Introducing Apache Tomcat 6

Mladen Turk
Red Hat, Inc.
Agenda

• A brief history of time

• Tomcat versions

• Tomcat 6
What is Tomcat?

• Core features
  – Servlet container
  – JSP container
  – HTTP server

• Additional features
  – Native web server integration (AJP)
  – Basic application server features (JNDI, JDBC, …)
  – Clustering and session replication
What is Tomcat?

• Started at Sun Microsystems
  – Reference Java Servlet and JSP implementation
  – Donated to Apache Software Foundation in 1999

• ASF TLP (Top-level Project)
  – Since 2005
  – Before developed under Jakarta umbrella
public MessageBytes getClone() {
    try {
        return (MessageBytes) this.clone();
    } catch (Exception ex) {
        return null;
    }
}

public boolean isNull() {
    // should we check also hasStrValue ???
    return byteC.isNull() && charC.isNull() && ! hasStrValue;
    // bytes==null && strValue==null;
}
What is Tomcat?

- Actively developed
  - In average 1 release/month
  - 10+ active committers
  - 600,000 downloads/month (without counting mirrors)
Tomcat versions

- Major version related to Servlet/JSP Spec
  - JVM required depends on the JVM required by spec
  - Spec maintained by JCP (Java Community Process)

<table>
<thead>
<tr>
<th>Servlet/JSP Spec</th>
<th>Appliance Tomcat version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5</td>
<td>6.0.x</td>
</tr>
<tr>
<td>2.4</td>
<td>5.5.x</td>
</tr>
<tr>
<td>2.3</td>
<td>4.1.x</td>
</tr>
<tr>
<td>2.2</td>
<td>3.3.x</td>
</tr>
</tbody>
</table>
Tomcat release process

• **Follows the ASF release guidelines**
  – Needs 3 +1 votes from the committers
  – When “enough” work is done
  – Release manager is volunteer

• **Urgent releases**
  – Security related issues
  – Not discussed publicly during report-patch-release process
  – Private email list
Enough Tire Kicking!

- Can we look under the hood?
Servlet 2.5 Specification

- **JSR-000154**
  - Created via Java Community Process ([http://jcp.org](http://jcp.org))
- **JSE 5.0 dependency**
  - JDK 5.0 is minimum platform requirement
- **Allow multiple url-pattern in servlet-mapping**
  ```
  <servlet-mapping>
    <servlet-name>OneFancyServlet</servlet-name>
    <url-pattern>/foo/*</url-pattern>
    <url-pattern>/bar/*</url-pattern>
  </servlet-mapping>
  ```
Servlet 2.5 Specification

- **Annotations support**
  - Alternatives to some XML entries in web.xml
  - `@Resource`:
    - Class or variable “resource injection” request
  - `@Resources`:
    - Similar as `@Resource` but holds the array or `@Resource` annotations
  - `@RunAs`:
    - Used to replace `<run-as>` entries in web.xml
  - `@DeclareRoles`:
    - Alternative to `<security-role>` in web.xml

- Come from JSR 250 (Common Annotations for the Java Platform)
Tomcat 6 Source

- Restructured code base
  - No multiple repositories
  - Reduced duplicate code
  - Removed obsolete items
  - Reduced distribution size
Tomcat 6 Distribution

- Restructured distribution
  - `/server/*` and `/shared/*` are gone!
  - `/lib/*` location of common libraries
  - Windows service wrapper for WIN64 (EMT64 and IA64)
  - Reduced distribution size
Tomcat 6 Distribution

• **Additional components**
  – Downloadable from `/extras/*`

• **Full commons-logging implementation**
  – Tomcat uses a package renamed `commons-logging` API implementation which is hard-coded to use the `java.util.logging` API.

• **Web Services support (JSR 109)**
  – Tomcat provides factories for JSR 109 which may be used to resolve web services references.
  Place the generated `catalina-ws.jar` as well as `jaxrpc.jar` and `wsdl4j.jar` (or another implementation of JSR 109) in the Tomcat `/lib/` folder.
Tomcat 6 Distribution

- **Windows Installer**
  - Installs and setup the Tomcat as a Windows service
  - Optionally download native tcnative-1.dll component located at heanet.ie
Running Tomcat 6

- Use provided OS mechanism
  - On Windows use `tomcat6.exe` and `tomcat6w.exe`
  - The famous `./catalina.sh run`

```
mturk@rh5x64:~/builds/apache-tomcat-6.0.12/bin

mturk@rh5x64 $ ./catalina.sh run
Using CATALINA_BASE: /home/mturk/builds/apache-tomcat-6.0.12
Using CATALINA_HOME: /home/mturk/builds/apache-tomcat-6.0.12
Using CATALINA_TMPDIR: /home/mturk/builds/apache-tomcat-6.0.12/temp
Using JRE_HOME: /opt/jdk1.5.0_11
Apr 28, 2007 1:33:37 PM org.apache.catalina.core.AprLifecycleListener init
INFO: The Apache Tomcat Native library which allows optimal performance in production environments
was not found on the java.library.path: /opt/jdk1.5.0_11/jre/lib/amd64/server:/opt/jdk1.5.0_11/jre/lib/amd64:
INFO: Initializing Coyote HTTP/1.1 on http-0080
INFO: Initialization processed in 1124 ms
INFO: Starting service Catalina
INFO: Starting Servlet Engine: Apache Tomcat/6.0.12
INFO: Starting Coyote HTTP/1.1 on http-0080
Apr 28, 2007 1:33:40 PM org.apache.jk.common.ChannelSocket init
INFO: JK: ajp13 listening on /0.0.0.0:8009
Apr 28, 2007 1:33:41 PM org.apache.jk.server.JkMain start
INFO: Jk running ID=0 time=0/529  config=null
INFO: Server startup in 4268 ms
```
Running Tomcat 6

- APR was not found message

```
mturk@rh5x64 bin]$ ./catalina.sh run
Using CATALINA_BASE: /home/mturk/builds/apache-tomcat-6.0.12
Using CATALINA_HOME: /home/mturk/builds/apache-tomcat-6.0.12
Using CATALINA_TMPDIR: /home/mturk/builds/apache-tomcat-6.0.12/temp
Using JRE_HOME: /opt/jdk1.5.0_11
Apr 28, 2007 1:33:37 PM org.apache.catalina.core.AprLifecycleListener init
INFO: The Apache Tomcat Native Library which allows optimal performance in production environments was not found on the java.library.path: /opt/jdk1.5.0_11/jre/lib/amd64/server:/opt/jdk1.5.0_11/jre/lib/smd64:/opt/jdk1.5.0_11/jre/.:/lib/amd64:
INFO: Initializing Coyote HTTP/1.1 on http-8080
INFO: Initialization processed in 1124 ms
INFO: Starting service Catalina
INFO: Starting Servlet Engine: Apache Tomcat/6.0.12
INFO: Starting Coyote HTTP/1.1 on http-8080
Apr 28, 2007 1:33:40 PM org.apache.jk.common.ChannelSocket init
INFO: JK: ajp13 listening on /0.0.0.0:8009
Apr 28, 2007 1:33:41 PM org.apache.jk.server.JkMain start
INFO: Jk running ID=0 time=0/529 config=null
INFO: Server startup in 4268 ms
```

- Fallbacks automatically to the standard Java IO implementation
Running with APR

- **Uses Tomcat Native (tcnative)**
  - Wrapper over Apache Portable Runtime (APR)
  - Replaces common JSE network stack
Running with APR

- Build or download native package
  - Needs to be in `LD_LIBRARY_PATH` or `PATH` on Windows
How Native works

- The traditional model is Thread per connection
  - Each client is assigned one thread during the connection lifetime.
How Native works

- Uses Thread per request
  - During KeepAlive thread is assigned to next client
How native works

- **Higher concurrency**
  - Able to handle 10000+ concurrent clients
How native works

- **Sendfile support**
  - Uses OS feature for zero-copy.
The benefits of using Native

- **Performance**
  - Uses zero-copy for static content
  - Uses OpenSSL instead JSSE
  - Zero GC – no Java objects are created

- **Scalability**
  - Thread per request allows more HTTP/1.1 concurrent connections
  - Lower CPU and memory usage
Some numbers

- **50 concurrent users**
  - Log scale: zoom on 4MB file serving
Some numbers

- Sendfile lowers down the memory and CPU usage
Tomcat Manager

Server Status

**Server Information**

<table>
<thead>
<tr>
<th>Tomcat Version</th>
<th>JVM Version</th>
<th>JVM Vendor</th>
<th>OS Name</th>
<th>OS Version</th>
<th>OS Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache Tomcat/6.0.12</td>
<td>1.5.8.11-b03</td>
<td>Sun Microsystems Inc.</td>
<td>Linux</td>
<td>2.6.18-1.1.elfslan</td>
<td>amd64</td>
</tr>
</tbody>
</table>

*Physical memory:* 932.16 MB *Available memory:* 5.99 MB *Total page file:* 194338 MB *Free page file:* 1933.00 MB *Memory load:* 1.00
*Process kernel time:* 0.6 s *Process user time:* 0.0 s

*Free memory:* 3.59 MB *Total memory:* 8.71 MB *Max memory:* 81.25 MB

**ajp-8009**

Max threads: 40 Current thread count: 0 Current thread busy: 0 Kept alive sockets count: 0
Max processing time: 6 ms Processing time: 0.6 ms Request count: 2414 Bytes received: 0.00 MB Bytes sent: 0.99 MB

**Stage** | **Time** | **B Sent** | **B Recv** | **Client** | **VHost** | **Request**
---|---|---|---|---|---|---
[Table content]

*Done*
Tomcat NIO Connector

• Developed on APR principles
  – Use JSE NIO features
  – No need for extra native package
  – Not for static content delivery and SSL

• Allows thread per connection model
  – Something NIO wasn’t designed for
  – Extends the standard blocking Servlet model
Tomcat Advanced IO

- **The Comet**
  - Process servlets asynchronously!
  - Receive events when data is available for reading on the connection.
  - Not usable with AJP (for now)
Comet events

- **The Servlet is awaken on event**
  - Implement the `org.apache.catalina.CometProcessor` interface
  - The servlet is valid until the **END** event

- **BEGIN**
  - Called at the beginning of the processing of the connection.
  - Valid until the **END** or **ERROR** events.

- **READ**
  - Indicates that input data is available.
  - It is not allowed to attempt reading data from the request object outside of the execution of this method.

- **END**
  - Indicates the end of the processing
  - Can be fired by application reload
Tomcat 6

- **Documentation**

- **Download**
  - [http://tomcat.apache.org/download-60.cgi](http://tomcat.apache.org/download-60.cgi)

- **Support**
  - Tomcat users mailing list

- **Bug database**
  - We don’t make bugs but anyhow …
  - [http://tomcat.apache.org/bugreport.html](http://tomcat.apache.org/bugreport.html)
Q & A
This page intentionally left blank

Mladen Turk