High-End Java™ EE Platform-Based Application Servers for Enterprise Scale Business Suites

Peter Kulka

Solution Architect
SAP AG
peter.kulka@sap.com

Peter Emmel

Evangelist
SAP AG
peter.emmel@sap.com

TS-4830

Goal of Our Talk

What you will gain

Learn about challenges and mitigation strategies to run high-end mission-critical Java™ Platform, Enterprise Edition (Java EE) based Business Suites.

Agenda

Motivation and Challenges

Mitigation Strategies to address

- Hardware Failure/Disaster Recovery
- JVM™ Software Crashes
- Uncontrolled Resource Consumption

Conclusions

Agenda

Motivation and Challenges

Mitigation Strategies to address

- Hardware Failure/Disaster Recovery
- JVM™ Software Crashes
- Uncontrolled Resource Consumption

Conclusions

“We believe that the boat is unsinkable.”

Philip Franklin
Vice-President of White Star Line
8:00 a.m.
April 15th, 1912

Source: http://users.senet.com.au/~gittins/quotes.html

Core Definitions

- **“High-End”**—refers to the best and generally most expensive of a class of goods or services
- **“Mission-Critical”**—refers to systems, which are of crucial importance for an entire venture

Source: Wikipedia

Core Definitions (Cont.)

- Availability
  - A service is regarded as available when it can fulfill the tasks it has been designed for
- High-Availability (HA)
  - High availability is reached with additional measures as redundant components and should guarantee a continuity of work if one component fails
- Business Continuity
  - Business continuity can be reached with measures that guarantee a continuity of work even if an entire location fails (used here as a synonym for disaster recovery)

Source: SAP

Real World Example

Lufthansa

Goals

- Role-based and on-demand information provision for all employees
- Networked knowledge management across organizational barriers
- Ease of work through improved business processes
- Optimization of resource deployment and efficiency gains

Passage

Logistics

MRO

Catering

Tourism

IT Services

Financial Services

Top-tier international airline, with world-wide operations in cargo, catering, MRO, and IT consulting and development.

Key: MRO - Maintenance, Repair and Overhaul

Technical View

Lufthansa

- SAP enterprise portal
  - High-End and Mission-Critical
  - Licensed users: > 73,000
  - Productive system: 2 x Sun V880, 1 x Sun V480
  - 34 Java VMs (in productive system)
  - Java VM details
    - 2.3 GB heap space per JVM software
    - ~ 30 additional Lufthansa applications

Source: Lufthansa

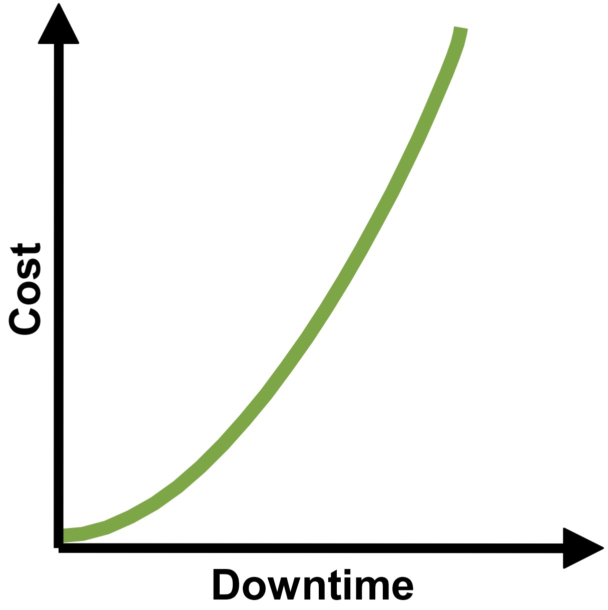
Scaling High Availability

The high availability scale

Availability	Downtime per week	Weekly downtime usable for (examples)	Downtime per year	Yearly downtime usable for ….
99,9999%	0,6 sec	??	30 sec	??
99,999%	6 sec	Weekly fast switchover	5 min	1 yearly restart (?)
99,99%	1 min	Daily fast switchover	52 min	One offline software maintenance per year
99,9%	10 min	1 Weekly restart	8 hours 45 min	One offline backup per year
99%	1h 40 min	Offline software maintenance	87,5 hours	
90%	16h 48 min	1 Offline backup per week	36 days	

ACHIEVABLE TODAY

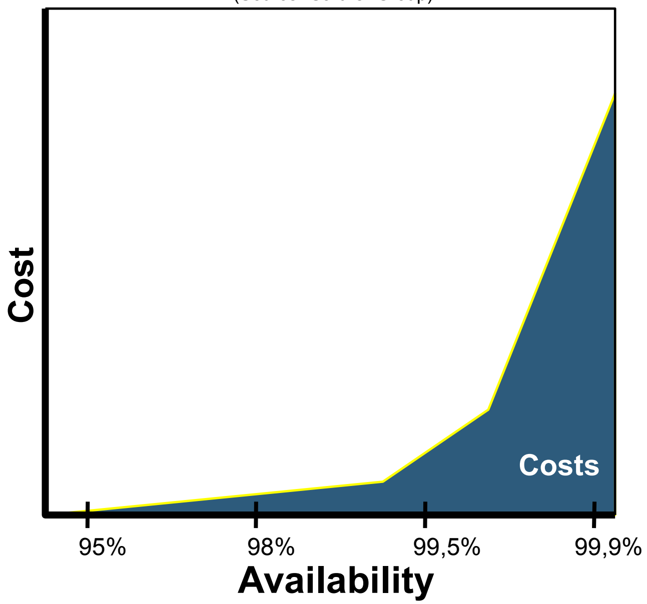
Cost of Downtime vs. Availability



- Downtime costs not linear
- For longer downtimes costs increase progressively
- Car industry example: system downtime for 1 minute causes the whole production to stop

Availability Cost Curve

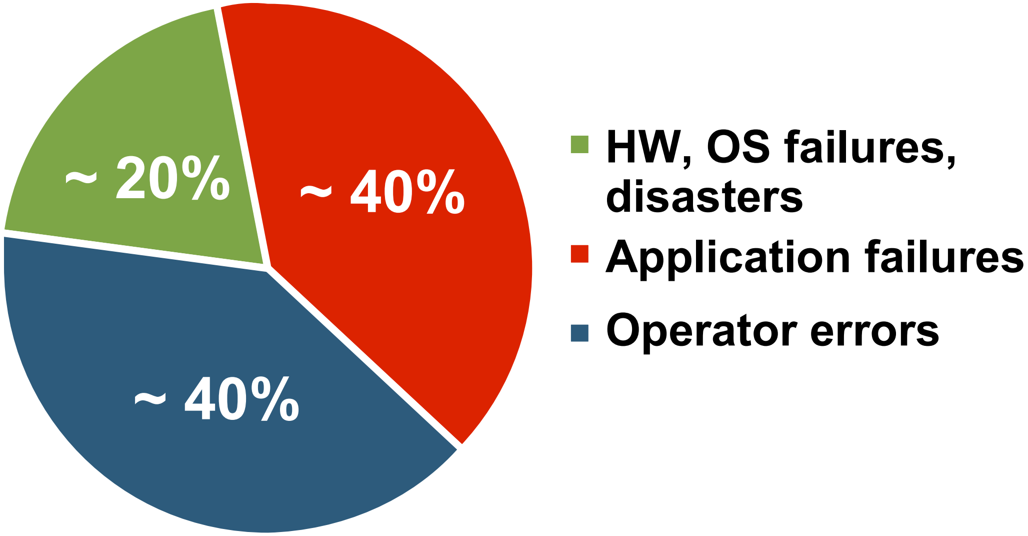
(Source: Gartner Group)



- Redundant components
- System management tools
- IT staff
- Guaranteed SLA
- Disaster recovery site

Source: SAP Technology Consulting

Causes for Downtimes



- HW, OS failures and disaster impact to be avoided by eliminating Single Points of Failure (SPOF)
- Application failures to be tackled by mitigation strategies
- Operator errors to be addressed by automated system management processes!

Source: Discussion with CIOs of leading enterprises

First Conclusions

- Enterprises demand for business continuity.
- High Availability strategies are typically beyond the scope of application development.
- Additional measures have to be taken to protect the application server for high-end enterprise scale computing

So what are effective mitigation strategies?

Agenda

Motivation and Challenges

Mitigation Strategies to address

- Hardware Failure/Disaster Recovery
- JVM Software Crashes
- Uncontrolled Resource Consumption

Conclusions

Agenda

Motivation and Challenges

Mitigation Strategies to address

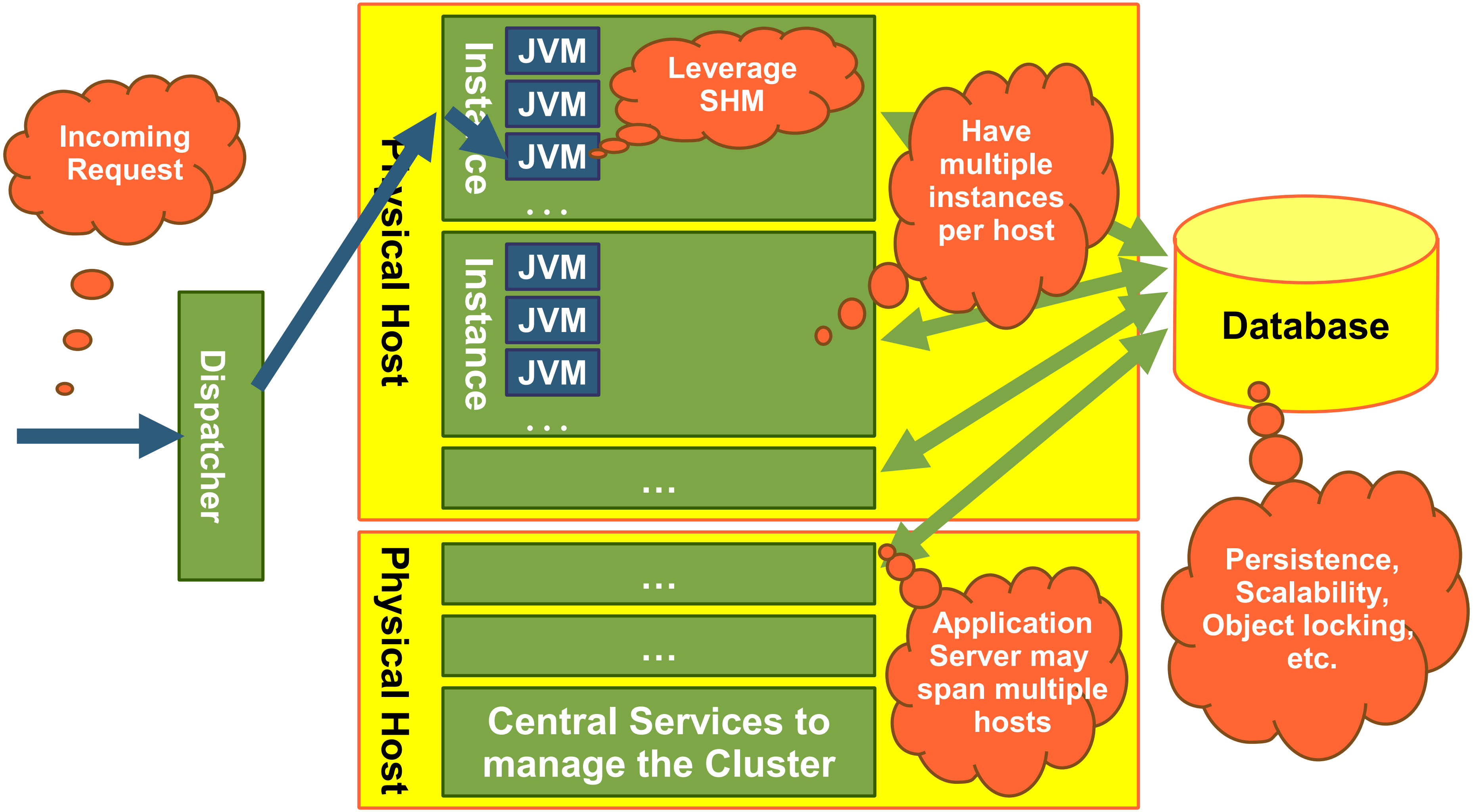
- **Hardware Failure/Disaster Recovery**
- JVM Software Crashes
- Uncontrolled Resource Consumption

Conclusions

Hardware Failure/Disaster Recovery

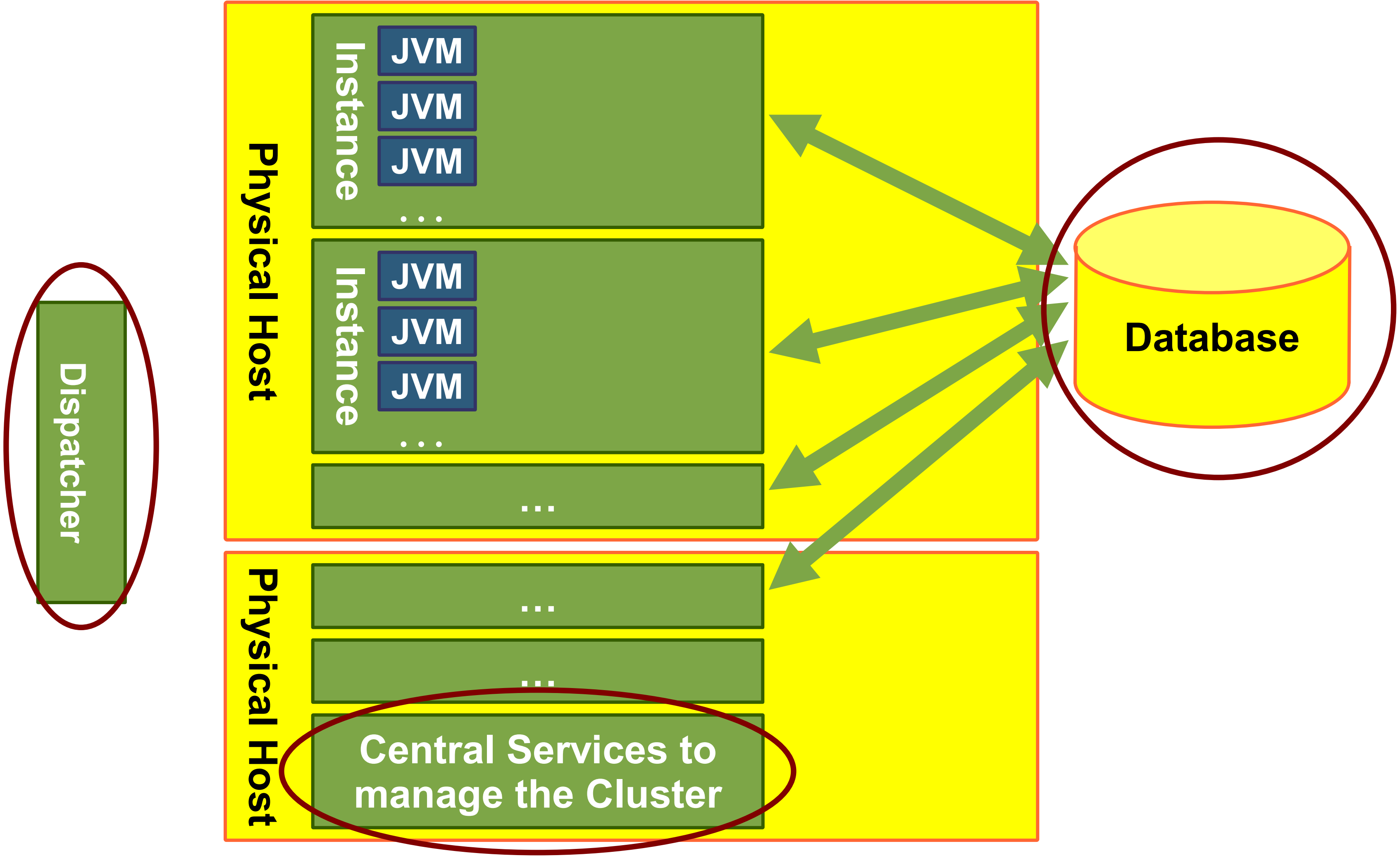
- A generic Server Infrastructure
- Securing Single Point of Failures
- Disaster Recovery

A Generic Server Infrastructure


Incoming Request
Dispatcher
Physical Host
Instance
JVM
JVM
JVM
. . .
Leverage SHM
Instance
JVM
JVM
JVM
. . .
…
Have multiple instances per host
Database
Physical Host
…
…
Central Services to manage the Cluster
Application Server may span multiple hosts
Persistence, Scalability, Object locking, etc.

Server Infrastructure Components

Single Points of Failure (SPOF)



How to Secure Single Point of Failure

Single Point of Failure	Technical Possibilities to eliminate the Single Point of Failure
Central Database	• Switch-over solutions
Central Services/ Dispatcher	• Switch-over solutions • Special Care: Services with local persistence
Central File share	• Cluster file share by switch-over solution • NFS file share • Highly available file system (Hardware based highly available Storage Solution)

Disaster Recovery

- Effective strategies include:
  - All dynamic changes must be replicated (on database and/or file system level) to remote locations
  - Business view: synchronization of application data with remote site
  - Never works out-of-the-box, project work!

Agenda

Motivation and Challenges

Mitigation Strategies to address

- Hardware Failure/Disaster Recovery
- **JVM Software Crashes**
- Uncontrolled Resource Consumption

Conclusions

JVM Software Crashes Caused by Application Failures

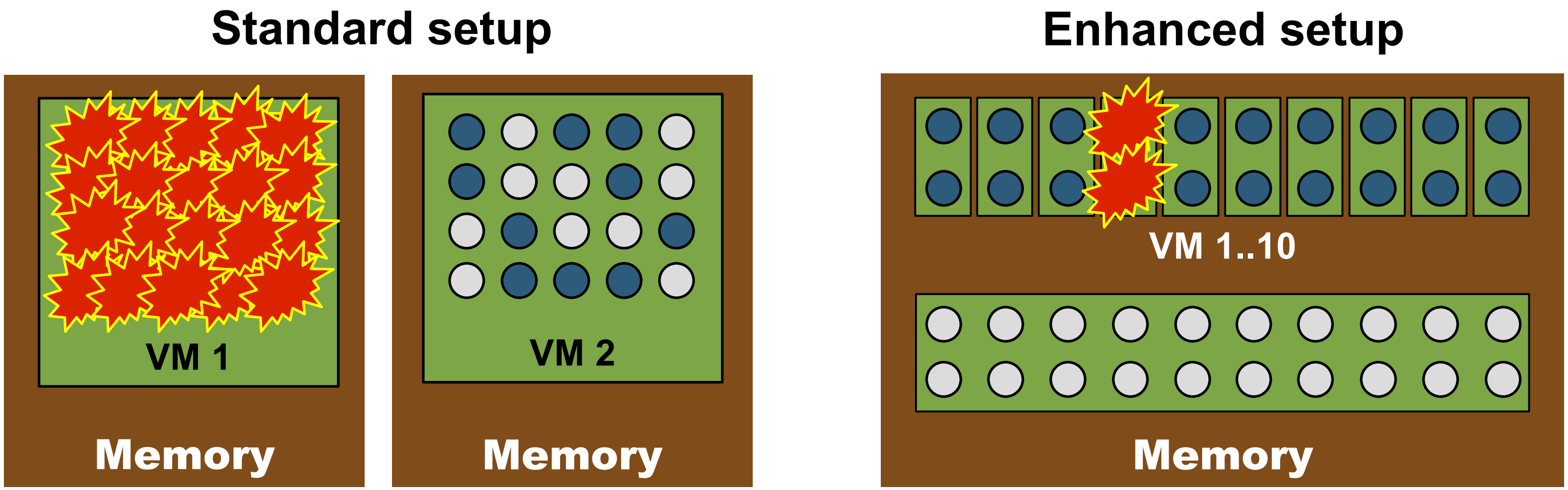
- Reduce the impact of JVM software Crashes
- Regular monitoring/reporting
- Post mortem JVM software Analysis

Reduce the Impact of VM Crashes

- User Session Isolation
  - Reduce number of user session in each JVM software
  - Separation of active and inactive sessions
- Application Isolation
  - Protect applications/scenarios from each other
  - Possible approaches:
    - Enhanced load balancing (in the dispatcher)
    - WSRP-based portal content
    - Web Services
    - Message queue separation
    - JSR 121: Application Isolation API

User Session Isolation

The following picture illustrates both concepts



Standard setup

Enhanced setup

VM 1

VM 2

VM 1..10

Memory

Memory

Memory

○ = Inactive Session

● = Active Session

See also: TS-8486 Unbreakable and Self-Adaptive Java™ EE Application Service

Regular Monitoring/Reporting

- Regular monitoring of critical data
  - E.g. memory consumption, load, response times, statistics, cache size, etc.
  - Integration with Common Information Model (CIM)
- Reporting
  - Early watch—focus on the past:
    - Compare monitored data with predefined thresholds and throw alerts
  - Forecast analysis—focus on the future:
    - Predict potential VM crashes in the future by examining/analysing historical data
- Future direction: cross component monitoring

Post Mortem VM Analysis

- Heap dump (of live objects and classes)
  - J2SE™ 1.2: Java VM Profiling Interface (JVM PI)
  - J2SE 5.0 (Tiger): Java VM Tool Interface (JVM TI)
  - Java SE 6 (Mustang):
    - Build-in heap dumper/jmap (analyse out-of-memory errors)
    - Generate heap dumps from core file (not always possible due to possible heap corruptions)
- Enhanced monitoring
  - Keep monitoring data outside the VMs, e.g. SHM, FS, DB, etc.
- Flight recorder
  - Record the last 5–10 minutes of relevant data, e.g. VM, Garbage Collector, DB pool, etc.

Agenda

Motivation and Challenges

Mitigation Strategies to address

- Hardware Failure/Disaster Recovery
- JVM Software Crashes
- **Uncontrolled Resource Consumption**

Conclusions

Uncontrolled Load and Resource Consumption

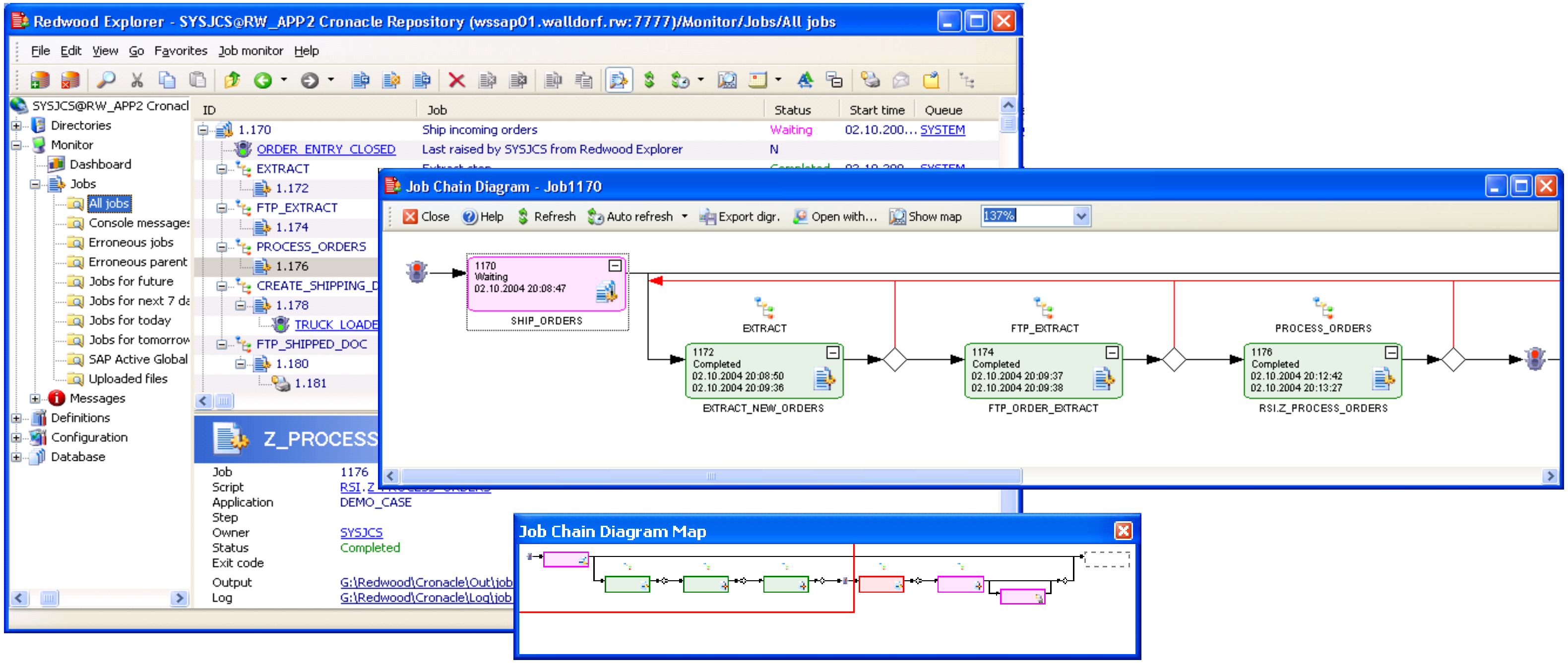
- Memory leaks
- Reduce/control the number of threads
- Server adaptation

Memory Leaks

- Entire application footprint is not clear to the developer
- Memory leaks are back! (but they are now called unnecessary/unintentional object retention)
  - No GC strategy can solve the issue of memory leaks
  - Sometime not obvious, e.g. WeakHashMap
    J2SE 5.0 API Specification:
    “The value objects in a WeakHashMap are held by ordinary strong references. Thus care should be taken to ensure that value objects do not strongly refer to their own keys, either directly or indirectly, since that will prevent the keys from being discarded.”
- Monitoring/profiling of special resources, e.g. sessions, buffers, pools, caches
  - E.g. Java Technology Heap Analysis Tool (jhat)
- Challenge when debugging memory leaks: complexity and large number of applications
- Mitigation: Node aging

Reduce/Control the Number of Threads

- Thread management
- Asynchronous processing, e.g. through Java Message Service (JMS)
- Implementing Jobs, but what is a Job?
- Local and enterprise wide scheduling



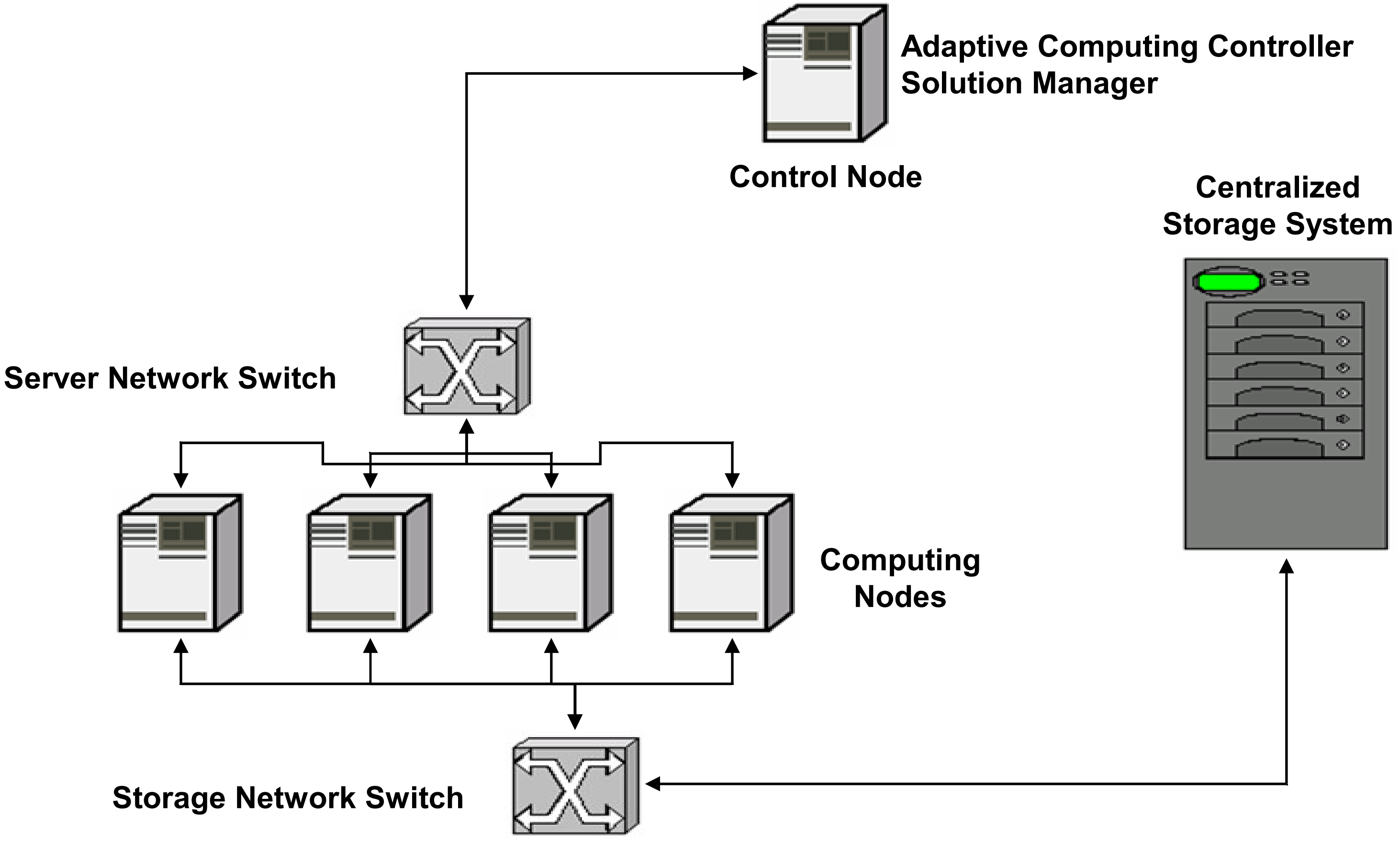
ID	Job	Status	Start time	Queue
1.170	Ship incoming orders	Waiting	02.10.200...	SYSTEM
ORDER_ENTRY_CLOSED	Last raised by SYSJCS from Redwood Explorer	N		
EXTRACT				
1.172				
FTP_EXTRACT				
1.174				
PROCESS_ORDERS				
1.176				
CREATE_SHIPPING_D				
1.178				
TRUCK_LOADE				
FTP_SHIPPED_DOC				
1.180				
1.181				

Z_PROCESS

Job 1176
Script RSI.Z
Application DEMO_CASE
Step
Owner SYSJCS
Status Completed
Exit code
Output G:\Redwood\Cronacle\Out\job
Log G:\Redwood\Cronacle\Log\job

Server Adaptation

Adaptive computing



Agenda

Motivation and Challenges

Mitigation Strategies to address

- Hardware Failure/Disaster Recovery
- JVM Software Crashes
- Uncontrolled Resource Consumption

Conclusions

Agenda

- Motivation and Challenges
- Mitigation Strategies to address
  - Hardware Failure/Disaster Recovery
  - JVM Software Crashes
  - Uncontrolled Resource Consumption
- **Conclusions**

Summary

- High-availability is a must to ensure business continuity of mission-critical business suites!
- Promising Risk Mitigation Strategies include:
  - Minimize impact of a Java VM Crash!
  - Control unexpected workloads!

For More Information

- SAP NetWeaver Application Server Java Technology: https://www.sdn.sap.com/irj/sdn/developerareas/java
- Norbert Kuck: Increasing the Robustness of the Java™ Virtual Machine, 2005 JavaOneSM Conference, TS-7179
- Thomas Smits: Unbreakable Java, JDJ Volume 9 Issue 12 (12/2004), http://jdj.sys-con.com/read/47362.htm
- Henning Blohm, Dirk Marwinski: Unbreakable and Self-Adaptive Java™ EE Application Service, JavaOne Conference 2006, TS-8486

Q&A

Peter Kulka—SAP AG

Peter Emmel—SAP AG

Thank You!

High-End Java™ EE Platform-Based Application Servers for Enterprise Scale Business Suites

Peter Kulka

Solution Architect
SAP AG
peter.kulka@sap.com

Peter Emmel

Evangelist
SAP AG
peter.emmel@sap.com

TS-4830