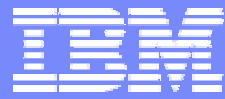


IBM Software Group

2006 B2B Customer Conference

B2B – Catch the Next Wave

Tuning WebSphere Partner Gateway

Tawab Hamidi & Tuan Dang

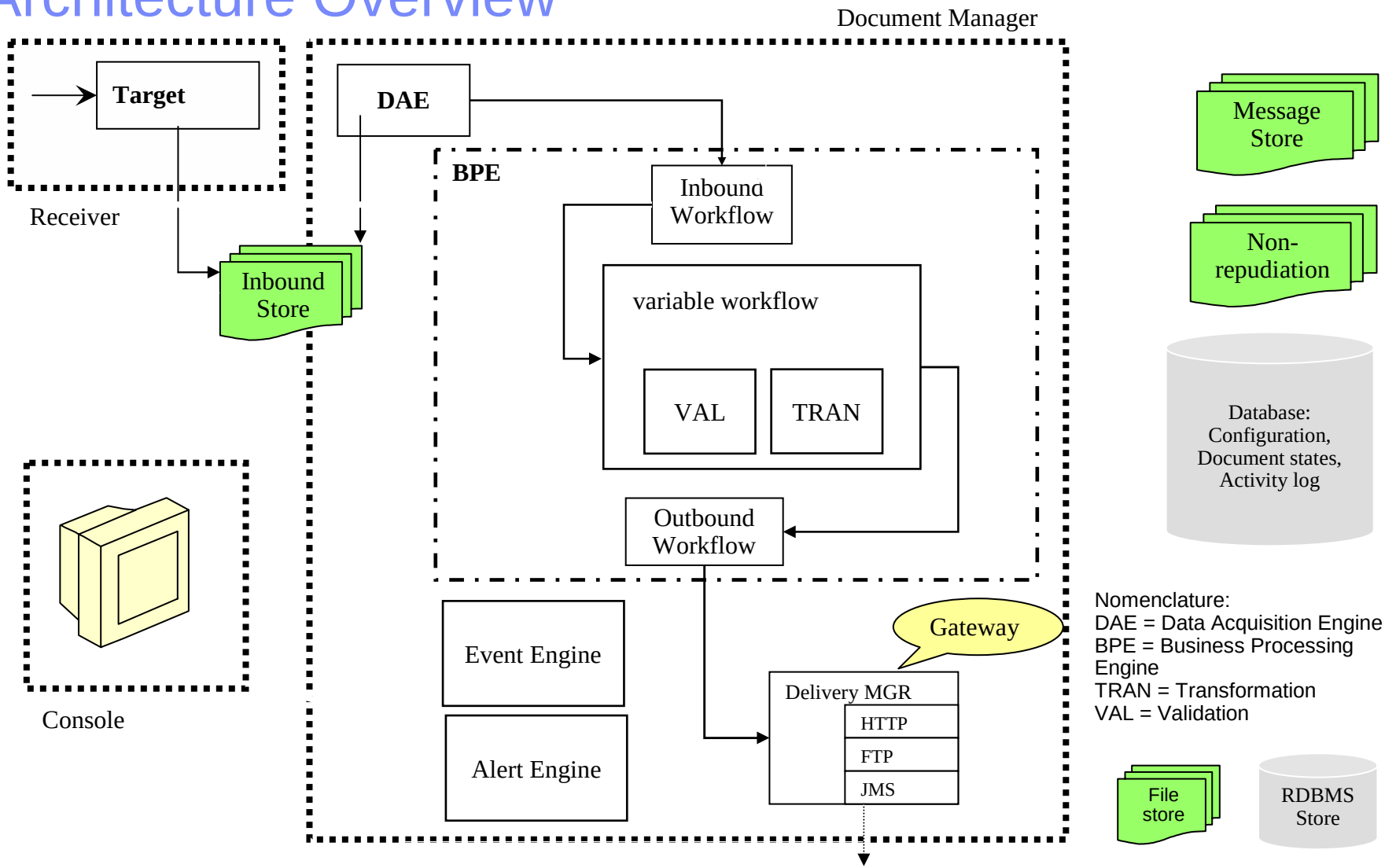
WebSphere. software

ON DEMAND BUSINESS

Objectives

- How can we tune the various components of WebSphere Partner Gateway
  - Document Manager
  - Database (DB2 UDB)
  - I/O subsystem
  - WebSphere MQ

Architecture Overview



Connection Cache Expiration (1)

- For each document, WebSphere Partner Gateway needs to look up the connection between the trading partners
- Expensive database operation
- Connection memory cache
  - Time based : cache entries expire after 10 seconds
  - Maximum of 20 cache entries

Connection Cache Expiration (2)

- bcg.channelCache.expiry
  - Value is in seconds
  - Increase the time that cache entries stay valid
- bcg.channelCache.maxSize
  - Increase the size of the cache

Tuning Document Processing

- 3 document “pipelines”
  - Main
  - Synchronous
  - Signals
- For each, can control
  - How often inbound file store is checked
  - How many files are processed
  - How many threads are assigned to process the work

Reading Documents

- bcg.inbound_poll_interval.xxxx
  - How often we'll check for new documents
  - Value is in milliseconds
- bcg.in_thread_count.xxxx
  - How many threads to pick up files from inbound repository
  - Start with value equal to the number of CPUs, with 2 at minimum
- bcg.inbound_files_per_pass.xxx
  - How many files to read in on each poll interval

Processing Documents

- bcg.bpe_thread_count.xxxx
  - How many threads should the BPE use
  - Start at twice the number of CPUs

Event Engine

- Central place where we process all events
- Input is via WebSphere MQ queue DATALOGQ
- bcg.logReceiver.initialNumberOfReceivers
  - How many threads to use to read from queue
  - Start with value equal to number of CPUs, 2 at minimum

Alert Engine

- Central place where we process all alerts
- Input is via WebSphere MQ queues ALERTEVENTQ and ALERTQ
- bcg.eventAlertQReceiver.initialNumberOfReceivers
  - How many threads to use to read from ALERTEVENTQ
  - Start with value equal to number of CPUs
- Bcg.alertQReceiver.initialNumberOfReceivers
  - How many threads to use to read from ALERTQ
  - Start with value equal to number of CPUs

Filtering Events

- Bcg.event_log_exclude
  - Value is comma delimited list of events to exclude
  - Reduces amount of queue traffic between BPE, event and alert engines

Non Repudiation (1)

- Bcg.nonrep.inbound-doc
  - Value is true/false
  - If false, do not non-repudiate inbound documents
- Bcg.nonrep.outbound-doc
  - Value is true/false
  - If false, do not non-repudiate outbound document

Non Repudiation (2)

- Potentially 4 copies of a document are copied to the file system
  - 2 for inbound and outbound non-repudiation
  - 2 for in the message store for document viewing
- Depending on your non-repudiation requirements, save on I/O processing by turning non-repudiation off

State Engines

- BPE contains an AS state engine and a RosettaNet state engine
- Configured as JMX Mbeans
- …/router/lib/config/router-was-jmx-container.xml
- Mbean definitions
  - <mbean name="RosettaNetStateEngine …
  - <mbean name="ASStateEngine …
- Stop state engine by commenting out the Mbean entry

Controlling The Document Manager

- Use the thread and polling properties to increase or throttle the throughput
- If events and document logs can lag, throttle the amount of receivers for the event and alert engines
- If a protocol is not used
  - Don’t allocate threads, e.g bcg.in_thread_count.signal=0
  - Don’t start the state engine

Database – Tuan Dang

- WebSphere Partner Gateway supports
  - DB2 UDB v8.2
  - Oracle 9i release 2, 9.2.0.6
  - Oracle 10g release 1, 10.1.0.3
- Will concentrate on DB2 v8.2

DB2 Problem Determination

- Look for SQLxxxx errors in the WebSphere Partner Gateway logs
  - Bcg_console.log, bcg_router.log, bcg_receiver.log
- DB2's error log , file DB2DIAG.log
  - On Windows, sqllib/<instance name,DB2>/
  - On Unix, sqllib/db2dump

DB2 Problem Determination

- What is the current configuration ?
  - db2level
  - db2 get database configuration for <db name>

    Current database configuration parameter values
  - db2 get database manager configuration

    Current instance configuration parameter values

DB2 Problem Determination

- Mismatch between database sort heap and instance sort heap threshold
- Most common symptom :
  - In db2diag.log, message “unable to get memory for sort”
- The sort heap sets the maximum number of memory pages to be used for sorting for a database
- The sort heap threshold is the maximum memory used for sorts at any one time by all the databases in an instance

DB2 Problem Determination

- Sort Heap Threshold value should be a multiple of the largest sort heap value of the databases in the instance
  - At least, 2 times the sort heap value
- WebSphere Partner Gateway creates its database with sort heap set to 16K pages

DB2 Problem Determination

- To change sort heap threshold
  - db2 update dbm cfg using sheapthres 32768
- To change sort heap for a database
  1. db2 connect to <dbname> user <id> using <password>
  2. db2 update db cfg using sortheap 16384

DB2 Reorgchk/Runstats (1)

- DB2 keeps statistics on tables so that it can optimize access logic (the access plan)
- In a new database, most tables are empty or contain very small amount of data. So, the access plan is to scan all rows in the table
- As tables fill up, performance suffers as scans take longer

DB2 reorgchk/runstats (2)

- Run a workload through WebSphere Partner Gateway for some period of time
- Then execute the following commands
  - db2 connect to <db name>
  - db2 -v reorgchk update statistics on table all
  - db2 connect reset
  - db2rbind <db name> -l logfile all -u <id> -p <password>
  - db2stop
  - db2start

DB2 reorgchk/runstats (3)

- db2 -v reorgchk update statistics on table all
  - This command checks if table storage needs to be reorganized
  - At the same time, it will update the statistics for the tables

DB2 reorgchk/runstats (4)

- db2rbind
  - This command rebinds all application packages, functions and stored procedures in the database
  - During this step, all access plans are re-calculated using the updated statistics from the REORGCHK step
  - If there is enough statistics data, the access plans will switch from table scans to using indexes defined on the table
- db2stop / db2start
  - Restart the database to make sure all updates take effect

DB2 reorgchk/runstats (5)

- REORGCHK
  - As data is inserted into a table over time, the table storage can become fragmented
  - REORGCHK will determine if any table or index needs to be reorganized or cleaned up

DB2 reorgchk/runstats (6)

- RUNSTATS
  - This command updates the statistics for one table
  - Can be less time consuming than REORGCHK

DB2 reorgchk/runstats (7)

- Run REORGCHK or RUNSTATS on a regular basis
- Will need to tradeoff between time and resources needed to run these commands versus potential performance degradation over time
- Remember to rebind so that access plan can take advantage of updated statistics

DB2 Monitoring

- Need to measure what your database is doing
- These numbers will help in problem determination and performance management
- Execute on a regular basis
- 2 types of monitor
  - Snapshot – activity for a given point in time
  - Event – usage over a period of time

DB2 Monitor Switches (1)

BUFFERPOOL	Buffer pool usage statistics
LOCK	Number of locks and deadlocks
SORT	Sort activity and overflows
STATEMENT	SQL statements usage statistics
TABLE	Table read and write usage statistics
UOW	Unit of work measures: start/stop time, status

DB2 Monitor Switches (2)

- To turn monitor switches on
  - db2 update monitor switches using <switch name> on

DB2 Snapshots

- To get monitor snapshot data
  - db2 get snapshot for bufferpools on <db name>
  - db2 get snapshot for locks on <db name>
  - db2 get snapshot for dynamic sql on <db name>
  - db2 get snapshot for tables on <db name>
  - db2 get snapshot for applications on <db name>
  - db2 get snapshot for all on <db name>

DB2 Buffer Pool

- A buffer pool is the “in memory” work area for the database
- The database server uses the buffer pool when performing any transactional activity (reads, writes, updates, deletes, etc …)
- Data is copied to and from buffer pools as needed using IOServers and IOCleaners

DB2 Buffer Pool Monitoring (1)

- Effectiveness is measured by how frequently requested data is already in the buffer pool
- Hit ratio
  - (1 – (pool physical reads) / (pool logical reads) * 100
- The closer the hit ratio is to 100, the lower the disk I/O overhead

DB2 Buffer Pool Monitoring (2)

- IO Servers are the processes that do prefetch and asynchronous I/O from disk to the buffer pool
- DB2 recommends that you have one or two more than the number of physical devices on which the database resides
- You can adjust the number of IO Servers based on
  - Pool data reads vs Pool async data reads
- Configure parameter NUM_IOSERVERS

DB2 Buffer Pool Monitoring (3)

- IO Cleaners are the processes that write changed data from the buffer pool to disk before space is requested by database agents
- DB2 recommends that you have from one to the number of physical devices used for the database
- You can adjust the number of IO Cleaners based on
  - Pool data writes versus Pool async data writes
- Configure parameter NUM_IOCLEANERS

DB2 Buffer Pool Monitoring (4)

- On a dedicated database server, the rule of thumb is to use 75% of main memory for buffer pools
- Access plans take buffer pool size into account so consider rebinding if you modify the buffer pool size
- Watch out for page swapping if your system has limited memory or if other applications are memory intensive

DB2 Buffer Pool Monitoring (5)

- To set the Buffer pool size
  - db2 alter buffer pool <buffer pool name> size <number>
- Note: as of v8.2, you can no longer use the BUFFPAGES configuration parameter

DB2 Lock Monitoring (1)

- A database has a finite memory area for locks
- If this memory area cannot hold the amount of locks requested, the database will free up memory by replacing row locks with table locks
- Lock escalation thus impacts performance by reducing concurrency
- We want to minimize escalations and lock wait time

DB2 Lock Monitoring (2)

- The snapshot will indicate
  - Number of lock escalation
  - Lock list memory in use
  - Amount of time waiting for locks
- From this data, you can configure parameters
  - LOCKLIST – the total amount of space allocated for locks
  - MAXLOCKS – the percentage of LOCKLIST used by an application

DB2 Sort Monitoring

- Snapshot will indicate
  - Total Private Sort heap allocated
- Use this value to optimize configuration parameter SORTHEAP

DB2 Tablespaces

- A tablespace is the logical storage device for database objects
- 2 types
  - System Managed (SMS)
  - Database Managed (DMS)

DB2 SMS TableSpace

- Default used by WebSphere Partner Gateway
- Size is automatically managed by DB2
  - DB2 will automatically increase size as needed
  - Usually maps to a file on the file system

DB2 DMS Tablespace (1)

- Managed by user
- User needs to monitor space consumption and increase size as needed
- Much faster than SMS tablespace
  - Does not have to map to file system
  - Can bypass operating system I/O module
  - Can do raw I/O on actual physical device

DB2 DMS Tablespace (2)

- When running DBLoader installer, instruct installer to not run the SQL database creation scripts
- Modify file Create_db2.sql with tablespace commands

I/O

- Consider using RAID arrays
- Separate database and WebSphere Partner Gateway file stores
  - Separate physical drives
  - For database, separate out transaction logs

DB2 Transaction Logs

- Holds uncommitted data
- Usually among the most I/O intensive DB2 components
- Isolate from rest of system by assigning its own physical device
- Database configuration parameter NEWLOGPATH

WebSphere MQ (1)

- WebSphere Partner Gateway uses WebSphere MQ queues to communicate between its components
- Problems
  - Transaction rollback
  - Queues filling up too fast

WebSphere MQ (2)

- Transaction rollback
- WebSphere Partner Gateway taking too long between commits and WebSphere MQ transaction logs fill up
- Increase log sizes
  - <MQ dir>/qmgrs/<qmgr name>/qm.ini
  - LogPrimaryFiles=62
  - LogSecondaryFiles=2
  - LogFilesPages=2048
  - LogBufferPages=128

WebSphere MQ (3)

- When processing EDI interchanges with multiple transactions, many events are generated and queued on DATALOGQ to the event engine
  - At least 4 events per transaction
  - Event Engine does not keep and queue fills up
- Increase queue depth
  - At least queues DATALOGQ, DELIVERYMANAGERQ, ALERTQ, ALERTEVENTQ
- Monitor queues on regular basis

Miscellaneous

- Out of memory condition while processing large documents or interchanges with multiple transactions
  - Increase Java Virtual Machine heap size
  - Default is 256 Megs
  - Use scripts

    <install root>/scripts/bcgQueryJVMHeapAttrs.jacl

    <install root>/scripts/bcgSetJVMHeapAttrs.jacl

References (1)

- WebSphere Partner Gateway
  - InfoCenter

    http://www-306.ibm.com/software/integration/wspartnergateway/library/infocenter/
  - Redbook

    http://www.redbooks.ibm.com/redpieces/abstracts/sg247109.html?Open
- Database
  - InfoCenter

    http://publib.boulder.ibm.com/infocenter/db2help/index.jsp
  - Developer Works

    http://www-128.ibm.com/developerworks/db2
- WebSphere MQ
  - InfoCenter

    http://www-306.ibm.com/software/integration/wmq/support/
- WebSphere Application Server
  - InfoCenter

    http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp

Summary

- Tuning of database is most beneficial step
- Use Thread and MBean properties to regulate Document Manager
- Monitor on regular basis