August 2017 (Cognos Analytics 11.0.7)

IBM Cognos Analytics 11 (CA 11) Performance Tuning and Monitoring

Performance Team

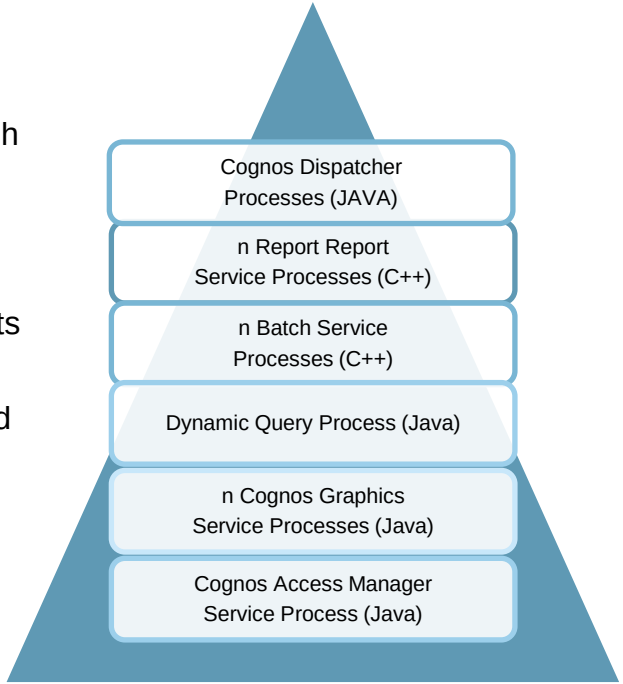
IBM Cognos Analytics and System Performance

- All customers desire the capability to squeeze the maximum performance out of their IBM Cognos Analytics (CA) investment.
- IBM Cognos Analytics is but a part of a complex software and hardware environment. A single bottleneck in either the software or hardware has a ripple effect for the entire system.
- We have compiled a list of the most common software and hardware bottlenecks and how to monitor the system for them.
  - Process Threading:
    - Too few threads and queuing can occur
  - Process Memory Sizing:
    - Too little memory and OutOfMemory conditions can exist
    - Too little memory and Garbage Collection can become costly
- Nothing in the deck can minimize the impact of a sub-optimal Analytics model or poorly authored report spec.

IBM Cognos Analytics Basic Architecture

The following Services account for the core Analytics processes seen in system process views and are generally tunable.

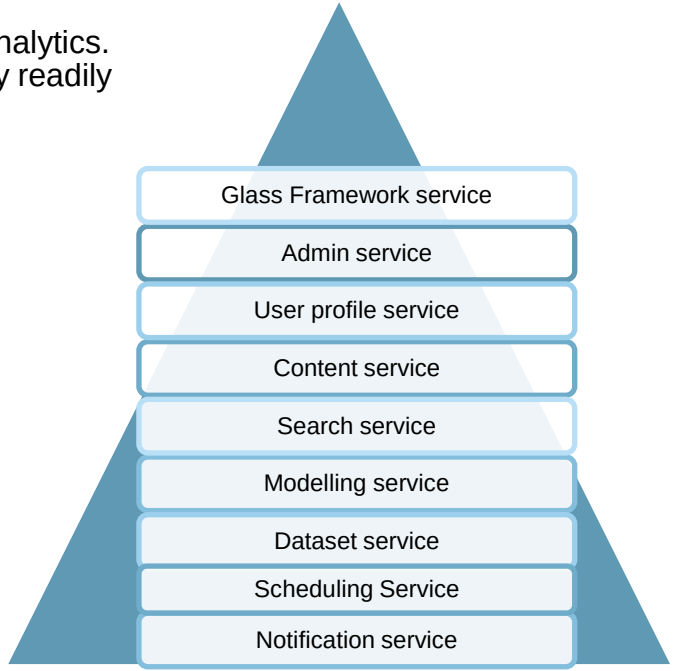
- **Cognos Dispatcher** – Java application responsible for routing requests through the Analytics system and managing Analytics content
- **Report Service** – C++ application that manages interactive requests to execute reports (Process name BIBusTKServerMain.exe)
- **Batch Report Service** – C++ application that manages non-interactive requests to execute reports (Process name: BIBusTKServerMain.exe)
- **Dynamic Query** – Java application that manages Dynamic Query requests and returns the result to the requesting Service
- **Cognos Graphics Service** – Java application that produces graphics on behalf of the other services (Process name: cgsLauncher.exe)
- **Cognos Access Manager Service** – Java application that handles user authentication, authorization, and encryption (Process name: CAM_LPSvr.exe)



IBM Cognos Analytics Micro-Services

The following java/javascript based micro-services were new in Cognos Analytics. They are embedded in the other java processes and are not necessarily readily exposed for tuning.

- **Glass Framework**
- **Admin Service**
- **User Profile Service**
- **Content Search Service**
- **Modelling Service**
- **Dataset Service**
- **Scheduling Service**
- **Notification Service**



- All CA micro services run in the “dispatcher” jvm
  - Installed with all “app-server” install

Cognos Analytics Processes (Windows)

Cognos Analytics 11.0.7 system under load (Windows Platform):

Process	PID	CPU	Private Bytes	Working Set	Command Line	Description
cogbootstrapservice.exe	21184		15,004 K	26,224 K	"D:\IBM\cognos\c10_64_HeliosR7_rs\bin64\cogbootstrapservice.exe"	IBM Cognos Products: btsv
java.exe	21420	2.74	18,682,988 K	5,502,488 K	..\jre/bin/java.exe -XX:+UseCompressedOops -XX:+UseCompressedOops -Djava.net.preferIPv4Stack=true -Xmx16384m -Xms8192m -Xms16384m -Xms16384m -Xgc...	Java(TM) Platform SE binary
conhost.exe	40408	< 0.01	1,424 K	6,036 K	\??\C:\WINDOWS\system32\conhost.exe 0x4	Console Window Host
cgsLauncher.exe	14316		17,533,380 K	801,404 K	D:\IBM\cognos\c10_64_HeliosR7_rs\bin64\cgsLauncher.exe -option COG_ROOT=D:/IBM/cognos/c10_64_HeliosR7_rs idleTimeLimitSec=900 -vmargs Xmx16g X...	IBM Cognos Products: cgs
conhost.exe	23060		1,168 K	5,372 K	\??\C:\WINDOWS\system32\conhost.exe 0x4	Console Window Host
java.exe	24860	2.86	17,970,788 K	7,528,440 K	D:\IBM\cognos\c10_64_HeliosR7_rs\jre\bin\java.exe -Xms16384m -Xmx16384m -Xmnx6144m -Xmns6144m -server -Xscmx100m -Xshareclasses:cachedir=../javash...	Java(TM) Platform SE binary
conhost.exe	26944		1,176 K	5,424 K	\??\C:\WINDOWS\system32\conhost.exe 0x4	Console Window Host
BIBusTKServerMain.exe	22120	0.01	295,976 K	319,900 K	D:\IBM\cognos\c10_64_HeliosR7_rs\bin\BIBusTKServerMain.exe threads=20 camssl=false COG_ROOT=D:/IBM/cognos/c10_64_HeliosR7_rs -../bibustkserver/B...	IBM Cognos Products: cclmisc
conhost.exe	24092		1,176 K	5,372 K	\??\C:\WINDOWS\system32\conhost.exe 0x4	Console Window Host
BmtMDProviderMain.exe	40696	< 0.01	16,152 K	28,836 K	D:\IBM\cognos\c10_64_HeliosR7_rs\bin\BmtMDProviderMain.exe threads=10 camssl=false COG_ROOT=D:/IBM/cognos/c10_64_HeliosR7_rs idleTimeLimitSec=...	BmtMDProviderMain
conhost.exe	29500		1,164 K	5,360 K	\??\C:\WINDOWS\system32\conhost.exe 0x4	Console Window Host
BIBusTKServerMain.exe	12900	1.36	286,536 K	308,052 K	D:\IBM\cognos\c10_64_HeliosR7_rs\bin\BIBusTKServerMain.exe threads=20 camssl=false COG_ROOT=D:/IBM/cognos/c10_64_HeliosR7_rs -../bibustkserver/B...	IBM Cognos Products: cclmisc
conhost.exe	18292		1,168 K	5,376 K	\??\C:\WINDOWS\system32\conhost.exe 0x4	Console Window Host
BIBusTKServerMain.exe	2640	0.72	271,616 K	294,112 K	D:\IBM\cognos\c10_64_HeliosR7_rs\bin\BIBusTKServerMain.exe threads=20 camssl=false COG_ROOT=D:/IBM/cognos/c10_64_HeliosR7_rs -../bibustkserver/B...	IBM Cognos Products: cclmisc
conhost.exe	30076		1,168 K	5,372 K	\??\C:\WINDOWS\system32\conhost.exe 0x4	Console Window Host
BIBusTKServerMain.exe	20608	< 0.01	272,420 K	292,672 K	D:\IBM\cognos\c10_64_HeliosR7_rs\bin\BIBusTKServerMain.exe threads=20 camssl=false COG_ROOT=D:/IBM/cognos/c10_64_HeliosR7_rs -../bibustkserver/B...	IBM Cognos Products: cclmisc
conhost.exe	37332		1,160 K	5,372 K	\??\C:\WINDOWS\system32\conhost.exe 0x4	Console Window Host
BIBusTKServerMain.exe	28492	1.35	258,332 K	283,752 K	D:\IBM\cognos\c10_64_HeliosR7_rs\bin\BIBusTKServerMain.exe threads=20 camssl=false COG_ROOT=D:/IBM/cognos/c10_64_HeliosR7_rs -../bibustkserver/B...	IBM Cognos Products: cclmisc
conhost.exe	32828		1,172 K	5,376 K	\??\C:\WINDOWS\system32\conhost.exe 0x4	Console Window Host
BIBusTKServerMain.exe	31068	0.65	248,980 K	272,252 K	D:\IBM\cognos\c10_64_HeliosR7_rs\bin\BIBusTKServerMain.exe threads=20 camssl=false COG_ROOT=D:/IBM/cognos/c10_64_HeliosR7_rs -../bibustkserver/B...	IBM Cognos Products: cclmisc
conhost.exe	23652		1,328 K	5,428 K	\??\C:\WINDOWS\system32\conhost.exe 0x4	Console Window Host
BIBusTKServerMain.exe	24584	1.00	267,652 K	288,432 K	D:\IBM\cognos\c10_64_HeliosR7_rs\bin\BIBusTKServerMain.exe threads=20 camssl=false COG_ROOT=D:/IBM/cognos/c10_64_HeliosR7_rs -../bibustkserver/B...	IBM Cognos Products: cclmisc
conhost.exe	17600		1,332 K	5,420 K	\??\C:\WINDOWS\system32\conhost.exe 0x4	Console Window Host
BIBusTKServerMain.exe	2724	0.48	270,448 K	292,516 K	D:\IBM\cognos\c10_64_HeliosR7_rs\bin\BIBusTKServerMain.exe threads=20 camssl=false COG_ROOT=D:/IBM/cognos/c10_64_HeliosR7_rs -../bibustkserver/B...	IBM Cognos Products: cclmisc
conhost.exe	37856		1,332 K	5,424 K	\??\C:\WINDOWS\system32\conhost.exe 0x4	Console Window Host
BIBusTKServerMain.exe	36056	0.75	233,224 K	259,136 K	D:\IBM\cognos\c10_64_HeliosR7_rs\bin\BIBusTKServerMain.exe threads=20 camssl=false COG_ROOT=D:/IBM/cognos/c10_64_HeliosR7_rs -../bibustkserver/B...	IBM Cognos Products: cclmisc
conhost.exe	32824		1,332 K	5,420 K	\??\C:\WINDOWS\system32\conhost.exe 0x4	Console Window Host

Cognos Analytics Processes (Unix / Linux)

Cognos Analytics 11.0.7 system under load (Power/Linux):

Nmon:

```
nmonq14iqqqqqq[H for help]qqqHostname=drl-pipitqqqqRefresh= 2secs qqq21:17.25qqqqqqq
x Top Processes Procs=165 mode=4 (1=Basic, 3=Perf 4=Size 5=I/O)qqqqqqqqqqqqqqqqqqqqqqq
x  PID    %CPU  Size   Res    Res    Res   Res Shared   Faults Command
x          Used    KB    Set  Text  Data   Lib    KB  Min  Maj
x    2638 118.5 12865984 5560704    64 12677376     0 40832 6172    0 java
x    4194  97.2 12578240 5145152    64 12288704     0 83520   86    0 java
x    4079  25.8 11179392 4373504    64 10987456     0 43584  260    0 java
x    5089  70.9 910336 627968    64 761152     0 94528   31    0 BIBusTKServerMa
x    5039  91.7 941824 612672    64 739904     0 95872  623    0 BIBusTKServerMa
x    5014  89.7 859520 587776    64 710336     0 94912  132    0 BIBusTKServerMa
x    4564 102.6 857472 582592    64 708288     0 95232    6    0 BIBusTKServerMa
x     483   0.0 93248 72192   384  1088     0 69952    0    0 systemd-journal
x    3753   0.0 1140160 55296    64 1071872     0 33536    0    0 CAM_LPSvr
x    9125   0.0 297152 52160   640 156672     0 47744    0    0 rsyslogd
x    2637   0.0 204352 34368   256 145152     0 23296    0    0 cogbootstrapser
x    9114   0.0 441792 33024    64 305600     0 14080    0    0 tuned
```

```
esperf    2637     1  0 21:11 ?        00:00:00 /data_1/perf/IBM/cognos/auto/./bin64/./cogbootstrapservice -install_root=/data_1/perf/IBM/cognos/auto/./bin64/.. -java_home=/data_1/perf/IBM/cognos/auto/jre -serviceClass=WebSphereLiberty runAsDaemon
esperf    2638  2637 99 21:11 ?        00:11:56 /data_1/perf/IBM/cognos/auto/jre/bin/java -Xgcpolicy:gencon -Xmx8192m -Xms8192m -Xcompressedrefs -Xdump:heap+system:none -Xdump:system:events=gpf+abort,range=1..2,request=serial+compact+prepwalk -Xdump:system:events=systhrow+throw,filter=java/lang/OutOfMemory*,range=1..2,request=serial+compact+prepwalk -Xmn4096m -Xdisableexplicitgc -Djdk.map.althashing.threshold=512 -verbose:gc -javaagent:/data_1/perf/IBM/cognos/auto/./bin64/../wlp/bin/tools/ws-javaagent.jar -jar /data_1/perf/IBM/cognos/auto/./bin64/../wlp/bin/tools/ws-server.jar cognosserver
esperf    3753  2638  0 21:13 ?        00:00:01 /data_1/perf/IBM/cognos/auto/bin64/CAM_LPSvr camssl=false cam=true idleTimeLimitSec=300 COG_ROOT=/data_1/perf/IBM/cognos/auto threads=67 ccithreadstacksize=1048576
esperf    4048  2638  0 21:13 ?        00:00:00 /bin/sh /data_1/perf/IBM/cognos/auto/bin64/cgsServer.sh useNonAsync COG_ROOT=/data_1/perf/IBM/cognos/auto idleTimeLimitSec=900
esperf    4079  4048 48 21:13 ?        00:02:14 /data_1/perf/IBM/cognos/auto/jre/bin/java -Djava.awt.headless=true -Xmx8g -Xms8g -Xmn4g -Xcompressedrefs -Xgcpolicy:gencon -DuseNonAsync -classpath ../webapps/p2pd/WEB-INF/lib/p2pd.jar:../webapps/p2pd/WEB-INF/lib/commons-httpclient-3.1.jar:../webapps/p2pd/WEB-INF/lib/logkit-1.2.jar:../webapps/p2pd/WEB-INF/lib/dom4j-1.6.1.jar:../webapps/p2pd/WEB-INF/lib/jcam_crypto.jar:../webapps/p2pd/WEB-INF/lib/commons-pool-1.3.jar:../webapps/p2pd/WEB-INF/lib/commons-logging-1.1.jar:../webapps/p2pd/WEB-INF/lib/commons-codec-1.3.jar:../webapps/p2pd/WEB-INF/lib/commons-logging-api-1.1.jar:../webapps/p2pd/WEB-INF/lib/commons-logging-adapters-1.1.jar:../webapps/p2pd/WEB-INF/lib/cognosipf.jar:../webapps/p2pd/WEB-INF/lib/log4j-1.2.17.jar:../webapps/p2pd/WEB-INF/lib/cclcfgapi.jar:../webapps/p2pd/WEB-INF/lib/CognosIPF.jar:../webapps/p2pd/WEB-INF/lib/bcprov-jdk14-145.jar:../bin/jcam_jni.jar:../webapps/p2pd/WEB-INF/lib/../classes:../webapps/p2pd/WEB-INF/lib/cgsService.jar:../webapps/p2pd/WEB-INF/lib/cclcoreutil.jar:../webapps/p2pd/WEB-INF/lib/i18nj.jar:cgsJava.jar:openviz3.jar:particles-cognos.jar:../webapps/p2pd/WEB-INF/lib/jaxen-1.1.1.jar:../webapps/p2pd/WEB-INF/lib/JSON4J.jar:../webapps/p2pd/WEB-INF/lib/icu4j.jar:../webapps/p2pd/WEB-INF/lib/UC.jar:../webapps/p2pd/WEB-INF/lib/visOutputServiceCore.jar:../webapps/p2pd/WEB-INF/lib/visOutputService.jar:../webapps/p2pd/WEB-INF/lib/chartAdapters.jar:../webapps/p2pd/WEB-INF/lib/pdfbox-1.7.1.jar:../webapps/p2pd/WEB-INF/lib/commons-lang-2.6.jar com.ibm.cgsBus.CGSServer COG_ROOT=/data_1/perf/IBM/cognos/auto idleTimeLimitSec=900
esperf    4194  2638 99 21:13 ?        00:04:44 /data_1/perf/IBM/cognos/auto/jre/bin/java -Xms8192m -Xmx8192m -Xmnx4096m -Xmns4096m -server -Xjit:disableLoopVersioner -Xscmx100m -Xshareclasses:cachedir=../javasharedresources,name=cognos10%u,nonfatal -Xmso512K -Xcompressedrefs -verbose:gc -Xverbosegclog:/data_1/perf/IBM/cognos/auto/logs/XQE/dq_verbosegc_%Y%m%d.%H%M%S.%pid.log,1,1000 -Xgcpolicy:gencon -Xdump:tool:events=systhrow,filter=java/lang/OutOfMemoryError,exec=rm -f /data_1/perf/IBM/cognos/auto/data/.xqe.%pid > /dev/null -DCOG_ROOT=/data_1/perf/IBM/cognos/auto -DParentProcessID=2638 -Dcubingservices_home=/data_1/perf/IBM/cognos/auto/v5dataserver -Dcubingservices_data=/data_1/perf/IBM/cognos/auto/data -DParentProcessPort=47599 -Djava.library.path=/data_1/perf/IBM/cognos/auto/bin64: -DenableTraceServer -ea -Dorg.dom4j.QName.singleton.strategy=org.dom4j.util.PerThreadSingleton -Dorg.dom4j.DocumentFactory.singleton.strategy=org.dom4j.util.PerThreadSingleton -javaagent:bin/tools/ws-javaagent.jar -jar bin/tools/ws-server.jar dataset-service
esperf    4564  2638 56 21:13 ?        00:02:22 /data_1/perf/IBM/cognos/auto/bin64/BIBusTKServerMain threads=20 camssl=false COG_ROOT=/data_1/perf/IBM/cognos/auto -../bibustkserver/BIBusTKServerMain4j.xml idleTimeLimitSec=900 lightThreadStackSize=327680 lighthreads=10
esperf    5014  2638 75 21:14 ?        00:02:26 /data_1/perf/IBM/cognos/auto/bin64/BIBusTKServerMain threads=20 camssl=false COG_ROOT=/data_1/perf/IBM/cognos/auto -../bibustkserver/BIBusTKServerMain4j.xml idleTimeLimitSec=900 lightThreadStackSize=327680 lighthreads=10
esperf    5039  2638 67 21:14 ?        00:02:11 /data_1/perf/IBM/cognos/auto/bin64/BIBusTKServerMain threads=20 camssl=false COG_ROOT=/data_1/perf/IBM/cognos/auto -../bibustkserver/BIBusTKServerMain4j.xml idleTimeLimitSec=900 lightThreadStackSize=327680 lighthreads=10
esperf    5089  2638 66 21:14 ?        00:02:08 /data_1/perf/IBM/cognos/auto/bin64/BIBusTKServerMain threads=20 camssl=false COG_ROOT=/data_1/perf/IBM/cognos/auto -../bibustkserver/BIBusTKServerMain4j.xml idleTimeLimitSec=900 lightThreadStackSize=327680 lighthreads=10
```

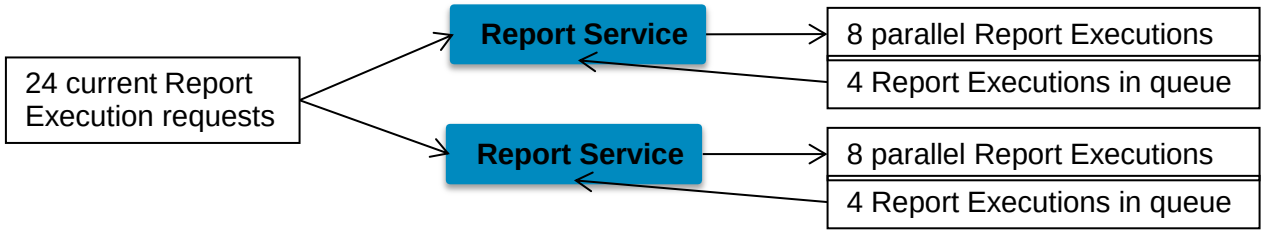
Report Service Tuning

Key things to considering when tuning Report Service:

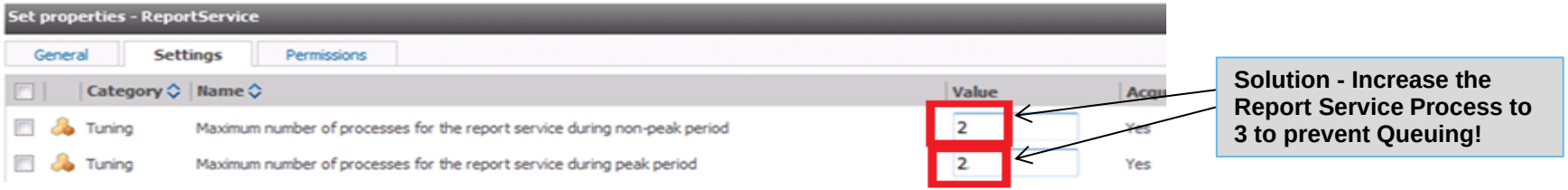
- If Report Service processes are not enough for the work load on the system than queuing can occur.
- Queuing of requests negatively impacts the Analytics system,
  - Would affect both Interactive Report Service and Batch Report Service.
  - Affects Dynamic and Compatible Reporting Engines.
- Monitoring via Cognos Administration

Report Service Tuning

- The threading model in Cognos Analytics 11 Report Server defaults to 8 low affinity threads and 2 high affinity threads
- With 2 Report Server processes there is 16 low affinity thread (default value).
- If 24 requests are issued to report service, 16 get served and 8 get queued.

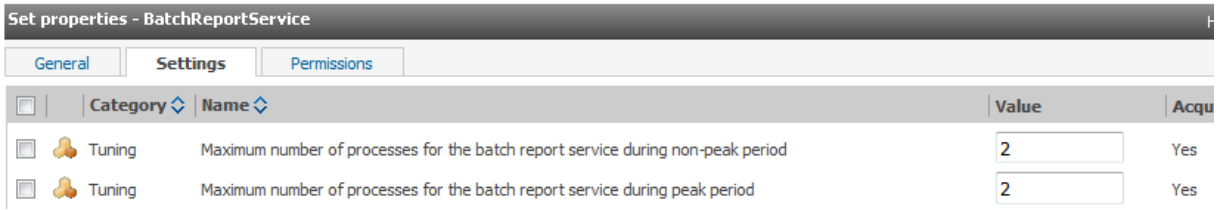


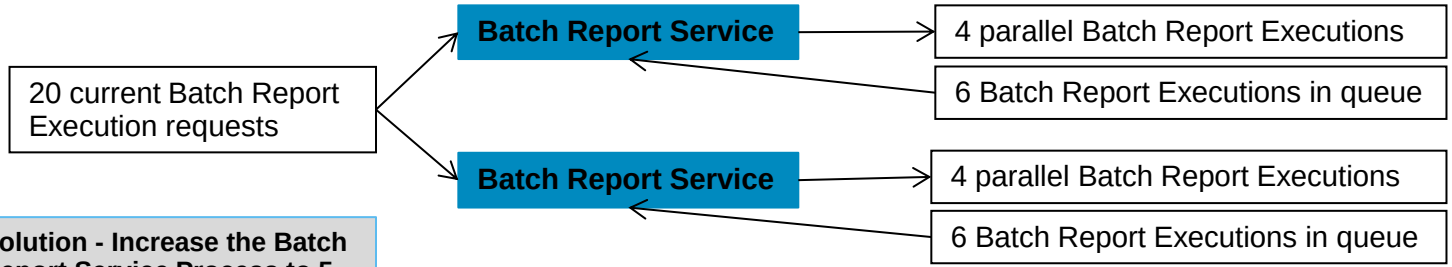
- Increase the number of Report Server processes through Cognos Administration:



Batch Report Service Tuning

- Similar to the Report Service. The Batch Report Service handles job report execution. The number of processes has a significant affect on batch report execution
  - Too few Batch Report Service processes will lead to report execution requests waiting in the Queue.
- Set in IBM Cognos Administration:

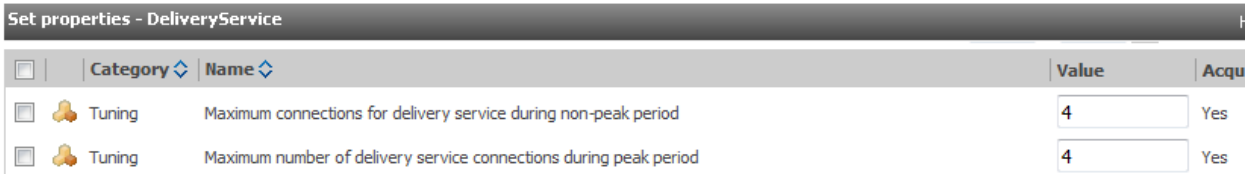




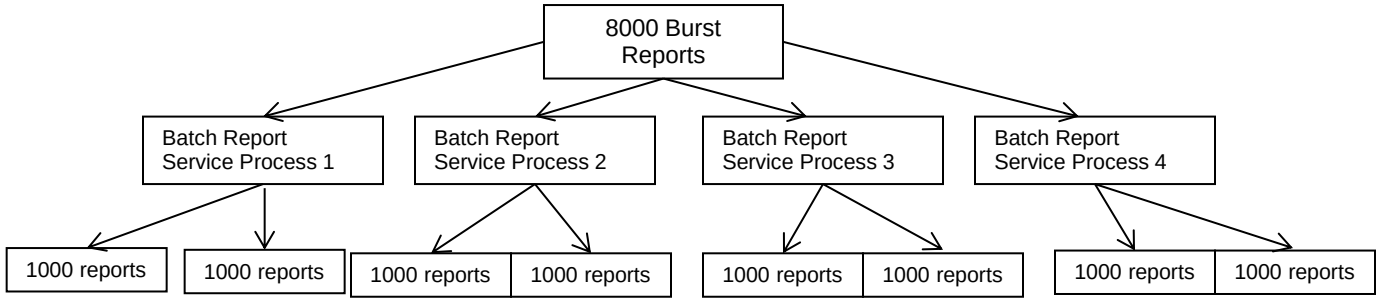
Solution - Increase the Batch Report Service Process to 5 to prevent Queuing!

Batch Report Service Tuning

- The number of Delivery Service connections may need to be increased for heavy Batch environments that write to disk:

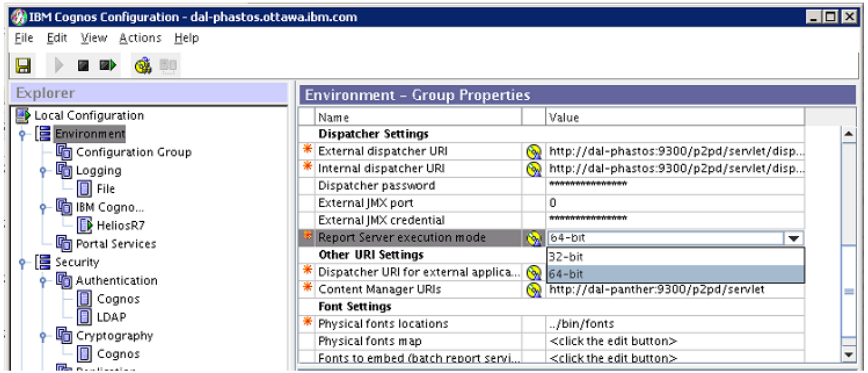


- By default in CA11, Burst Report execution are as follows.
  - The work gets “chunked” across multiple Batch processes.
  - Assume a Burst Reports with 8000 report and 4 Batch Report processes. The work would be chunked across 8 Batch Report Service threads.



Report Server: 32-bit vs. 64-bit

- Two Query Engines in Cognos Analytics: Compatible and Dynamic.
- Report Server Configurations Supported:
  - Compatible: 32-bit Report Server
  - Dynamic : 32-bit Report Server

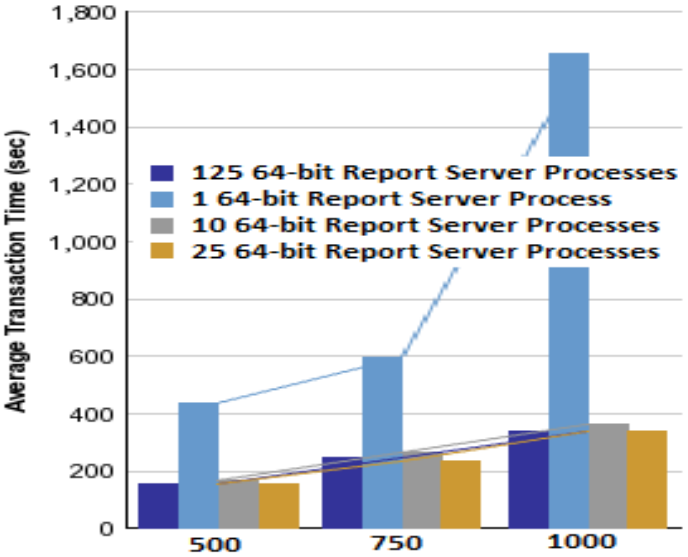


- User end performance comparable between the two Report Server Configurations.
- 64-bit Report Server could reduce the memory footprint on a system by using less report server processes.
- 32-bit Report Server process is Large Address aware. This means that now the 32 bit applications can grow to a larger size (up to 4 GB OS dependent).

Report Server: 64-bit

The memory footprint of Report Server can be reduced by varying the affinity and session cache settings for the number of 64 bit Report Server processes. For example, these configurations handle the same number of work:

- 125 processes, 8 low and 2 high threads, session cache of 20
  - Memory: ~350MB per process = over 40GB
- 25 processes, 40 low and 10 high threads, session cache of 100
  - Memory: ~700MB per process = under 20 GB
- 10 processes, 100 low and 25 high threads, session cache of 250
  - Memory: ~1.3GB per process = under 15GB



- Calculate the session cache by taking the low and high affinity threads and multiple by two
- Affinity settings configured in Cognos Administration
- Session configured in the rsvpproperties.xml on disk in the configuration folder

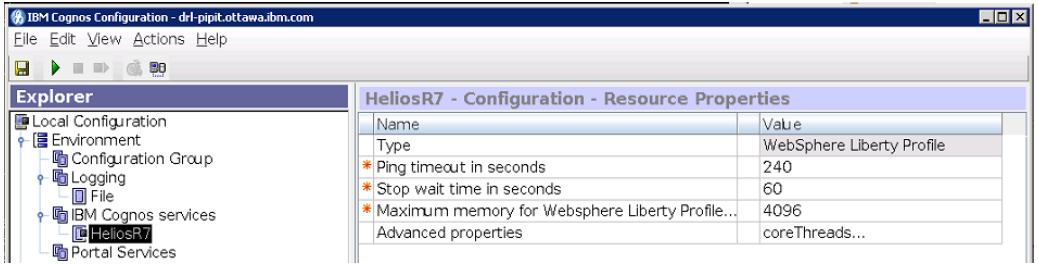
WebSphere Liberty Profiles (WLP) for Analytics

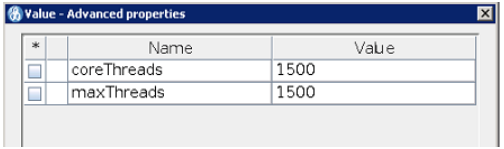
- The WebSphere Liberty Profiles bundled in Cognos Analytics 11.0.7 is Liberty version 16.0.0.4
  - Located in the CA installation at <root>/wlp
- There are two WebSphere Liberty Profiles
  - CognosServer (Dispatcher JVM): <root>/wlp/usr/servers/cognosserver
  - Dataset-Service (CA 11.0.6+) (Query Service JVM): <root>/wlp/usr/servers/dataset-service

- Non-optimal settings for the JVM's can lead to:
  - OutOfMemory conditions
  - Frequent or long Garbage Collection pauses
  - 'timeout' or 'unresponsive' application
  - Overall poor performance

WLP for CA: CognosServer

- The CognosServer JVM is configured in **Cognos Configuration**
- The default WLP settings for CA 11 CognosServer (Dispatcher) java process:
  - JVM Settings
    - Initial Heap Size (Xms) = 1GB or 1024MB
    - Maximum Heap Size (Xmx) = 4GB or 4096MB
    - Xgcpolicy:gencon is set by default
      - http://javaeesupportpatterns.blogspot.com/2012/03/ibm-jvm-tuning-gencon-gc-policy.html
    - Xcompressedrefs is set by default (IBM Java in the <root>/bin64/bootstrap_wlp_<os>.xml)
  - Thread Pools: coreThreads (100) / maxThreads (1500)





WLP for CA: Dataset-Service

- The dataset-service JVM is configured in **Cognos Adminstration**
- The default WLP settings for CA 11 dataset-service (Query Service) java process:
  - JVM Settings
    - Initial Heap Size (Xms) = 1GB or 1024MB
    - Maximum Heap Size (Xmx) = 4GB or 4096MB
    - Xgcpolicy:gencon is set by default
      - http://javaeesupportpatterns.blogspot.com/2012/03/ibm-jvm-tuning-gencon-gc-policy.html
    - Xcompressedrefs is set by default

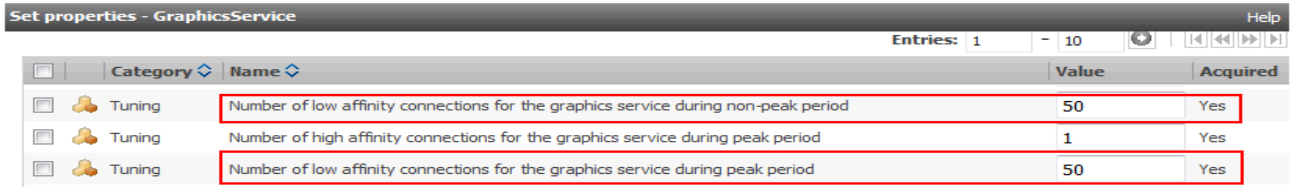
Category	Name	Value	Acquired
Tuning	Minimum query execution time before a result set is considered for caching (milliseconds)	50	Yes
Tuning	Initial JVM heap size for the query service (MB) (Requires QueryService restart)	1024	Yes
Tuning	JVM heap size limit for the query service (MB) (Requires QueryService restart)	8192	Yes
Tuning	Initial JVM nursery size (MB) (Requires QueryService restart)	0	Yes
Tuning	JVM nursery size limit (MB) (Requires QueryService restart)	0	Yes
Tuning	JVM garbage collection policy (Requires QueryService restart)	Generational	Yes
Tuning	Additional JVM arguments for the query service (Requires QueryService restart)		Yes
Tuning	Number of garbage collection cycles output to the verbose log (Requires QueryService restart)	1000	Yes
Tuning	Disable JVM verbose garbage collection logging (Requires QueryService restart)		Yes

Dataset-Service / Query Service

- Threading is handled dynamically by the Query Engine. The number of Interactive Report Services and Batch Report Services affect the thread count.
- DQ automatically provides logs (dq_verbosegc_<timestamp>.log) to help determine if your values are set correctly and allow for easy debugging.
  - Located in the <root>/logs/XQE folder
- **Dynamic Cubes**
  - Cognos Dynamic Cubes is an extension of IBM Cognos Dynamic Query that leverages substantial in-memory data assets as well as aggregate awareness in order to achieve high performance interactive analysis & reporting over terabytes of warehouse data.
  - The JVM must be tuned correctly in order to obtain optimal performance depending on:
    - The size of the data it consumes
    - The expected workload
  - Sizing Recommendations for Dynamic Cubes is available online: http://www.ibm.com/developerworks/library/ba-pp-performance-cognos10-page635/

Cognos Graphics Service (CGS) Tuning

- Threading for the Cognos Graphics Service is configured in Cognos Administration: 50 threads per process.



- Default JVM size of 1GB with no JVM arguments applied.

Unix / Linux

- JVM values and arguments configured in cgsServer.sh in the <root>/bin and <root>/bin64 locations. JVM arguments are set after $JAVA_OPTS: $JAVA_OPTS –Xmx2g –Xms2g –Xmn1g –Xcompressedrefs –Xgcpolicy:gencon

Windows

- JVM values and arguments configured in cgsService.xml in <root>/webapps/p2pd/WEB-INF/services. JVM arguments are set between child-proc-cmd tags after vmargs (2 places in the same file):

<child-proc-cmd>-vmargs</child-proc-cmd>
<child-proc-cmd>Xmx2g</child-proc-cmd>
<child-proc-cmd>Xms2g</child-proc-cmd>
<child-proc-cmd>Xmn1g</child-proc-cmd>
<child-proc-cmd>Xcompressedrefs</child-proc-cmd>
<child-proc-cmd>Xgcpolicy:gencon</child-proc-cmd>

Cognos Analytics Logging

- Cognos Analytics logging changed in CA 11.0.7.
  - The purpose of this logging change is to help in diagnosing complex problems by collecting related logs from easily accessible web-based menus
  - Before enabling diagnostic logging, the Cognos Administrator can review and update size limit and maximum number of files to keep
  - Enable through the CA portal at Manage > Configuration > System, select the “Diagnostic Logging” tab
- https://www.ibm.com/communities/analytics/cognos-analytics-blog/how-to-use-logging-in-cognos-analytics-11-0-7/

Here is quick overview on some of the main log files.

Log file	Changes	Description
cogaudit.log	*New*	Renamed from previous ‘cogserver.log’. Contains the audit information
cognosserver.log	*New*	Log messages from dispatcher (cognosserver liberty server)
dataset-service.log	*New*	Log messages from dataset service (dataset-service liberty server)
p2pd_messages.log	unchanged	Log standard error and output from cognosserver
dq_messages.log	unchanged	Log standard error and output from dataset-service
pogo_<date>.log	replaced	Log data are now written into one of new files
*session_<id>.log	*New*	Generated when enabling session logging. Log either session_<id>_cognosserver or session_<id>_dataset-service

Performance in Cognos Analytics 11

New Features

Data Sets

- Dataset testing was conducted using FM modeled data developed against EAPPS data base (DB2) (~200,000 rows x 48 Columns)
- The below configuration was used for both interactive and not interactive testing using both external object store and a traditional db2 content store.
- ***Interactive Reporting:*** The reports being executed in the background were a mix of reports, including a dashboard based on the dataset. Data set refresh was performed in parallel while interactive use load was executing.
- ***Non-Interactive Reporting:*** The reports being executed in the background were not in anyway associated to the dataset. Data set refresh was performed in parallel while non-interactive use load was executing.

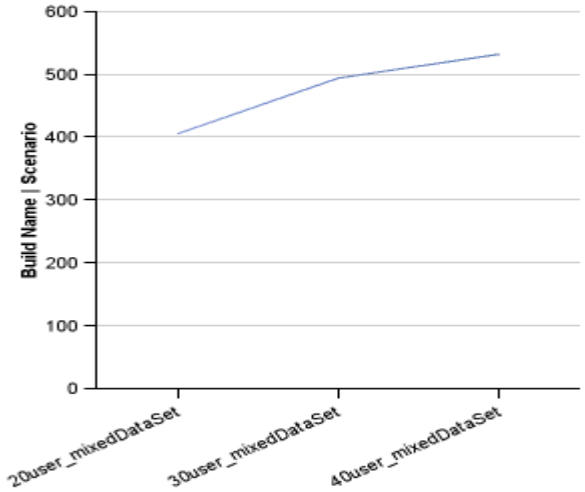
BI Configuration: 1 Content Manager, 1 Report Server writing to a DB2 Content Store DB or External Object store

Content Manager	Dispatcher JVM	8GB
	Core Threads	1500/1500
Report Server	Dispatcher JVM	8GB
	Core Threads	1500/1500
	Dynamic Query JVM	8GB
	Cognos Graphics Service	8GB
	Report Service	10/10 processes
	Batch Report Service	32/32 processes

Data Sets – Interactive Reports

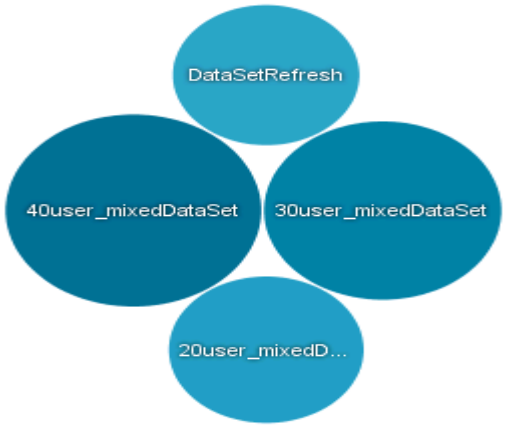
- Testing mixed interactive reports while performing a dataset refresh in the background, there appears to be no performance impact to interactive users or reports.
- Upon investigation, there does however appear to be a performance impact on the dataset refresh times. As users were added, both reporting and the dataset refresh scaled predictably.
- The results were similar when using both a db2 content store or external content store

Interactive Load



Dataset Refresh Times while Under Load

	datasetRefreshTime
DataSetRefresh	387.86
20user_mixedDataSet	405.15
30user_mixedDataSet	493.49
40user_mixedDataSet	531.47



Data Sets: Non-Interactive Reports

- Testing homogeneous non-interactive reports while performing a dataset refresh in the background, there appears to be no performance impact to interactive users or non-interactive jobs.
- Upon investigation, as with interactive reports, there does however appear to be a performance impact on the dataset refresh times. Dataset refresh was performed in parallel while non-interactive use load was executing.
- The results were similar when using both a db2 content store or external content store

Non-Interactive Load

	heliosR4_16090915_CM_base	heliosR4_16090915_DataSetRefresh_bkgnd	
	average	average1	Diff
1user_customer_burst_40k_pdf_cm_dqm	1,544	1,368	11.4%
1user_masterJob_40_large_html_cm_classic	1,530	1,581	-3.3%
1user_masterJob_40_large_html_cm_dqm	1,619	1,620	-0.1%
1user_masterJob_40_large_pdf_cm_classic	1,576	1,571	0.3%
1user_masterJob_40_large_pdf_cm_dqm	1,584	1,603	-1.2%

Interactive Dataset Refresh Times while Under Load

Scenario	Datase Refresh(s)
DataSetRefresh	372.38
1user_customer_burst_40k_pdf_cm_dqm	408.21
1user_masterJob_40_large_html_cm_dqm	541.03
1user_masterJob_40_large_pdf_cm_classic	556.73
1user_masterJob_40_large_pdf_cm_dqm	557.19

Search

4GB JVM vs 8GB JVM

- Search service is handled by the Content Manager Dispatcher JVM process. By default the dispatcher JVM process is set to 4GB, so by increasing the JVM Memory of a factor of 2X from the default value of 4GB we are seeing an increase in transaction performance over default as user load increases. Overall performance increase from 15%-21% or 0.2 seconds (low user load) – 15 seconds (high user load)

	16042614_HeliosR2_Search_100KObj_JVM4GB_WP_Scale								16042614_HeliosR2_Search_100KObj_JVM8GB_WP_Scale							
	Σ Averages	Std Dev	Margin of Error	Base Lower Range	Base Upper Range	Pass	Fail	% Pass	Σ Average	% Diff (sec)	Std Dev	Margin of Error	Build Lower Range	Build Upper Range	Pass	Fail	% Pass
10user_testScenario	0.76	0.52	+/- 0.02	0.75	0.78	19,870	0	100.00%	0.75	2.24% (0.0s)	0.63	+/- 0.01	0.74	0.76	20,360	0	100.00%
20user_testScenario	1.31	0.96	+/- 0.03	1.29	1.34	23,440	0	100.00%	1.14	15.31% (0.2s)	0.88	+/- 0.01	1.13	1.15	26,925	0	100.00%
40user_testScenario	3.43	1.52	+/- 0.05	3.38	3.48	18,115	0	100.00%	2.83	21.18% (0.6s)	1.30	+/- 0.02	2.81	2.85	21,920	0	100.00%
80user_testScenario	8.94	2.61	+/- 0.10	8.85	9.04	14,195	0	100.00%	7.45	20.10% (1.5s)	3.51	+/- 0.05	7.39	7.50	17,010	0	100.00%
160user_testScenario	20.42	5.25	+/- 0.20	20.22	20.62	12,835	0	100.00%	18.65	9.47% (1.8s)	4.68	+/- 0.08	18.58	18.73	14,030	0	100.00%
320user_testScenario	47.88	15.75	+/- 0.64	47.24	48.52	11,660	0	100.00%	43.98	8.88% (3.9s)	14.49	+/- 0.25	43.73	44.23	12,850	0	100.00%
640user_testScenario	114.01	59.08	+/- 2.42	111.58	116.43	11,425	0	100.00%	98.98	15.18% (15.0s)	45.71	+/- 0.78	98.20	99.76	13,145	0	100.00%
Total	**196.77**	**85.69**	**3.45**	**193.31**	**200.22**	**111,540**	**0**		**173.78**		**71.20**	**0.39**	**172.58**	**174.98**	**126,240**	**0**	

4GB JVM vs 16GB JVM

- By Increasing the JVM Memory of a factor of 4X from the default value of 4GB we are seeing an increase in transaction performance over default as user load increases. Overall performance increase from 16%-44% or 0.3 seconds (low user load) – 16.1 seconds (high user load)

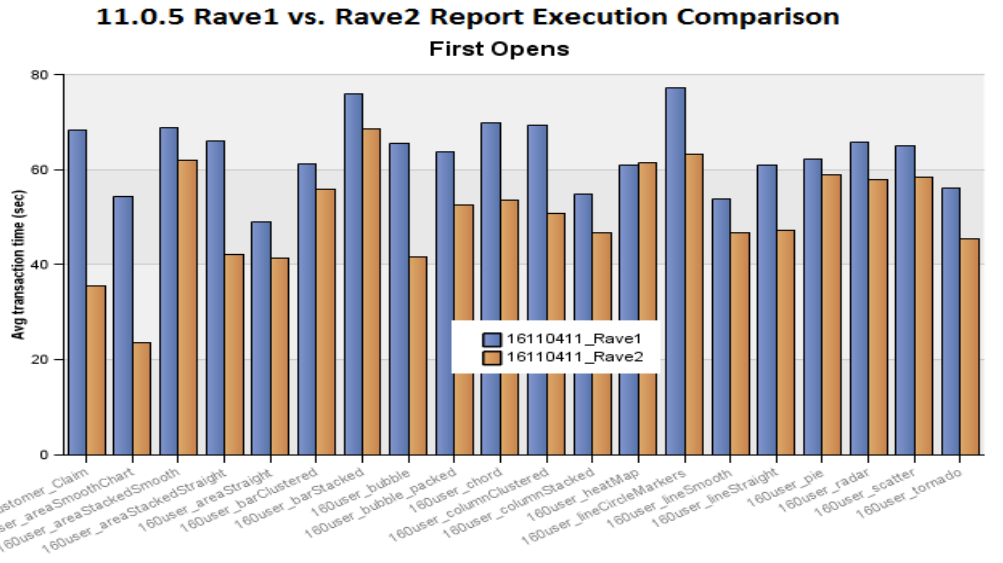
Therefore, if using high concurrent user load while performing searches we would recommend a JVM setting of at least 8GB for optimal performance.

Dashboards, Stories and Guided Journeys

- Dashboards and stories are tested against content that is created off of datasets, modules and uploaded files.
- Dashboards and stories use only the java processes. They do not use the BIBusTKServerMain process.

Charting: RAVE1 vs. RAVE2

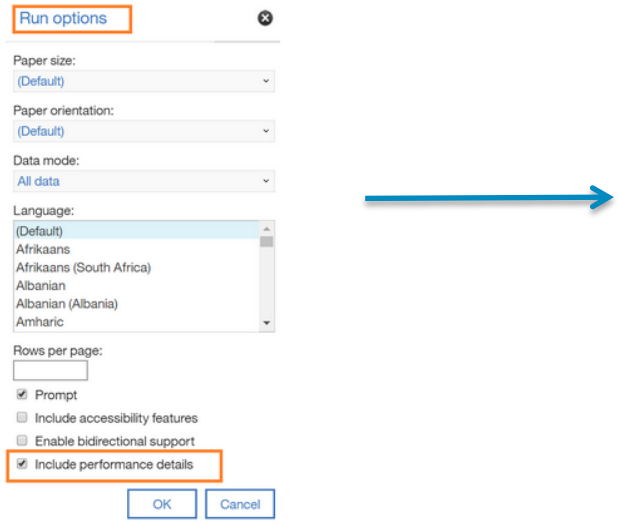
- In general Rave2 performance is better than Rave1.
- Most of the performance gains were seen with the first opens for the test cases.
- Memory usage is identical between Rave1 and Rave2.
- %CPU increased for some Rave2 reports. This is due to the cgsLauncher process.

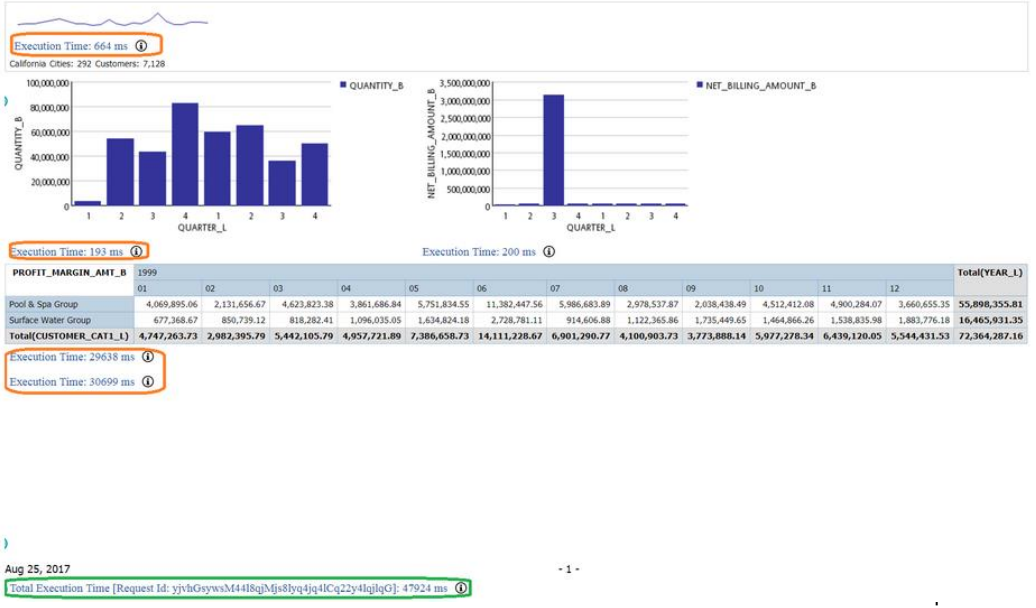


Monitoring Cognos Analytics

Interactive Performance Assistant (IPA)

- New in Cognos Analytics 11.0.7!
- Enable and use Interactive Performance Assistant to view query and rendering performance on report objects in Cognos Analytics.
- https://www.ibm.com/communities/analytics/cognos-analytics-blog/interactively-view-report-performance/





Monitoring Tools: IBM JVM Tools

IBM Monitoring and Diagnostic Tools

IBM provides tooling and documentation to assist in the understanding, monitoring, and problem diagnosis of applications and deployments running IBM Runtime Environments.

- **Health Center**
  The Health Center is a lightweight tool that monitors active IBM Virtual Machines with minimal performance overhead. The Health Center suggests live tuning recommendations for Garbage Collection, profiles methods including call stacks, and highlights contended locks.
  Learn more about the Health Center.

- **Interactive Diagnostic Data Explorer**
  Interactive Diagnostic Data Explorer (IDDE) is the strategic tool for allowing interactive analysis of JVM problems using post mortem artifacts such as core files or javacores. It is lightweight, allowing you to quickly get information from the artifact you are investigating where you aren't sure what the problem is and you want to avoid launching resource intensive analysis.
  Learn more about the IDDE.

- **Memory Analyzer**
  The Memory Analyzer brings the diagnostic capabilities of the Eclipse Memory Analyzer Tool (MAT) to the IBM Virtual Machines for Java. Memory Analyzer extends Eclipse MAT using the Diagnostic Tool Framework for Java (DTFJ) which enables Java heap analysis using operating system level dumps and IBM Portable Heap Dumps (PHD).
  Learn more about the Memory Analyzer.
  Download the Memory Analyzer using the IBM Support Assistant Workbench.

- **IBM Extensions for Memory Analyzer**
  The IBM Extensions for Memory Analyzer offer both additional capabilities for debugging generic Java applications, and capabilities for debugging specific IBM software products by building knowledge of those products into the extensions.
  Learn more about the IBM Extensions for Memory Analyzer.

- **Garbage Collection and Memory Visualizer**
  The Garbage Collection and Memory Visualizer (GCMV) helps visualise the memory footprint, garbage collection behaviour and performance of Java and Node.js applications using verbose garbage collection output. The Garbage Collection and Memory Visualizer provides recommendations for tuning the application.
  Learn more GCMV.

- **Dump Analyzer**
  The Dump Analyzer analyzes a formatted system dump and produces a report including suggestions on how to resolve the problem.
  Learn more about the Dump Analyzer.
  Download Dump Analyzer using the IBM Support Assistant Workbench.

- **IBM Diagnostic Tool Framework for Java**
  The Diagnostic Tool Framework for Java (DTFJ) is a Java application programming interface (API) used to support the building of Java diagnostics tools.
  Learn more about the IBM Diagnostic Tool Framework for Java.

www.ibm.com/developerworks/java/jdk/tools

Monitoring Tools: GC Logs

JVM Memory and GC Policies

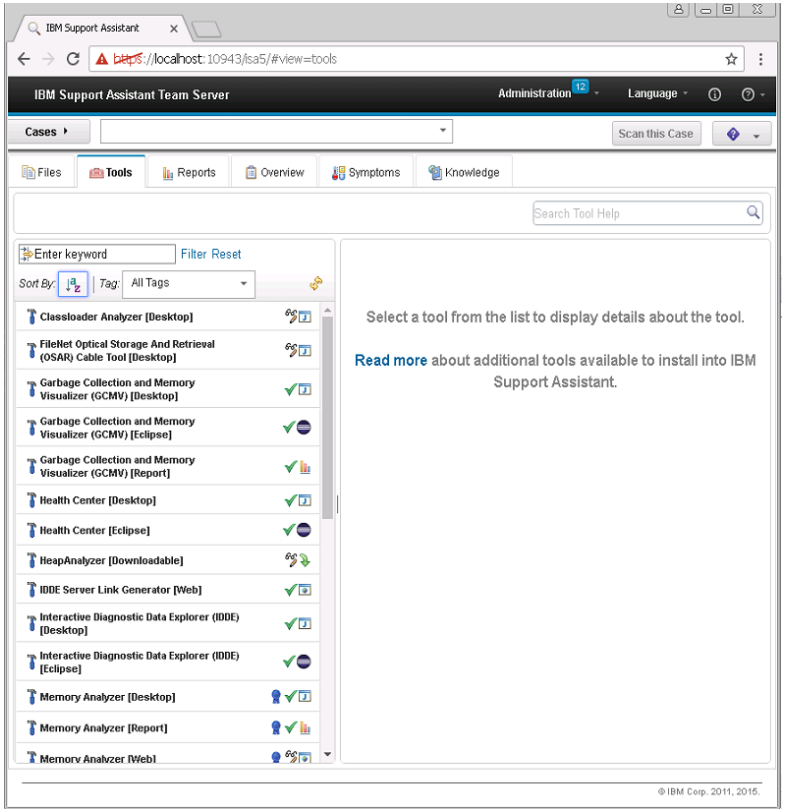
- Enabling GC logging is a low impact method of measuring JVM sizing and Garbage Collection policies. Undersized JVMs can lead to OOM situations or excessive garbage collections and high JVM pause times.
- For Dispatcher and CM, edit <root>/bin64\bootstrap_wlp_<OS>.xml and add the following line to the Java arg list
- For CGS, edit \webapps\p2pd\WEB-INF\services\cgsService.xml or cgsServer.sh on UNIX and add to the JVM arguments (in two locations in the file!)

```
<param condName="${java_vendor}" condValue="IBM">-Xverbosegclog:../logs/disp_gc.log</param>
```

```
<child-proc-cmd>Xverbosegclog:../logs/cgs_gc.log</child-proc-cmd>
```

- For Dynamic Query and Dynamic Cubes, GC Logging is on by default in a file named dq_verbosegc_%timeStamp%.log

Monitoring Tools: IBM Support Assistant



www-01.ibm.com/software/support/isa/

The desired toolset is ‘IBM Monitoring and Diagnostic Tools for Java’

- IBM Monitoring and Diagnostic Tools for Java™ - Dump Analyzer
- IBM Monitoring and Diagnostic Tools for Java™ - Garbage Collection and Memory Visualizer for ISAv4
- IBM Monitoring and Diagnostic Tools for Java™ - Health Center
- IBM Monitoring and Diagnostic Tools for Java™ - Interactive Diagnostic Data Explorer
- IBM Monitoring and Diagnostic Tools for Java™ - Memory Analyzer
- IBM Monitoring and Diagnostic Tools for Java™ - Memory Analyzer 64bit

Quick steps to use the tool:

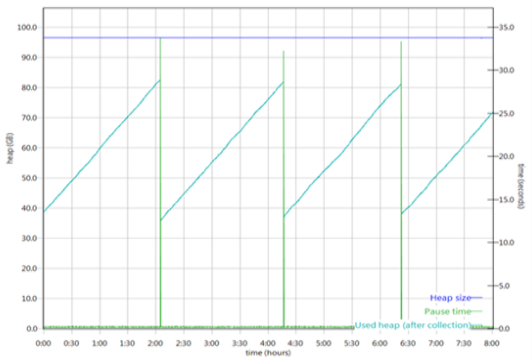
- Highlight the ‘Garbage Collection and Memory Visualizer’ tool and hit ‘Launch’ and browse to the garbage collection log using the ‘Remote Artifact Browser’

- Arguably the most useful view is ‘Heap size’, ‘Pause time’, and ‘Used heap (after collection)’ chosen from the File menu ‘VGC pause’ and ‘VGC heap’

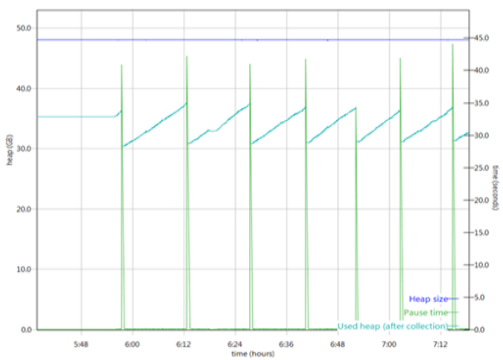
IBM Support Assistant: GC Logging Analysis

- Classic example of the IBM Support Assistant helping determine ‘ideal’ JVM size
  - Left graph shows a properly tuned JVM with GC occurring approximately every 2 hours with a pause time of under 35 seconds.
  - Right graph illustrates a JVM running an undersized Java Heap. GC occurs every 10 minutes with a pause time of over 40 seconds.

Tuned JVM (using Xms/Xmx = 100 GB)



Small JVM (using Xms/Xmx = 50 GB)



Healthy Analytics System

Summary

Concurrent collection count	1
Forced collection count	0
GC Mode	gencon
Global collections - Mean garbage collection pause (ms)	543
Global collections - Mean interval between collections (ms)	1534407
Global collections - Number of collections	1
Global collections - Total amount tenured (MB)	421
Largest memory request (bytes)	6813808
Number of collections triggered by allocation failure	16
Nursery collections - Mean garbage collection pause (ms)	469
Nursery collections - Mean interval between collections (ms)	163418
Nursery collections - Number of collections	16
Nursery collections - Total amount flipped (MB)	1615
Nursery collections - Total amount tenured (MB)	433
Proportion of time spent in garbage collection pauses (%)	0.62
Proportion of time spent unpaused (%)	99.38
Rate of garbage collection (MB/minutes)	127

Operating System Monitoring

- Important to use tools that allow for unattended monitoring of resource utilization over time. For example:
  - PerfMon for Windows
  - Nmon on AIX/Linux (use Nmon Analyser to process Nmon output)
- Read the Cognos Analytics documentation for any OS specific settings that may need to be applied.
- Bottlenecks in the system resources can lead to frustration in the Analytics community due to:
  - Inconsistent performance
  - Unexpected error messages
- As hardware and software evolve, system bottlenecks tend to shift. The four most common system bottlenecks are:
  - CPU
  - Memory for both the entire system and key Analytics processes
  - Network utilization
  - Disk for read, writes, and waits

Operating System Monitoring

- If CPU is a bottleneck:
  - Shift Analytics services to other servers in the system that may have available CPU
  - Consider adding additional CPU resources or an additional server
  - Monitor Run Queue
- If memory is a bottleneck:
  - Turn off services and processes to free up memory
  - Add more memory to the system or shift Analytics services to servers with more available RAM
  - Do not rely on Virtual Memory. Absolute performance killer!
- If network appears to be a bottleneck:
  - Check that the NIC card is using the full bandwidth available
  - Ensure the server resolves localhost locally first and not to the DNS first (netsvc.conf)
  - Ensure routers in the Analytics system are not overtaxed
- If disk might be a bottleneck:
  - Check that file system logging is turned off or minimized (mount)
  - Consider using fast storage to host disk intensive Analytics services