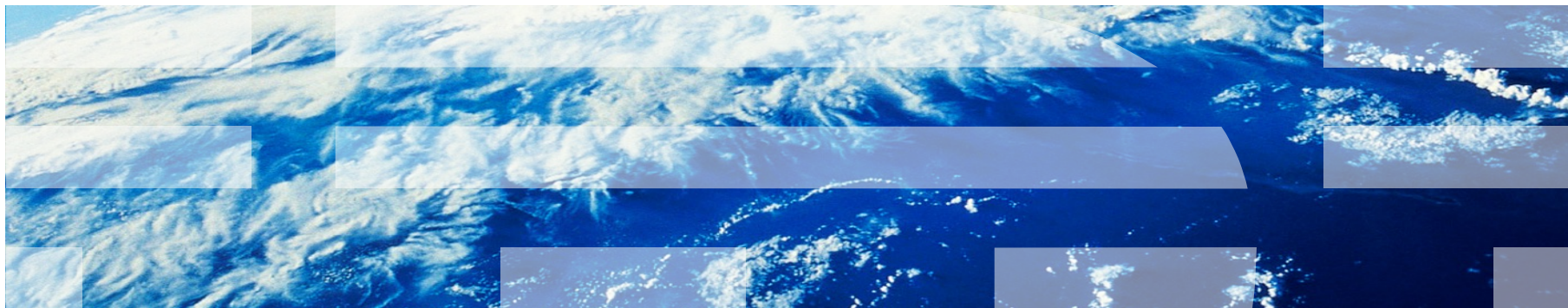


IBM Operational Decision Manager V8.11.x

Tuning Guide

Authors : Pierre-André Paumelle, Nicolas Peulvast



© IBM Corporation 2022. All Rights Reserved.

This presentation has been prepared by IBM and individual employees and reflect their own views. It is provided for informational purposes only, and is neither intended to, nor shall have the effect of being, legal or other guidance or advice to any readers. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other materials. Nothing contained in this presentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries. For a complete list of IBM trademarks, see www.ibm.com/legal/copytrade.shtml
AIX, CICS, CICSplex, DB2, DB2 Universal Database, i5/OS, IBM, the IBM logo, IMS, iSeries, Lotus, OMEGAMON, OS/390, Parallel Sysplex, pureXML, Rational, RCAF, Redbooks, Sametime, Smart SOA, System i, System i5, System z, Tivoli, WebSphere, and z/OS.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Product Versions

- IBM Operational Decision Manager V8.11.0
- WebSphere Application Server ND 9.0.0.9
- WebSphere Liberty 21.0.0.9
- Java 8 sr6
- DB2 V11
- Cloud Pak for Business Automation 21.0.3 on OpenShift 4.7
- Oracle 12.1.0.2.0 Enterprise Edition

Rule Execution Server Tuning Recommendations

Basic Tuning : Rule Execution Server

- The **TraceLevel** flag should be set to Severe or Warning in the production environment. That improves performance. This property is accessible in the resource adaptor or in the ra.xml.
- The **enableInterceptor** flag should be unset to avoid a performance cost if you don't need interceptor. This property is accessible in the resource adaptor or in the ra.xml.
- Check the version of the ruleSession library.
- Java2security should be off if possible especially on JEE.
- Verify that your Application Server is not in Debug mode
- The optimal pool size depends on two dimensions:
 - the number of possible concurrent connections
 - the number of rulesets to execute.
 - The first performance measurement should be performed with the default size. The limit is reached by the memory consumption. A sizing method is available at :

<http://www-01.ibm.com/support/docview.wss?uid=swg21400803>

Basic Tuning : Rule Execution Server (Classic Rule Engine)

JVM Tuning

Memory

- Additional memory can improve performance but the recommended starting heap size is:
 - 1.2 GB per ODM server for 32 bit systems
 - 4GB to 8GB per ODM server for 64 bit systems

Garbage Collector

- gencon is the policy which gives the best results
- Limit the number of threads for garbage collector if you have more than 6 cores
 - -Xgcpolicy:gencon -Xgcthreads6

JIT configuration

- The JVM supplied with WAS 8.0.0.5 and further versions generates more JIT code than previous versions, which can lead to longer warm-up times and lower throughput.
- The JIT can take time to compile a large ruleset, but this behaviour can be modified by increasing the cache or by limiting the inlining on ODM classes.
 - -Xjit:{ilog/rules/engine/sequential/generated/*}

Limit the inlining for one of the ODM package

Basic Tuning : Rule Execution Server (Decision Engine)

JVM Tuning

Memory

- Additional memory can improve performance but the recommended starting heap size is:
 - 1GB per ODM server for 32 bit systems
 - 1GB per ODM server for 64 bit systems
- The new Decision Engine use less memory than Classic Rule Engine
- For existing Rule Projects that want to use the new Decision Engine, we recommend to make a new memory tuning, so it will reduce allocated memory on the server.

Garbage Collector

- gencon is the policy which gives the best results
- Limit the number of threads for garbage collector if you have more than 6 cores
 - `-Xgcpolicy:gencon -Xgcthreads6`

JIT configuration

- No particular JIT configuration is needed.

Assertion configuration

- Decision Engine is using assertions. So, disable assertions could be useful (`-da` or `-da:com.ibm.rules.* vmargs level`)

Basic Tuning : Rule Execution Server

- There are two main ways to optimize Decision Warehouse:
 - Decision Warehouse is configurable at a ruleset base with filter properties. With this, you can choose the information to save.
 - Decision Warehouse is customizable. Further customization can be done through the Decision Warehouse extension point for better execution performance. You can define how data can be persisted or queried through the extension point. An asynchronous version of decision warehouse for WAS8 is available as a contribution at:

<http://www-01.ibm.com/support/docview.wss?uid=swg21433167>

- There are two main ways to optimize BAI Emitter:
 - BAI emitter is configurable at a ruleset base with filter properties. With this, you can choose the information to send.
 - The BAI emitter is configurable at RES plugin level see the ra.xml or the helm. See [BAI tuning guide](#)

Advanced Tuning : Rule Execution Server

- Tune the execution trace, Decision Warehouse and BAI by filtering.
- Limit the size of your XOM to useful classes
- A ruleset on an XML XOM should be configured to run in multiple simultaneous executions. This tuning should be done at two levels:
 - By configuring the pool of XML document drivers at ruleset level. This is configured using the ruleset property `ruleset.xmlDocumentDriverPool.maxSize` (default value 1).
 - By configuring the connection factory property `documentBuilderPoolProperties` with a value like `{pool.maxSize=20}` (default value 10), as described at:
https://www.ibm.com/support/knowledgecenter/SSQP76_8.9.2/com.ibm.odm.dserver.rules.res.managing/topics/con_res_config_rule_session_conx_pools.html
- Use the ruleset caching information (XU dump) from the Rule Execution Server console to get the status.
- DecisionID : if the DecisionID is set in the request, so it will optimize the response time.
- In case of batch application, you should consider stateful `ruleSession` API to ensure the same ruleset for the complete batch execution and to avoid the search of an engine for each single rule execution.

Advanced Tuning : Rule Execution Server

- The XU dump could be generated every x seconds by configuring the connection factory (ra.xml in JSE case)
 - xuDumpSchedulerEnabled should be set to true
 - xuDumpSchedulerInterval is by default to 10000.
 - Every 10s the RES saves a XU dump.
- This configuration has an impact on performance and should be set for debug purpose only
- When activated, the XU will create a directory named 'xudumps' and put all dumps in it. It is not working on TWAS but on Liberty

Advanced tuning: Rule Execution Server

- Minimizing the overhead of ruleset parsing
 - Parsing occurs at first invocation of the ruleset. To minimize the affect of this, the ruleset can be parsed before the first execution. This can be done with a dummy execution or a call of the `loadUptodateRuleset` (Management Session API).
 - When a new version of a ruleset is deployed a parse is launched which blocks the executions. This can be fixed by using the asynchronous parsing.
 - Ruleset is used infrequently, and removed from the pool running other rulesets. This can be fixed by setting `ruleset.maxIdleTime` to a number > 0

Advanced tuning: Rule Execution Server

Tuning of a RES JSE

Tomcat, Liberty, JSE application

- The connection manager should set a timeout to avoid the exception pool full when the RES is overloaded.

```
<config-property>  
<config-property-name>defaultConnectionManagerProperties</config-property-name>  
<config-property-type>java.lang.String</config-property-type>  
<config-property-value>pool.maxSize=50,pool.waitTimeout=3000</config-property-value>  
</config-property>
```

Advanced tuning: Rule Execution Server

Tuning on WebSphere Liberty Profile

- The tcpOptions element (tcpOptions soReuseAddr="true") is added in the server configuration to enable immediate rebind to a port with no active listener and improve the throughput of the server.
- The JSE RES is embedded in every applications using ODM.
- The pool size, the traceLevel and the enableinterceptor should be configured in every application (htds, customer application) .
- Enableinterceptor should be unset

Advanced tuning: Rule Execution Server

- Using Rule Execution Server with Apache Spark Batch
 - The procedure to use ODM in Spark is described in this article:
 - <https://developer.ibm.com/odm/docs/solutions/odm-and-analytics/odm-business-rules-with-apache-spark-batch-operations/>
 - Within SPARK the access to the errors or the exceptions is difficult
 - Before using a ruleset and a code inside SPARK you must validate it in a standalone JSE environment
 - Verify that you don't set any ruleset property related to Decision Warehouse.
 - Verify the classpath used inside Spark

Decision Center
Tuning Recommendations

Basic Tuning : Decision Center

JVM Tuning

Memory

- Additional memory can improve performance but the recommended starting heap size is:
 - 4GB to 16GB per ODM DC server for 64 bit systems
 - 8GB is good starting point

Garbage Collector

- gencon is the policy which gives the best results
- Limit the number of threads for garbage collector if you have more than 6 cores
 - -Xgcpolicy:gencon -Xgcthreads6

▪ Flag to support UTF8

- -Dclient.encoding.override=UTF-8

▪ Java2security should be off if possible especially on JEE

▪ Apply the session management configuration for WAS described at

- [Logging out results in a WebSphere Application Server error message](#)

Basic Tuning : Decision Center

- Limit memory consumption
 - Use Automatic build to avoid a ruleset generation cost at “first ruleset generation”.
 - Set in Administration > Settings “Type of storage to use as a temporary buffer when generating a ruleset archive” to **File** instead of **Memory**.
- Decision Engine Bytecode generation
 - For Decision Center, you must have enable the XOM management to be able to activate the bytecode generation
- Debug a ruleset
 - Enable in Administration > Settings ” Include debug information for Decision Engine archives “
- Diagnose a performance issue
 - Enable SQL Logging in Administration > settings
 - Tune the execution duration of the queries to Log

Basic Tuning : Decision Center

■ Business Console Stream configuration

- You can configure it at profil level. It is really useful before the synchronization of a project otherwise this user will follow every artifact of this project

The screenshot shows a 'User Profile' window for a user named 'rtsAdmin'. The window has a dark blue header with the title 'User Profile' and a close button. Below the header, there is a user icon and the name 'rtsAdmin'. Underneath the name, it says 'Groups: rtsAdministrator, rtsConfigManager, rtsInstaller, rtsUser'. There is a 'Change' link next to the user icon. On the left side, there are three sections: 'Display', 'Grid columns', and 'Automatically follow'. Each section has a list of items with corresponding toggle switches on the right. The 'Automatically follow' section is currently expanded, showing a list of items that the user can follow.

Section	Item	Toggle
Display	• Rules, releases, or activities I create	<input type="checkbox"/>
	• Rules I edit	<input type="checkbox"/>
Grid columns	• New rules that I am authorized to edit	<input type="checkbox"/>
	• Releases or activities that I am an approver of	<input type="checkbox"/>
Automatically follow	• Releases or activities that I am the owner of	<input type="checkbox"/>
	• Activities in which I am an author or tester	<input type="checkbox"/>
	• New projects that I am authorized to see	<input type="checkbox"/>
	• New branches that I am authorized to see	<input type="checkbox"/>

Basic Tuning : Decision Center

- Use out-of-the-box features of the product and limit the number of customizations as much as possible: all the provided benchmarks are done on a customization-free product;
- If you meet some performance issue, try to reproduce it without your customizations before opening a PMR in the ODM Support;
- Having custom value provider in your authoring environment is known to be a performance killer feature in the product (authoring and compilation time).
- If you have to interact with the backend, be sure that the session is not used in multi-threaded mode as that could lead to an important additional overhead due to the synchronized blocks in the code.

Advanced Tuning : Decision Center

- Configuration of Solr

- To deactivate Solr :
 - Disable in Administration > Settings “Enable indexing for textual search”
- To limit the impact of Solr
 - Tune the refresh interval in Administration > Settings

Advanced Tuning : Decision Center

■ Decision center cache and clustering

- If the Decision Center Cache is put in a local directory the command API “DELETE/[v1/build/discard.cache](#)” will clean the local directory only, not every caches of each node.
- To avoid that issue you can:
 - Put the cache on a shared disk
 - Run “DELETE/[v1/build/discard.cache](#)” on each node of the cluster

Advanced Tuning : Decision Center

- In case of Error 500.. No session available in the current thread, you should call `lirSession.beginUsage`
 - You have to set the cookie expiration along with the session timeout.
 - Select the current browser session option.

[Enterprise Applications](#) > [teamserver-WAS85](#) > [Session management](#) > [Cookies](#)

Use this page to specify cookie settings for Hypertext Transfer Protocol (HTTP) session management.

Configuration

General Properties

Cookie name

☐ Restrict cookies to HTTPS sessions

☒ Set session cookies to HTTPOnly to help prevent cross-site scripting attacks

Cookie domain

Cookie maximum age

☒ Current browser session

☐ Set maximum age
 seconds

Cookie path

Advanced Tuning : Decision Center

■ Configuration of the datasource

– Datasource custom properties to Set

- Additional properties → WebSphere Application Server -> data source properties
- Prepared statement cache size: 100 (Increase the size the prepared statement cache to 100).
- webSphereDefaultIsolationLevel property should be set to 2
- Use this data source in container managed persistence (CMP) must be unsetted

– Configure the pool:

- Connection timeout: 30
- Maximum connections: 10
- Minimum connections: 5
- Reap time : 60
- Unused timeout : 90
- Aged timeout: 120
- Purge policy : FailingConnectionOnly

– This configuration is a starting point for performance tuning

Advanced Tuning : Decision Center

■ Configuration of the datasource

– Datasource pool size

- The pool size depends on the number of concurrent users.
- A user is using 5 concurrent connections to the database
 - a) Nb users < 5 you should have a pool size of 10
 - b) $5 < \text{nb users} < 10$ you should have a pool size of 20
 - c) $10 < \text{nb users} < 20$ you should have a pool size of 30
 - d) Nb users > 20 the pool size should not be over 50.

Advanced Tuning : Decision Center

■ Datasource configuration examples:

– With JBOSS

```
<datasource jta="false" jndi-name="java:/jdbc/ilogDataSource" pool-name="DCDatasource" enabled="true" use-ccm="false" statistics-enabled="false">
```

```
  <connection-url>jdbc:oracle:thin:@localhost:1521:BD</connection-url>
```

```
  <driver-class>oracle.jdbc.OracleDriver</driver-class>
```

```
  <driver>oracle</driver>
```

```
  <transaction-isolation>TRANSACTION_READ_COMMITTED</transaction-isolation>
```

```
  <validation>
```

```
    <validate-on-match>>false</validate-on-match>
```

```
    <background-validation>>false</background-validation>
```

```
  </validation>
```

```
  <statement>
```

```
    <prepared-statement-cache-size>100</prepared-statement-cache-size>
```

```
    <share-prepared-statements>true</share-prepared-statements>
```

```
  </statement>
```

```
</datasource>
```

Advanced Tuning : Decision Center

■ Datasource configuration examples:

– With LIBERTY

```
<dataSource id="jdbc/ilogDataSource"
  jndiName="jdbc/ilogDataSource"
  statementCacheSize="100"
  isolationLevel="TRANSACTION_READ_COMMITTED"
  purgePolicy="FailingConnectionOnly"
  type="javax.sql.XADataSource">
```

– With TOMCAT

```
<Resource auth="Container"
  defaultTransactionIsolation="READ_COMMITTED"
  driverClassName="com.ibm.db2.jcc.DB2Driver"
  name="jdbc/ilogDataSource"
  username="db2admin" password="password"
  jdbcInterceptors="StatementCache(prepared=true,callable=false,max=100)"
  maxActive="10"
  maxIdle="50"
  minIdle="10"
  suspectTimeout="60"
  timeBetweenEvictionRunsMillis="30000"
  type="javax.sql.DataSource"
  url="jdbc:URL;" />
```

Advanced Tuning : Decision Center

■ Configuration of DB2 (LUW)

– The user space page size, should be at 32Kb. A 4Kb page size is a performance killer.

- `CREATE TABLESPACE USERSPACE1 PAGESIZE 32K BUFFERPOOL BP32K;`
- `CREATE SYSTEM TEMPORARY TABLESPACE USERSPACE1TMPTS PAGESIZE 32K BUFFERPOOL BP32K;`

– To increase the transaction log and optimize DB2 you could use that type of ddl commands:

- `connect to DatabaseName;`
- `AUTOCONFIGURE USING MEM_PERCENT 80 APPLY DB AND DBM;`
- `ALTER BUFFERPOOL IBMDEFAULTBP IMMEDIATE SIZE AUTOMATIC;`
- `ALTER TABLESPACE USERSPACE1 FILE SYSTEM CACHING;`
- `UPDATE DB CFG FOR DatabaseName USING LOGFILSIZ 65535;`
- `UPDATE DB CFG FOR DatabaseName using LOGPRIMARY 100;`
- `UPDATE DB CFG FOR DatabaseName using LOGSECOND 100;`
- You could also send the transaction log on another disk sample on windows
`UPDATE DB CFG FOR DatabaseName USING NEWLOGPATH F:\;`

– To increase the performance in multiuser mode for DB2 10.5 and ODM 8.10.2 and before

- Add `db2set DB2_SKIPDELETED=YES` and `db2set DB2_SKIPINSERTED=YES`
- You should check them for the user used by Decision Center. (use `db2set` to check)

Advanced Tuning : Decision Center

■ Configuration of Oracle

- The number of open cursor should be $> \text{number of max connections} * \text{Prepared statement cache size}$
 - In our case $> 10 * 100 = 1000$
 - A way to configure that : **alter system set open_cursors = 1100**
 - A way get the number of open cursors: **select value from v\$parameter where name = 'open_cursors';**
- You should let the database analyses the tables to improve the performance
- You could use Oracle Enterprise Management to follow or to force the optimization adapted to your repository.
- A performance test should be done at the end of the analyse by Oracle of all the tables.
- To force a full computation of statistics
 - **EXECUTE DBMS_STATS.GATHER_SCHEMA_STATS('SchemaName');**
 - **alter system flush shared_pool;**

Advanced Tuning : Decision Center

- Configuration for Microsoft SQL Server
 - Set in the installation manager the configuration parameter `teamserver.concurrencyGateEnabled = false`
 - Prepared Statement cache should be set at 100
 - A possible optimization is to alter the database with the following command:
 - `ALTER DATABASE <basename> SET READ_COMMITTED_SNAPSHOT ON;`

Advanced Tuning : Decision Center

- Configuration for PostgreSQL
 - Set in the installation manager the configuration parameter `teamserver.concurrencyGateEnabled = false`
 - Prepared Statement cache should be set at 100
- Configuration for MySQL
 - Should upgrade to 5.7 or more. Version 5.6 is really slow with Decision Center
 - Set in the installation manager the configuration parameter `teamserver.concurrencyGateEnabled = false`
 - Prepared Statement cache should be set at 100

Decision Center
Database Migration

Database migration of Decision Center

▪ Step 1 Creation of the new schema

- You should get the extensions used in the old version
- The files to get (BRDX and BRMX) are available in the diagnostic tab or via the following requests:
 - `select * from ODMRTSADMIN88.RTSRESOURCE where name = 'teamserver.extensionModel';`
 - `select * from ODMRTSADMIN88.RTSRESOURCE where name = 'teamserver.extensionData';`

▪ Step 2

Use decisioncenter-api to generate the script to create the schema

- `curl -X POST "https://Myserver:12443/decisioncenter-api/v1/DBAdmin/createschema" -H "accept: */*" -H "Content-Type: multipart/form-data" -F "extensionDataFile=@biggerExtension.brdx;type=" -F "extensionModelFile=@biggerExtension.brmx;type="`

Execution

- `curl -X POST "https://Myserver:12443/decisioncenter-api/v1/DBAdmin/execute" -H "accept: */*" -H "Content-Type: multipart/form-data" -F "sqlScriptFile=@script.sql;type=application/sql"`

Upload of extensions

- `curl -X POST "https://Myserver:12443/decisioncenter-api/v1/DBAdmin/uploadextensionmodel" -H "accept: application/json;charset=UTF-8" -H "Content-Type: multipart/form-data" -F "extensionDataFile=@biggerExtension.brdx;type=" -F "extensionModelFile=@biggerExtension.brmx;type="`

Database migration of Decision Center

▪ Step 3

Generation of Role creation script

- curl -X GET "<https://Myserver:12443/decisioncenter-api/v1/DBAdmin/migrationrole?oldSchemaName=RTS86>" -H "accept: */*" -
- a sql file is generated.
- Run this script with your DB tool.
- If the database used is Oracle you should start Decision Center on the new database to finalize the role initialisation.

Database migration of Decision Center

▪ Step 4

Generation of migration script

– curl -X GET "<https://Myserver:12443/decisioncenter-api/v1/DBAdmin/migration?oldSchemaName=RTS86>" -H "accept: */*"

– a sql file is generated.

– Insure that Decision Center is stopped (especially in Oracle case) on new and old schema

– Run this script.

▪ Step 5 **Recompute** the statistics after the migration before any connection of **Decision Center**

– On Oracle :

- EXECUTE DBMS_STATS.GATHER_SCHEMA_STATS('SchemaName') ;
- alter system flush shared_pool;

▪ Step 6 Restart Decision Center on the new schema

Decision Center Dimensions

Decision Center dimensions

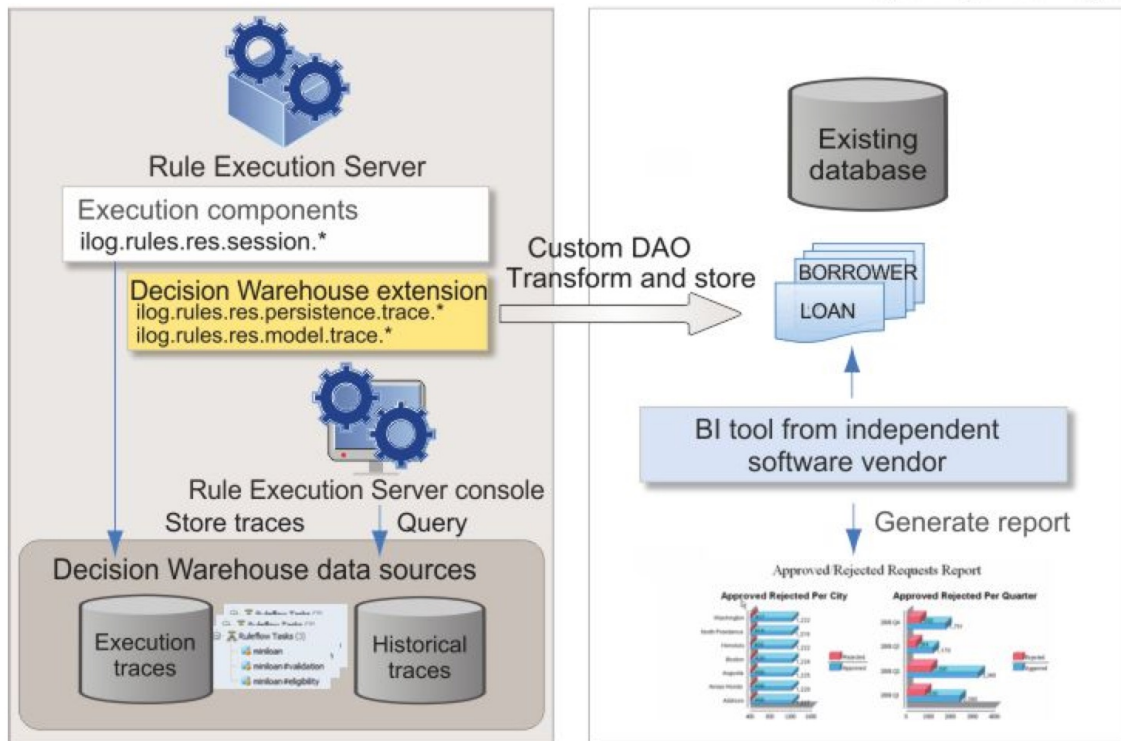
- A Decision table manageable and editable should not have
 - more than 5000 rows
 - more than 50 000 cells
- A Ruleflow manageable should be limited to less than 100 tasks
- A Decision Service with more than 20 projects will be slow to load
- A big ruleset with more 50 000 rules (one row of DT is a rule) could imply a specific Decision Center to avoid timeouts in ruleset generation or Rule tests, sometimes small is beautiful.
- Limit the number of change activities to 200 in a release (DGF)

Decision Warehouse Tuning Recommendations

Decision Warehouse Overview

Enterprise applications
Operational system

Reporting and analysis



Decision Warehouse execution

- Decision Warehouse execution is divided into the following actions:
 - This mode executes the rule with the execution trace on; this into the performance cost.
 - At the end of execution, a serialization of input and output parameters is done (BOM serialization).
 - A complete serialization to XML of the execution trace.
 - A database insertion.

Performance of Decision Warehouse

- In general, the performance of Decision Warehouse depends on the following factors:
 - The performance of generating execution trace.
 - The serialization performance, which divides into two principal cases for the input/output parameter serialization to BOM (depending on the size of the ruleset parameters) and the serialization of the execution trace (depending on the ruleset characteristics, such as the number of rules, the number of rules fired, and so on).
 - The performance of the database server and the network used to access it.

Optimization of Decision Warehouse

- Decision Warehouse is configured inside the ruleset properties. With this, you can choose the information to save. This is accessible in RES Console
- You can remove the BOM serialization. See *Optimizing Decision Warehouse* at the following website:

https://www.ibm.com/support/knowledgecenter/en/SSQP76_8.9.2/com.ibm.odm.dserver.rules.res.managing/topics/tpc_res_config_optim_dw_intro.html

Optimization of Decision Warehouse

- The performance of the Decision Warehouse depends on the database.
 - For example, the database should be optimized to handle CLOB data, not in a generic way, but for the specific case of the ruleset.
 - Further customization can be done through the extension point for better execution performance. You can define how data can be persisted or queried through the extension point.
 - You can also implement your own asynchronous logging of the data captured with Decision Warehouse.

Optimization of Decision Warehouse

■ Asynchronous Decision Warehouse

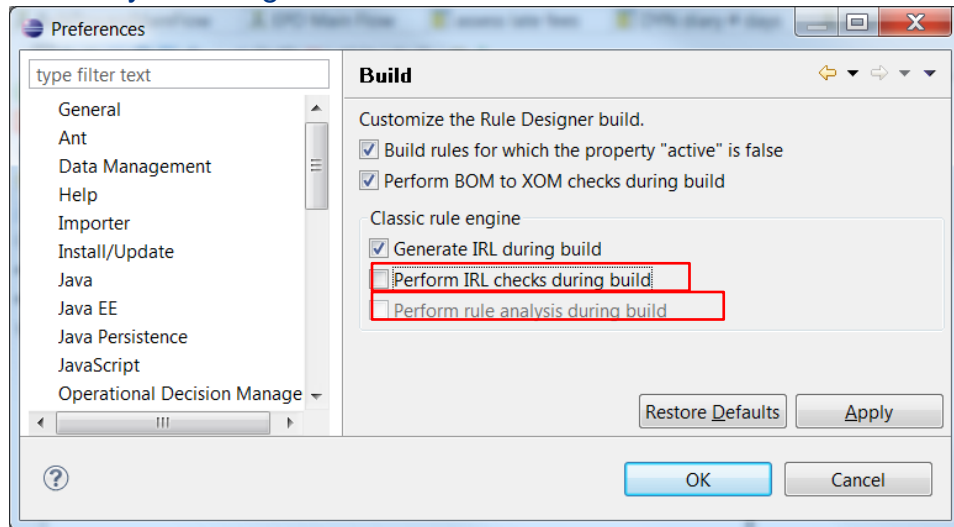
- An asynchronous version of decision warehouse on WAS is available as a contribution at the following website:
 - <http://www-01.ibm.com/support/docview.wss?uid=swg21433167>
 - This implementation is using the WorkManager API to manage the insertion in the database separately from execution
- There is also a sample in the product which shows the reference architecture for using the extension point with JMS

Rule Designer

Tuning Recommendations

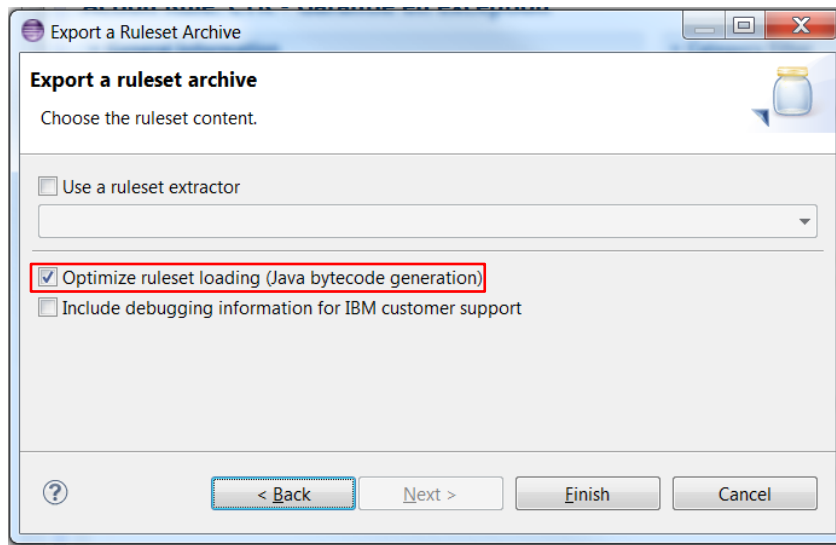
Basic Tuning : Rule Designer

- Configuration to get better build performance
 - Uncheck the 'build automatically' option on large rule projects (eclipse options)
 - On large rule projects close "Rule Project Map" view before the build
 - Rule Designer Menu> Windows> Preferences> Rule Designer > Build
 - Remove perform IRL checks during build
 - Perform rule analysis during build



Basic Tuning : Rule Designer (Decision Engine)

- Configuration to get better loading time with Decision Engine
 - Rule Explorer with selected Rule Project> Right clic> Export> Ruleset Archive
 - For best loading time, leave checked (default) the “Optimize ruleset loading (Java Bytecode generation). To have DE Archive format archive, just uncheck it.



VMWare ESX

Tuning Recommendations for Java

Key indicator of a CPU issue

- **CPU Ready Time should be less 2%**
 - Otherwise there is a performance issue at CPU level.
 - In general this is due to :
 - An overload of the ESX
 - A VM with too many vCPUs

Tuning of an ESX

- 1 Do not allocate too many vCPU on one machine otherwise the impact will on that machine but also on all machines on the same hypervisor
 - <http://www.gabesvirtualworld.com/how-too-many-vcpus-can-negatively-affect-your-performance/>
 - <https://www.virtualease.fr/vmware-mecanismes-de-gestion-cpu/> (in French)
- 2 Desactivate not mandatory processes like indexation of disks or screen saver, becareful with the scheduling of antivirus
- 3 For Database desactivate the antivirus scans on these files You should follow the VMWare best practices.
- 4 Install the drivers "Paravirtual" in the VMs => SCSI controller Paravirtual and network VMXNET3 that a real advantage on CPU consumption , and network bandwidth
- 5 Install VmwareTools and up to date them

Tuning of an ESX

- 6 Configure the hypervisors on mode "Full Performance" in the BIOS + mode "High performance" In the "Power Management Settings" on the ESX
- 7 Pre-allocation of the VMs ' disks (the most critical) In this case the start is slower but the I/O performance are much better
- 8 Group on the same hypervisor the machines which need to communicate a lot (You need to add some "DRS" rules for that), By this way we don't use the network for this communication
- 9 If possible group the VM with the same number of vCPU on the same hypervisors. One important point to notice the smaller VMs are always a better priority when the ESX got a contention at CPU level

Tuning of an ESX

- 10 Be careful the ratio vCPU/pCPU, on a hypervisor should not be more than 2 for production machines (1 for critical machines / 2 for prod / 3 maximum)

You need to compute the total number of vCPU allocated on the machines of the same hypervisor divided the number of physical cores

- 11 On Storage side, the access to the datastores should use several fiber links and several fiber routers the usage of some SSD disks for cache on the NAS are really useful (hybrid mode)
- 12 Follow the "Performance Best Practices VMWARE"

<http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/vmware-perfbest-practices-vsphere6-0-white-paper.pdf>

- 13 "Best Practices Java VMWARE"

<http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/enterprise-java-applications-on-vmware-best-practices-guide.pdf>

BAI for ODM

Tuning Recommendations
BAI Emitter

Tuning : BAI for ODM Memory consideration

- The BAI plugin does not have impact on the performance of a ruleset if it is not used
- You can tune per ruleset what type of BAI event will be sent through ruleset properties
- The BAI plugin is using Kafka so it has impact on the CPU and Memory used by the RES
 - Additional memory can improve performance but the recommended starting heap size is:
 - 4 GB per ODM server using BAI for 64 bit systems
- To limit the memory consumption of the BAI plugin we must use the following configuration
`kafka.max.in.flight.requests.per.connection=1`

Tuning : BAI for ODM CPU and topology

- The usage of BAI emitter could increase the CPU consumption. This consumption depends on the performance of your ruleset.
- The horizontal scalability of the BAI Emitter in the RES is really good
- The vertical scalability of the BAI Emitter in the RES is limited around 5k TPS on a full event
- In consequence several small RES have better performance than a single big one RES using BAI emitter
 - Test of HTDS (JSON) with a ruleset with a response time <1ms with 20 concurrent users
 - With a single RES the performance was limited around 5.6k TPS
 - Using two RES the performance was around 11k TPS
 - So you should run several small RES instead of a big one using BAI

Basic Tuning : BAI for ODM

- For optimal performance, the BAI cluster and the Kafka server should be HA and properly sized to be on par with the ODM event emission rate.
- If it is not possible to put in place a proper HA Kafka server, the performance of ODM can drop when Kafka is down.
 - To mitigate the performance impact in such a case, the configuration flags to use are
`kafka.max.block.ms=1, kafka.acks=0`
 - These settings increase the risk of event loss and should only be used in situations where it is not feasible to ensure the Kafka server is HA and properly sized.
- A customer can modify the KAFKA configuration to adapt to his use case constraints: No event message lost, HA, Latency.....

Basic Tuning : BAI for ODM

- You must change the default configuration of the ra.xml to avoid errors due to timeout:
defaultConnectionManagerProperties
 - To add `pool.waitTimeout=3000` to the `pool.maxSize` used by the customer
 - Example: `pool.maxSize=50,pool.waitTimeout=3000`

- You should change the default Kafka client behavior on HTDS, depending on you Business case: Perf first or reliability first:
 - `kafka.max.block.ms=1`, mandatory to manage KAFKA not available
 - `kafka.acks=0`, in order to have no acknowledgement of message reception, it improves performance but `kafka.acks=1` improves reliability
 - `kafka.max.in.flight.requests.per.connection=1`, to limit the impact of a (xu) pool size and memory usage

Basic Tuning : BAI for ODM

- If your ODM Emitter is generating high frequency events, the BAI configuration must be scaled appropriately, in particular for the processing of ODM events.
- Otherwise, ODM events may not be processed fast enough and your BAI Decision Dashboard may be not relevant to the current state. So you should configure BAI to scale for your case:
 - Kafka should be HA and properly sized
 - Increase the parallelism of the ODM Flink job
 - Increase the number of Kafka topics partitions
 - Scale Elasticsearch and Kibana.
- See "Scaling an IBM BAI system"
 - https://www.ibm.com/support/knowledgecenter/en/SSYHZ8_19.0.x/com.ibm.dba.bai/topics/tsk_bai_scaling.html
- Tuning of KAFKA is key :
 - https://www.cloudera.com/documentation/enterprise/latest/topics/kafka_performance_large_messages.html

ODM on Open Shift

Tuning Recommendations

Tuning : ODM on Open Shift

- The default configuration of a route in Open Shift limits the horizontal scalability of ODM
 - if you have two pods only one is used
 - If we use directly the port of the service the throughput is load balanced on the two pods and the horizontal scalability is ok.

- How to tune a route with OKD 3.11?
 - The tuning should be set at router level in general in the default namespace
 - By default it is mono-thread and it is using source as load balancing policy.
 - Edit the router properties
 - Change the load balancing policy to round robin by adding a property `ROUTER_TCP_BALANCE_SCHEME` with roundrobin as value
 - Change the number of threads by editing the property `ROUTER_THREADS` and set it to 4 or more

Tuning : ODM on Open Shift

- How to tune a route with OCP 4.x?
 - The tuning should be set at route level by adding an annotation
 - `haproxy.router.openshift.io/balance roundrobin`
 - This annotation is useful for the ODM runtime route when the application which calls ODM is on a limited number of IP addresses.

Tuning : ODM on Open Shift- Zen tuning in CP4BA 21.0.3

- CP4BA 21.0.3 introduces the ZEN reverse proxy
- This reverse proxy manages external routes
- With default configuration the Throughput limit is 3200 TPS if you have several sources of requests
- To Increase this limit, you should modify ibm-nginx deployment
 - Use Vertical scalability by increasing the cpu from 400m to 4 or more. Useful if a single source of requests needs more than 1600 TPS
 - Use horizontal scalability by increasing the number of replicas. Useful if there are several sources of requests (load balancing policy at **source**)

Tuning : ODM on Open Shift

- The load balancing policies documentation

- <https://docs.okd.io/latest/architecture/networking/routes.html#load-balancing>
- <https://docs.openshift.com/container-platform/4.6/networking/routes/route-configuration.html>

- Useful documentation

- https://docs.okd.io/3.11/scaling_performance/routing_optimization.html
- https://docs.openshift.com/container-platform/4.6/scalability_and_performance/recommended-host-practices.html#infrastructure-moving-router