



10 Tricks to Ensure Your Oracle Coherence Cluster is Not a “Black Box” in Production

Tom Lubinski, Founder and CTO
SL Corporation

July 2010



About SL Corporation

- A Leader in Real-time Application Performance Monitoring
- Located near San Francisco, CA with office in Tokyo
- RTView platform for APM and operational visibility
- Oracle Coherence Monitor (OCM) and Viewer (OCV) based on RTView



RTView – Partial Customer List



SL - Background

Extensive background in real-time process monitoring

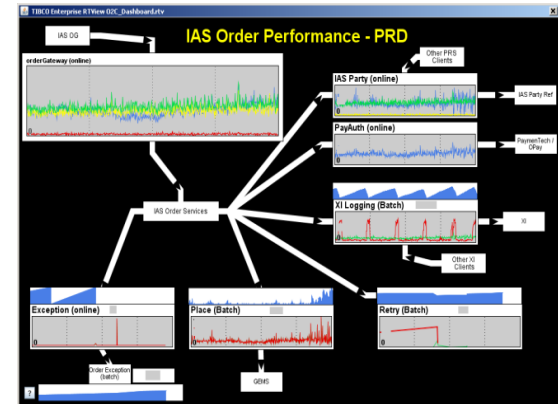
Large volumes of dynamic data

Visualization technologies

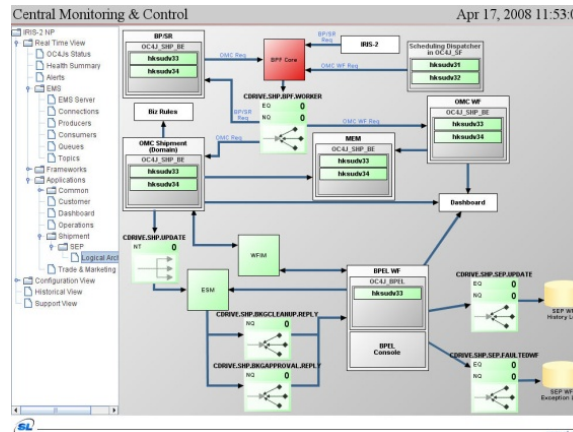
Mission-critical applications



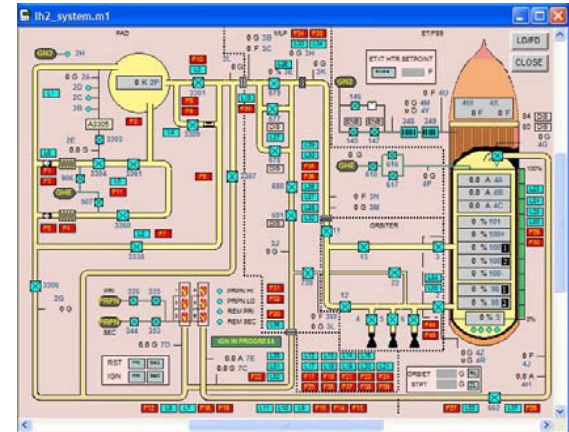
Connecticut Valley Power Grid Management System



Critical Tax Season Applications at Intuit



OOCL World Wide Shipment Tracking



NASA Space Shuttle Launch Control System

■ Disclaimer (1)

- **Built Coherence monitoring product in 2006**
- **We've learned a lot**
- **We don't know everything**
- **We could be wrong**

■ Disclaimer (2)

- This stuff is *boring* **-Z Z Z-**
- Questions, banter, even heckling
are welcome



Why Monitor Coherence (and Applications) ?

*... so they stay up and are
fast and reliable*

(of course)



What typically happens ...

The Three Phases of a Coherence Project

Why Monitor Coherence (and Applications) ?

Your Coherence Vision

Faster, better, smarter applications

Large volumes of in-memory data

Fast caching to buffer database

In-location parallel processing



Why Monitor Coherence (and Applications) ?

Development

Download, install, experiment

Initial results promising

Difficult to test full user and data load

Pressure to deploy quickly

Monitoring takes back seat



Why Monitor Coherence (and Applications) ?

Reality

User and data load causes problem

Latency, bottlenecks, timeouts

Did Coherence survive my changes ?

Which app is causing excessive load ?

What events lead up to the problem ?

How do I tell Oracle what happened ?

Difficult to change once in production





There is an ...
ALTERNATIVE *Reality*

“Black Box” is opened up

Alerts notify users of problems

Monitoring metrics at your fingertips

Event history shows what happened

Life is good



~~Ten~~ *Eleven* Things You Can Do

■ Ten Things You Can Do

0 - Change the mindset (most important):

~~Monitoring is a
"Nice-to-have"~~

← NO !!!

Monitoring is a
"Must-have"

← YES !!!

No ... you can't just get by with log files

■ Ten Things You Can Do

1 – It's your code ... not Coherence (usually)

"There must be a bug in the compiler" – Donn Combelic

Understand how your code executes in the cluster

Memory dies ... you replace the memory

App dies ... you change your code (or config)

All "ops" can do is restart the cluster

No "one size fits all" monitoring

■ Ten Things You Can Do

2 – Know JMX and its *sluggish* nature

Coherence is high-performance, JMX is not

Node / Cluster / Service MBean counts low

Cache / Storage counts can be huge ... > 20,000

Monitor the monitor – with JMX easy to get bad data

Separate node for JMX MBeanServer

Separate hardware for monitoring

Don't put management=all on every node

■ Ten Things You Can Do

3 – Understand Network Metrics

Good communication critical to Coherence

Run the datagram test to get throughput baseline

Storage / Process node behaviors different

See Oracle documents re: buffers, switches, etc

Ten Things You Can Do

3 – Understand Network Metrics

Success rates published in JMX are asymptotic:

$$\text{Pub Succ Rate} = \text{Pkts Resent} / \text{Pkts Sent}$$


Location Δ	Id	StorageFlag	Pkts Sent	Delta	Pkts Resent	Delta	Pub Succ Rate	Delta Succ Rate
ProcessTest01	3	0	1,244,750	6,652	16	0	1.00	1.00
ProcessTest02	4	0	15,347	97	1	0	1.00	1.00
RTViewOCMDDataServer	5	0	20,960	142	3	0	1.00	1.00
StoreTest01	1	1	827,372	0	8	0	1.00	1.00
StoreTest02	2	1	842,768	20,109	2,955	2,948	1.00	0.85



Instead, calculate rate from deltas:

$$\text{Delta Succ Rate} = \text{Delta Pkts Resent} / \text{Delta Pkts Sent}$$

Cluster tries to self-correct by reducing rates

Ten Things You Can Do

4 – Identify the Players (Roles)

Location Δ	Id	StorageFlag
ProcessTest01	3	0
ProcessTest02	4	0
RTViewOCMDataServer	5	0
StoreTest01	1	1
StoreTest02	2	1

Track nodes across restarts – use member names
(ids change on restart)

Proxy services on separate nodes
(node metrics apply to proxy only)

Ten Things You Can Do

4 – Identify the Players (Roles)

Do cache operations (get/put) on non-storage nodes
–localstorage=false



← Process Nodes request history

← Storage Nodes request history –
rogue process node cannot be identified here

■ Ten Things You Can Do

5 – Configure Services for Monitoring

Define one service for each important cache
(or group of related caches)

Service Metrics:

- Cpu load
- Requests
- Messages
- Task Backlog
- Requests Pending
- etc.

Cache Metrics:

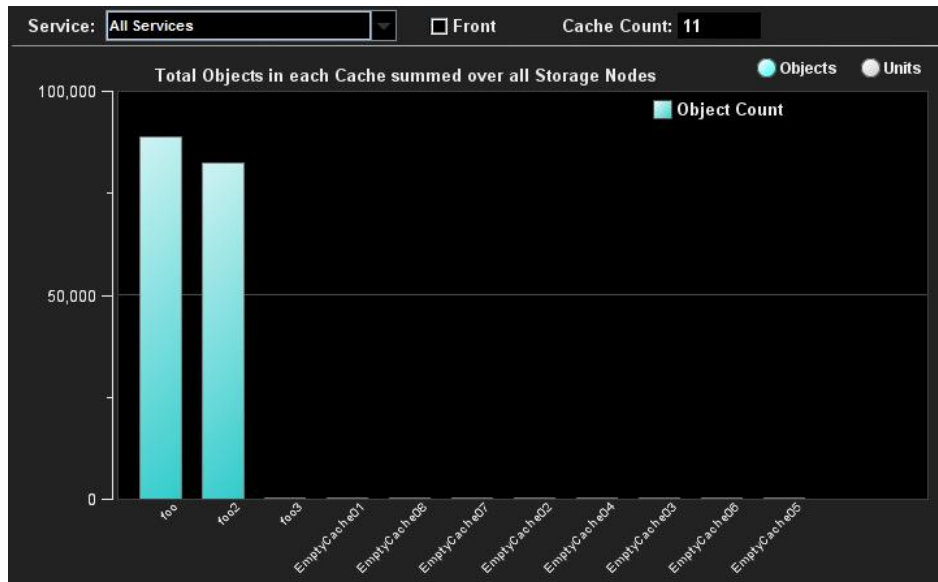
- Total Gets/Hits/Misses
- Total Puts
- Hit Rate
- Store Writes/Reads
- etc.

Especially important for entry processors

Ten Things You Can Do

5 – Configure Services for Monitoring

With a few large caches and many small caches -
Use heterogeneous scaling to reduce MBean counts



$MBean\ count = \#nodes * \#caches * 2\ for\ each\ service$

$100\ nodes\ X\ 100\ caches = 20,000\ MBeans$

By running service on smaller # of nodes, fewer MBeans

■ Ten Things You Can Do

6 – Monitor Capacity Carefully

Avoid Out-Of-Memory Errors at all costs

Setup node death on OOME

No more than 30% heap for data

Avoid swapping: total JVM heap < physical memory

■ Ten Things You Can Do

6 – Monitor Capacity Carefully

Binary Unit Calculator useful for cache sizes

Use High Units to limit cache sizes

Binary units not helpful for front caches

Backup data and index sizes not shown

■ Ten Things You Can Do

7 – Stomp Out Cluster Abuse

Service is like database, Cache is like table – Gene Gleyzer

Dynamic Caches != Scratch Space

Create/destroy cache = costly MBean registration

Ten Things You Can Do

7 – Stomp Out Cluster Abuse

Hundreds of Caches costly to monitor = many MBeans

E.g. Cache / Storage MBeans for 10 nodes with 375 caches

JMX MBean Test - Single Row MBeans Debug

Current Refresh Policy: New Refresh Policy:

JMX Query Times

Key ▲	Last Row Count	Last Col Count	Last Exec Time	Avg Exec Time
local Coherence:type=Cache,* 0 *-1 *-	3,753	43	4.24419	4.37048
local Coherence:type=Management 0 RefreshP...	1	1	0.0001	0.00016
local Coherence:type=Node,* 0 *-1 *-	10	58	0.02038	0.03942
local Coherence:type=Service,* 0 *-1 *-	19	40	0.03445	0.03204
local Coherence:type=StorageManager,* 0 *-1 *-	3,753	16	3.40452	3.09635
local RTView:name=DataServerConnections 0 *...	0	0	0.00008	0.00016
local RTView:name=Manager 0 *-1 *-	1	5	0.00011	0.00025

Total Last Execution Time: **7.704 s** Avg: **7.539 s**

Cannot respond any faster than you can get the data

■ Ten Things You Can Do

8 – Refine Monitoring During Testing

Monday Morning Problem – validate after changes

Develop monitoring for your load tests – automated reports

Use monitoring to get familiar with failure modes during test

Separate batch vs. operational scenarios

■ Ten Things You Can Do

9 – Monitor Your Resources

The effect your app has on resources (cpu, mem, network)

CPU monitoring using external tools

Java BCI tools like Wiley useful

Network signature tools

End user experience monitoring tools

Develop holistic view of app and Coherence component

■ Ten Things You Can Do

10 – Instrument Your Applications

Coherence JMX provides a lot of info, but much is missing

No info about time taken to perform operations or the counts

No way to differentiate requests in the cluster clients

Custom JMX beans useful to augment Coherence

■ Ten Things You Can Do

10 – Instrument Your Applications

Use *custom-mbeans.xml* to specify MBeans in app

```
<mbeans>
  <mbean id="10">
    <mbean-query>MyDemoData:* </mbean-query>
    <mbean-name>type=MyDemoData</mbean-name>
    <enabled>>true</enabled>
  </mbean>
</mbeans>
```

Coherence collects MBeans from each client

■ Ten Things You Can Do

Final Notes

Donn did find one bug in the compiler

Happy clusters are all alike.
Every unhappy cluster is unhappy in its own way.

- var. Tolstoy

Keep your cluster singing ...

It's up to you ... not "ops"



For more information, or to request an evaluation of
RTView for APM | Oracle Coherence Monitor,
please visit www.sl.com.

- Or -

Contact Marie Hoch at:
(415) 927-8510 / marie@sl.com