IBM Watson IoT

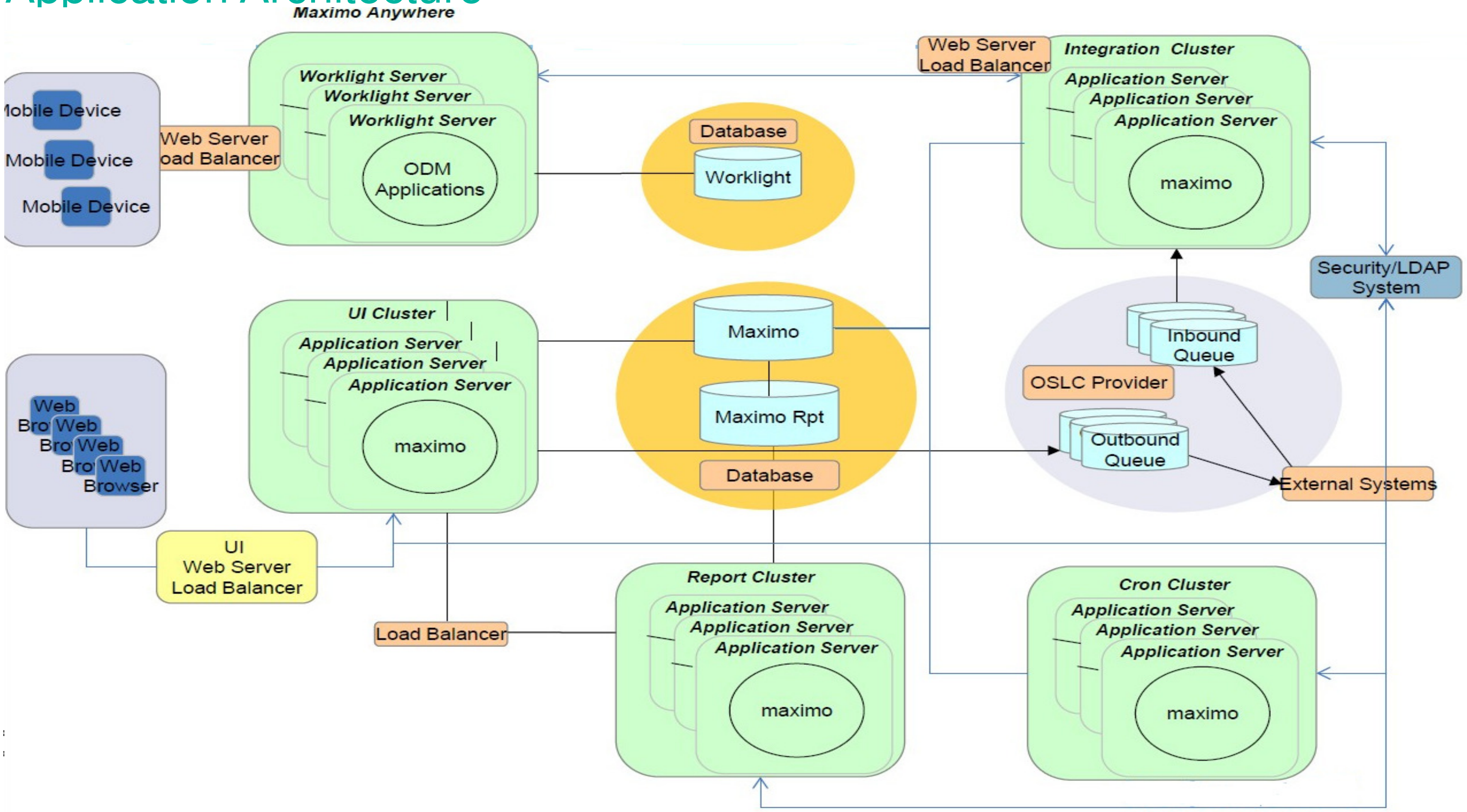
Maximo Performance Tuning and Best Practices

April 2016

Agenda

- Maximo Performance
- Maximo Performance Troubleshooting
- Tools
- Resources
- Q&A

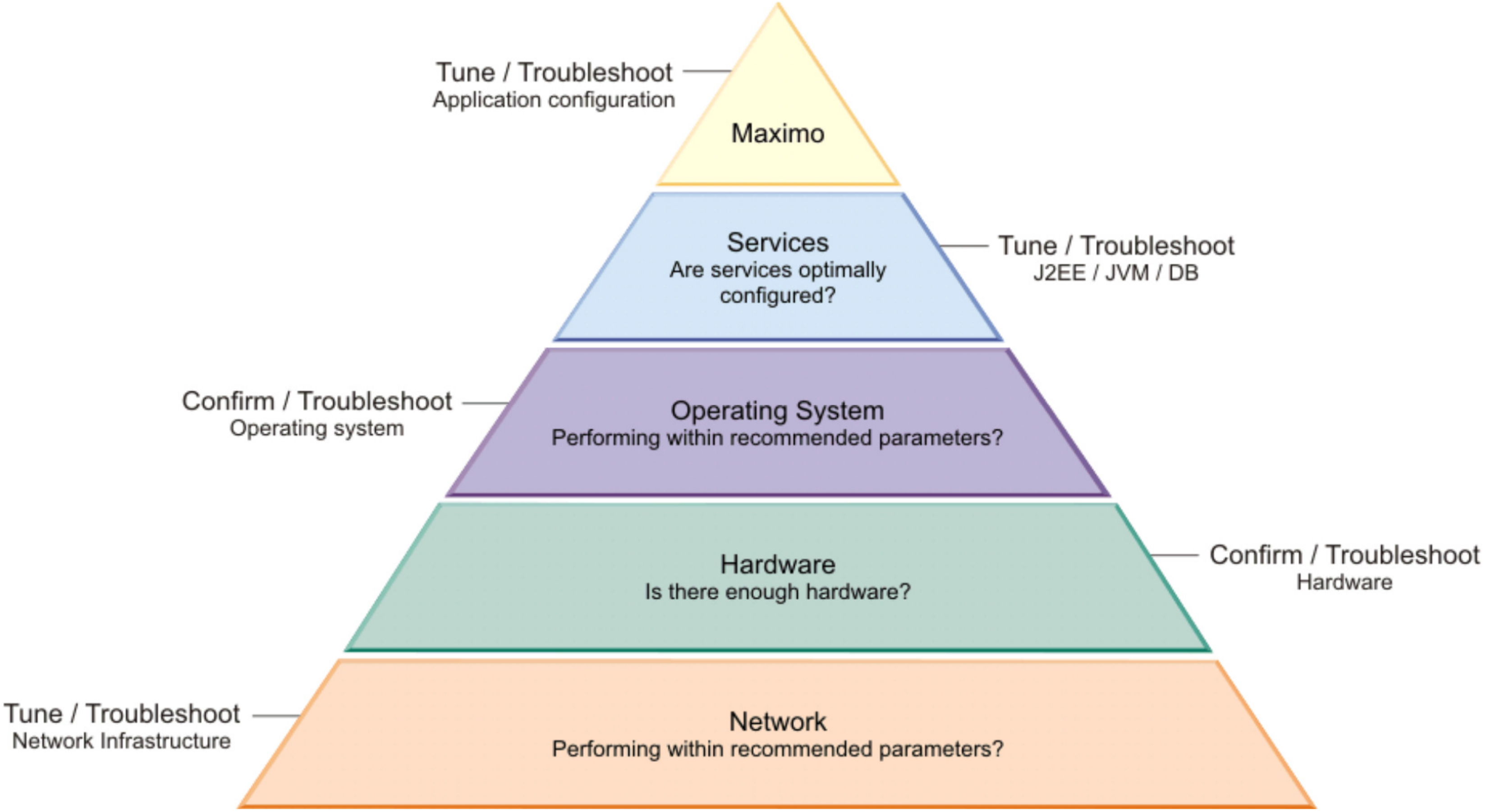
Application Architecture



Factors in System Performance

- System architecture setup
  - Hardware/processor/memory
  - Operating System performance
- Application server configuration
- Scheduled tasks (cron tasks)
- Reporting
- Integration with other systems using the integration framework
- Network
- Bandwidth/Latency
- Load balancing
- Database tuning
- SQL/Index tuning
- Application Configurations
  - Default behavior
  - Environment settings
  - Screen design
  - Validation/sync settings
- Client workstation configuration
- Miscellaneous performance improvement tips
- Troubleshooting

Maximo Infrastructure



Tune / Troubleshoot
Application configuration

Maximo

Services
Are services optimally configured?

Tune / Troubleshoot
J2EE / JVM / DB

Confirm / Troubleshoot
Operating system

Operating System
Performing within recommended parameters?

Hardware
Is there enough hardware?

Confirm / Troubleshoot
Hardware

Tune / Troubleshoot
Network Infrastructure

Network
Performing within recommended parameters?

Architecture - Sizing

- Application Server
  - 64-bit
  - JVMs Clustering
  - Memory – Both physical and JVM memory
  - Current fix packs
- Database Server
  - Disk Usage sizing
  - CPU sizing
  - Database memory
- Reporting Server
- LDAP/Directory Server
- Integration Component
  - Transaction rates
  - Clustering
- Network
  - Bandwidth
  - Latency

Hardware/Operating System

- Using the minimum hardware requirements may produce the minimum acceptable application performance

- Using virtual technology to maximize hardware availability can reduce overall performance (VM overhead, shared resources)

- Use physical memory. Sharing memory and swapping to disk has a performance impact

- Use Operating System tuning to maximize throughput (temp/swap space, background processing, disk/storage tuning)

Application Server

- Each application server vendor will have their own requirements and recommendations related to supporting infrastructure and settings. There is no “one size fits all”

IBM WebSphere

- Generic JVM Argument : -Dsun.rmi.dgc.ackTimeout=10000

Oracle WebLogic

For both platforms - Memory settings: Initial: 4096m / Maximum: 4096m

Note: Some implementations may be using 6GB.

More memory is not always a good answer. Too much memory can slow GC (garbage collection).

Application Server

- Application Servers should always be setup in a cluster
  - JVMs per cluster based on load
- UI Cluster
  - End user logins
  - 1 CPU per JVM
  - 4-6 GB heap
  - 50-75 users per JVM
- Crontask Cluster
  - Background tasks
  - 1 CPU per JVM
  - 4-6 GB heap
  - At least 2 JVMs
- Integration Cluster
  - Integration transactions
  - 1 CPU per JVM
  - 4-6 GB heap
  - At least 2 JVMs

Application Server

- Report Cluster
  - Report Execution
  - 1 CPU per JVM
  - 4-6 GB heap
  - At least 2 JVMs

- JVM tuning
  - Heap
  - Garbage Collection
  - Load Balancing

- Use replication to populate and synchronize a reporting database

Application Server

- Per JVM – allocate an additional 1GB of memory for internal usage
  - 4GB JVM needs 5GB total memory
- Allocate 25% additional memory for the Operating System
- Take into account all processes that might be started on the physical/virtual server
  - Eg. JVM, node manager, deployment manager, web server, anti-virus, etc.

Application Server

- JVM arguments
  - -Dsun.rmi.dgc.ackTimeout
    - This represents how often the server checks for objects that can be garbage collected. The time value is in milliseconds
    - The best practice for this is 10000 (10 seconds). The default value is 10 minutes
    - Setting this value too low can increase CPU usage
  - -Xdisableexplicitgc
    - This property should be set to disable explicit garbage collection, this disables all System.gc() calls from initiating garbage collection
  - -Xmn
    - The nursery size should be set to 25% of the maximum heap. This property should be used along side the –Xgcpolicgy:gencon in which the garbage collection policy places objects in separate areas of the heap based on lifetime

Application Server

- JVM arguments
  - -Xgcpolicy:gencon
    - The gencon policy is now the default policy in WebSphere 8.5.5 and is set out of the box when doing a Maximo install. This is the recommended policy. The gencon policy places objects into the heap in separate areas based on how long they have been active. The heap is split into a nursery based on the –Xmn setting. The nursery can be garbage collected without the overhead of cleaning up the entire heap at the same time.
  - -Xgcpolicy:optthruput
    - This used to be the default policy for WebSphere, in version 8 and forward the default policy has been changed to gencon. Optthruput works based on multiple passes. First it goes through all reachable objects and marks it as active data, it then passes through a second time to clean up any non marked objects, freeing up memory.
    - This can not be used along with –Xmn (nursery) parameter.

Application Server

- Thread pool
  - Thread pools are a collection of threads on a server that can be reused when the JVM creates a request rather then create a new thread with each new request.
- Best Practices
  - Minimum number of threads in the pool should be set to 20 for the default thread pool and 120 for the WebContainer thread pool
  - The thread inactivity timeout defines how long a thread can be inactive before the thread is returned to the pool. The default thread pool should be set to 30000 and WebContainer thread pool to 60000
  - Enable Allow thread allocation beyond the maximum size. This defines if a thread can be created past the maximum thread pool value
  - Thread pool settings are defined for each individual server.

Database

- Each database vendor will have their own requirements and recommendations related to supporting infrastructure and settings. There is no “one size fits all”
- Tune database as per install and best practices documents
  - Specific settings are documented
    - Oracle
      - Use cursor_sharing=force
      - The more memory the better
    - DB2
      - Lock timeout
      - Memory settings
      - Schedule regular maintenance for reorganization and runstats
      - For DB2 9.5 and later set DB2_WORKLOAD=MAXIMO
    - SQL Server
      - Turn off page level locking
      - Use Maximo properties for SQL Server
      - Do not try to use with more than 250 concurrent users

Database

- Database tuning is the most important performance improvement task
  - Maximo is extremely database centric
  - Use database tools to find long running queries and recommended indexes

- Analyze database memory and user I/O
  - Maximo fetches a lot of data into application server
  - Increase system memory. Helps reduce user I/O

- Separate tablespaces and mount points to optimize I/O

Database

- Setup an index statistics update schedule
  - Weekly update helps in better performance
- Purge / Archive data
  - Transactional data needs archiving plus purging
  - Purge on a regular basis
    - E.G. Workflow transaction data, login tracking, start center
- Add indexes as they are needed
  - Do not be scared to add indexes
  - In Maximo, 80% is data fetch, 20% is data insert/update
  - Lack of right indexes causes more performance problem than slowing down on insert/update
- Sequences (Oracle & DB2)
  - Enable sequence caching
  - Increase cache size for sequences to 50 (except maxseq)
  - Increase cache size for maxseq sequence to 500

Database

- High Availability Database Setup
  - DB2 provides HACMP and HADR
  - Oracle has RAC
  - SQL Server High Availability
  - High Availability helps in failover

- Separate reporting database helps reduce load on the primary database

- Multi-site setup use the property to improve SQL timing
  - mxe.db.useSiteInListQuery=1
  - This includes the list of sites the user has access in the SQL statement instead of figuring out through complex SQL against multiple tables

Database

- 1 CPU for every 100 database connections

- Environment of 400 concurrent users, total database connections may be double
  - 400 concurrent users requires at least 8 CPUs
  - Consider peak load
  - Heavy reporting usage will increase database load and requires more database CPU

- 2GB SGA memory for every 100 database connections

- 400 concurrent users = 8GB of SGA

The Gray Area

- Tuning the application is sometimes a cross between application and middleware settings. This occurs with both database server and application server.
- Tuning the database server and then sending it bad queries will still result in poor performance
  - Make sure any application properties for connecting to the database are set
  - Use the approved versions of JDBC drivers
  - Use logging best practices - too much SQL logging impact results
- Tuning the application server and then putting too much load on it will still result in poor performance
  - Separate functionality into dedicated JVMs (UI, Cron, Report, Integration)
  - Limit concurrent users to 75 per dedicated UI JVM

Querying/SQL

- Search Methodologies – Default to “EXACT”
  - If possible, change WILD card searches to EXACT
  - Educate users to use % wild card option in search fields
  - Reduces ‘like’ searches and greatly improved database query times
- Set most ALN fields to UPPER
- Educate users on ad-hoc queries
  - Check for newly created user queries and optimize
- Identify long running reports
  - Review SQL in reports
- Escalations
- Conditions
- Start Center result sets
- Doclinks query
- KPIs
- List panel order by queries
  - Can remove order by in presentation XMLs

Reporting

- Reporting Cluster (BROS)
- Reporting Database
- Identify long running reports from report usage and tune them
- Can identify poor performing ad hoc reports
- Scheduled report cron tasks
  - Execute in Cron Cluster
  - Separate these out of UI servers
  - Ensure long running scheduled reports are not submitted at the same time
- Property mxe.report.birt.maxconcurrentrun defines how many concurrent reports can run on a given Maximo JVM
  - Default value is 5 – good starting point
  - Recommendations
    - 2 * num of CPU
    - Depends on the complexity of the reports

Integration Framework

- Separate sequential inbound and sequential outbound queues for integrated system
- Limit the number of message-driven beans (MDBs) on the continuous queue
- Exception-handling queues
  - ‘bad’ data redirect out of continuous queue
- XML/HTTP instead of Simple Object Access Protocol (SOAP) web services
  - Less overhead and better performer under load
- File import for XML documents containing multiple records

Network

- 50ms or faster round-trip packet response between client and application server
- Low bandwidth or high latency network
    - Citrix or Terminal Server
- Compression techniques
    - HTTP Compression
        - Web server
    - Hardware Compression
        - Network appliances – Juniper and Riverbed
        - Provide compression and caching
    - Gzip Compression
        - Maximo configuration
- Acceleration
- Caching
    - MaxAge – enabled by default on the application server
- Configuring asynchronous and client side validation can improve the users experience. Good screen design can reduce traffic

Miscellaneous

- Task Focused Screen Design – Create small screens
- Off Hours WO Generation, Reorder, Crons, and Reporting
- Reduce start center data retrieval
  - Keep the default start center simple to no data fetch portlets
  - There can be multiple start centers – the first one displayed should be simple to avoid excessive data fetch
  - Be sure that queries used on the Start Centers return small numbers of rows and are well tuned
- Client Browser Configuration – Page refresh to Automatic
- Database connection pooling
  - Lower properties values
- Limit query times
  - System properties
- Limit number of fields on a single screen / develop silo (business unit) based screens with only the fields required for that silo
- Configure client side and asynchronous validation to limit real time interaction with the database server

Miscellaneous

- Tune DOCLINKS relationships or disable DOCLINKS
- Limit the number of objects that can be fetched from the database in a single result set (system properties)
- Carefully consider business requirements for frequency of Cron Tasks and Escalations
- Tune cron tasks and Escalations
  - Adjust start timings to spread the load
  - Load balance across multiple JVMs
    - New in Maximo 7.6 – cron task load balancing

Troubleshooting

- Analyze and clearly define reported problem
  - Understand what is being reported. “Application is slow” is not a definition. Find out what is slow. See it for yourself if possible
- Determine/define what is “successful resolution”
  - Don't allow client perception to interfere with a solution. Get a clear definition of what is acceptable so you have a measure for success
- Focus on technologies that might impact issue
  - If users at one plant are affected but users at another plant are not affected, it is likely not a server problem. Focus on items that can impact the affected area

Troubleshooting - Logging

- Keep Maximo as well as application server logging to minimal level unless investigating a problem
  - Set everything except Maximo root logger to ERROR
  - Maximo logging adds 5% + overhead
- Clean up log folders from filling up
  - Watch out for heap dumps as these files are very large
- Since logging changes can be applied dynamically, turn logging on when needed and turn it off when done – Logging application
- SQL logging to collect all SQL statements
- New in Maximo 7.6 – logging marker
- If analyzing memory usage
  - Use verbose GC logging. Clean up old log files

Troubleshooting - Properties

- System Properties used as part of logging
  - Enabled through System Properties application. Dynamic, can be enabled without re-building application EAR file
- Performance properties
  - mxe.mbocount
    - Capture snapshot intervals of memory usage
  - mxe.db.logSQLTimeLimit
    - Captures SQL statements that take longer than the specified amount of time to return
  - mxe.db.fetchResultLogLimit
    - Captures the number of objects an MBO loads while it is processing
- mxe.db.logSQLPlan (Oracle only)
- mxe.db.sqlTableScanExclude (Oracle only)
- FetchStopLimit
  - Enable and use until you reach a stable state. Returns an error to the end user.
    - mxe.db.fetchStopLimitEnabled
    - mxe.db.fetchResultStopLimit
    - mxe.db.fetchResultStopLimit.OBJECTNAME
    - mxe.db.fetchStopExclusion
- FetchResultLogLimit
  - This greatly helps in identifying excessive data fetch
  - Use this to identify the code that is fetching excessive data

Troubleshooting

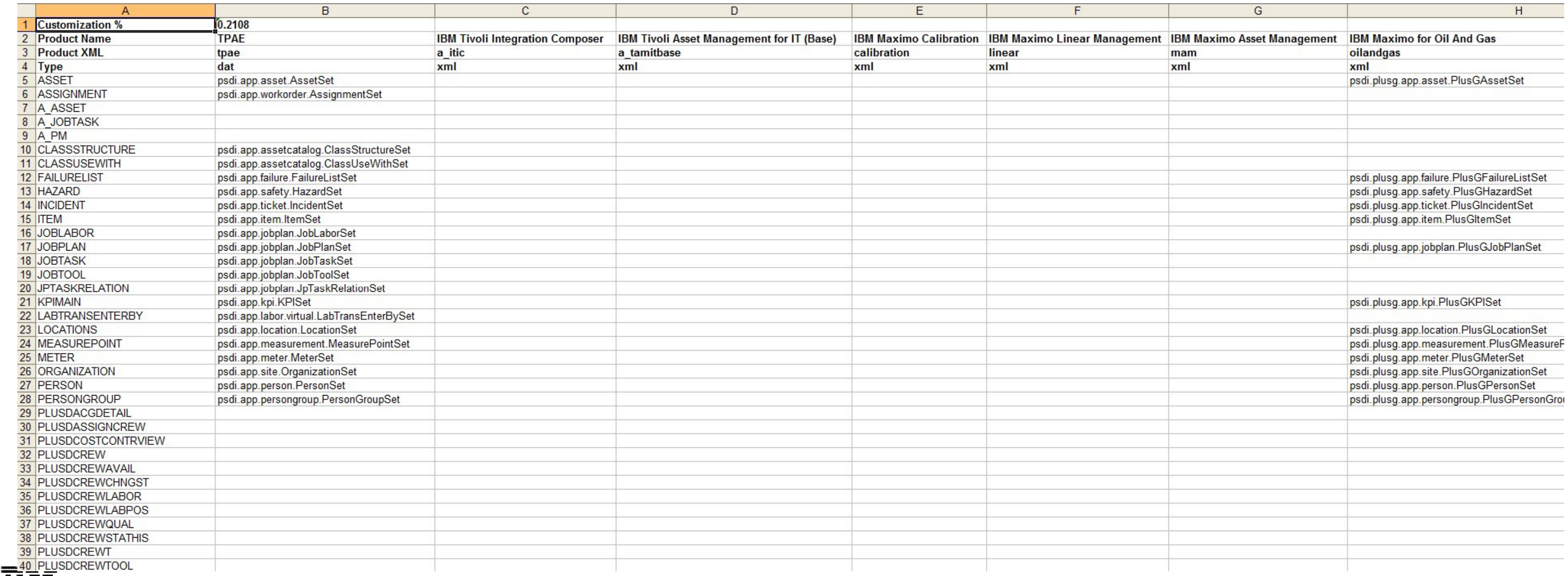
- Enable Connection Pool Watchdog to monitor database connection pool
  - Enabled in the Loggings application
- Close long running database connections
  - Maximo 7.5.0.3 properties introduced
- Use database tools to find long running queries and recommended indexes
- Test queries through SQL development tools and add required indexes
  - Always make sure the indexes are added through Maximo Database Configuration to retain during fix pack installation and upgrades
- Network pings and trace routes from client to Maximo server
- Monitor CPU and memory usage

Tools

- Integrity Checker to find Maximo schema problems that can affect applications as well as performance
  - Integrity Checker can be found in the Maximo tools folder (integrityui)
  - Integrity Check should be done in development phase to production phase
- LA Test Fix Tool
  - Report of limited availability fixes (one-off) installed
  - LA test fix tool can be found in the Maximo tools folder (LATestFixReportWriter)
- IBM Thread and Monitor Dump Analyzer for Java
  - Identification of hangs, deadlocks, resource contention and bottlenecks in Java threads
  - Classpath
  - Java and Garbage Collection parameters
- Heap Analyzer
  - Possible heap leak detection
- IBM Support Assistant (ISA)
  - Add-on tools for heap and thread analysis
- Real world load testing

Tools

- Customization Detection Tool
  - Report the lists all the services, object, and attribute classes that have been extended
  - Displays extensions (class customizations) in a class hierarchy structure
  - Shows percentage customized
  - Reports on services, objects and attributes

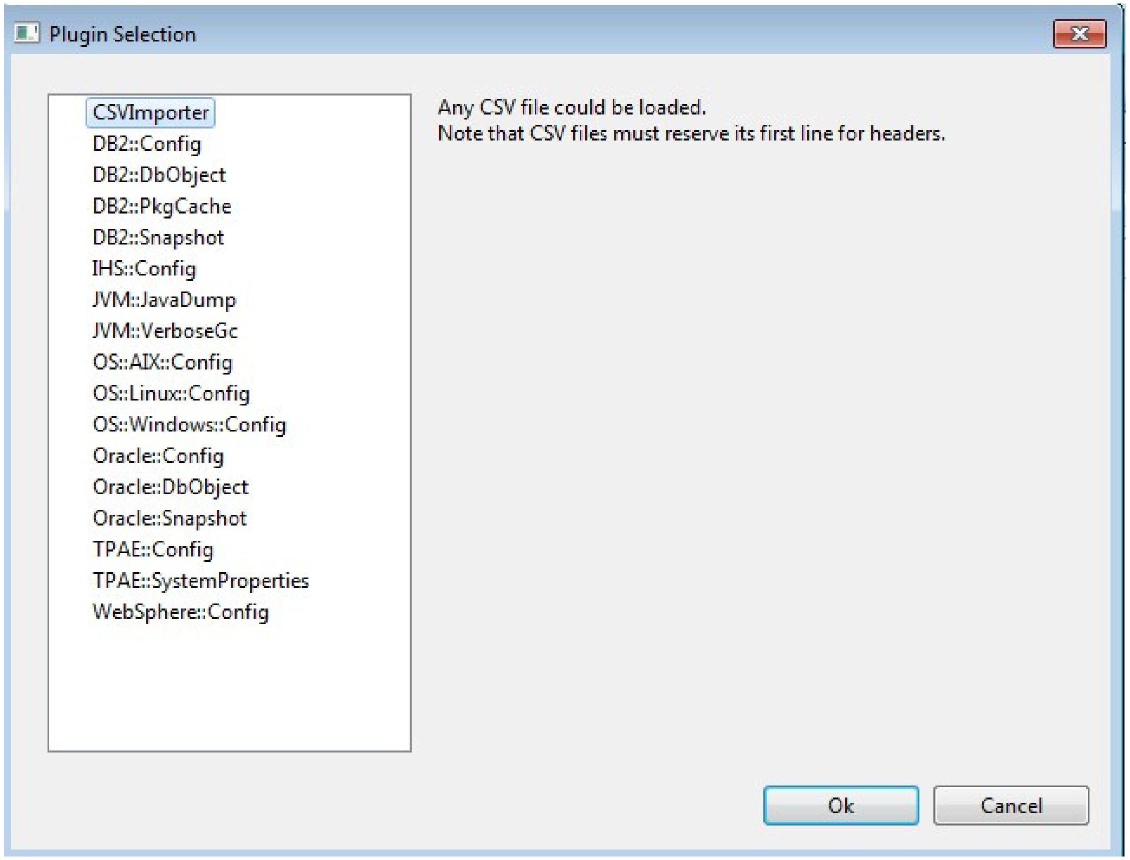


	A	B	C	D	E	F	G	H
1	**Customization %**	0.2108						
2	**Product Name**	**TPAE**	**IBM Tivoli Integration Composer**	**IBM Tivoli Asset Management for IT (Base)**	**IBM Maximo Calibration**	**IBM Maximo Linear Management**	**IBM Maximo Asset Management**	**IBM Maximo for Oil And Gas**
3	**Product XML**	**tpae**	**a_itic**	**a_tamitbase**	**calibration**	**linear**	**mam**	**oilandgas**
4	**Type**	**dat**	**xml**	**xml**	**xml**	**xml**	**xml**	**xml**
5	ASSET	psdi.app.asset.AssetSet						psdi.plusg.app.asset.PlusGAssetSet
6	ASSIGNMENT	psdi.app.workorder.AssignmentSet						
7	A_ASSET							
8	A_JOBTASK							
9	A_PM							
10	CLASSSTRUCTURE	psdi.app.assetcatalog.ClassStructureSet						
11	CLASSUSEWITH	psdi.app.assetcatalog.ClassUseWithSet						
12	FAILURELIST	psdi.app.failure.FailureListSet						psdi.plusg.app.failure.PlusGFailureListSet
13	HAZARD	psdi.app.safety.HazardSet						psdi.plusg.app.safety.PlusGHazardSet
14	INCIDENT	psdi.app.ticket.IncidentSet						psdi.plusg.app.ticket.PlusGIncidentSet
15	ITEM	psdi.app.item.ItemSet						psdi.plusg.app.item.PlusGItemSet
16	JOBLABOR	psdi.app.jobplan.JobLaborSet						
17	JOBPLAN	psdi.app.jobplan.JobPlanSet						psdi.plusg.app.jobplan.PlusGJobPlanSet
18	JOBTASK	psdi.app.jobplan.JobTaskSet						
19	JOBTOOL	psdi.app.jobplan.JobToolSet						
20	JPTASKRELATION	psdi.app.jobplan.JpTaskRelationSet						
21	KPIMAIN	psdi.app.kpi.KPISet						psdi.plusg.app.kpi.PlusGKPISet
22	LABTRANSENTERBY	psdi.app.labor.virtual.LabTransEnterBySet						
23	LOCATIONS	psdi.app.location.LocationSet						psdi.plusg.app.location.PlusGLocationSet
24	MEASUREPOINT	psdi.app.measurement.MeasurePointSet						psdi.plusg.app.measurement.PlusGMeasureF
25	METER	psdi.app.meter.MeterSet						psdi.plusg.app.meter.PlusGMeterSet
26	ORGANIZATION	psdi.app.site.OrganizationSet						psdi.plusg.app.site.PlusGOrganizationSet
27	PERSON	psdi.app.person.PersonSet						psdi.plusg.app.person.PlusGPersonSet
28	PERSONGROUP	psdi.app.persongroup.PersonGroupSet						psdi.plusg.app.persongroup.PlusGPersonGro
29	PLUSDACGDETAIL							
30	PLUSDASSIGNCREW							
31	PLUSDCOSTCONTRVIEW							
32	PLUSDCREW							
33	PLUSDCREWAVAIL							
34	PLUSDCREWCHNGST							
35	PLUSDCREWLABOR							
36	PLUSDCREWLABPOS							
37	PLUSDCREWQUAL							
38	PLUSDCREWSTATHIS							
39	PLUSDCREWT							
40	PLUSDCREWTOOL							

Object Example

Tools

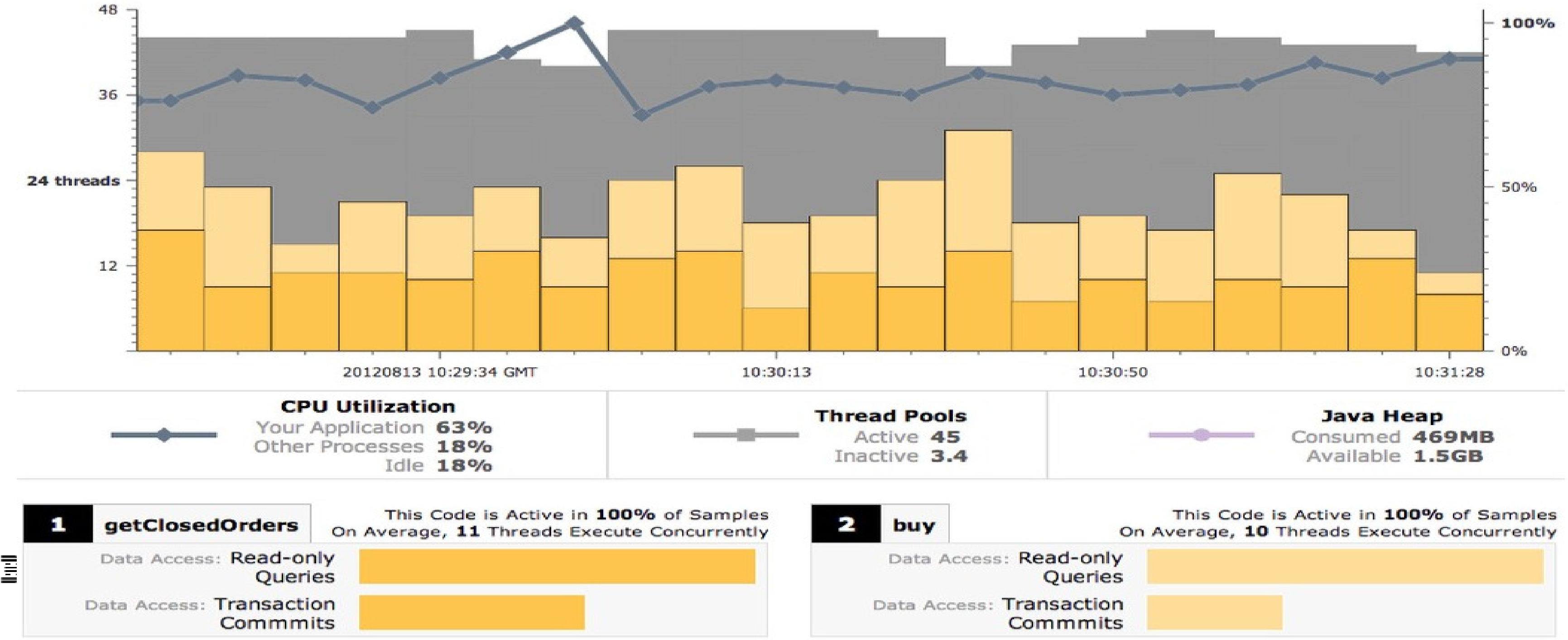
- Performance Analysis Suite
  - Help analyze performance problems
  - Plug-ins
  - TPAE:Config plugin
    - Performance white paper compliance checking
    - Dashboard tab shows a high level result of performance compliance checking result



Any CSV file could be loaded.
Note that CSV files must reserve its first line for headers.

Tools

- Wait Tool
  - Identifies the bottlenecked resources, and shows the code that drives this load.
  - Works on any Java application
  - No system of agents
  - Output is an HTML report



CPU Utilization
Your Application **63%**
Other Processes **18%**
Idle **18%**

Thread Pools
Active **45**
Inactive **3.4**

Java Heap
Consumed **469MB**
Available **1.5GB**

1 **getClosedOrders**
This Code is Active in **100%** of Samples
On Average, **11** Threads Execute Concurrently
Data Access: Read-only Queries
Data Access: Transaction Commmits

2 **buy**
This Code is Active in **100%** of Samples
On Average, **10** Threads Execute Concurrently
Data Access: Read-only Queries
Data Access: Transaction Commmits

Tools

- Maximo Management Interface (MMI)
  - New in Maximo 7.6 – http://hostname:port/maximo/oslc/members
  - Integrated into Maximo framework
  - Real time access to the environment
  - Extendable
  - Provides input to monitoring tools
  - Standardized JSON output
  - Set of REST APIs that enable HTTP-based clients to access information that is related to the current state of Maximo
  - The Maximo server processes that are part of the deployment, potentially in a multi-clustered setup
  - Information about resources, such as the JVM, memory, operating system, application server, JVM threads, database connection pool, class or classloader, MBO count, event topic tree, JMS Queue connectivity, cache information

Resources

Application Server
IBM WebSphere Tuning (can cause application to hold objects too long if not set correctly)
http://www.ibm.com/support/docview.wss?rs=3214&uid=swg21261874

Oracle WebLogic Tuning
http://www.ibm.com/support/docview.wss?rs=3214&uid=swg21261853

Using Multiple JVMs to Support Users and Functionality
http://www-01.ibm.com/support/docview.wss?uid=swg21329219

Network Performance
Network caching and compression properties
http://www.ibm.com/support/docview.wss?rs=3214&uid=swg21262009

Browser Caching (Helps with Wide Area Network -WAN- performance and high latency)
http://www.ibm.com/support/docview.wss?rs=3214&uid=swg21292557

Reporting
BIRT Reporting Performance document link
Maximo 7.5 https://www-304.ibm.com/support/entdocview.wss?uid=swg21305031
Maximo 7.6 http://www-01.ibm.com/support/docview.wss?uid=swg21693526

Resources

Database Server Performance

SQL Server Index Tuning

http://www.ibm.com/support/docview.wss?rs=3214&uid=swg21261979

SQL Server Minimizing Locks

http://www.ibm.com/support/docview.wss?rs=3214&uid=swg21296072

SQL Server Row Versioning

http://www.ibm.com/support/docview.wss?rs=3214&uid=swg21440598

Maximo 7.5 on SQL Server Performance

http://www-01.ibm.com/support/docview.wss?uid=swg21592716

SQL Server Turning of Page Locking

http://www-01.ibm.com/support/docview.wss?uid=swg21268567

Oracle Cursor Sharing

http://www.ibm.com/support/docview.wss?rs=3214&uid=swg21262959

Resources

Database Server Performance

DB2 Parameter Tuning

https://www-304.ibm.com/support/docview.wss?uid=swg21451593

Database Connection Leak Resolution

http://www-01.ibm.com/support/docview.wss?uid=swg21622010

Resources

Application Performance

Understanding Search Methodologies

http://www-01.ibm.com/support/docview.wss?uid=swg21321289

Start Center Performance

http://www-01.ibm.com/support/docview.wss?uid=swg21405898

Improving Performance Using Small User Interface (UI) Screens

http://www-01.ibm.com/support/docview.wss?uid=swg21420509

Disable Data Validation

http://www-01.ibm.com/support/docview.wss?uid=swg21516856

Resources

Troubleshooting

Debug properties
http://www-01.ibm.com/support/docview.wss?uid=swg21291250

Collecting Data/Must Gather – General Information
http://www-01.ibm.com/support/docview.wss?uid=swg21313647

Collecting Data/Must Gather – Performance
http://www-01.ibm.com/support/docview.wss?uid=swg21313341

Logging
http://www-01.ibm.com/support/docview.wss?uid=swg21264064

Logging Appenders
http://www-01.ibm.com/support/docview.wss?uid=swg21446599

Using fetch stop limits to prevent out-of-memory errors
http://www-304.ibm.com/support/docview.wss?uid=swg21412865

Maximo Performance Monitor Configuration
https://www-304.ibm.com/support/docview.wss?uid=swg21448706

Resources

Best Practices

Maximo 7.1 Performance

http://www-01.ibm.com/support/docview.wss?uid=swg21440192

Maximo 7.5 Performance Best Practices

http://www.ibm.com/support/docview.wss?uid=swg21591070

Maximo 7.6 Performance Best Practices

https://www.ibm.com/developerworks/community/groups/service/html/communityview?communityUuid=a9ba1efe-b731-4317-9724-a181d6155e3a#fullpageWidgetId=W5f281fe58c09_49c7_9fa4_e094f86b7e98&file=e0291480-2b4f-4366-bb01-e6e7360cd033

Maximo Wiki

https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20Maximo%20Asset%20Management

Maximo Wiki Performance Page

https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20Maximo%20Asset%20Management/page/Performance%20and%20tuning

Resources

Tools

IBM Support Assistant

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

IBM Thread and Monitor Dump Analyzer for Java

https://www.ibm.com/developerworks/community/groups/service/html/communityview?communityUuid=2245aa39-fa5c-4475-b891-14c205f7333c

Heap Analyzer

https://www.ibm.com/developerworks/community/groups/service/html/communityview?communityUuid=4544bafe-c7a2-455f-9d43-eb866ea60091

Customization Detection Tool

https://www-304.ibm.com/software/brandcatalog/ismlibrary/details?catalog.label=1TW10MA4S

IBM Performance Analysis Suite

https://www.ibm.com/developerworks/community/groups/community/perfanalyst

Wait Tool

https://wait.ibm.com/

Maximo Management Interface (MMI)

https://www.ibm.com/support/knowledgecenter/SSLKT6_7.6.0/com.ibm.mam.doc/troubleshooting/c_mmi.html

https://www.ibm.com/developerworks/community/blogs/d6565698-694d-442b-a26b-c89892fa0c02/entry/Maximo_Management_Interface_MMI_API_overview_and_how_to_use_it?lang=en

https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Wb8e8390134d1_4377_be1e_cf6a57f4ea75/page/Maximo%20Management%20Interface

IBM **Watson IoT**

Thank You

IBM