Ajax Push (a.k.a. Comet) With Java™ Business Integration (JBI)

Andreas Egloff

Chief Architect for Open ESB
Sun Microsystems, Inc.
http://open-esb.org

TS-8434

Goal of This Talk

Learn how to push your critical business information out to Ajax clients—with lower latency and without compromising scalability or performance

Agenda

Basics of Ajax Push

Architecture challenges

Technology solutions

How does JBI fit into the picture?

Demo

Potential drawbacks and pitfalls

Agenda

Basics of Ajax Push

Architecture challenges

Technology solutions

How does JBI fit into the picture?

Demo

Potential drawbacks and pitfalls

What Is Ajax Push Good For?

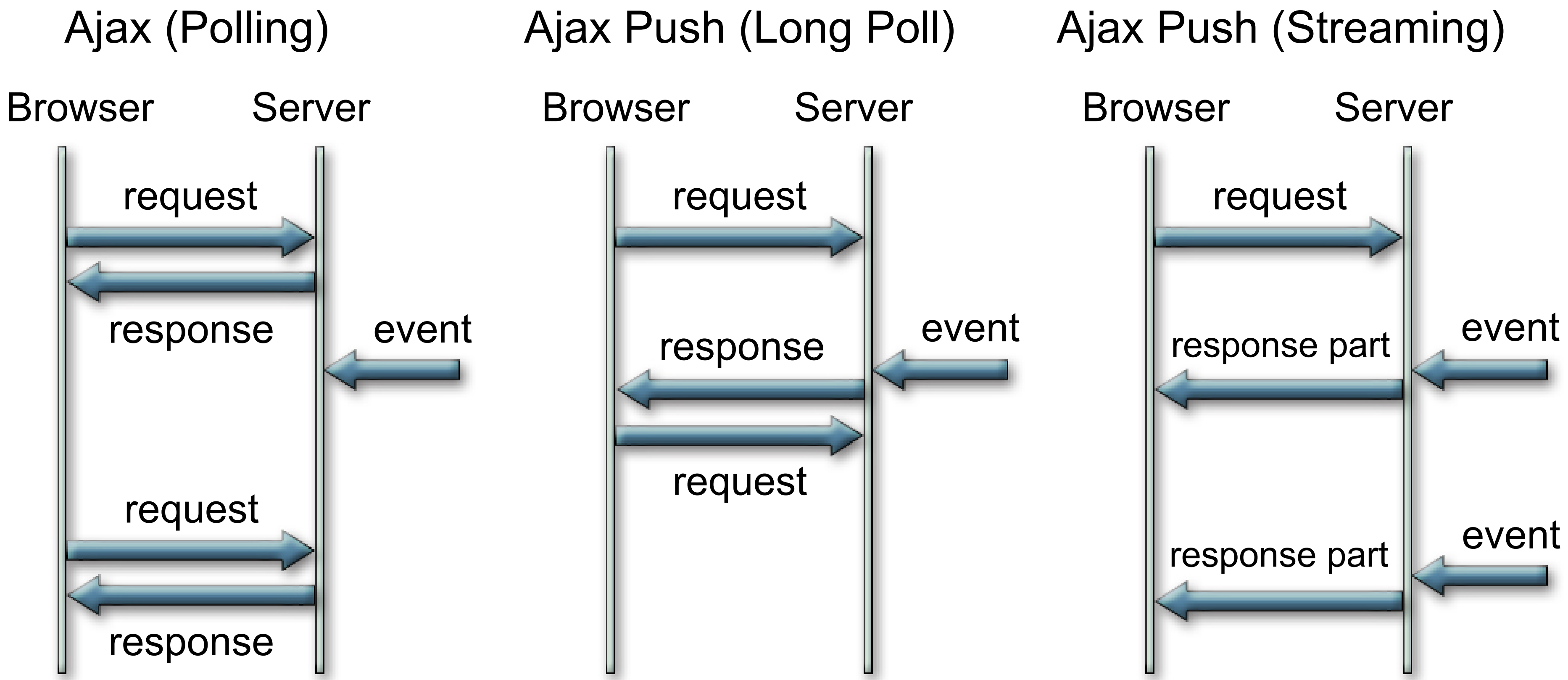
- Use it to create highly responsive, event-driven applications in a browser
  - Keep clients up-to-date with data arriving or changing on the server, without frequent polling
- Pros
  - Lower latency, not dependent on polling frequency
  - Server and network do not have to deal with frequent polling requests to check for updates
- Example applications
  - Gmail and Google Talk
  - Meebo
  - JotLive
  - KnowNow
  - Many more…

How Does “Push” to the Browser Work?

- Deliver data over a previously opened connection
  - Always “keep a connection open”; do not respond to the initiating request until event occurred
  - Streaming is an option by sending response in multiple parts and not closing the connection in between

How Does “Push” to the Browser Work?

Standard Ajax compared to Ajax Push options



Agenda

Basics of Ajax Push

Architecture challenges

Technology solutions

How does JBI fit into the picture?

Demo

Potential drawbacks and pitfalls

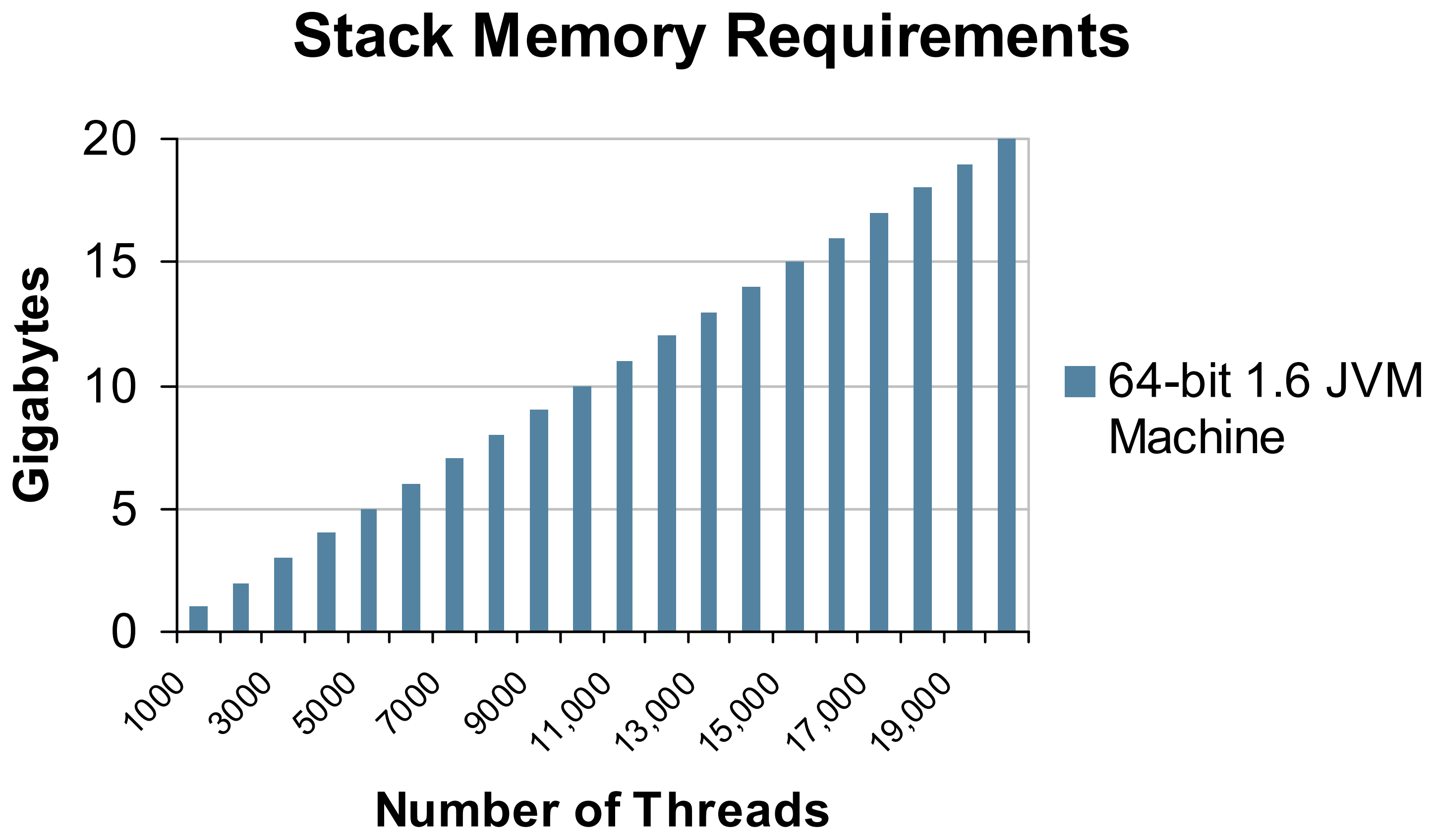
Architecture Challenges

- Using blocking, synchronous technology will result in a blocked thread for each open connection that is “waiting”
  - Every blocked thread will consume memory
  - This lowers scalability and can affect performance
  - To get the Java Virtual Machine (JVM™) to scale to 10,000 threads and up needs specific tuning and is not an efficient way of solving this
- Servlets 2.5 are an example of blocking, synchronous technology

The terms “Java Virtual Machine” and “JVM” mean a Virtual Machine for the Java™ platform.

Architecture Challenges

Affect of blocking threads (default thread stack size)



Agenda

Basics of Ajax Push

Architecture challenges

Technology solutions

How does JBI fit into the picture?

Demo

Potential drawbacks and pitfalls

Technology Solutions

- Use new I/O (NIO) non-blocking sockets to avoid blocking a thread per connection
- Use technology that supports asynchronous request processing
  - Release the original request thread while waiting for an event
  - May process the event/response on another thread than the original request
- Advantages
  - Number of clients is primarily limited by the number of open sockets a platform can support
  - Could have all clients (e.g., 10,000) “waiting” without any threads processing or blocked

Agenda

Basics of Ajax Push

Architecture challenges

Technology solutions

How does JBI fit into the picture?

Demo

Potential drawbacks and pitfalls

How Does Java Business Integration (JBI) Fit Into the Picture?

- The JBI 1.0 (JSR 208) framework provides the basis for building loosely coupled composite applications by combining services
- JBI is inherently asynchronous
  - Components exchange requests and responses through in-memory queues (the normalized message router)
  - Complements the asynchronous nature of Ajax Push

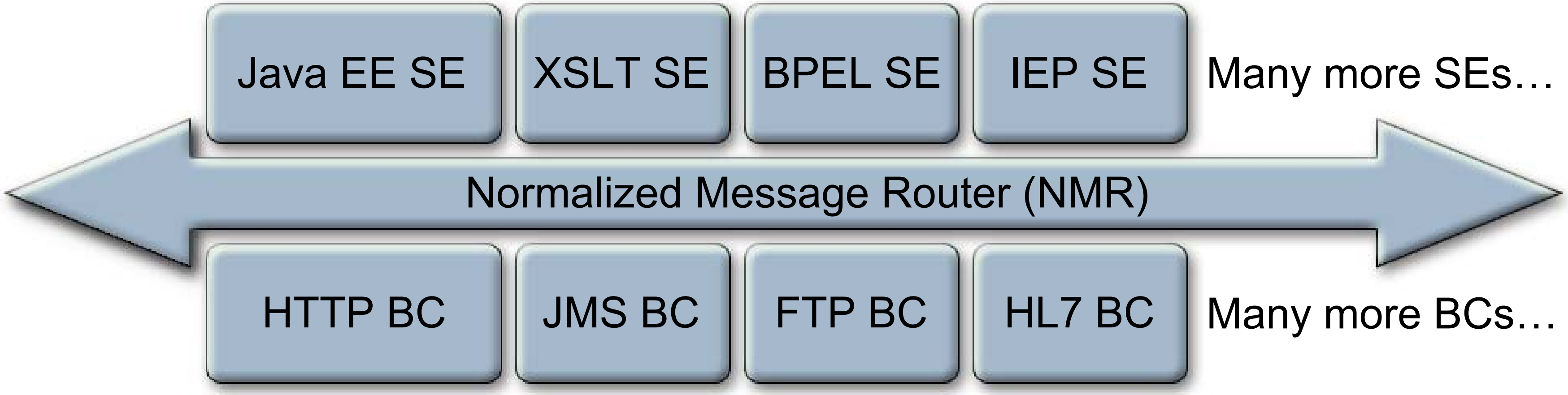
JSR = Java Specification Request

Why Should I Care About JBI?

- JBI provides a standard to plug in server-side containers for business logic and transformations, e.g.:
  - XSLT engine
  - BPEL engine
  - JavaScript™ engine
  - YASU rules engine
  - Many more…
- Choose the best-of-breed containers from the provider(s) you like
- Because JBI components are interoperable you can chain, mix, and match the right technologies to solve your business problems
- External connectivity (binding components) is separate from the business logic containers (service engines); hence every container has access to all the pluggable external connectivity capabilities

JBI Key Components

- Service Engines (SE)—pluggable business logic containers
- Binding Components (BC)—pluggable external connectivity
- Normalized message router (NMR)—standard way of exchanging messages between components



Java EE = Java Platform, Enterprise Edition | Java SE = Java Platform, Standard Edition | JMS = Java Message Service

HTTP Connector in JBI

- The Java EE platform 5 SDK includes a JBI implementation; a sub-set of the open source Project Open Enterprise Service Bus (Open ESB)
- Open ESB includes a fully featured HTTP binding component (BC)
  - Built on the “Grizzly” high-performance NIO framework and uses non-blocking sockets
  - “Grizzly” supports asynchronous request processing and does not block threads
  - “Grizzly” even supports non-blocking sockets when using SSL to secure the connection
  - Leverages a re-architected Java API for XML Web Services (JAX-WS) stack that supports separate threads to process the request and response
- This ensures that the HTTP BC takes full advantage of the asynchronous nature of JBI

Considerations for Service Engines

- To handle the “wait” for an event in Ajax Push, choose a container/language that does not have to block
- Some inherently do not block
  - e.g., BPEL “receive” activity can wait for events without blocking a thread
- Continuations are another options
  - e.g., JavaScript™ technology “continuation” in the “Rhino” open source implementation

DEMO

Ajax Push With JBI

Agenda

Basics of Ajax Push

Architecture challenges

Technology solutions

How does JBI fit into the picture?

Demo

Potential drawbacks and pitfalls

Potential Drawbacks and Pitfalls

- Beware of flooding clients with too many events
- Firewalls may terminate connections after a certain amount of time
    - Solution: re-establish connection after tear-down or at certain intervals
- The HTTP 1.1 specification suggests a limit of two simultaneous connections from a client to the same host
    - Some use a separate host name for the "Push" connection
- In security terms the attack surface is very similar to standard Ajax applications, for denial of service (DoS) the "wait" is a consideration
- Simple "Long Poll" Ajax Push is portable across browsers

Potential Drawbacks and Pitfalls—With Streaming

- Streaming is more challenging
  - Portability issue to different browsers and XMLHttpRequest (XHR)
    - IE, for example, does not make data available until connection close
    - Some solutions use IFrames instead for portability
  - With streaming data will accumulate, release memory regularly
  - Primitive proxies may buffer data in a way that interferes with streaming

Summary

- Pushing data to the web client can form a crucial tool in the developer’s arsenal when latency is important
- “Long-Poll” is straight forward and portable
- “Streaming” today is more complex and has portability implications
- JBI complements the asynchronous nature of Ajax Push and gives the user the freedom to mix and match the right server side tools to solve their business problems

For More Information

- Cross-references to other sessions
  - **TS-8897:** Designing Service Collaborations: The Design of “Wire”-Centric Integration
  - **TS-8683:** Introduction to CASA: An Open Source Composite Applications Editor
- Links
  - Open ESB community
    http://open-esb.org
  - Grizzly NIO framework
    http://grizzly.dev.java.net
  - Andi’s Blog
    http://blogs.sun.com/andi

Q&A

Andreas Egloff

Ajax Push (a.k.a. Comet) With Java™ Business Integration (JBI)

Andreas Egloff

Product Line Architect for Alaska
Sun Microsystems, Inc.
http://open-esb.org

TS-8434