IBM DB2 Performance Solutions

HENNIE MYNHARDT
mynhardt@us.ibm.com

Henry Yim
yimhenry@us.ibm.com
Please note

IBM’s statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM’s sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

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.
Agenda

- IBM DB2 Tools – Solution Packs overview
- DB2 Performance Solution
- OM/PE – what is new and how it addresses your pain points
- DB2 Query Monitor – what is new and tuning with key metrics
- Close / Q&A
History - Before you were born (some of you)

- NT (Not There)
- Leap of Faith
- Low hanging fruits and cheap
- Hire and Acquire
- Can’t count tools
- Solution Packs
- Extra-Autonomic functions via Solution Packs only
- More integration
DB2 Solution Overview

“More data, reduce costs”
Optimize, control manage & automate
How do we get the best control over DB2 utility processing?
How do we save CPU & Elapsed time while improving availability?
How can we smartly automate our DB2 utilities execution?

“Protect Data Eliminate CPU”
Insure, modernize optimize & protect
How do we maximize our storage for use with DB2 for z/OS?
Can we guarantee recovery of DB2 data & objects from app errors to disaster recovery?
How do we reduce CPU and maintain performance?

“Improving the bottom-line”
Identify, diagnose solve & prevent
How do we zero in on any performance issues affecting our profitability?
How can we avoid performance issues and not impact SLA or users?

“Limited Resources”
Navigate, manage change & track
How do we become more efficient in our day-to-day tasks?
How do we ensure the integrity of our DB2 for z/OS assets throughout the application lifecycle?
How do we do more with less resources?
Solution Overview - Product Components

**DB2 Utilities Solution Pack**
- DB2 Automation Tool
- DB2 HPU
- DB2 Sort
- DB2 Utilities Enhancement Tool

Supercharge the IBM utilities

**DB2 Performance Solution Pack**
- DB2 SQL Performance Analyzer
- Tivoli Omegamon XE for DB2 PE
- Optim Query Workload Tuner
- DB2 Query Monitor

Master the performance lifecycle

- One PID per solution pack
- Price is discounted with the solution pack
- Build more intelligence into when and how actions are performed???

**DB2 Fast Copy Solution Pack**
- DB2 Cloning Tool
- DB2 Recovery Expert

Superior avail & cost for copy, backup & recovery operations

**DB2 Administration Solution Pack**
- DB2 Administration Tool
- DB2 Object Comparison Tool
- DB2 Table Editor
- Optim Configuration Manager

Manage objects & schema
DB2 Performance Solution

- Prevent performance problems before they occur
  - Use performance data for pre-emptive analysis and capacity planning
  - Thread situations and SQL exception or alert thresholds
  - Help developers identify query hot spots, tune queries, and validate results

- Improve application performance
  - Collect and understand application performance
  - Get query recommendations, optimize statistics, create appropriate indexes
  - Optimize results for entire workloads, not just single queries

- Ensure SLAs and user satisfaction are achieved
  - Pinpoint and isolate problems to correct instantly
  - End-to-end

- Integrate business priorities directly
  - Monitor KPIs to better reflect end user experience
  - Monitor and report on transaction response-time service objectives
  - Allocate resources according to business priorities

- **Save hours of staff time and stress**
  - Isolate problems to correct a problem area instantly

“Improving the bottom-line”

More companies are facing growth of data volume, but not staff.

Need:

- Smarter approach w/ less resources
- Proactively optimize performance
- Keep up more w/ users, growing & changing requests
DB2 Performance Solution Pack

- **OMEGAMON for DB2 Performance Expert on z/OS**
  - Full range system and application monitoring and reporting
  - End-to-end transaction tracking via Extended Insight, Stored Procedure monitoring
  - IBM DB2 Analytics Accelerator online and batch reporting

- **DB2 Query Monitor for z/OS**
  - Dedicated SQL monitoring; ISPF and web interfaces
  - Current and historical views of query detail throughout DB2 subsystems with ability to set alerts for early warning
  - Integration with Optim Query Workload Tuner
  - IBM DB2 Analytics Accelerator query analysis and drill down

- **Optim Query Workload Tuner for DB2 for z/OS**
  - Eclipse-based SQL performance tuning and expert advice
  - Integration with OMEGAMON XE for DB2 PE and Query Monitor
  - Single or workload tuning via expert advisors to provide tuning recommendations; Access Plan Comparison

- **DB2 SQL Performance Analyzer for z/OS**
  - ISPF-based SQL performance analysis
  - Cost estimate and explain function with “what if” analysis
Performance Solution Pack – Usage Scenarios

- **Fixing a web application slowdown**
  1. Extended Insight feature of OMPE identifies where the slowdown is occurring and QM enables drill-down of SQL
  2. Problem SQL passed to Query Workload Tuner and Statistics Advisor is run to determine stats are out of date
  3. Once stats are updated, OMPE and QM are used to validate normal performance of the application

- **Identifying and fixing poorly performing queries**
  1. Collection of high CPU SQL statements identified in QM passed to Query Workload Tuner & analyzed as a workload
  2. Advisors are run against the workload and recommendations are implemented to improve performance
  3. Once adjustments are completed, applications are re-deployed with OMPE and QM used to validate performance

- **Tuning queries during application development**
  1. App developers leverage OQWT and/or SQL PA to fine tune SQL statements
  2. OMPE and QM used to validate performance once apps deployed into production
OMEGAMON XE for DB2 Performance Expert - Evolution

Driving factors:
- Customer Requirements
- DB2 V9, V10, V11 exploitation
OMEGAMON DB2 and DB2 Tools available to all customers in DB2 11 ESP

- DB2 10 program success making Tools available to ESP customers
- Repeat story for DB2 11 utilizing new OMEGAMON DB2 features
  - The Spreadsheet Input Generator – easy to use and key to analyzing DB2 11 application performance as compared to DB2 10 workload

What is your experience with OMPE spreadsheet capability?

- Existing OMEGAMON user
  - “It has been top-dollar! We love it!! I estimate less than 30 minutes from workload finished to spreadsheet ready to upload!!

- Non-OMEGAMON user
  - Overall, my current experience using the Batch reporting and CSV generation has been very positive"
OMEGAMON XE for DB2 Performance Expert V520

- Reduced footprint improvements
  - Notable OMEGAMON performance gains
    - zIIP offload
    - Code path reduction
    - Storage reduction

- DB2 V11 support

- PARMGEN (ICAT can be imported, but eventually going away)

- Other enhancements and PTF’s
  - e3270
  - New methodologies/techniques in monitoring
    - Monitor and resolve issues w/o using performance traces and control blocks via SQL dashboard and statement cache
OMEGAMON DB2 Performance Expert competitive differentiators

- **Support for IBM Data Analytics Accelerator**
  - If you want to monitor IDAA, bring up OMPE to the latest release

- **Monitoring STORED PROCEDURES**
  - IBM is seeing a steady migration to stored procedure exploitation especially in distributed application development:
    - Centralization of program logic to tried-and-true legacy centers of competency ensure accuracy and consistency
    - Centralization may also create single points of failure and / or bottlenecks

- **Monitoring End-to-end : DISTRIBUTED APPLICATIONS**
  - Businesses have fully embraced distributed technologies to reduce costs and improve customer satisfaction by giving clients control of their own data
Support for IBM Data Analytics Accelerator

- **IBM Data Analytics Accelerator reporting supported via OMPE**
  - Batch statistics trace/report of IBM Data Analytics Accelerator used by DB2 subsystem
  - Batch accounting of applications with IBM Data Analytics Accelerator accelerated SQL queries with the accelerator specific performance metrics
  - Batch record trace reporting on single DB2 trace record with accelerator specific metrics
  - Statistics metrics in real-time

```plaintext
| PROJECT ID: | ACCEL01 |
| SERVER ID: | EME03 |
| STATE: | ONLINE |

### ACC 001234

<table>
<thead>
<tr>
<th>ACTION</th>
<th>COUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXECUTION</td>
<td>100</td>
</tr>
<tr>
<td>REQUESTS</td>
<td>500</td>
</tr>
</tbody>
</table>

### ACC 002345

<table>
<thead>
<tr>
<th>ACTION</th>
<th>COUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXECUTION</td>
<td>150</td>
</tr>
<tr>
<td>REQUESTS</td>
<td>750</td>
</tr>
</tbody>
</table>

### ACC 003456

<table>
<thead>
<tr>
<th>ACTION</th>
<th>COUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXECUTION</td>
<td>200</td>
</tr>
<tr>
<td>REQUESTS</td>
<td>1000</td>
</tr>
</tbody>
</table>
```

### ACC 004567

<table>
<thead>
<tr>
<th>ACTION</th>
<th>COUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXECUTION</td>
<td>250</td>
</tr>
<tr>
<td>REQUESTS</td>
<td>1250</td>
</tr>
</tbody>
</table>

### ACC 005678

<table>
<thead>
<tr>
<th>ACTION</th>
<th>COUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXECUTION</td>
<td>300</td>
</tr>
<tr>
<td>REQUESTS</td>
<td>1500</td>
</tr>
</tbody>
</table>

### ACC 006789

<table>
<thead>
<tr>
<th>ACTION</th>
<th>COUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXECUTION</td>
<td>350</td>
</tr>
<tr>
<td>REQUESTS</td>
<td>1750</td>
</tr>
</tbody>
</table>

### ACC 007890

<table>
<thead>
<tr>
<th>ACTION</th>
<th>COUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXECUTION</td>
<td>400</td>
</tr>
<tr>
<td>REQUESTS</td>
<td>2000</td>
</tr>
</tbody>
</table>

### ACC 008901

<table>
<thead>
<tr>
<th>ACTION</th>
<th>COUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXECUTION</td>
<td>450</td>
</tr>
<tr>
<td>REQUESTS</td>
<td>2250</td>
</tr>
</tbody>
</table>

### ACC 009012

<table>
<thead>
<tr>
<th>ACTION</th>
<th>COUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXECUTION</td>
<td>500</td>
</tr>
<tr>
<td>REQUESTS</td>
<td>2500</td>
</tr>
</tbody>
</table>
```
OMEGAMON DB2 Stored procedure monitoring

- Fundamental reporting in accounting reports / by plan and package
- Gaining more insights into stored procedure than can be accomplished with batch
- New trace in DB2 V10

Show the SQL executed by a SP

Supports nested SPs
End-to-end monitoring - Where is my distributed application spending its time?

- **OMEGAMON XE for DB2 PE’s Extended Insight** is an advanced way to monitor the database workload (SQL) of your applications, end-to-end
  - Get total response times and response time breakdown (appl, app server, driver, network, data server) per defined workload/cluster (e.g. per system, application, user)
  - Compare workload from various servers / applications
  - Select a time period for analysis
  - Get top SQL statements per defined workload
  - Identify top clients contributing in the workload
Extended Insight Analysis Dashboard

Dynamically change the time period and duration for analysis

Compare response time break down for different workloads

Workload Clusters represent client applications (pre-defined or custom)
IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS - Summary of Key Capabilities

Advanced offerings
- PERFORMANCE EXPERT ONLY: Extended Insight and the stored procedure monitoring
- Support for IBM Data Analytics Accelerator
- FULL Day-one support for DB2 versions

Productivity
- Enhanced 3270 user interface
- Enterprise-wide middleware monitoring
- Advanced mechanisms for complex, cross-discipline situations & alerting

- Integration with SQL analysis tools like Optim tools and SQL/PA
- Extensions to DSG support to include zPARMs
- DB2 Connect monitoring

Total cost of ownership (TCO) reduction
- zIIP offload for improved resource consumption
- Robust buffer pool analysis
- Extensive object analysis

Plan and prevent
- Trending, capacity planning and historical analysis via a performance database
- Complete historical view of DB2 threads and transactions
- Industry leading batch reporting
The Architectural Overview - the components
End User Interfaces

OMEGAMON XE GUI Interface (TEP)
- Real time and historical
- Automation & alerts – Situations & Policies
- Integration with other Omegamon Monitors
- Plex level information (CF, n-way)

OMEGAMON VTAM Classic
- 3270 Interface command interface
- Real Time & Historical (NTH)
- Object analysis
- Application Trace

Enhanced 3270
- All Omegamon products

PE GUI
- GUI client interface
- Performance Warehouse (ROT, Expert)
- Bufferpool Analysis

ISPF
- Exception threshold specification & profiling
- Trace collection
- Report tailoring and JCL generation

OMEGAMON CUA

Extended Insight Analysis & SQL Dashboard for:
- End-to-end SQL monitoring
- Stored procedure monitoring
Enhanced 3270 user interface

- Easier navigation and fewer screen interactions to accomplish tasks, instead of a series of green screens to gather and display information.
- Transactions can be linked across multiple sysplexes; no need to move between multiple screens and monitors.
- Centralized management environment (Tivoli OMEGamon Manager) from which various activities can be launched.
- Color highlighting to flag problems for quicker resolution.
- TEPS like consolidated data across entire OMEGamon family.

e-3270 is a “green screen” GUI that provides an enterprise view of information supported across the entire OMEGamon family.

To us, these new IBM 3270 screens accomplish functionally the same thing as their GUI equivalents. The 3270 “graphical “green screen” approach offers the same information that a GUI screen offers — only displayed slightly differently. What IBM has done with its “GUI on a green screen” is it has allowed some mainframe managers to stay within their 3270 comfort zone while getting the same display benefits that GUI mainframe managers get using graphically-oriented products “ – Clabby Analytics.

Enterprise management summary can be created that shows what is happening across all active Sysplexes.
Systems Management from Tivoli
System z Portfolio
Redesigned OMEGAMON moving to simplified architecture with less footprint

Current OMEGAMON Architecture

- Classic 3270
- CUA 3270
- TEMAS
- TEP GUI

Enhanced OMEGAMON Architecture

- TEMAS
- TEP GUI

Both architectures support:
- IMS
- z/OS
- CICS
- Storage
- MfN
- DB2
- Messaging

Current OMEGAMON Architecture:
- Multiple Address Spaces across each OMEGAMON family

Enhanced OMEGAMON Architecture:
- Single Manager across OMEGAMON family
OMEGAMON Classic (VTAM) end user interface

the Summary of DB2 Activity

with optional Data Sharing group wide view

<table>
<thead>
<tr>
<th>DSYS</th>
<th>Connections</th>
<th>Threads</th>
<th>CPU</th>
<th>Getpage Rate</th>
<th>Elapsed Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSAS+DBAS+IRLM+DIST CPU</td>
<td>0</td>
<td>0</td>
<td>0.0%</td>
<td>0.0/sec</td>
<td>00:00:00:00</td>
</tr>
<tr>
<td>Create Thread Rate</td>
<td>0</td>
<td>0</td>
<td>0.0%</td>
<td>0.0/sec</td>
<td>00:00:00:00</td>
</tr>
<tr>
<td>Synch Read I/O Rate</td>
<td>35.0K/sec</td>
<td>1.7M/sec</td>
<td>21.0/sec</td>
<td>0.0/sec</td>
<td>00:00:00:00</td>
</tr>
<tr>
<td>Update Request Rate</td>
<td>726.63</td>
<td>0</td>
<td>0.0%</td>
<td>0.0/sec</td>
<td>00:00:00:00</td>
</tr>
<tr>
<td>Getpages/Read I/O</td>
<td>0</td>
<td>0</td>
<td>0.0%</td>
<td>0.0/sec</td>
<td>00:00:00:00</td>
</tr>
<tr>
<td>Current Lock Suspensions</td>
<td>0</td>
<td>0</td>
<td>0.0%</td>
<td>0.0/sec</td>
<td>00:00:00:00</td>
</tr>
<tr>
<td>Locking Deadlocks</td>
<td>0</td>
<td>0</td>
<td>0.0%</td>
<td>0.0/sec</td>
<td>00:00:00:00</td>
</tr>
<tr>
<td>Connection Type</td>
<td>Connections</td>
<td>Threads</td>
<td>CPU</td>
<td>Getpage Rate</td>
<td>Elapsed Time</td>
</tr>
<tr>
<td>IMS</td>
<td>0</td>
<td>0</td>
<td>0.0%</td>
<td>0.0/sec</td>
<td>00:00:00:00</td>
</tr>
<tr>
<td>CICS</td>
<td>0</td>
<td>0</td>
<td>0.0%</td>
<td>0.0/sec</td>
<td>00:00:00:00</td>
</tr>
<tr>
<td>TSO Foreground</td>
<td>0</td>
<td>0</td>
<td>0.0%</td>
<td>0.0/sec</td>
<td>00:00:00:00</td>
</tr>
<tr>
<td>Batch</td>
<td>1</td>
<td>4</td>
<td>0.0%</td>
<td>0.0/sec</td>
<td>00:00:00:00</td>
</tr>
<tr>
<td>Utilities</td>
<td>0</td>
<td>0</td>
<td>0.0%</td>
<td>0.0/sec</td>
<td>00:00:00:00</td>
</tr>
<tr>
<td>Distributed</td>
<td>1</td>
<td>4</td>
<td>0.0%</td>
<td>0.0/sec</td>
<td>00:00:00:00</td>
</tr>
<tr>
<td>Stored Procedures</td>
<td>0</td>
<td>0</td>
<td>0.0%</td>
<td>0.0/sec</td>
<td>00:00:00:00</td>
</tr>
<tr>
<td>All Connections</td>
<td>0</td>
<td>0</td>
<td>0.0%</td>
<td>0.0/sec</td>
<td>00:00:00:00</td>
</tr>
</tbody>
</table>

Change ‘S’ to ‘G’ to obtain Data Sharing Group-wide view
OMEGAMON Classic (VTAM) end user interface

DSG Support - zoom-in to remote thread

Horizontal navigation within thread detail (*)
Fields in exception are highlighted.

Command driven panel invocation (panels are customizable)
You can detect exceptional situations/events, see details and Expert Advise, and you can Take Action.
ISPF interface

Command ===>

Select one of the following.

1. Create and execute reporting commands
2. View online DB2 activity - Classic Interface
3. View online DB2 activity - PE ISPF OLM
4. Maintain parameter data sets
5. Customize report and trace layouts
6. Exception profiling
**The “PE Client”**

A java application which connect via TCP/IP directly with the data collector. It provides a graphical end user interface:

- Deep dive real-time monitoring, like Classic (3270)
- Unique functions like “Snapshot History” and “PWH analysis”
- Sub second response time
End-to-end SQL & Application monitoring
(Extended Insight Analysis Dashboard)

Extended Insight Analysis Dashboard: OMP1D911

- Back
  Locate the source of performance problems, determine how these problems affect different parts of the workload, and analyze the performance of individual SQL statements, clients, and partitions.

Response Time Details: 9.152.205.30

**Graph & Grid**

- Selected layer: No layer selected
- Show Maximum

**SQL Statements & Clients**

- Show highest
- by Average Data Server Time (sec)

<table>
<thead>
<tr>
<th>Statement Text</th>
<th>Statement Executions</th>
<th>Average Data Server Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT 'PUT_40K' AS 'VALUE', '100519' AS 'TIME', '1' AS 'STMTNR', '100K0' AS 'LENGTH', 'O' AS 'CL', 'O' AS 'TD', 'false' AS 'TABNEWLINE', COUNT(*) AS COUNT FROM LOQD0002 WHERE A=000100 OR A=000100 OR...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.584</td>
<td></td>
</tr>
<tr>
<td>SELECT 'PUT_40K' AS 'VALUE', '100519' AS 'TIME', '1' AS 'STMTNR', '100K0' AS 'LENGTH', 'O' AS 'CL', 'O' AS 'TD', 'false' AS 'TABNEWLINE', COUNT(*) AS COUNT FROM LOQD0002 WHERE A=000100 OR A=000100 OR...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.474</td>
<td></td>
</tr>
<tr>
<td>SELECT 'PUT_40K' AS 'VALUE', '100519' AS 'TIME', '1' AS 'STMTNR', '100K0' AS 'LENGTH', 'O' AS 'CL', 'O' AS 'TD', 'false' AS 'TABNEWLINE', COUNT(*) AS COUNT FROM LOQD0002 WHERE A=000100 OR A=000100 OR...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.510</td>
<td></td>
</tr>
<tr>
<td>SELECT 'PUT_40K' AS 'VALUE', '100519' AS 'TIME', '1' AS 'STMTNR', '100K0' AS 'LENGTH', 'O' AS 'CL', 'O' AS 'TD', 'false' AS 'TABNEWLINE', COUNT(*) AS COUNT FROM LOQD0002 WHERE A=000100 OR A=000100 OR...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1.393</td>
<td></td>
</tr>
<tr>
<td>DROP</td>
<td>1</td>
<td>1.027</td>
</tr>
</tbody>
</table>

**Detail Area for SQL Statements**

- Statement Information

- Package name: N/P
- Section number: 0
- Package Consistency token: N/P
- Package Version: N/P
- Collection: N/P

- Java class: N/P
- Java package: N/P
- Method: N/P
- Source line number: N/P
- Build version: N/P
- Source expression: N/P
- Method signature: N/P
- Application Name: N/P
- Metadata file: N/P

- Transfer Volume

- Average bytes transferred locally: 0 bytes
- Average bytes transferred remotely: 41,369 KB
- Average rows returned: 0
- Average number of round trips: 1

**Statement Performance**

- Number of Executions: 1
- Average end-to-end elapsed time: 0.466 sec
- Average client time: 0 sec
- Average driver time: 0 sec
- Average network time: 0.018 sec
- Average data server time: 0.474 sec

**Statement Time Distribution (%)**

- Client time: 97.33%
- Driver time: 2.67%
- Network time: 0%
- Data server time: 0%

**Statement Outcome**

- Failure rate (with negative SQL code): 0.0%
- First SQL code: N/P
Where Is My Problem?
OMEGAMON for DB2 Extended Insight capability

Extended Insight looks beyond the database
- See where transactions spend time
- Monitor workload response time
- View database time spent analysis

Most database monitors focus only on the database
Where is my problem?

=> End-to-end monitoring with SQL level deep dive - Extended Insight

- … tells me which application it is, and ultimately which business function

- … where I can measure what my application/user is really experiencing. Tells me which components are involved and where my application/SQL is spending its time

- … tells me where the SQL statement is coded and let’s me change it if necessary
One major pain point - What is happening outside of DB2? (e.g. with remote applications)

- DB2 for z/OS
- OMPE STC (data collector, agent, TEMS)
- Network
- Customer’s application Host Batch, CICS, etc.
- Application Server e.g. WebSphere
- Customer’s Distributed Application Programs
- Data Server Drivers (JCC, CLI, .NET)
- z/OS

3270 Classic, CUA, GUI TEP, PE client
How does it work?

DB2 for z/OS

OMPE STC (data collector)

Repository Database

Linux, zLinux, Unix, Windows

Can run as stand-alone web UI

z/OS

Customer's Application program

N
e
t
w
o
r
k

Application Server e.g. WebSphere

Data Server Drivers (JCC, CLI, .NET)

EI Client Software

Classic, CUA, PE client

Color coding:
- Orange = OMEGAMON XE for DB2 PE V510
- Green = DB2 for z/OS and DB2 Data Server drivers

TEP
OMPE on z/OS TEP
Navigation to the Extended Insight Analysis Dashboard

Navigation to the OMPE workspace with the E2E SQL monitoring information
OMPE on z/OS TEP
Navigation to the Extended Insight Analysis Dashboard
OMEGAMON DB2 PE 5.2 Extended Insight
Zoom into selected workload and see the TOP SQL list

Top SQL statements executed by Java or CLI applications like SAP, Cognos, DataStage or WebSphere

Zoom in on a selected SQL

Detailed End-to-End Response Time
Who or what caused the Spikes?
zoom in!

=> Move cursor over the spike to see the date and time, and narrow down …
Who or what caused the Spikes?

... narrow down the observation period to this end time and select a 2 minutes duration!
Production test with increased workload

=> It is not DB2!

Response Time Details: RDB2

- Average End-to-End Response Time:
- Overall average response time per transaction: 0.014 sec
- Maximum response time: 17.176 sec
- Maximum Time of running transactions: 01:02.133 min
- Number of executions: 376,102
- Statements: 376,338
- Statement Failure Rate: 0%

Time Distribution (%):
- Client time: 78.570%
- Network time: 14.230%
- Data server time: 7.140%
Production test with increased workload

=> It is not DB2!
How a monitor can help? *proactive and efficient*

- Only a few people have really time to sit in front of a monitor and watch performance
  - Be proactive rather than reactive – use monitors to notify you when there are issues – take advantage of monitoring capabilities:
    - Capture exceptions
    - Send alerts when necessary

- Historical Data
  - Report and analyze DB2 accounting and statistics trace data
  - Set up a PDB or a PWH to collect historical data for trending and analysis
  - Load accounting and statistics trace data into a repository

**Note:**
- PDB into IDAA
- Analysis Requirement to join OMPE and QM PDB
- Workshop for OMPE and QM
- Some functions only via Solution Pack only
Summary of functions

- **Integrated cross zSeries monitoring**
  - OMEGAMON 3270 UI
  - Browser UI (TEP/Extended Insight)
  - In addition: DB2 PM/PE ISPF and PE client GUI

- **Batch Reports**
  - Statistics
  - Accounting
  - Subsystem Parameters
  - Locking
  - SQL Activity
  - I/O Activity
  - Utility
  - Audit
  - Record Trace
  - Explain

- **Performance Warehouse**
  Performance database with Rules-of-thumb and “expert” SQL queries

- **The Buffer Pool Analysis function**
  (included with the “PE” product offering)

- **Real-Time monitoring**
  - Thread and Statistics Details
    - Data Sharing Support
    - Exception processing
      - Snapshot history
      - Object analysis
      - Near-term history
      - PWH processing

- **Distributed Application Monitoring**
  - Extended Insight (End-to-End SQL)
    (available with the “PE” offering)
  - DB2 Connect (gateway) monitoring

- **One collector per LPAR**
  - Only one started task per LPAR needed
  - One separate collector subtask per DB2 subsystem
DB2 Query Monitor
New Features

- DB2 V11 exploitation
- Literal stripping
- Additional delay statistic on objects
- SQL Execution Count now used for averages
- New Browser Features
- Additional zIIP enablement
- Collector code path optimization
  - FETCH intensive workload optimization
- Improved Offload Process
- Improved Messages
- New operation commands
Installation and Customization Improvements (V3.2)

1) DB2 Data Access Collector for z/OS
   • Program Number 5639-OLC
     • FMID HCQC110
   • Introduced with the following products
     • DB2 Query Monitor for z/OS V3R2
     • InfoSphere Guardium S-TAP for DB2 on z/OS V9R1
     • InfoSphere Optim Workload Replay V2R1
   • Delivery vehicle for future collector maintenance and enhancements

2) Tools Customizer for z/OS
   ‣ New to V3R2
   ‣ Standardizes QM with other tool install processes
   ‣ Discovery of QM V3R1 setup
   ‣ Creates all the necessary jobs to customize QM

3) QM V3.2
Response Time Factors

General transaction flow
DB2 Times Terminology - Accounting Class 1, 2, 3
Accounting Class 1 Data (Thread allocation to termination)

For local applications DB2 provides an accumulative timer which includes both CPU spent in application and CPU spent in DB2. Activity time is very close to Class 1 elapsed time. For local applications, Class 1 (elapsed - CPU) time can point to application inefficiencies. Class 1 elapsed time is not meaningful for distributed and thread reuse applications.

<table>
<thead>
<tr>
<th>AVERAGE</th>
<th>APPL (CL.1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELAPSED TIME</td>
<td>0.012122</td>
</tr>
<tr>
<td>NONNESTED</td>
<td>0.012122</td>
</tr>
<tr>
<td>STORED PROC</td>
<td>0.000000</td>
</tr>
<tr>
<td>UDF</td>
<td>0.000000</td>
</tr>
<tr>
<td>TRIGGER</td>
<td>0.000000</td>
</tr>
<tr>
<td>CP CPU TIME</td>
<td>0.001482</td>
</tr>
<tr>
<td>AGENT</td>
<td>0.001482</td>
</tr>
<tr>
<td>NONNESTED</td>
<td>0.001482</td>
</tr>
<tr>
<td>STORED PROC</td>
<td>0.000000</td>
</tr>
<tr>
<td>UDF</td>
<td>0.000000</td>
</tr>
<tr>
<td>TRIGGER</td>
<td>0.000000</td>
</tr>
<tr>
<td>PAR. TASKS</td>
<td>0.000000</td>
</tr>
</tbody>
</table>
Where is the most time spent?

Need to analyze time distribution: where is the time really spent?

- Application logic inefficiency
  - often combined with Class 2 CPU << Class 1 CPU
- Network problems
- Class 2 not active all the time
Where is the time spent within DB2?

- Turn off expensive traces
- Inefficient access paths
  - Explain

- What is the largest contributor
  - Class 3 and 8 analysis
A Tuning Hierarchy

- Sysplex
- LPAR
- DB2 SSID
- SQL
OMPE vs. QM

- **OMPE**
  - Reports on DB2 accounting data
  - On-line is thread based
  - Excellent batch reporting from trace data
    - Plan
    - Package
  - Includes application Class 1 time

- **QM**
  - Doesn’t use trace data
  - Is completely SQL statement based
  - Provides on-line and batch reports
  - Does not include application Class 1 time, nor Class 2 time for a thread
“Expensive” or Poorly Tuned SQL?

- **Expensive SQL**
  - High total CPU usage
  - High execution count
  - High resource utilization

- **Poorly Tuned SQL**
  - Excessive resource utilization
    - Excessive I/O
      - Buffer pool hit ratios
      - Synchronous I/O
      - Asynchronous I/O
        - Prefetch
    - Excessive memory usage
      - Usually shows in getpage counts

- What is your query tuning objective
  - Reduced CPU usage or reduced elapsed time?
Key Metrics

- Focus is generally on CPU Time
- Dynamic SQL
  - High PREPARE cost
    - Dynamic statement cache usage
  - Use Stored Procedures when possible
- GetPage requests
  - A key metric to watch
  - Can be misleading
- Number of data rows
  - Required to determine the result set
  - Returned to the application program
- Elapsed time
  - Focus on reducing delays
Key Metrics to Monitor (CPU Usage)

- **CPU utilization**
  - What’s the total CPU consumed by an SQL statement?
    - By hour?
    - By day?
    - By week, month, quarter, year?
  - Has the CPU usage changed?
    - Increased or decreased?
    - Application changes?
    - DB2 upgrade?
      - New release?
      - Maintenance?
  - CPU to elapsed time ratio
    - Nirvana is 1:1
    - Very well tuned system is 1:1.5
    - Varies from LPAR to LPAR
Key Metrics to Monitor (CPU Usage)

- GetPage count
  - General rule
    - High GetPages increases CPU utilization
  - DB2 data is compression can affect this rule
    - Read Robert Catterall’s blog “GETPAGEs are not ALWAYS the Key Determinant of SQL Statement CPU Time” (http://robertsdb2blog.blogspot.com/2013/08/db2-for-zos-getpages-are-not-always-key.html)
  - Result set size is not the only factor
    - How many pages need to be accessed to determine the result set?
Key Metrics to Monitor (Elapsed Time)

- May want to look at the number of synchronous reads compared to the number of asynchronous reads or GETPAGE requests.
- A high number of synchronous reads may result in DB2 suspensions which may cause extended elapsed times.
- Synchronous writes are also expensive. Often done as a result of DB2 having to externalize changed pages to DASD during a physical close or when changed pages are in the bufferpool after 2 checkpoints.
- Synchronous writes also occur if the Data Manager critical threshold (95%) is reached.
Key Metrics to Monitor (Not Accounted Time)

- The following formula defines DB2® Class 2 Not Accounted Time when no parallelism is involved:
  - DB2 Class 2 Not Accounted Time = DB2 Class 2 Elapsed time - (DB2 Class 2 CPU time + DB2 Class 3 suspension time)
- In production systems, the DB2 Class 2 Not Accounted time is usually very small or negligible. It represents time that DB2 is unable to account for.
  - If you see significant DB2 Class 2 Not Accounted time, it could be the result of many causes.
  - The causes of DB2 Class 2 Not Accounted time will change through future maintenance and DB2 releases as DB2 tries to account for waiting time, without causing performance overhead.
Potential Causes of High Not Accounted Time

- Too much detailed online tracing, or problems with some vendor performance monitors. This situation is usually the primary cause of high not-accounted-for time on systems that are not CPU-constrained.
- Running in a high z/OS paging environment and waiting for storage allocation.
- Running in a very high CPU utilization environment and waiting for CPU cycles.
- A non-dedicated LPAR losing CP when there are multiple LPARs sharing one processor.
- Frequent gathering of dataset statistics (SMF 46 Type 2 records)
- DD consolidation (z/OS parm DDCONS=YES or/and DETAIL) overhead - APAR I107124
- Delays for CF Lock requests when the lock structures use system managed duplexing
Usage of monitoring profiles – monitor by exception

- Monitoring profiles define specific workloads to be monitored
  - May INCLUDE or EXCLUDE only those workloads to be monitored
  - Specify multiple workload definitions within a single profile
  - Workloads may be qualified by SSID, Plan, Program, etc.
  - User specified thresholds determine what level of resource consumption constitutes an exception

- Profiles provide Exception Monitoring and Alerting
  - Information relating to exceptional SQL activity is collected and contains a complete picture of the environment at the time of the event
  - Collected information can be used to trigger Exceptions and Alerts

- Profiles can be changed for a monitoring agent while the agent is still collecting information
Use Exceptions and Alerts

- Exceptions and Alerts are completed SQL statements which exceed specified thresholds
  - Thresholds are specified in the Monitoring Profile
- Information for the Exception or Alert is saved
  - Exceptions are written to VSAM files
  - Exceptions can be offloaded to DB2 tables later
  - Alerts are stored in the CAE server
- Supported thresholds include
  - CPU Time
  - Elapsed Time
  - Getpages
  - SQLCALL count
- Negative SQLCODES are exceptions
  - All or Specific SQLCODES may be excluded
- The monitoring profile specifies whether an Exception also generates an Alert
- This is the way to find that ‘one time’ occurrence
Viewing Exceptions and Alerts in QM

- **View Exceptions in ISPF or Web Client**
  - Single record of all detail about a single SQL Statement
  - Designed to be viewed on demand
  - Ability to drill-down for more detail
  - Saved in VSAM files

- **View Alerts in Web Client**
  - Alerts are viewed from the Message Board
  - Relates Alert messages for Root cause
  - Designed to be responded to with an action
  - Saved on CAE Server
DB2 Query Monitor User Interfaces

- **ISPF**
  - View all collected SQL data from stand-alone DB2 subsystems and DB2 Data Sharing groups
  - View current activity and summarized historical data
  - View exception events
  - Configure Monitoring Profiles

- **Web Browser Client**
  - Full-featured Web Browser Client that provides all ISPF functionality plus additional features
  - View DB2 activity across enterprise from a single browser console
  - Browse Alerts in Message Board
  - Provides real time alerts to exceptional events on monitored DB2 subsystems via automated Response
  - Perform Root Cause Analysis
  - Define Action / Response configuration such as WTO’s, pager notifications, e-mails, and corrective actions
  - View the off-loaded metrics
DB2 QM ISPF UI Only, non-Data Sharing
DB2 QM Architecture
Data Sharing View

ISPF VIEW
- DS GROUP DSGA
- DBA1, DBA2
- DS GROUP DSGB
- DBB1, DBB2

offloaded performance history DB

CAE AGENT
- started task

CQM Collector
- started task

z/OS LPAR1

Monitored DB2
- DS member DBA2
- DS Group DSGA

Monitored DB2
- DS member DBB2
- DS Group DSGB

Interval VSAM

Offload Utility

z/OS LPAR2

Monitored DB2
- DS member DBA1
- DS Group DSGA

Monitored DB2
- DS member DBB1
- DS Group DSGB

Interval VSAM

WEB BROWSER VIEW
- DS GROUP DSGA
- DBA1, DBA2
- DS GROUP DSGB
- DBB1, DBB2
- HISTORY DB

CAE SERVER
- WINDOWS or USS

CAE AGENT
- started task

CQM Collector
- started task

SMA

HISTORY DB
## QM System Selection

<table>
<thead>
<tr>
<th>Option</th>
<th>====&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>C:S-Select</td>
<td></td>
</tr>
</tbody>
</table>

### QM Subsystems

<table>
<thead>
<tr>
<th>CMD</th>
<th>QM</th>
<th>VER</th>
<th>DB2</th>
<th>VER</th>
<th>DS GROUP</th>
<th>MSTATUS</th>
<th>CURRENT INTERVAL STRT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>QMR1</td>
<td>310</td>
<td>I9A2</td>
<td>V09R1M0</td>
<td>IDS2</td>
<td>ACTIVE</td>
<td>06/06/2011 - 23:59:58</td>
</tr>
</tbody>
</table>

### Data Sharing

<table>
<thead>
<tr>
<th>DS GROUP</th>
<th>VER</th>
<th>DB2</th>
<th>VER</th>
<th>MSTATUS</th>
<th>CURRENT INTERVAL STRT</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDS2</td>
<td>310</td>
<td>I9A2</td>
<td>V09R1M0</td>
<td>ACTIVE</td>
<td>06/07/2011 - 3:59:58</td>
</tr>
<tr>
<td>310 I9B2</td>
<td>V09R1M0</td>
<td>ACTIVE</td>
<td>06/07/2011 - 3:59:58</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Plan List

<table>
<thead>
<tr>
<th>CQMP</th>
<th>Elapsed</th>
<th>%Elapsed</th>
<th>Avg Elapsed</th>
<th>CPU</th>
<th>%CPU</th>
<th>Avg CPU</th>
<th>&lt;%CPU&gt;</th>
<th>Delay</th>
<th>%Delay</th>
<th>Avg Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>CQMP002</td>
<td>0.087397</td>
<td>0.54</td>
<td>0.038447</td>
<td>0.003083</td>
<td>0.00</td>
<td>0.000016</td>
<td>0.38</td>
<td>0.035967</td>
<td>0.22</td>
<td>0.017427</td>
</tr>
<tr>
<td>CQMP002A</td>
<td>0.000881</td>
<td>0.00</td>
<td>0.00004</td>
<td>0.000680</td>
<td>0.00</td>
<td>0.000031</td>
<td>74.91</td>
<td>0.000000</td>
<td>0.00</td>
<td>0.000000</td>
</tr>
<tr>
<td>CQMP002B</td>
<td>0.000716</td>
<td>0.00</td>
<td>0.000034</td>
<td>0.000592</td>
<td>0.00</td>
<td>0.000028</td>
<td>92.74</td>
<td>0.000000</td>
<td>0.00</td>
<td>0.000000</td>
</tr>
<tr>
<td>CQMP002C</td>
<td>0.000006</td>
<td>0.00</td>
<td>0.000000</td>
<td>0.000020</td>
<td>0.00</td>
<td>0.000000</td>
<td>77.25</td>
<td>0.000000</td>
<td>0.00</td>
<td>0.000000</td>
</tr>
<tr>
<td>CQMP002D</td>
<td>0.000070</td>
<td>0.00</td>
<td>0.000003</td>
<td>0.000569</td>
<td>0.00</td>
<td>0.000000</td>
<td>83.20</td>
<td>0.000000</td>
<td>0.00</td>
<td>0.000000</td>
</tr>
<tr>
<td>CQMP002E</td>
<td>0.0000628</td>
<td>0.00</td>
<td>0.000003</td>
<td>0.000568</td>
<td>0.00</td>
<td>0.000027</td>
<td>90.50</td>
<td>0.000000</td>
<td>0.00</td>
<td>0.000000</td>
</tr>
<tr>
<td>CQMP002F</td>
<td>0.0000719</td>
<td>0.00</td>
<td>0.000003</td>
<td>0.000575</td>
<td>0.00</td>
<td>0.000027</td>
<td>75.98</td>
<td>0.000000</td>
<td>0.00</td>
<td>0.000000</td>
</tr>
<tr>
<td>CQMP002G</td>
<td>0.0000703</td>
<td>0.00</td>
<td>0.000003</td>
<td>0.000552</td>
<td>0.00</td>
<td>0.000026</td>
<td>78.58</td>
<td>0.000000</td>
<td>0.00</td>
<td>0.000000</td>
</tr>
<tr>
<td>CQMP002H</td>
<td>0.0000665</td>
<td>0.00</td>
<td>0.000003</td>
<td>0.000549</td>
<td>0.00</td>
<td>0.000025</td>
<td>90.17</td>
<td>0.000000</td>
<td>0.00</td>
<td>0.000000</td>
</tr>
<tr>
<td>CQMP002I</td>
<td>0.000049</td>
<td>0.00</td>
<td>0.000000</td>
<td>0.000531</td>
<td>0.00</td>
<td>0.000020</td>
<td>65.66</td>
<td>0.000000</td>
<td>0.00</td>
<td>0.000000</td>
</tr>
<tr>
<td>CQMP002J</td>
<td>0.0000504</td>
<td>0.00</td>
<td>0.000000</td>
<td>0.000566</td>
<td>0.00</td>
<td>0.000026</td>
<td>85.29</td>
<td>0.000000</td>
<td>0.00</td>
<td>0.000000</td>
</tr>
<tr>
<td>CQMP002K</td>
<td>0.0000712</td>
<td>0.00</td>
<td>0.000003</td>
<td>0.000621</td>
<td>0.00</td>
<td>0.000025</td>
<td>87.41</td>
<td>0.000000</td>
<td>0.00</td>
<td>0.000000</td>
</tr>
<tr>
<td>CQMP002L</td>
<td>0.0000732</td>
<td>0.00</td>
<td>0.000003</td>
<td>0.000611</td>
<td>0.00</td>
<td>0.000029</td>
<td>85.48</td>
<td>0.000000</td>
<td>0.00</td>
<td>0.000000</td>
</tr>
<tr>
<td>CQMP002M</td>
<td>0.0000789</td>
<td>0.00</td>
<td>0.000003</td>
<td>0.000089</td>
<td>0.00</td>
<td>0.000028</td>
<td>76.19</td>
<td>0.000000</td>
<td>0.00</td>
<td>0.000000</td>
</tr>
<tr>
<td>CQMP002N</td>
<td>0.0000703</td>
<td>0.00</td>
<td>0.000003</td>
<td>0.000612</td>
<td>0.00</td>
<td>0.000029</td>
<td>85.04</td>
<td>0.000000</td>
<td>0.00</td>
<td>0.000000</td>
</tr>
<tr>
<td>CQMP002O</td>
<td>0.210612</td>
<td>0.02</td>
<td>0.002240</td>
<td>0.027909</td>
<td>0.02</td>
<td>0.000051</td>
<td>2.40</td>
<td>0.50990</td>
<td>0.11</td>
<td>0.000147</td>
</tr>
<tr>
<td>DNS001</td>
<td>0.0070975</td>
<td>0.02</td>
<td>0.004356</td>
<td>0.004975</td>
<td>0.00</td>
<td>0.000036</td>
<td>45.24</td>
<td>0.052857</td>
<td>0.42</td>
<td>0.070849</td>
</tr>
</tbody>
</table>
Details and Pie Chart for Selected Row
Synergy from Historical PDB (Performance Data Base) by OMPE and QM
QM PDB

QM VSAM Backstore per Interval

Optional Offload of selected VSAM Backstore

QM SQL Archive DB on Interval aggregated level

- Identifiers for correlation of data
- Aggregation level, valid time range – Interval consideration

QM PDB

PDB on ”Interval“ aggregated Plan/Pkg level

PDB with Plan/pkg raw data

CQM23_SUMM_METRICS
CQM23_SUMM_OBJECTS
CQM23_SUMM_TEXT
CQM23_INTERVALS
CQM23_STMT_TYPES
CQM21_EXCEPTIONS
CQM23_EXCP_CALLS
CQM23_EXCP_HOSTV
CQM23_EXCP_OBJS
CQM23_EXCP_TEXT
CQM23_DB2_COMMANDS
CQM23_SQLCODES
CQM23_SQLCODE_DET
CQM23_SQLCODE_TEXT

METRDATA - summary level information related to SQL call execut.
OBSDATA - summary object level data
TEXTDATA - summary level SQL text data
INTERVALS – QM interval information
SQL statement type and description
EXCPINDX/ EXCPDATA – except. data SQL calls, txt, SQLCA, host vars
EXCPINDX/ EXCPDATA – except. data SQL calls information
EXCPINDX/ EXCPDATA – except. host variables information
EXCPINDX/ EXCPDATA – except. objects information
EXCPINDX/ EXCPDATA – except. text information
DB2CMDS – DB2 Commands
SQLCDATA - negative SQLCODE information
SQLCDATA - details about negative SQLCODE
SQLCDATA - text for negative SQLCODEs
OMPE Accounting PDB key fields – OMPE PDB

### SAVE data aggregated by „concatenated key“ and „interval“

```sql
CREATE TABLE DB2PMSACCT_GENERAL
(DB2PM_REL SMALLINT NOT NULL WITH DEFAULT,
 DB2_REL CHAR(2) NOT NULL WITH DEFAULT,
 LOCAL_LOCATION VARCHAR(128) NOT NULL WITH DEFAULT,
 GROUP_NAME CHAR(8) NOT NULL WITH DEFAULT,
 SUBSYSTEM_ID CHAR(4) NOT NULL WITH DEFAULT,
 MEMBER_NAME CHAR(8) NOT NULL WITH DEFAULT,
 REQ_LOCATION VARCHAR(128) NOT NULL WITH DEFAULT,
 CONNECT_TYPE CHAR(8) NOT NULL WITH DEFAULT,
 CONNECT_ID CHAR(8) NOT NULL WITH DEFAULT,
 CORRNAME CHAR(8) NOT NULL WITH DEFAULT,
 CORRNUMBER CHAR(8) NOT NULL WITH DEFAULT,
 PLAN_NAME CHAR(8) NOT NULL WITH DEFAULT,
 INTERVAL_TIME TIMESTAMP NOT NULL WITH DEFAULT,
 MAINPACK VARCHAR(128) NOT NULL WITH DEFAULT,
 THREAD_TYPE CHAR(8) NOT NULL WITH DEFAULT,
 CLIENT_ENDUSER VARCHAR(128) NOT NULL WITH DEFAULT,
 CLIENT_WSNNAME VARCHAR(255) NOT NULL WITH DEFAULT,
 CLIENT_TRANSACTION VARCHAR(255) NOT NULL WITH DEFAULT,
...
```

CORRNAME and CORRNUMBER (each 8 Bytes) is derived from the original 12 Byte DB2 CORRID

### Accounting raw data (FILE)

```sql
CREATE TABLE DB2PMFACCT_GENERAL
(DB2PM_REL SMALLINT NOT NULL WITH DEFAULT,
 DB2_REL CHAR(2) NOT NULL WITH DEFAULT,
 LOCAL_LOCATION VARCHAR(128) NOT NULL WITH DEFAULT,
 GROUP_NAME CHAR(8) NOT NULL WITH DEFAULT,
 SUBSYSTEM_ID CHAR(4) NOT NULL WITH DEFAULT,
 MEMBER_NAME CHAR(8) NOT NULL WITH DEFAULT,
 NET_ID CHAR(8) NOT NULL WITH DEFAULT,
 LUNAME CHAR(8) NOT NULL WITH DEFAULT,
 INSTANCE_NBR CHAR(12) NOT NULL WITH DEFAULT,
 LUW_SEQNO SMALLINT NOT NULL WITH DEFAULT,
 REQ_LOCATION VARCHAR(128) NOT NULL,
 REQUEST_PRODUCT_ID CHAR(8) NOT NULL WITH DEFAULT,
 CONNECT_TYPE CHAR(8) NOT NULL WITH DEFAULT,
 CONNECT_ID CHAR(8) NOT NULL WITH DEFAULT,
 CORRNAME CHAR(8) NOT NULL WITH DEFAULT,
 CORRNUMBER CHAR(8) NOT NULL WITH DEFAULT,
 MAINPACK VARCHAR(128) NOT NULL WITH DEFAULT,
 CICS_NET_ID CHAR(8) NOT NULL WITH DEFAULT,
 CICS_LUNAME CHAR(8) NOT NULL WITH DEFAULT,
 CICS_INSTANCE_NBR CHAR(12) NOT NULL WITH DEFAULT,
 WLM_SERVICE_CLASS CHAR(8) NOT NULL WITH DEFAULT,
 CLIENT_ENDUSER VARCHAR(128) NOT NULL WITH DEFAULT,
 CLIENT_WSNNAME VARCHAR(255) NOT NULL WITH DEFAULT,
 CLIENT_TRANSACTION VARCHAR(255) NOT NULL WITH DEFAULT,
 CLASS1_TIME_BEG TIMESTAMP NOT NULL,
 CLASS1_TIME_END TIMESTAMP NOT NULL,
...
```
### QM DB2 tables

#### CREATE TABLE SYSTOOLS.CQM32_INTERVALS

<table>
<thead>
<tr>
<th>Column</th>
<th>Data Type</th>
<th>Nullability</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMFID</td>
<td>CHAR(04)</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>CQM_SUBSYSTEM</td>
<td>CHAR(04)</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>INTERVAL_NUMBER</td>
<td>INTEGER</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>INTERVAL_START</td>
<td>TIMESTAMP</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>DB2_SUBSYSTEM</td>
<td>CHAR(04)</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>DB2_VERSION</td>
<td>CHAR(03)</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>SMFID</td>
<td>CHAR(04)</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>CQM_SUBSYSTEM</td>
<td>CHAR(04)</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>INTERVAL_NUMBER</td>
<td>INTEGER</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>INTERVAL_START</td>
<td>TIMESTAMP</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>DB2_SUBSYSTEM</td>
<td>CHAR(04)</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>DB2_VERSION</td>
<td>CHAR(03)</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>METRICS_TOKEN</td>
<td>CHAR(08)</td>
<td>NOT NULL FOR BIT DATA</td>
</tr>
<tr>
<td>METRICS_TIMESTAMP</td>
<td>TIMESTAMP</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>TEXT_TOKEN</td>
<td>CHAR(10)</td>
<td>NOT NULL FOR BIT DATA</td>
</tr>
<tr>
<td>CONSISTENCY_TOKEN</td>
<td>CHAR(08)</td>
<td>NOT NULL FOR BIT DATA</td>
</tr>
<tr>
<td>PLAN</td>
<td>CHAR(08)</td>
<td>NOT NULL FOR BIT DATA</td>
</tr>
<tr>
<td>COLLECTION</td>
<td>VARCHAR(128)</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>PROGRAM</td>
<td>VARCHAR(128)</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>SECTION</td>
<td>SMALLINT</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>STMT</td>
<td>INTEGER</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>TYPE</td>
<td>SMALLINT</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>CORRID</td>
<td>CHAR(12)</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>AUTHID</td>
<td>CHAR(08)</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>WORKSTATION_USER</td>
<td>CHAR(16)</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>WORKSTATION_TRAN</td>
<td>CHAR(32)</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>WORKSTATION_NAME</td>
<td>CHAR(18)</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>INTERVAL_END</td>
<td>TIMESTAMP</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>ZIIP_CPU_TIME</td>
<td>DECIMAL(15,6)</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>DB2_GROUP_NAME</td>
<td>CHAR(08)</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>STATEMENT_COUNT</td>
<td>INTEGER</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>EXECUTION_COUNT</td>
<td>INTEGER</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>CURRENT_SCHEMA</td>
<td>VARCHAR(128)</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>ACCELERATOR</td>
<td>CHAR(08)</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
</tbody>
</table>

IN SYSTOOLS.CQM32INT;

#### CREATE TABLE SYSTOOLS.CQM32_SUMM_METRICS

<table>
<thead>
<tr>
<th>Column</th>
<th>Data Type</th>
<th>Nullability</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMFID</td>
<td>CHAR(04)</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>CQM_SUBSYSTEM</td>
<td>CHAR(04)</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>INTERVAL_NUMBER</td>
<td>INTEGER</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>INTERVAL_START</td>
<td>TIMESTAMP</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>DB2_SUBSYSTEM</td>
<td>CHAR(04)</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>DB2_VERSION</td>
<td>CHAR(03)</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>METRICS_TOKEN</td>
<td>CHAR(08)</td>
<td>NOT NULL FOR BIT DATA</td>
</tr>
<tr>
<td>METRICS_TIMESTAMP</td>
<td>TIMESTAMP</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>TEXT_TOKEN</td>
<td>CHAR(10)</td>
<td>NOT NULL FOR BIT DATA</td>
</tr>
<tr>
<td>CONSISTENCY_TOKEN</td>
<td>CHAR(08)</td>
<td>NOT NULL FOR BIT DATA</td>
</tr>
<tr>
<td>PLAN</td>
<td>CHAR(08)</td>
<td>NOT NULL FOR BIT DATA</td>
</tr>
<tr>
<td>COLLECTION</td>
<td>VARCHAR(128)</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>PROGRAM</td>
<td>VARCHAR(128)</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>SECTION</td>
<td>SMALLINT</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>STMT</td>
<td>INTEGER</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>TYPE</td>
<td>SMALLINT</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>CORRID</td>
<td>CHAR(12)</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>AUTHID</td>
<td>CHAR(08)</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>WORKSTATION_USER</td>
<td>CHAR(16)</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>WORKSTATION_TRAN</td>
<td>CHAR(32)</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>WORKSTATION_NAME</td>
<td>CHAR(18)</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>INTERVAL_END</td>
<td>TIMESTAMP</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>ZIIP_CPU_TIME</td>
<td>DECIMAL(15,6)</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>DB2_GROUP_NAME</td>
<td>CHAR(08)</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>STATEMENT_COUNT</td>
<td>INTEGER</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>EXECUTION_COUNT</td>
<td>INTEGER</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>CURRENT_SCHEMA</td>
<td>VARCHAR(128)</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>ACCELERATOR</td>
<td>CHAR(08)</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
</tbody>
</table>

IN SYSTOOLS.CQM32MET;

#### CREATE TABLE SYSTOOLS.CQM32_SUMM_OBJECTS

<table>
<thead>
<tr>
<th>Column</th>
<th>Data Type</th>
<th>Nullability</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMFID</td>
<td>CHAR(04)</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>CQM_SUBSYSTEM</td>
<td>CHAR(04)</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>INTERVAL_NUMBER</td>
<td>INTEGER</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>INTERVAL_START</td>
<td>TIMESTAMP</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>METRICS_TOKEN</td>
<td>CHAR(08)</td>
<td>NOT NULL FOR BIT DATA</td>
</tr>
<tr>
<td>METRICS_TIMESTAMP</td>
<td>TIMESTAMP</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>DBID</td>
<td>SMALLINT</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>OBID</td>
<td>SMALLINT</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>PSID</td>
<td>SMALLINT</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>BUFFERPOOL_NORM</td>
<td>CHAR(06)</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>BUFFERPOOL_NUM</td>
<td>SMALLINT</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>INTERVAL_END</td>
<td>TIMESTAMP</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>DB2_SUBSYSTEM</td>
<td>CHAR(4)</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>DB2_VERSION</td>
<td>CHAR(3)</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
<tr>
<td>DB2_GROUP_NAME</td>
<td>CHAR(08)</td>
<td>NOT NULL WITH DEFAULT</td>
</tr>
</tbody>
</table>

IN SYSTOOLS.CQM32OBJ;

#### CREATE TABLE SYSTOOLS.CQM32_SUMM_TEXT

<table>
<thead>
<tr>
<th>Column</th>
<th>Data Type</th>
<th>Nullability</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMFID</td>
<td>CHAR(04)</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>CQM_SUBSYSTEM</td>
<td>CHAR(04)</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>INTERVAL_NUMBER</td>
<td>INTEGER</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>INTERVAL_START</td>
<td>TIMESTAMP</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>INTERVAL_END</td>
<td>TIMESTAMP</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>TEXT_TOKEN</td>
<td>CHAR(10)</td>
<td>NOT NULL FOR BIT DATA</td>
</tr>
<tr>
<td>TEXT_TIMESTAMP</td>
<td>TIMESTAMP</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>CCSID</td>
<td>INTEGER</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>ROW_ID</td>
<td>ROWID</td>
<td>GENERATED ALWAYS NOT NULL</td>
</tr>
<tr>
<td>SQLTEXT</td>
<td>CLOB</td>
<td>NOT NULL</td>
</tr>
</tbody>
</table>

IN SYSTOOLS.CQM32MT1;
Correlation of aggregated data between QM and OMPE

Thread level aggregated reporting

- Blue threads reported
- Red threads reported
- Green threads reported

SQL level aggregated reporting

- How to correlate/associate the executed SQL to threads or package or vice versa.
  - Shorter threads are running and longer “intervals” are used in QM
  - Aggregation map
  - the correlation between aggregated SQL and aggregated thread level.
  - Synchronization of start/end time with QM, OMPE, and perhaps RMF

- How to correlate the QM “SQL Exceptions” with the single OMPE thread execution.
Integrated Data Management combined with Integrated System

Information Management Tools

OMEGAMON XE for Mainframe Networks / for zNetview
OMEGAMON XE for CICS
OMEGAMON XE for IMS
OMEGAMON XE for DB2 PE
DB2 Query Monitor
pureQuery
Optim Query (Workload) Tuner
Data Studio
DB2 Path Checker
DB2 SQL PA
Feature Manager
Extended Insight
Optim Performance
ITCAM
OMEGAMON XE for z/OS

=> DWL or additionally using OMEGAMON DE (Dashboard Edition)
www.ibm.com/software/data/tools