Unhappily Ever After: Troubleshooting in Production Environment

Alexandre Rafalovitch

United Nations

arafalov@gmail.com

TS-1669

Learn how to analyze and resolve production Java^{TM} technology problems **without panic**

For the Next 60 Minutes

Why Are You Here? Why am I Here?

Descriptive, Not Prescriptive Approach

System Boundaries and Choke Points

Tools and Analysis Methods

Proficiency Across Platforms

Honing the Troubleshooting Skills

Looking Forward

java.sun.com/javaone/sf

Let's Talk About

Why Are You Here? Why am I Here?

Descriptive, Not Prescriptive Approach

System Boundaries and Choke Points

Tools and Analysis Methods

Proficiency Across Platforms

Honing the Troubleshooting Skills

Looking Forward

java.sun.com/javaone/sf

Why are You Here? Why am I Here?

- You
  - Programmer looking for better troubleshooting tools
  - Technical support trying to upgrade skills
  - Manager desiring to improve technical support
- Me
  - 3 years as BEA senior tech. support engineer (DRE)
  - Java technology professional since JDKTM1.0b2
  - Not a guru—just sharing the collected experience

Let's Talk About

Why Are You Here? Why am I Here?

Descriptive, Not Prescriptive Approach

System Boundaries and Choke Points

Tools and Analysis Methods

Proficiency Across Platforms

Honing the Troubleshooting Skills

Looking Forward

Descriptive, Not Prescriptive Approach

- Prescriptive approach is wishing for a silver bullet
  - One cannot ask developer to never do something
  - Most complex problems are emergent issues
  - Expensive software is great, but is usually too late

- Descriptive approach is about understanding
  - Somebody will always end up doing X
  - And sometimes it is just Dude, Where is my log file?
  - Even if you do not know how you got there, you still have to fix it

Descriptive, Not Prescriptive Approach

Problem with Programmer's Method

- System.out.println() is **not** your friend in production

- Recompile and restart is infrequently an option

- Autowiring is good until something gets miswired

- Did anyone tell you about the firewall?

- When the system is losing more per hour than you make per year—this is not the time to start reading APIs

- Things are getting better, but production still runs old stuff

Let's Talk About

Why Are You Here? Why am I Here?

Descriptive, Not Prescriptive Approach

System Boundaries and Choke Points

Tools and Analysis Methods

Proficiency Across Platforms

Honing the Troubleshooting Skills

Looking Forward

System Boundaries and Choke Points
Knowing Where to Look

- Modern programs are beyond cat, grep or sort
  - Require configuration files
  - Create logs
  - Run in grids and clusters

- Filesystem boundary (config, log, classpath)

- Network boundary (clusters, webapps, JDBC^{TM} software)

- Processor and memory (multithreading)

- Environment variables (OS/user specific)

- Configuration files (XML, properties, automagic)

Example Program

```java
public static void main(String[] args) throws Exception {
    Preferences prefs =
        Preferences.userRoot().node("Boundaries");
    int port = prefs.getInt("port", 8001);
    int idx=0;
    ServerSocket socketListener = new ServerSocket(port);
    Logger logger = Logger.getLogger("Boundaries");
    while(true) {
        Socket socket = socketListener.accept();
        logger.info("Accepted connection: " + idx);
        BufferedReader in = new BufferedReader(
            new InputStreamReader(socket.getInputStream()));
        FileWriter writer = new FileWriter(args[idx++]);
        String line;
        while ((line = in.readLine()) != null) {
            writer.write(line); writer.write('\n');
}}}
//missing something?
```

System Boundaries and Choke Points

Filesystem

- Ignore relative path puzzle, look at lower level

- Currently open files (logs, locks, jars)
  - More than you expect
  - Process Explorer on Windows, lsof on *nix

- Transient files and file search (configs, classpath)
  - The system is way busier than you expect
  - Performance lessons of classpath ordering
  - FileMon on Windows, trace/struss/dtrace on *nix

System Boundaries and Choke Points
Network

- Applications are becoming more and more chatty
  - Webapps with a browser as a platform
  - Webstart applications
  - AJAX (quantitative change)
  - Clustering
  - JDBC

- Most of the traffic is over HTTP

- A lot of troubleshooting information is available, but it is hard to see with all the layers on the stack
  - Ethernet, IP, TCP, HTTP, XML

System Boundaries and Choke Points
Processor and Memory

- Processes are becoming more instrumentable

- Still hard to look inside, but getting better

- Java VMs expose more information via JMX^{TM} technology

- Multithreading issues will become more prominent

- Java based thread dumps
  - There are problems with JIT and different Java VMs
  - Locking information was good in 1.4, incomplete in 5.0, improved in 6.0

System Boundaries and Choke Points

Environment Variables

- If something is not defined anywhere in the program, look in the environment
  - Java VM version
  - Default classpath
  - Extensions jars

- Different in Windows and UNIX®
  - Everything is in the files on UNIX
  - Windows can have it in files or in registry

java.sun.com/javaone/sf

System Boundaries and Choke Points
Configuration Files

- GUI configuration does not survive meeting the troubleshooting reality
  - Compare settings across servers
  - Ultimate authority, when something is wrong

- Usually there more files than expected
  - Tomcat has 23 XML files (9 types) + 3 properties

- Good news: configuration files are parsable
  - Parsable means they can be correlated

- Some things are not in the configuration files
  - Autowiring
  - Defaults

java.sun.com/javaone/sf

Let's Talk About

Why Are You Here? Why am I Here?

Descriptive, Not Prescriptive Approach

System Boundaries and Choke Points

Tools and Analysis Methods

Proficiency Across Platforms

Honing the Troubleshooting Skills

Looking Forward

java.sun.com/javaone/sf

Tools and Analysis Methods

Modern Java VM and its Advantages

- JDK^{TM} 5.0 and JDK 6.0 new troubleshooting tools
  - jconsole, jps, jhat, jmap, jstack, jstat

- If your Java VM does not cut it, look at others
  - BEA JRockit—memory leak detector, console

- Look at what your O/S comes with
  - DTrace on Solaris

- 3rd party tools
  - Easy install—too late for complex configurations
  - Minimum admin privileges—not always possible

Tools and Analysis Methods

Filesystem Boundary

- Currently open
  - Log files, active IO, leaking handles
  - Windows: ProcessExplorer/Handle from Sysinternals
  - *nix: lsof

- Solves
  - Resolves relative paths
  - Shows leaking handles
  - Default locations for log files

- From our example:

```
FileWriter writer = new FileWriter(args[idx++]);
```

java.sun.com/javaone/sf

Tools and Analysis Methods
Filesystem Boundary

- Files briefly accessed by the process
  - Configuration files, classpath checking, jsp reload
  - Windows: FileMon from Sysinternals
  - *nix: truss/strace
  - Solaris 10: dtrace

- Solves
  - Configuration files not where expected
  - Incorrect library version is picked up
  - File (JavaServer Pages^{TM}) changed but not reloaded

- From our example:
  - Let's look at classpath

Tools and Analysis Methods

- FileMon output showing classpath search

```
javaw.exe... QUER... C:\TEMP\extraclasspath                                          SUCCESS
javaw.exe... QUER... C:\Program Files\Java\jre1.5.0_06\lib\ext\dnsns.jar             SUCCESS
javaw.exe... QUER... C:\Program Files\Java\jre1.5.0_06\lib\ext\localedata.jar        SUCCESS
javaw.exe... QUER... C:\Program Files\Java\jre1.5.0_06\lib\ext\sunjce_provider.jar   SUCCESS
javaw.exe... QUER... C:\Program Files\Java\jre1.5.0_06\lib\ext\sunpkcs11.jar         SUCCESS
javaw.exe... QUER... C:\Projects\Eclipse\JavaOneExamples\example\Boundaries.class    SUCCESS
```

Tools and Analysis Methods
Network Boundary

- Currently open connections
  - Unknown configuration, leaking descriptors
  - Same approach as with currently open files
  - Windows: ProcessExplorer/TCPView from Sysinternals
  - *nix: lsof

- From our example:

ServerSocket socketListener = new ServerSocket(port)...

Socket socket = socketListener.accept();

Tools and Analysis Methods

- ProcessExplorer showing open/leaking sockets

java.sun.com/javaone/sf

Tools and Analysis Methods
Network Boundary

- Network traffic over time
  - webapps, applets, Cluster replication, JDBC, LDAP
  - Windows/*nix/*: Ethereal
    - Open source and multi-platform
    - Reads >20 tracer/tcpdump formats
    - Parses > 750 protocols (including HTTP and XML)
    - Custom capture/display filters
    - Displays both high and low level details as needed
    - Can be installed on client,server or spanning port
- An high/low level example
  - Connect to http://www.news.com

java.sun.com/javaone/sf

Tools and Analysis Methods


```
Frame 7 (490 bytes on wire, 490 bytes captured)
Ethernet II, Src: Netgear_64:ce:6e (00:0f:b5:64:ce:6e
Internet Protocol, Src: 216.239.115.148 (216.239.115.
Transmission Control Protocol, Src Port: http (80), D
Hypertext Transfer Protocol
    HTTP/1.1 302 Found\r\n
    Date: Mon, 27 Mar 2006 04:39:55 GMT\r\n
    Server: Apache/2\r\n
    Location: http://news.com.com/\r\n
    Content-Length: 204\r\n
    Keep-Alive: timeout=15, max=955\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html; charset=iso-8859-1\r\n
    \r\n
Line-based text data: text/html
    <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
    <html><head>
```

Tools and Analysis Methods
Processor and Memory

- Commercial tools are quite heavy for production

- New Java VM tools are much better

- Statistical tools are for trends, not troubleshooting

- Memory leaks/allocation issues

  - Use Java VM tools such as jconsole/jrockit profiler

- Processor issues

  - Deadlocks, livelocks, overly long execution

  - Thread-dumps are your friends, but not without tools

    - Different formats/capabilities for different versions/vendors

  - See my presentation from JavaONE^{SM} conference 2004 - TS-1646

Tools and Analysis Methods

Environmental Variables

- Environment is everything not defined explicitly
  - Common interesting variables
    - OS level
      - PATH, CLASSPATH (unexpanded), JAVA_HOME, TEMP
    - Java VM provided
      - Real classpath, Java VM versions
    - Software provided
      - Version/Patch
  - If different variables contradict, strange things happen
    - Path may override JAVA_HOME sometimes
  - Look for environment as logged by the application
  - If that fails, various process tools show OS level info

Tools and Analysis Methods
Configuration Files

- ## Common types
  - ### XML-parsable
  - ### name/value pairs (.properties)–usually parsable
  - ### Defaults and autowiring–problematic for maintenance

- ## Locations
  - ### Files on the filesystem
  - ### Registry
  - ### Inside the jars–hard to discover

- ## From our example

```
Preferences prefs = Preferences.userRoot().node("Boundaries");
int port = prefs.getInt("port", 8001);
```

Tools and analysis methods
Configuration Files

- Using RegMon (from Sysinternals)
  - Shows where Java 5 preferences are kept by default
  - Notice the leading / in **/Boundaries** branch

javaw.exe:3760	OpenKey	HKCU\Software\JavaSoft\Prefs	SUCCESS	Access: 0x4
javaw.exe:3760	CreateKey	HKCU\Software\JavaSoft\Prefs\/Boundaries	SUCCESS	Access: 0x20
javaw.exe:3760	CloseKey	HKCU\Software\JavaSoft\Prefs	SUCCESS	
javaw.exe:3760	CloseKey	HKCU\Software\JavaSoft\Prefs\/Boundaries	SUCCESS	
javaw.exe:3760	OpenKey	HKCU\Software\JavaSoft\Prefs\/Boundaries	SUCCESS	Access: 0x1
javaw.exe:3760	QueryValue	HKCU\Software\JavaSoft\Prefs\/Boundaries\port	SUCCESS	"8004"

Tools and Analysis Methods
Configuration Files

- Working with configuration files
  - Too long to read through
  - Processing extracts relevant information
  - Visualization highlights complex relationships

- Processing XML
  - XSLT/XQuery for serious use
  - XMLStarlet for prototyping

- Visualization
  - Graphviz for any A→B, B→D, C→D relations

java.sun.com/javaone/sf

Tools and Analysis Methods

Configuration Files

- ## XMLStarlet—Unix style toolkit for XML

  - ### Example: What ports tomcat listens on?

```
...\xmlstarlet-1.0.1\xml  sel -T -t
  -m //*[.//@port]
    -m ancestor::* -o -+ -b
    -v local-name()
    -o : -v @port
    -n
  server.xml


Server:8005
-+Service:
-+-+Connector:8080
-+-+Connector:8009
```
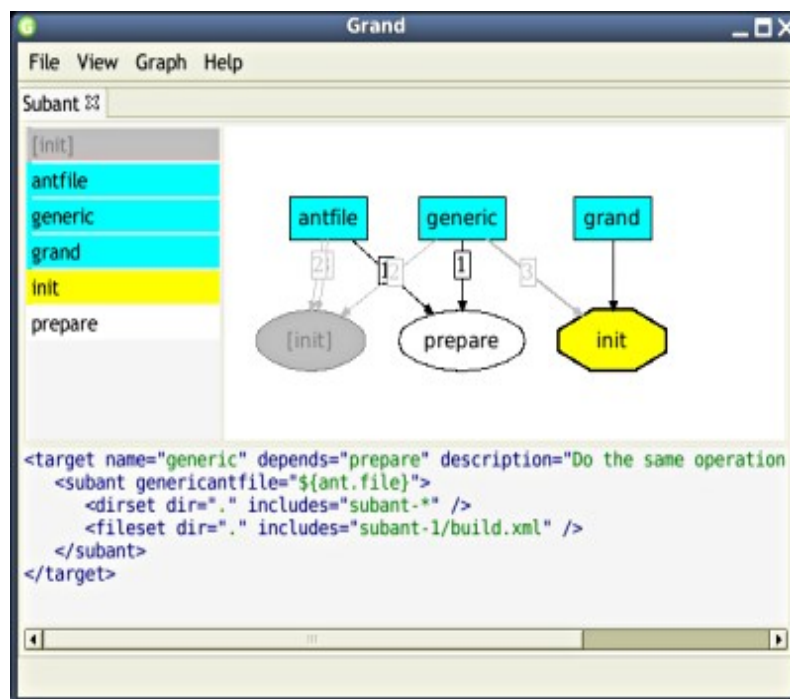
Tools and Analysis Methods
Configuration Files

- ANT config visualization—Grand from ggTools

Let's Talk About

Why Are You Here? Why am I Here?

Descriptive, Not Prescriptive Approach

System Boundaries and Choke Points

Tools and Analysis Methods

Proficiency Across Platforms

Honing the Troubleshooting Skills

Looking Forward

java.sun.com/javaone/sf

Proficiency Across Platforms

The Lazy Programmer

- If you work across multiple platforms
  - Do not learn multiple tools for the same task
  - Use the same editor—Vim/Emacs
    - I use Vim + OTF (script 634) + JAD (script 446)
  - Use Unix/Cygwin tools—grep, find, sort, uniq
  - Use the same XML processor—XMLStarlet
  - Use the same network analyser—Ethereal
  - Use the same image editor—Gimp

- Do not rely solely on super-environment (Eclipse)
  - It will most probably not be installed on production

Proficiency Across Platforms
The Lazy Programmer

- Vim + OTF (On-The-Fly highlighter)

Let's Talk About

Why Are You Here? Why am I Here?

Descriptive, Not Prescriptive Approach

System Boundaries and Choke Points

Tools and Analysis Methods

Proficiency Across Platforms

Honing the Troubleshooting Skills

Looking Forward

java.sun.com/javaone/sf

Honing the Troubleshooting Skills
Get it Before it Gets You

- Do you know what you run?
  - Confirm installed software version from log files
  - Find where **all** the configuration files are
  - Find where **all** the log files go to
  - Find out what the server does every 5 minutes

- Try flying blind
  - Deploy a program the normal way, **then**
  - Change a class and redeploy without restarting the server using tools installed in production **only**

- Read (not skim) and understand a log file
  - Tools like Splunk and Apache Chainsaw may help

Let's Talk About

Why Are You Here? Why am I Here?

Descriptive, Not Prescriptive Approach

System Boundaries and Choke Points

Tools and Analysis Methods

Proficiency Across Platforms

Honing the Troubleshooting Skills

Looking Forward

Looking Forward
Will it Get Easier?

- Things will get easier
  - Java VMs/OSs become more instrumentable
  - New commercial and open sources products appear
    - Splunk, Apache Chainsaw, Ethereal

- Things will get harder
  - Multiple processors—more synchronization problems
  - AJAX—control is no longer in one place
    - Synchronization is now on the client
    - Requests may or may not complete correctly
    - Browsers are different
  - SOA makes everything more distributed
    - More configuration, more log files, harder to troubleshoot

Summary

- Do not panic
- Remember the 5 boundary types
- Identify which boundary/choke point may have the answer
- Know the tools and how to use them
- Harmonize tools across all platforms
- Practice beforehand
- Share the knowledge

For More Information

- Articles
  - http://blogicblog.blogspot.com-my blog on this topic
- Tools
  - Sysinternals: http://www.sysinternals.com/
  - Vim: http://www.vim.org
  - Ethereal: http://www.ethereal.com/
  - XMLStarlet: http://xmlstar.sourceforge.net/
  - Graphviz: http://www.graphviz.org/
  - Apache Chainsaw: http://logging.apache.org/log4j/docs/chainsaw.html
  - Splunk: http://www.splunk.com/ (commercial)

Q&A

Alexandre Rafalovitch

Unhappily Ever After: Troubleshooting in Production Environment

Alexandre Rafalovitch

United Nations

arafalov@gmail.com

TS-1669