupFilter",
      "init"    : {}
   },

   {
      "channels": "/chat/*",
      "filter"   : "org.mortbay.cometd.filter.RegexFilter",
      "init"    : [
                  [ "[fF].ck","dang" ],
                  [ "teh ","the "]
               ]
   },

   {
      "channels": "/chat/**",
      "filter"   : "org.mortbay.cometd.filter.RegexFilter",
      "init"    : [
                  [ "[Mm]icrosoft", "Micro\\$oft" ],
                  [ ".*IE.*", null ]
               ]
   }
]
```


Security

- **Don't forget SSL**
  - **Is your async layer available in SSL?**
    - Jetty provides async SSL connector
  - **Does your SSL offloader handle long polling?**

- **Paranoid applications**
  - **Cometd can leak information**
    - Eg Stock portfolio
      - *An observer can use timing of encrypted packets to infer watch list*

Deploying the Application

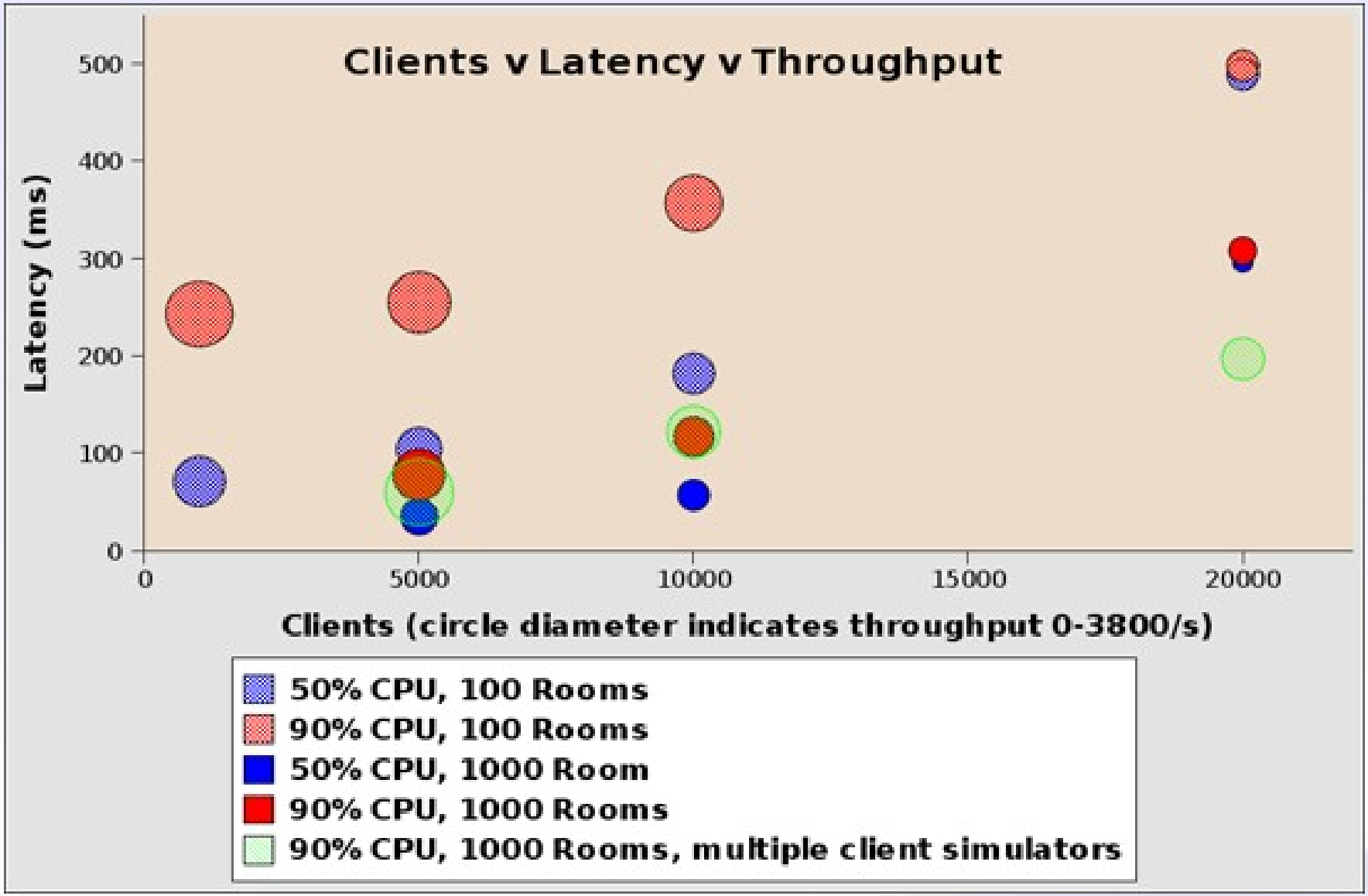
- Security
- Tuning
- Clustering
- Logging

Will it Scale?

- Many Cometd techniques allocate:
  - Thread per client (for long poll)
  - Buffers for streaming out (for forever-frame)
- If you want > 1000 users per node
  - Need >1000 threads, >0.5GB ram for stacks
  - Need > 125 MB IO Buffers (plus kernel buffers!)
- Jetty + Cometd/DWR/AMQ uses async techniques
  - Long polls do not need threads
  - Other scalability benefits (QoS, Asynchronous Web services)
- Make sure your comet will scale to your goals!

webtide

Cometd + Jetty Performance



Testing Times

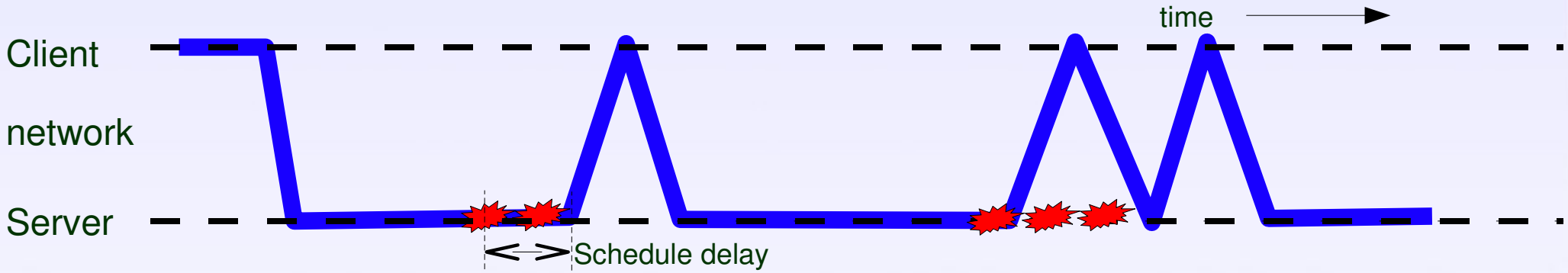
- **How to generate test load**
  - **Difficult to create a realistic load profile**
    - Consider playback of live data
  - **Difficult to simulate realistic network delays**
    - Delays, queuing, bursts can affect performance
  - **Difficult to generate simultaneous load**
    - Common test tools are blocking
- **Jetty has asynchronous test client**
  - **Can maintain 20,000 simultaneous connections!**
  - **Can generate reasonable load on 1000s of connections**
- **Cometd Clients**
  - **Javascript, java, perl**

Demo Test Client

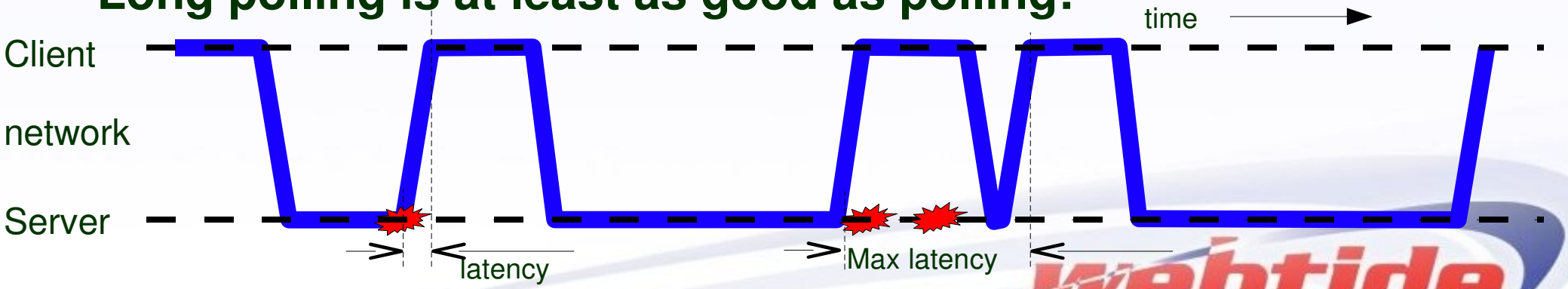
```
BayeuxClient client = new BayeuxClient(http,address,uri){
    public void deliver(Client from, Message message)
    {
        if (Bayeux.META_SUBSCRIBE.equals(message.get(Bayeux.CHANNEL_FIELD)) &&
            ((Boolean)message.get(Bayeux.SUCCESSFUL_FIELD)).booleanValue())
            _subscribed.incrementAndGet();
        super.deliver(from,message);
    }
};

MessageListener listener = new MessageListener() {
    public void deliver(Client fromClient, Client toClient, Message msg)
    {
        Object data=(Object)msg.get(AbstractBayeux.DATA_FIELD);
        if (data!=null){
            // ...
        }
    }

};
client.addListener(listener);

client.start();

client.subscribe("/chat/room");
```


Graceful Degradation under Load

- **Long polling naturally graceful**
  - **Stress leads to increased batching!**



- **Cometd has adjustable latency**
  - **Idea borrowed from DWR**
  - **Long polling is at least as good as polling!**



Graceful Degradation under Load

- **Jetty Optimization for single message**
  - **Preformated HTTP, Cometd message and Data content**

```
HTTP/1.1 200 OK
Date: Fri, 26 Sep 2008 19:56:17 GMT
Content-Type: text/json; charset=utf-8
Content-Length: 194
Server: Jetty(6.1.12.rc2)

[
 {"successful":true,
  "channel":"/meta/connect"},
 {"timestamp":"Fri, 26 Sep 2008 19:56:24 GMT",
  "channel":"/chat/demo",
  "data":{"user":"Elvis",
          "chat":"I'm leaving the building"}
 }
]
```

  - **Fantastic if only 1 message!**
    - Causes non graceful degradation if queues >= 2

Tuning

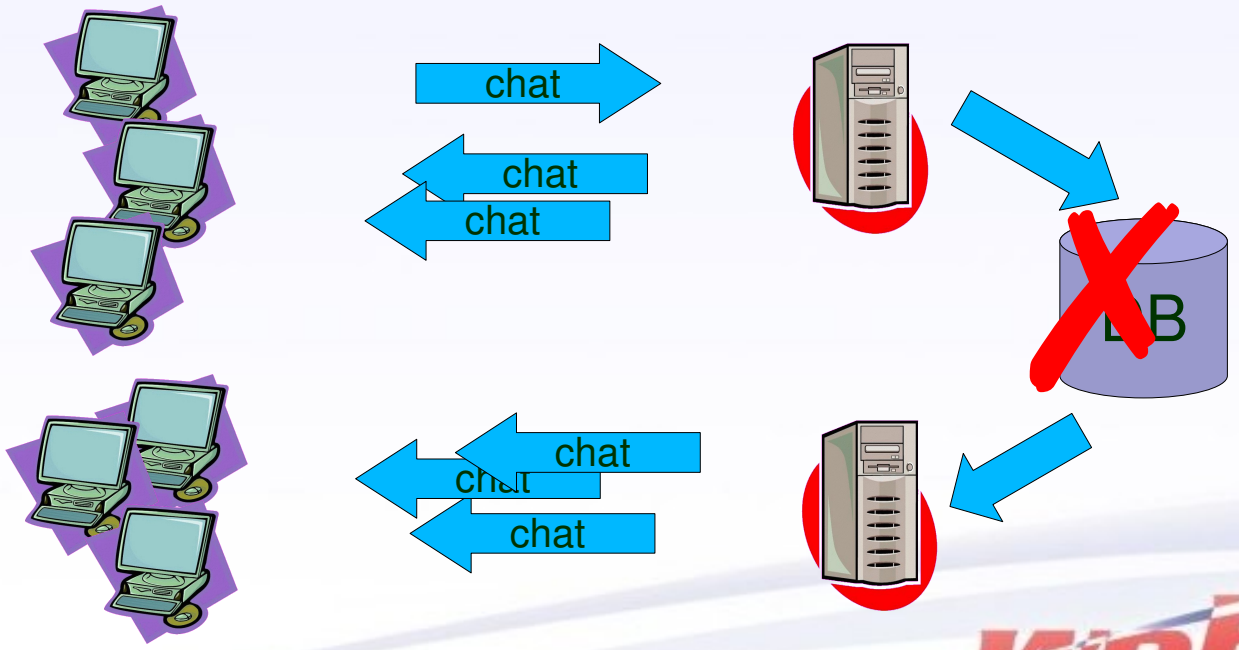
- **Header size**
  - **Thousands of waiting requests**
    - Thousands of allocated buffers
      - *Too big -Wastes memory*
      - *Too small -Some clients have errors!*
  - **Jetty allows tuning of**

```
Header buffer size
Request content buffer size (only allocated as needed)
Response content buffer size (only allocated as needed)
```

- **Gzip Filter**
  - **Some compression filters do not work with comet**

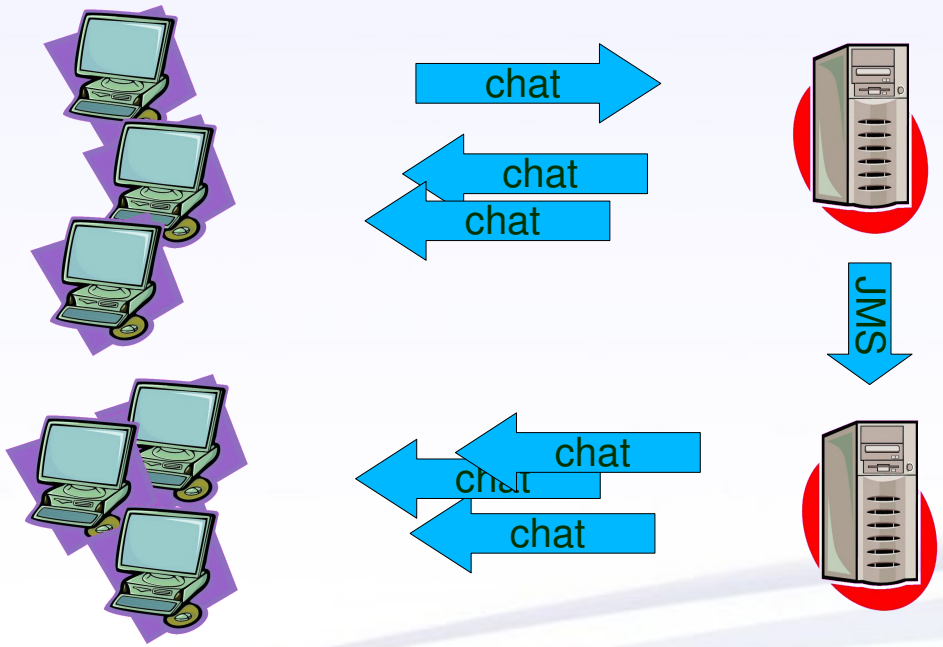
Clustering

- Standard session based clustering insufficient
- Don't use DB as a messaging transport



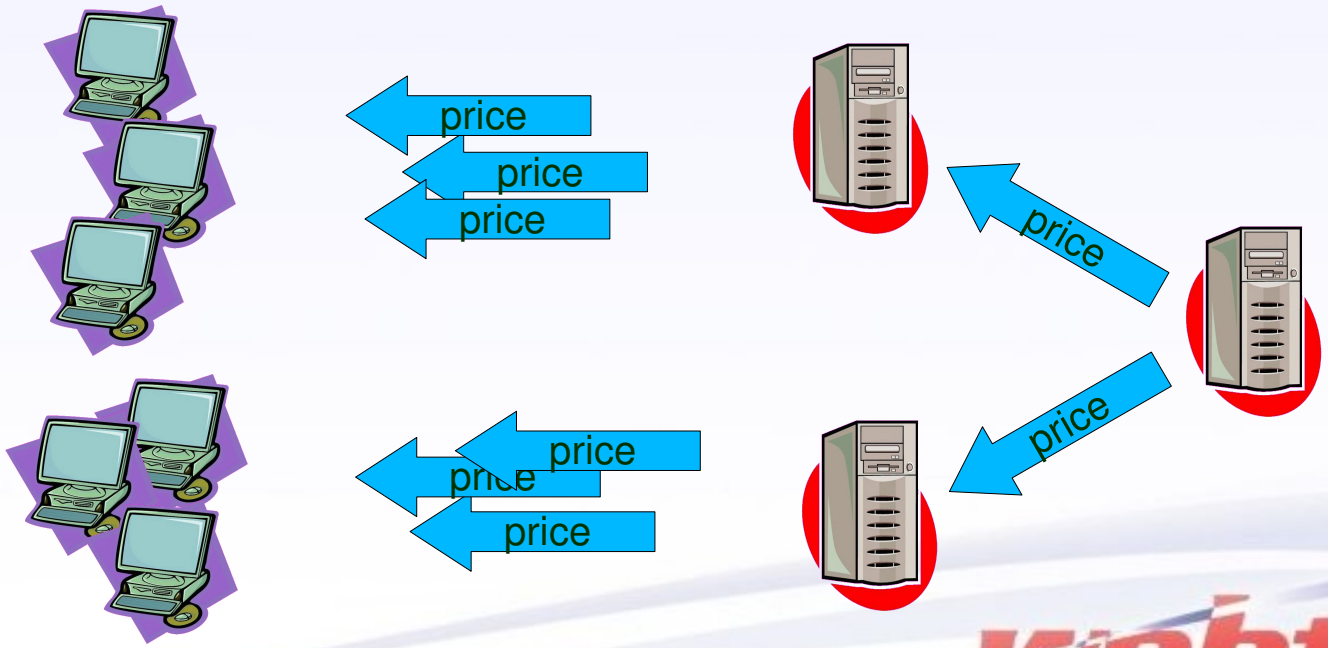
Clustering

- Standard session based clustering insufficient
- Don't use DB as a messaging transport
- JMS a good natural fit for chat-like comet



Clustering

- Standard session based clustering insufficient
- Don't use DB as a messaging transport
- JMS a good natural fit for chat-like comet
- Some apps cluster naturally (like portfolio)



Balancing the Load

- Difficult to find load balancers that
  - a) work
  - b) scale!
- Apache does not cut it
  - Zeus can handle connections, but no load
- Software load balancers that work:
  - Haproxy
  - Nginx
- Hardware (appliance) balancers
  - All have worked so far

Logging & Statistics

- **Long held requests can confuse stats**
  - **Jetty stats handlers updated for asynchronous long polls**
- **Polling can swamp logging**
  - **Jetty loggers can exclude patterns**
    - Images/*
    - css/*
    - /cometd/*

Production Problems

Image by www.stevezphotography.com

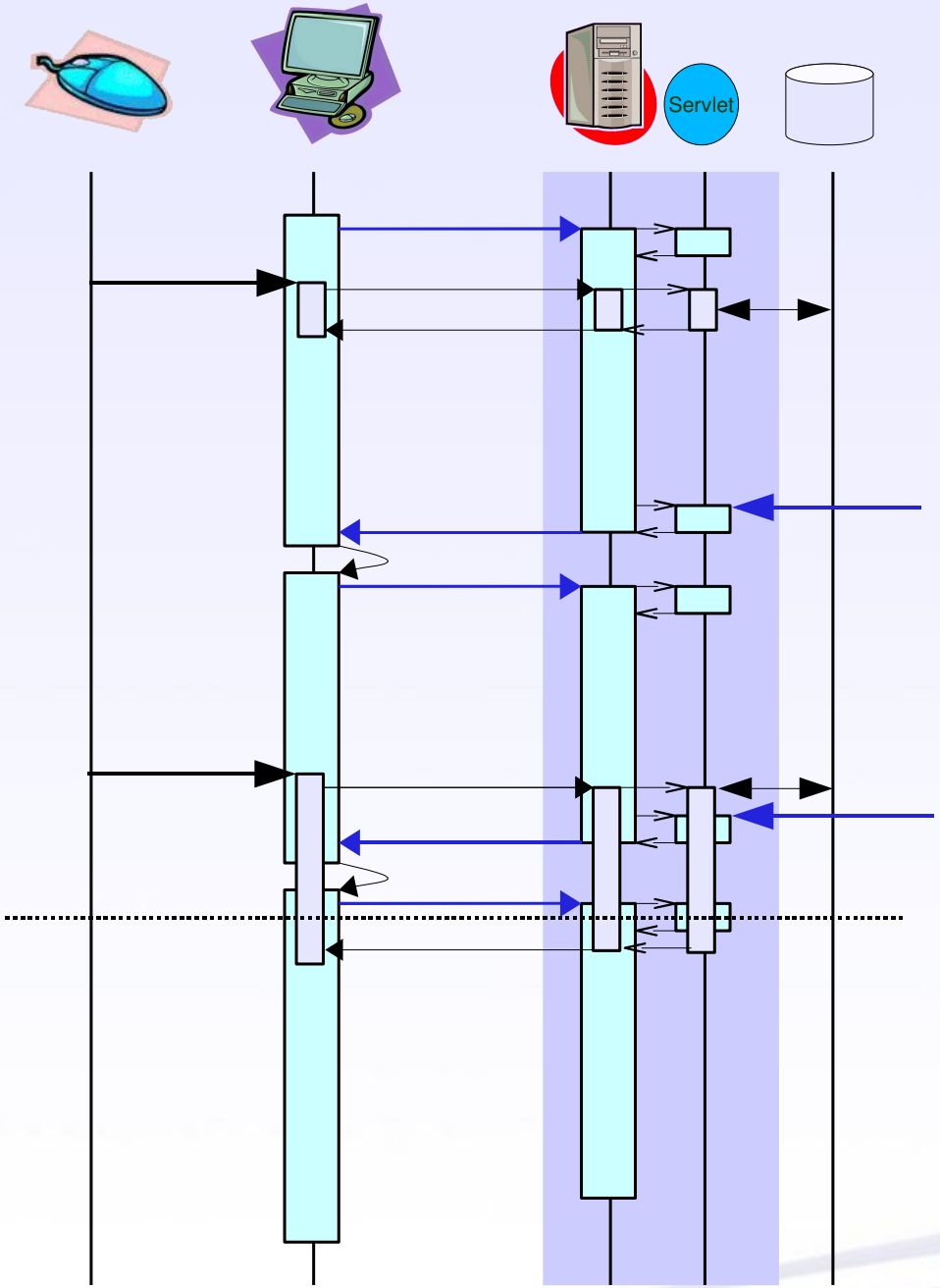
Problems in the Pipeline

- Consider a chat application with images
  - eg smiley substitution

- Or a comet game with lots of images

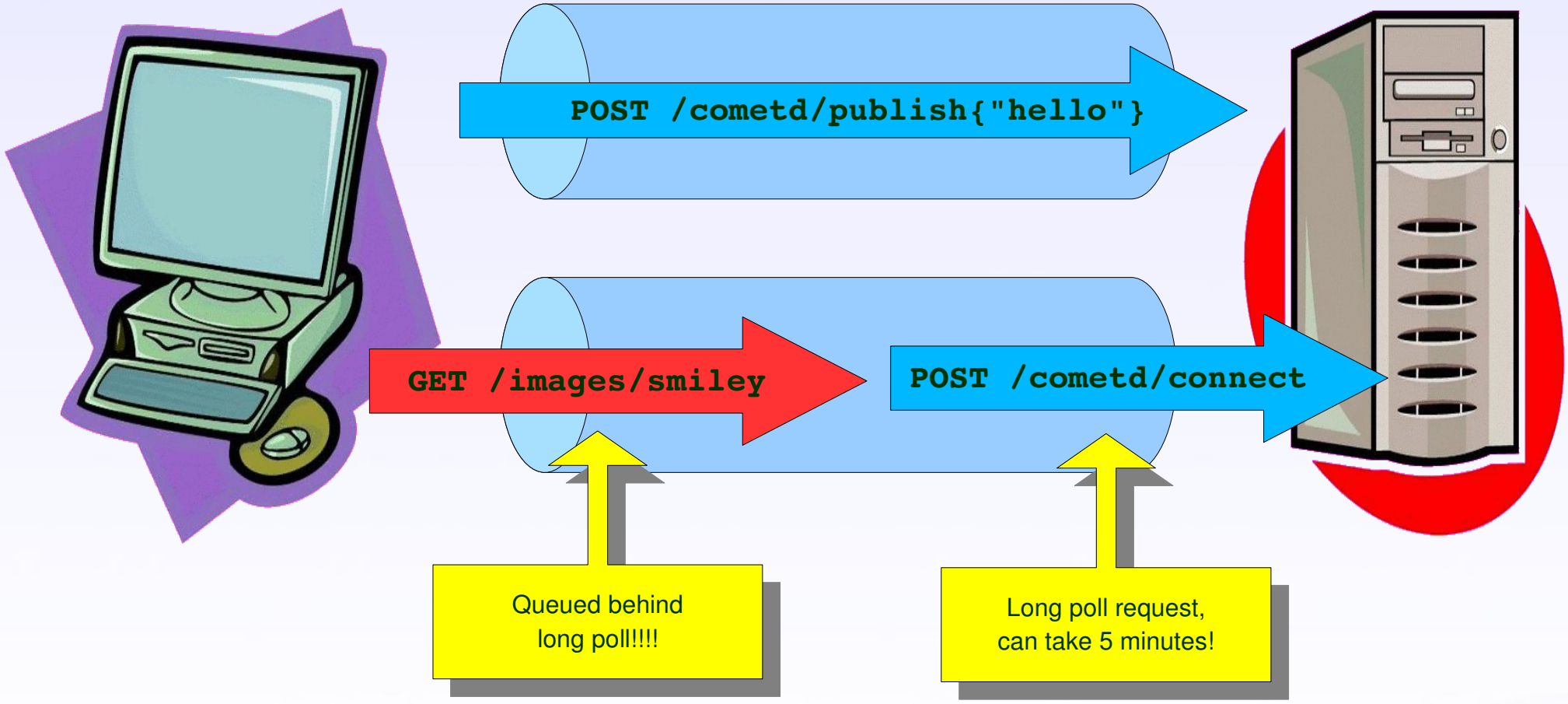
- Problem: Delay for Images to appear.
  - Up to 3 minute delays?

Problems in the Pipeline



- Image loads in pipeline behind long poll
  - If a inbound chat message provokes an image load
  - while outbound chat is being sent
  - Both connections in use
  - Image load may be pipelined behind long poll!

Problems in the Pipeline



Problems in the Pipeline

- **Control your GETs**
  - **Cometd manages GETs/POSTS**
    - Never more than 2 initiated.
  - **Difficult when mixing frameworks (eg Cometd + DWR)**
- **Use a different Domain for Images!**

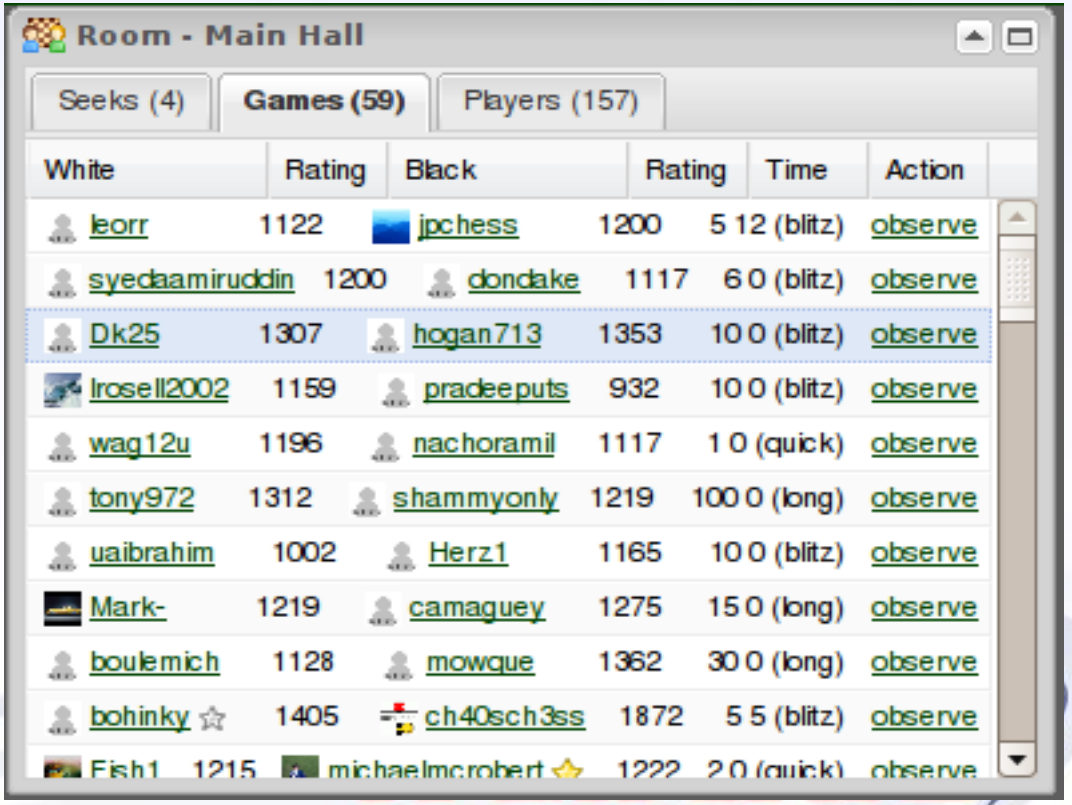
Chess Timeout

Chess Timeout

- Players "disconnected" during games
- Happens only with moderate load (>300 users)
- Server not loaded (plenty of available CPU)?
- Timeout is the cometd maxInterval
  - **Interval is the time between**
    - long poll response
    - next long poll request
- As if the client just didn't re connect???
- Happens only with IE!

Chess Timeout

- **100% CPU on IE**
  - Too slow to respond, hence timeouts
  - Caused by ext.js table updates of all rows (even hidden!)
- **Seeks/Games/Player tabs always updated!**
  - Issue with ext.js
    - Or IE?
  - Poor UI design?
    - Useless events sent
  - Poor load testing
    - Should have been caught in testing!



No going back with Safari

- Comet communication terminated on Safari
  - No timeouts
  - No error handlers invoked
  - As if all XHR were cancelled!

- Safari cancels all XHRs on the backbutton!
  - Even ones initiated in a parent frame!

No going back with Safari

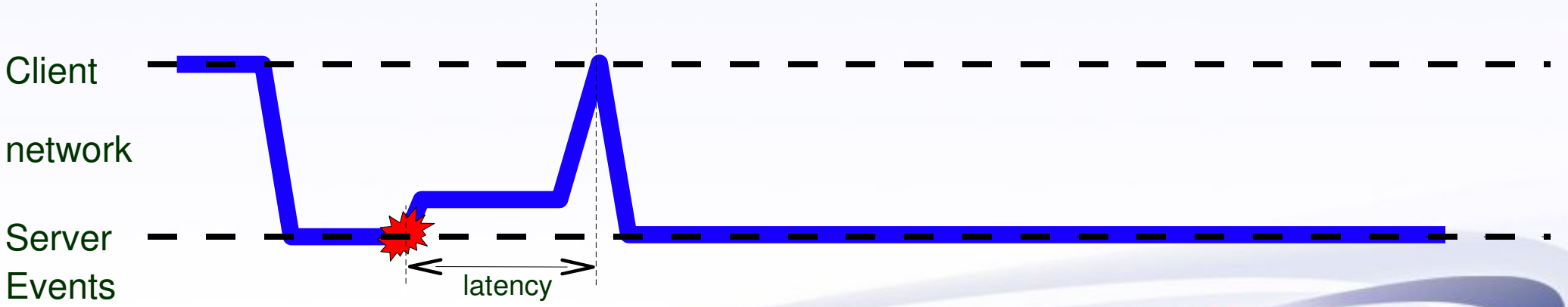
- "That's OK, I don't use iframes with content"
  - Back button handling uses iframes!
    - Implemented with hidden iframe!!!!
  - So you are stuffed!

- Workaround
  - force a page reload!
  - reinit cometd!

- yuck!!!!

Comet Developers Don't Scale

- **Comet is essentially asynchronous**
  - **Frameworks built on callbacks**
    - Assumed to execute in a short period of time
- **Deployed Application Timed out in Production.**
  - **Application developers didn't think of blocking**
  - **POJOs with getters that access the database**
    - Slowed conversion to JSON
  - **Worked OK on test bed with fast database**



webtide

Webtide Support for Jetty and Cometd

- **Expert Advice**
  - Get help avoiding problems rather than solving problems
  - Annual subscription for unlimited advice via private forum
  - Prearranged rate for small development jobs
- **Custom Development**
  - Application Development (outsource or collaborative)
  - Custom extension to open source projects
  - Sponsored development of open source
- **Production Support**
  - Rapid response from the experts
  - Active in all major timezones

Question and Answers?

- http://webtide.com
- http://cometd.org
- http://jetty.mortbay.org
- info@webtide.com

