

Messages in a Bottle(neck): AutoTrader.com's Image Processing System

Chris Kulinski
chris.kulinski@autotrader.com



What's In This Presentation Bottle?

- High-Level Business Case
- Old vs. New System
- 4 Best Practices for High-Volume Apps
- 6 Specific Implementation Tips
- Future Plans
- Summary



What is Image Processing?

- Retrieves, Scales and Formats Incoming Images
- 3.2 Million Auto Listings in Inventory
- 8.7 Million Auto Images
- It's a Batch-Processing, Easily Distributable Problem. It's Analogous to Many Other Business Applications in Your Organizations.



Business Case – Why Rebuild From Scratch?

- High-Quality Images are **Critical** to the Value Proposition
- In 2006, Our Product Plan Allows for **3x** Growth in Images per Dealership
- Current System Maxed Out at 600,000 Images per Day. New System Designed for 1 Million Images/Day Target
- Must Haves:
 - Visibility
 - Scalability
 - Stability
 - Reliability



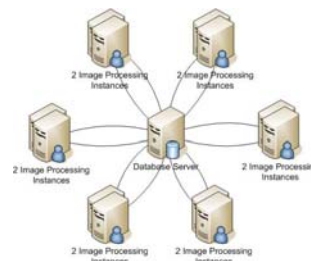
Old System

- Stand-Alone Java Application
- 13,800 Lines of Java Code
- 6 Production Servers – 12 Total Instances
- Crontabs Ran 27 Additional Perl and Shell Scripts! – 2,400 Extra LOC
 - Every 5 minutes we'd run checkForCoreFileAndRestart.sh
 - Every 1 hour we'd run restartAllServers.sh

Just a castaway, an island lost at sea, oh
Another lonely day, with no one here but me, oh
More loneliness than any man could bear
Rescue me before I fall into despair, oh
- Sting "Message In A Bottle" [Reggatta de Blanc](#)

Old System Diagram

- Decentralized – Each Instance Completely Autonomous



New System

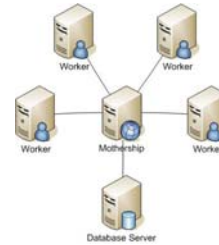
- 6,300 Lines of Java/JSP/HTML Code
 - Half the Size of the Old System
- 0 Perl/Shell Management Scripts
 - Stable and Reliable
 - Uptime Reported in Months – Not Hours
- 5 Production Servers
 - Less Hardware to Manage
- Handles 9+ Million Messages/Day – **9x** the Target!
- JBoss AS 4.0.2 at the Core
 - JMS (JBoss MQ)
 - JMX
 - J2EE SSBs & Timer Service



7

New System Diagram

- Centralized (but Distributed) Design



8

Old vs. New – Key Improvements

- JMS/MDB Threading Model Much More Stable Than Custom Thread Code
- Database now a “Mitigated” Bottleneck
 - We Can Scale JMS Workers at Will
 - Still a Bottleneck When Persisting Messages, but Much Less Significant
- Monitoring and Management are “Front & Center” Requirements (Visibility)
 - Leveraging Existing JMX, JMS M&M Components in JBoss
 - Using Web Container for Easy M&M Interface



9

BP #1: Leverage AppServer’s Core Strengths

- Be Critical of System Code in a Business Application
 - Leave Systems Code to the Application Server
- Watch Out For “Over Zealous” Identifiers
 - `new Thread();`
 - `System.runFinalization();`
 - `System.gc();`
- There Are Better Alternatives Than Custom Code for Batch Processing
 - Use JMS/MDBs for Threading
 - Use Timer Service for Scheduling



10

BP #2: Database’s Strength - Persistence

- Don’t Rely on Database Tables For Message Passing in a Transactional System
 - Ties System Scalability to Database Scalability
- Multithreaded Reads and Writes on Single Tables Creates *Hot Spots* of I/O
 - Consider Hash Partitions
- Critically Evaluate Every DB Call That’s Not Focused on Persistence



11

Tip #1: Consider Custom Persistence

- Standard JMS Container Persistence Isn’t Always The Best Fit
 - Serialized Objects are Stored in BLOBs
 - With 9+ Million Messages a Day, We Average 300,000 Failures on the Queue
 - Need Visibility on the Failures? – Write Custom Persistence
 - Use Simple SQL Queries for Reporting and Analysis of Message Failures



12

BP #3: Multi-Layer Monitoring

- The Most Common Monitoring?
 - Logfile Inspection ☹
- Out-Of-Box JMX Tools Don't Always Meet the Requirements
- Leverage the Web Container for Custom Monitoring
- Provide Layers for Specific Roles
 - Overview - Operations
 - Detailed - Application Support
 - Performance - Developer



13

Overview Monitoring

- Simple, High-Level Monitoring
- Shows the Overall Health of the System
- Fosters "*Peace-of-Mind*" for Operations
- Use Scripting Tools:
 - Perl
 - Curl
 - SSH/PS
 - SQL+

AutoTrader.com Image Processing (IPS) Status
Last Refresh Time: Sat May 13 10:41:15 2006

Queue	Count	Threshold
Waiting Messages - IP Queue	7	170000
Waiting Messages - TP Queue	125819	170000
Total Failed Messages	202025	100000
Total Deferred Messages	1342269	300000

Worker	Status	% CPU Used	Time Started
ipw=001 (motherhip)	OK	7.2%	May_04
ipw=001 (worker)	OK	5.2%	May_04
ipw=002 (worker)	OK	8.3%	May_04
ipw=003 (worker)	OK	7.8%	May_04
ipw=004 (worker)	OK	6.2%	May_04



14

Tip #2: View Messages in the Queue

- Standard JMX Queue Viewer is Poor
- Using JMS Queue Browser is Slow!
- Use JMX to Retrieve the Queue Contents
 - `org.jboss.mq.server.jmx.Queue.listMessages()`
 - Then Display In a Usable Interface

314768 Messages in the Queue

ID	Queue	Message	Timestamp	Sender	Receiver	Message Type	Message Size	Sender
488181273	2006-05-13 10:38:00



15

Tip #3: MBean Message Counter

- Example of a Useful "*Out-Of-Box*" MBean Operation
- Useful for Graphing General Traffic Curves

JMX MBean Operation Result listMessageCounter:story ()

Object Name	Value
...	...



16

Performance Monitoring

- Send RMI "HeartBeat" from Worker to Mothership
 - Timer Service Sends the Performance Stats
 - Currently Custom Framework
 - Critical to Continually Support High-Volumes of Messages

Performance Monitoring Dashboard

Metric	Value	Unit
...



17

BP #4: "Restructure" Management

- J2EE Specs Leave Management Up To Vendors
 - Timer Service Monitoring & Management Needs Improvement
 - JMS Monitoring Tools Are Lacking
- Best Opportunity of Value-Add for Vendors - JBoss ON?
- Until Then, Leverage JMX for Custom Runtime Configuration and Management



18

Detailed Status and Management

- Designed for Application Support
- Shows Specific Health of Individual Components
- Custom Interface for Timer Service and JMS Management Capabilities



19

Tip #4: Timer Service Improvements

- Use Utility Interface for Easy Monitoring and Management of the Timer Service
- Many Variations Seen In J2EE "Tips" Articles
 - Why Couldn't the Spec or Implementation Just Be Easy To Use?
- Could Use JMX Timer – But It's Not Well Integrated w/ EJBs
- Methods:
 - `TimerBeanInfo getInfo();`
 - `ejbTimeout();`
 - `scheduleNextInterval();`
 - `start();`
 - `stop();`



20

Tip #5: JMX Configs

- Use JMX for Runtime Configuration and Management
- Properties Files Seed Initial Values
- It's Saved Us a Few Times Already!
- Systems Developers Have Caught On – But Not Widely Used in Business Apps



21

Tip #6: Automate the Configuration

- JMS Setup Is Complicated
 - 8 Different XML Files for JBoss MQ and also JBoss Messaging
- There's Probably >1 App Using JMS In Your Organization
- Automate the JMS Configuration
 - Use Patch – You'll See Your Changes
 - Almost Every *service.xml File Changes With Every JBoss Release



22

JBoss Messaging 1.0 First Impressions

- Improved Performance
 - JBoss MQ is Meeting Our Current Needs
 - Pushes 9+ Million Messages per Day
- Clustering/HA Scheduled in 1.2
 - We'll Need For Scaling Beyond One "MOTHERSHIP"
- Configuration is Still Complex
 - Non-Trivial Effort To Apply Our Existing Custom JBoss MQ Config Changes
- Will Need to Rewrite our JBoss MQ Custom Monitoring & Management Code
 - Implementations Only Match at JMS Spec Level
 - Needed to Reference JBoss MQ Specific Classes to Achieve Our Visibility Requirements



23

Best Practices - Summary

1. Leverage Your Application Server's Strengths
 - It's System Code Done Right
2. Design to a Database's Strength
 - Persistence!
3. Provide Multi-Layer Monitoring
 - Overview
 - Detailed
 - Performance
4. Usable Management Functions



24

Implementation Tips - Summary

1. Consider JMS Custom Persistence
 - More Visibility Than Standard Version
2. Provide Better View of the JMS Queue
 - More Usable Than the Default
3. JMS Message Counter MBean
 - Useful JMS/JMX Default Operation
4. Improve on Timer Service
 - Otherwise, It's Painful to Use
5. Use JMX for Config/Management
 - Allows Flexibility at Runtime
6. Automate Your Configurations
 - Makes JMS Config a Little Less Complicated

25



The Big Finish

- Follow Best Practices to Efficiently Process High-Volumes of Messages
- Specific Implementation Tips Will Help Guide You to Success
- Make Management and Monitoring a "First Order" Requirement
 - But Be Prepared to Roll Your Own
- Tips or Best Practices From the Audience's Experience?

26

