



IBM Software Group

# Performance Tuning Support

## *Gathering Diagnostic Data*



This session will be recorded and a replay will be available on IBM.COM sites and possibly social media sites such as YouTube. When speaking, do not state any confidential information, your name, company name or any information that you do not want shared publicly in the replay. By speaking during this presentation, you assume liability for your comments.

© 2013 IBM Corporation

# AGENDA

- Must Gather – Why is it important?
- Performance environment – Set up
- Performance issues – What to gather?
- Generic information – checklist
- Use cases
- Links to upload files



## Must Gather – Why is it important?

- Performance PMRs require extensive data for analysis
- Insufficient diagnostic data leads to a delay in resolution
- Replicating performance issues on customer test environment consume time and resources
- Performance issues are likely to happen in Production environment, replication necessary in lower environment





# Common Performance issues – What to gather?

## ▶ **Servers/Services/APIs are slow**

- Expected response time/throughput
- Number of instances of server being run/ number of threads
- SQL DEBUG logs
- VERBOSE logs
- DB reports
- GC logs
- CPU picture on the boxes

**Tip-**

*Use the top command on Unix boxes to get CPU utilization details*

## ▶ **Servers are frozen or almost non responsive**

- 3-4 sets of thread dumps taken at intervals of 20 seconds.
  - kill -3 <pid>
- Results of blocking lock query
- DB reports
- GC logs



# Common Performance issues – What to gather?

## ▶ *Out of memory issues*

- Memory parameters used to run server
- Heap dumps and java core files
- GC logs
- Server logs that showcase error trace

**Tip –**

To produce a heapdump in the event of an `OutOfMemoryError`, use `XX:+HeapDumpOnOutOfMemoryError`

## ▶ *Memory leaks*

- GC logs to check the leak pattern
- 3 - 4 Forced heap dumps taken from the server every 30 minutes
  - For IBM JVM, set export `IBM_HEAPDUMP=true` and run `kill -3 <PID>`

## ▶ *DB Connection leaks*

- Details of database connection pooling
- Following properties to be added to customer overrides
  - `jdbcService.detect_leaked_connections=true`
  - `jdbcService.detect_leaked_statements_and_resultsets=true`



# Common Performance issues – What to gather?

## ▶ *Excessive blocking locks*

- DB reports
- Results of running the blocking locks sql
- Note - yfs.yfs.app.identifyconnection should be set to Y to identify which agents/servers are causing the locks
- Query available at - <http://www-01.ibm.com/support/docview.wss?uid=swg21614185>

## ▶ *Deadlocks*

- Execution plan for the query throwing the deadlock
- SQLDEBUG logs for 3-4 minutes, showcasing the deadlocks
- Deadlock trace from application
  - yfs.yfs.app.identifyconnection=Y in customer\_overrides.properties file
  - Add JVM parameter -DDLOCK\_LOG\_DIR to All IBM Sterling JVMs.  
Eg: -DDLOCK\_LOG\_DIR=/directory/where/deadlock/logs/should/go/.
  - If schema owner and DB login user are different ,add -DOWNER=<Schema Owner>
- Deadlock Reports from DB



# Common Performance issues – What to gather?

## ▶ *Index recommendations / Slow queries*

- Explain plan of the query
- Indexes present on the tables that are part of the query
- Number of records in the tables being queried
- DB reports / DB CPU

## ▶ *Issues with Hot SKU*

- HOT SKU properties
- Verbose Logs from the servers where the issue is seen
- Messages logged by Hot SKU –

Turning/retaining Item: [Acme:SKUA:EACH:A] into a hot sku as abnormal lock count has increased now to:4: YFSAvailHotSKUItem





# Generic Checklist

- ▶ FP version of the product
- ▶ If this issue has been seen earlier?
- ▶ Any changes made recently that could have caused this issue to occur?
- ▶ For issues related to JMS/ DB, please provide details of versions of JMS and DB
- ▶ For issues seen in Production, back up of server logs, GC logs et to be taken



# Use cases

## *reserveAvailableInventory taking more time than expected in Production*

- Logs were provided for single call of the API from test environment
- Slowness was not noticed in the logs provided with single call
- Support requested for logs under load where slowness was seen along with GC logs and CPU picture on the box
- Verbose logs under load were then provided
- GC logging was not enabled in Production
- Issue was then confirmed to be due to full GCs and recommendations made to avoid this





# Use cases

## *createOrder Integration server throwing OOM*

- Server was run with 2 GB heap
- Heap dumps from crash were provided
- Analysis showed only 500 MB of heap memory being utilized
- GC logs requested
- Nothing alarming deduced from the GC logs, available memory was again at 500 MB
- Issue with the cron job that was starting the server, reading the start up script from a different path





# Links to submit files to Support

## ▶ *Uploading files directly to ECuRep using FTP – for files > 2GB*

- From FTP client , ftp ftp://ftp.ecurep.ibm.com/
- Login as anonymous, Enter your e-mail address as the password.
- cd toibm/<IBM product>
- Enter the following command to enable binary mode for the FTP session:  
binary
- The file to be uploaded must use the format - xxxxx.bbb.ccc.yyy.zzz where
  - xxxxx is the PMR number
  - bbb is the Branch
  - ccc is the Country code
  - yyy is a unique file name
  - zzz is the file type or file extension
  - Example - 12345.055.000.collector01\_29\_Aug\_06.zip
- put file\_name - put 12345.055.000.collector01\_29\_Aug\_06.zip
- Quit to end the FTP session
- Refer - [http://www-05.ibm.com/de/support/ecurep/send\\_ftp.html](http://www-05.ibm.com/de/support/ecurep/send_ftp.html)



# Questions?

*For further queries, email me at –  
[Aparna\\_Subramanian@in.ibm.com](mailto:Aparna_Subramanian@in.ibm.com)*

