

DB2 Performance – A Primer

Bill Arledge – Principal Consultant

CA Technologies

Sept 14th, 2011


agility
made possible™



Agenda

- Performance Defined
- DB2 Instrumentation – Sources of performance metrics
- DB2 Performance Disciplines
 - System performance focus
 - Application performance focus
 - Detailed SQL analysis focus
- DB2 Performance Management Tools – just a discussion


What is Performance? In the General Sense

per·for·mance  *noun*

a : the ability to perform : EFFICIENCY
b : the manner in which a mechanism performs <engine performance>

per·form  *verb* \pə(r)-'fɔrm\
transitive verb

- 1 : to adhere to the terms of : FULFILL <perform a contract>
- 2 : CARRY OUT, DO
- 3 **a** : to do in a formal manner or according to prescribed ritual
b : to give a rendition of : PRESENT

ef·fi·cien·cy  *noun*

- 1 : the quality or degree of being efficient
- 2 **a** : efficient operation
b (1) : effective operation as measured by a comparison of production with cost (as in energy, time, and money) (2) : the ratio of the useful energy delivered by a dynamic system to the energy supplied to it

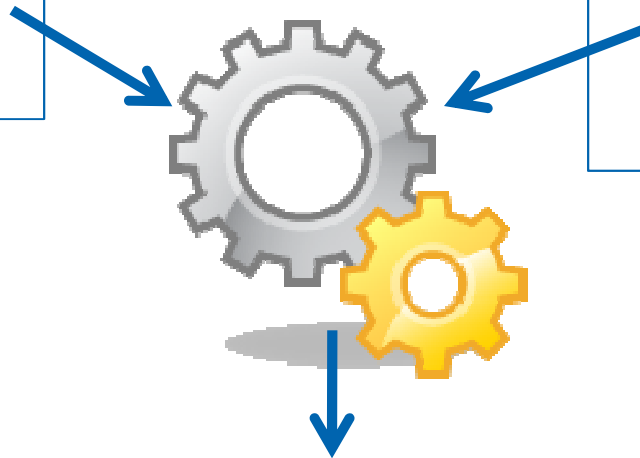
What is Performance? In the Mainframe Computing Sense

For a desired workload
measured by:

- Transactions/second
- SQL statements/second
- Commits/second
- Batch work per hour

With given resources
measured by :

- CPU capacity in MIPS,
MSUs or other measure
- Storage
- I/O subsystem
- Network bandwidth



Can we meet our objectives for:

- Response Time
- Transaction Workload
- Resource Utilization (CPU,I/O,etc.)

Performance Contracts

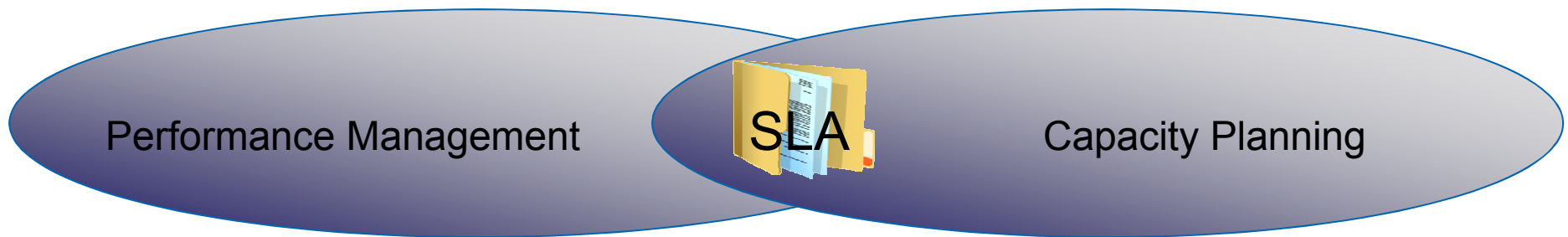
Setting Objectives

Service Level Agreements

- Binding formal agreement between the IT organization and a business unit
 - Often focused on a specific business critical application
- Documents expectations for application performance and availability
 - Average transaction response time given a maximum transaction rate;
 - Application availability for both online and batch processing
- Translate into the IT service level objectives and make performance management more effective

Managing to Meet Service Level Objectives

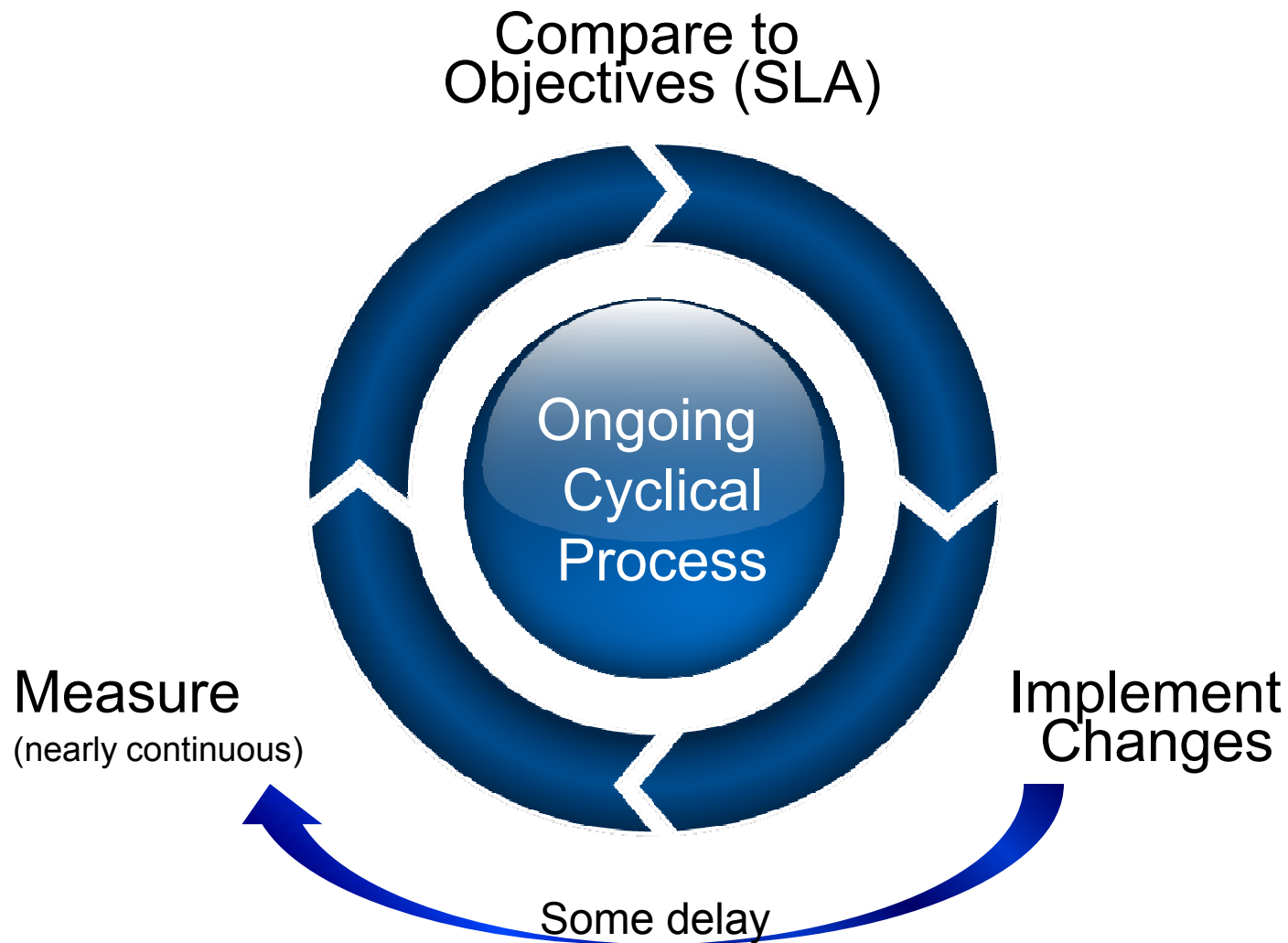
CPM (Capacity and Performance Management)



- Goal: Ensure IT systems can meet performance goals defined in the SLA
- Focus on current performance
- SLAs make performance management more effective

- Goal: Ensure adequate resources available to support critical workloads
- Focus on the future
- SLA determines application priority and importance

Performance Management Never Ending Task



Performance Management

What are We Measuring

Response Times

- Averages and high water marks defined generally as transaction start to transaction stop (end-to-end)
- Best way to describe transaction performance
- Include batch process end –to-end

Transaction Throughput Rates

- Measure of completed transactions for an elapsed time
- Also known as transaction rate

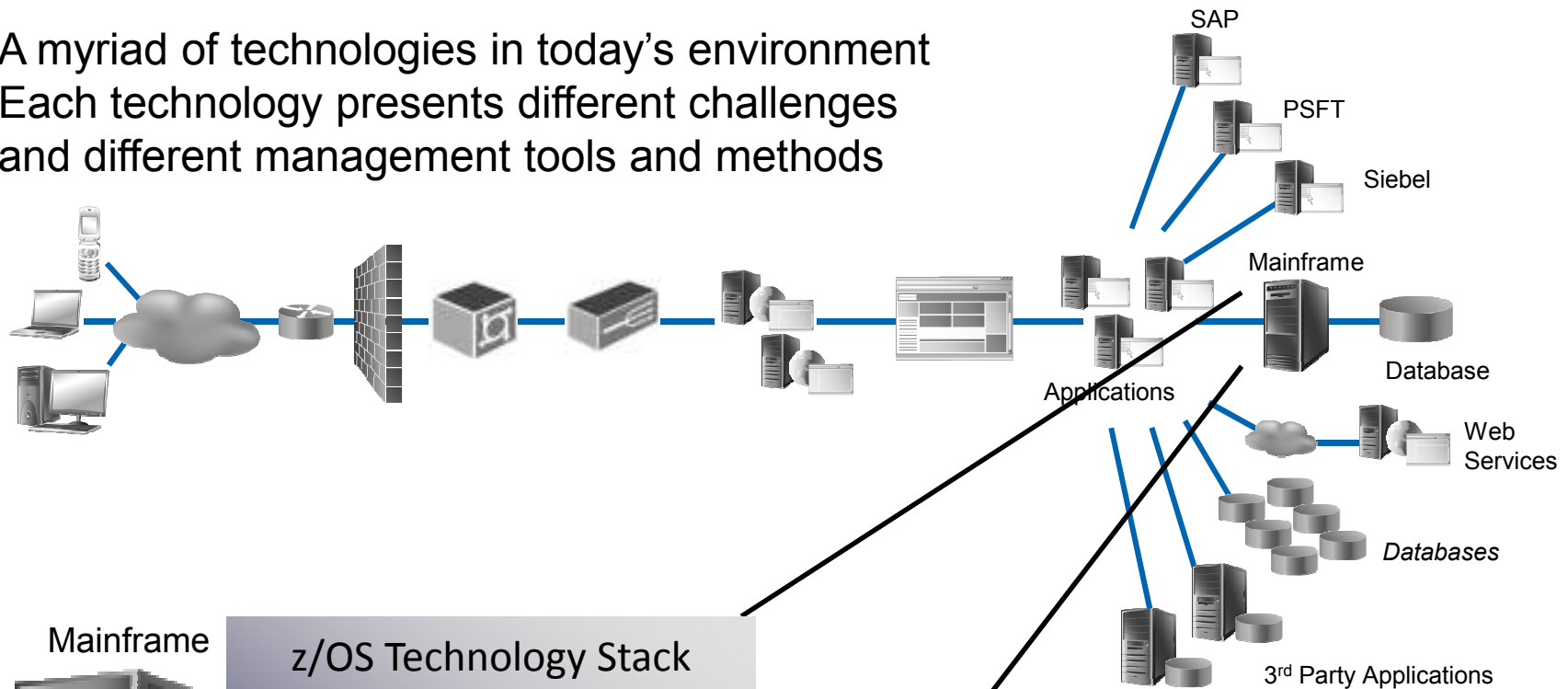
Resource Utilization

- CPU consumption
- I/O subsystem metrics
- Storage usage

Performance Management

End-to-End Performance Management

- A myriad of technologies in today's environment
- Each technology presents different challenges and different management tools and methods



Mainframe

z/OS Technology Stack

C	I	D	T	W
I	M	B	C	A
C	S	2	P	S
S			P	

- Mainframe has numerous technologies
 - Each with different metrics/tuning knobs
- Common technology for collecting and externalizing performance metrics

Performance Management

Collecting and Externalizing Performance Metrics

“If you can’t measure it, you can’t manage it”

Resource Measurement Facility (RMF)

- Provides performance and usage instrumentation of resources like processor, memory, disk, cache, workload, virtual storage, XCF and Coupling Facility

Systems Management Facility (SMF)

- Provides a standardized method for externalizing activity to datasets, including I/O, network activity, processor utilization, software usage, etc.
- RMF uses SMF to record data
- Basis for monitoring and automation utilities on z/OS
- Common usage for billing users

Performance Measurements

More on SMF

Numerous technologies write SMF data

- IBM z/OS internal components and program products
- Third-party vendor products

Products & Usage (some examples)

- Driven by product options
- Can be written to:
 - SMF datasets
 - Log streams
- Other product may externalize data using other options
 - IMS uses it's own log for example
- SMF data can be post processed by home-grown solutions

z/OS activities	SMF Record Types
DB2	100, 101, 102
CICS	110
WAS MQ	115, 116
WAS	120
RMF	70 - 79
RACF	80
Common Address Space Work	30
Workload by service class	72

Expanding on the Tuning Option

DB2 Performance Tuning Overview

DB2 Performance Management

Collecting Performance Data

Instrumentation Facility Component

- Records DB2 data and events (in IFCIDs)
 - IFCIDs are documented in *hlq.SDSNIVPD* at the very end of the DSNWMSGs member
 - Location changed in DB2 10
- Captured by trace facility with numerous types and options
- All traces (except performance) can be started by `-START TRACE` command or DSNZPARM setting

Trace Type	Trace Classes	IFCIDs	SMF Record Type
Global	1 - 11	IBM Support – 78 IFCIDs	N/A
Statistics	1 - 7	How systems and database services are used– 38 IFCIDs	100
Accounting	1-5,7-8,10	Application thread information – 73 IFCIDs	101
Audit	1 -11	DB2 security events – 21 IFCIDs	102
Monitor	1 – 10, 29	Online monitoring - 82 IFCIDs	N/A
Performance	1 – 22,23 – 29 (reserved)	Detailed event traces(222 IFCIDs)	102

DB2 Performance Management

Collecting Performance Data

- DB2 Trace Overhead
 - Statistics 2% per transaction/thread
 - Accounting 3 – 5 % per thread (can be higher depending on specific IFCIDs requested); can vary based amount of SQL processed
 - Monitor 2 – 5 %; can be higher if other trace classes are started (29 for example)
 - Audit 5% or less generally; increases for transactions that access audited tables
 - Performance 20 - 100% for each activity;
 - Global Can be as high as 100%
- -DISPLAY TRACE command output

```
DSNW127I  !S10A CURRENT TRACE ACTIVITY IS -
TNO TYPE   CLASS          DEST QUAL IFCID
01  STAT   01,03,04,05, SMF  NO
01          06
02  ACCTG  01,02           SMF  NO
03  MON    29            OP1  NO   003,023,024,025,044,045,090,091,095
03          096,105,107,172,196,197,199,203,204
03          205,206,207,208,209,210,221,224,235
03          236,239,250,258,313,318,335,337,343
04  MON    01,02,03,05, OP1  NO
04          07,08,09,10
05  MON    29            OP1  YES  006,007,008,009,020,044,045,073,095
05          096,177
06  AUDIT  01,02,03,04, SMF  NO
06          05,06,07,08
*****END OF DISPLAY TRACE SUMMARY DATA*****
DSN9022I  !S10A DSNWVCM1 '-DISPLAY TRACE' NORMAL COMPLETION
***
```

DB2 Performance Management

Best Practices for Managing Traces

Qualify all traces if possible

- AUTHID, Correlation ID, connection name
- End-user workstation ID, application, or transaction name; location
- Package, collection, or program

Enable minimal trace and audit classes you need at startup

- When specific problems occur you can start detailed traces to collect additional data

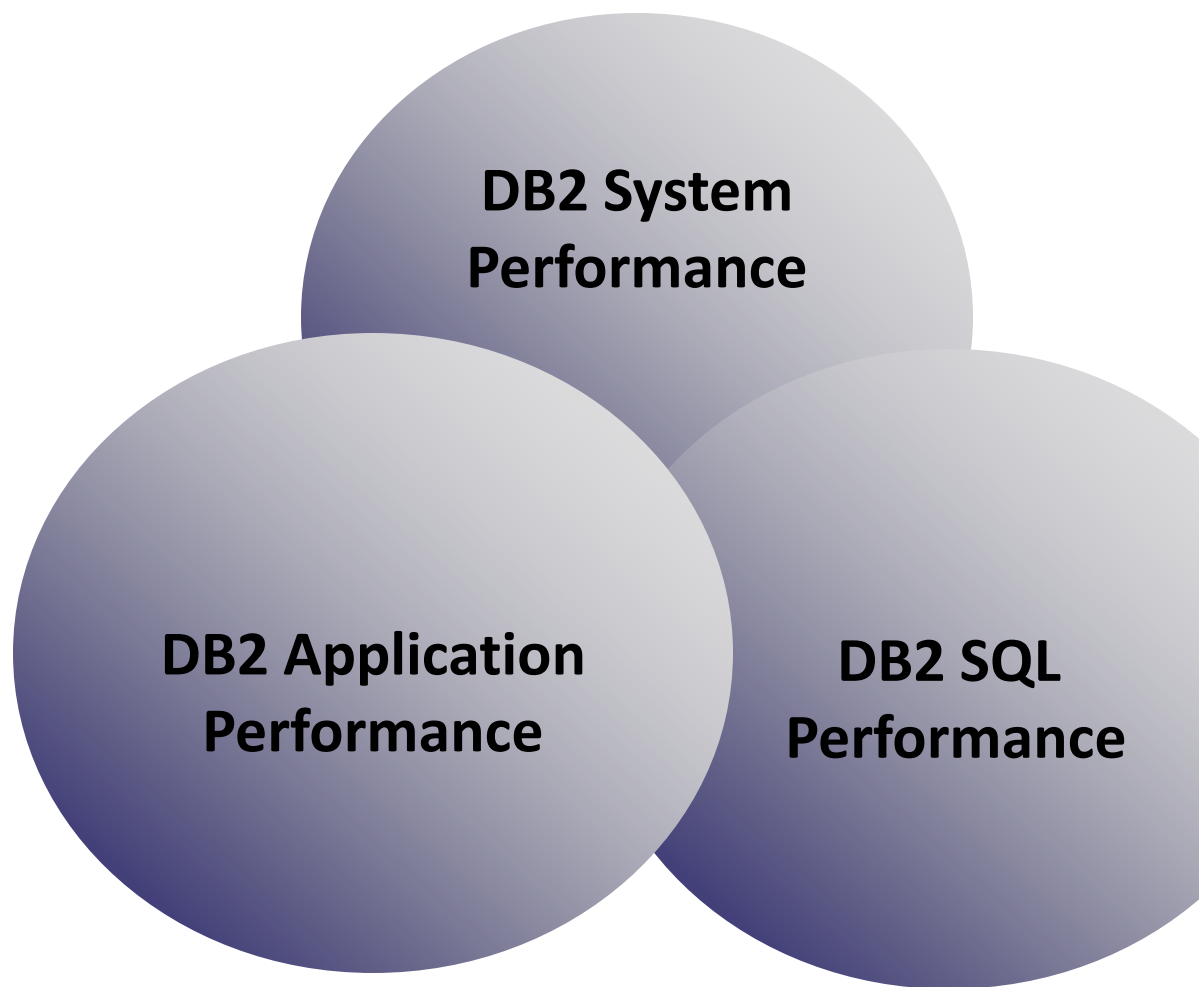
Start the global trace only if requested by IBM to work a problem

Use performance trace for short time periods and restrict usage to specific users or applications

With DB2 10 STATIME parameter no longer controls statistics IFCIDs 0001, 0002, 0202, 0217, and 0230 (these are now cut every minute)

- STATIME only impact IFCIDs 0105,0106, 0199, and 0365

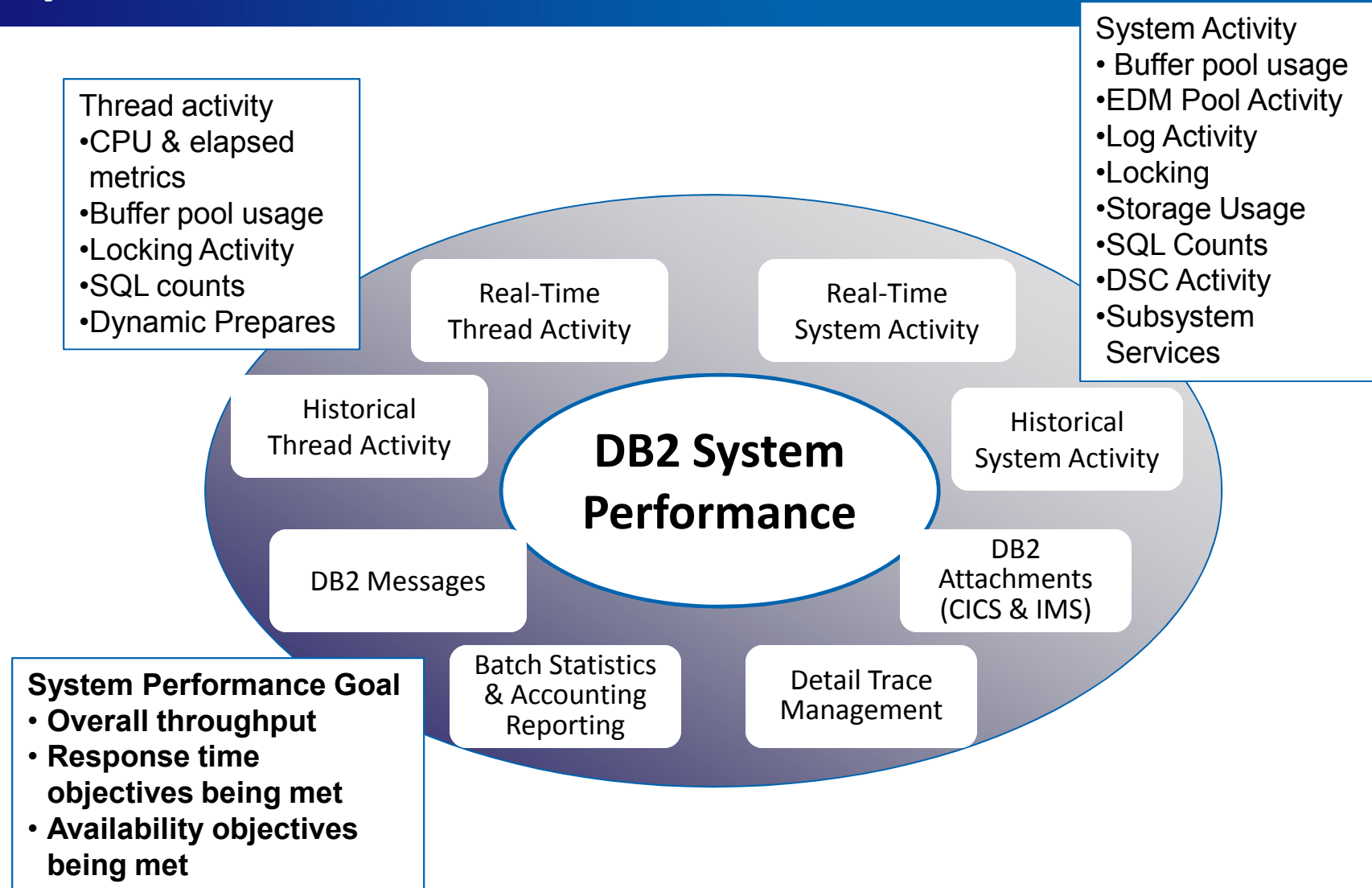
DB2 Performance Management Disciplines



- Performance foci
 - Largely driven by role in the organization
 - Application DBAs focus on application performance & SQL
 - Production DBAs, z/OS performance specialists focus on system performance
 - Focus on SQL tuning across the application lifecycle
 - No hard and fast rules as roles are often hybrid

DB2 Performance Management Disciplines

System Performance



DB2 Performance Management Disciplines

System Performance Data Collection

Monitor traces provide data for real-time statistics and accounting data

Monitor traces can be started for specific IFCIDs or by class

- If started by class a fixed set of IFCIDs are collection

MON	29	OP1	NO	003, 023, 024, 025, 044, 045, 090, 091, 095 096, 105, 107, 172, 196, 197, 199, 203, 204 205, 206, 207, 208, 209, 210, 221, 224, 235 236, 239, 250, 258, 313, 318, 335, 337, 343
MON	01, 02, 03, 05, 07, 08, 09, 10	OP1	NO	
MON	29	OP1	YES	006, 007, 008, 009, 020, 044, 045, 073, 095 096, 177

Some IFCID Examples by IFCID

00044 - LOCK SUSPEND OR IDENTIFY CALL IRLM

00197 - DB2 MESSAGE MONITORING

Some Monitor Class Examples

Monitor Class 03 – DB2 wait time for I/O, locks, resource usage information

Monitor Class 07 – Entry or exit from DB2 event signaling for package and DBRM accounting

DB2 Performance Management

How is Collected Performance Data Captured?

Monitor programs take advantage of the IFI (Instrumentation Facility Interface) using the following functions:

- READS to retrieve trace records synchronously and cause them to be returned immediately
- READA to retrieve trace records asynchronously; trace records accumulate and are then moved to the requesting monitor program
- COMMAND allows submission of DB2 commands
- WRITE can deliver information to a DB2 trace definition

Monitor programs can also collect data by reading control blocks in DB2 address spaces

- These programs always start some number of monitor traces to ensure that required control blocks are maintained by DB2
- Potential for better performance by reading control blocks directly rather than using the IFI

DB2 Performance Management

System Performance - Important Metrics

Workload Metrics

- # of Commits/Aborts – a good indicator of overall workload DB2 workload for the given interval; High number of aborts should be investigated
 - Main goal here is consistency – numbers should be consistent over time
- GETPAGE requests – also a good indicator of overall workload for this interval

Problems to watch for (not nearly a complete list)

- Thread issues – Watch for CREATE THREADs that are queued; may indicate need for larger CTHREAD or MAXDBAT values. DB2 10 dramatically increases the number of concurrent threads (2,000 -> 20,000)
- Buffer pool problems – Watch for low hit ratios and critical thresholds reached. May indicate need to increase buffer pool size or reallocate page sets to other pools.
- EDM Pool Problems – DB2 8 still has significant constraint for thread storage below the 2G line. DB2 9 reduced that and DB2 10 nearly eliminates. DBD, CT, and PT loads from disk are indicators your EDM pool may not be large enough. You might need to increase EDMPOOL size if possible.
- Locking problems – Lock suspensions and timeouts can be expected in a busy system. Deadlocks are generally issues that should be explored in more detail.
- Logging issues – Checkpoint frequency can create issues with logging. If you checkpoint too frequently you'll incur overhead, not frequently enough you impact restart times.

DB2 Performance Management

System Performance - Important Metrics

More problems to watch for (not nearly a complete list)

- Dynamic Statement Cache problems – Situations where SQL statements already cached are not being reused can have a significant hit on performance. Optimally statements found in cache and statements not found should be low. This can be highly dependent on application implementation techniques.

Need for Automated Exception Management

- As mentioned, the previous list is just a small sample of the system problems that can be encountered
- Automated exception detection and management is critical because you can't see everything.

DB2 Performance Management Disciplines

Application Performance

Application Performance Goals

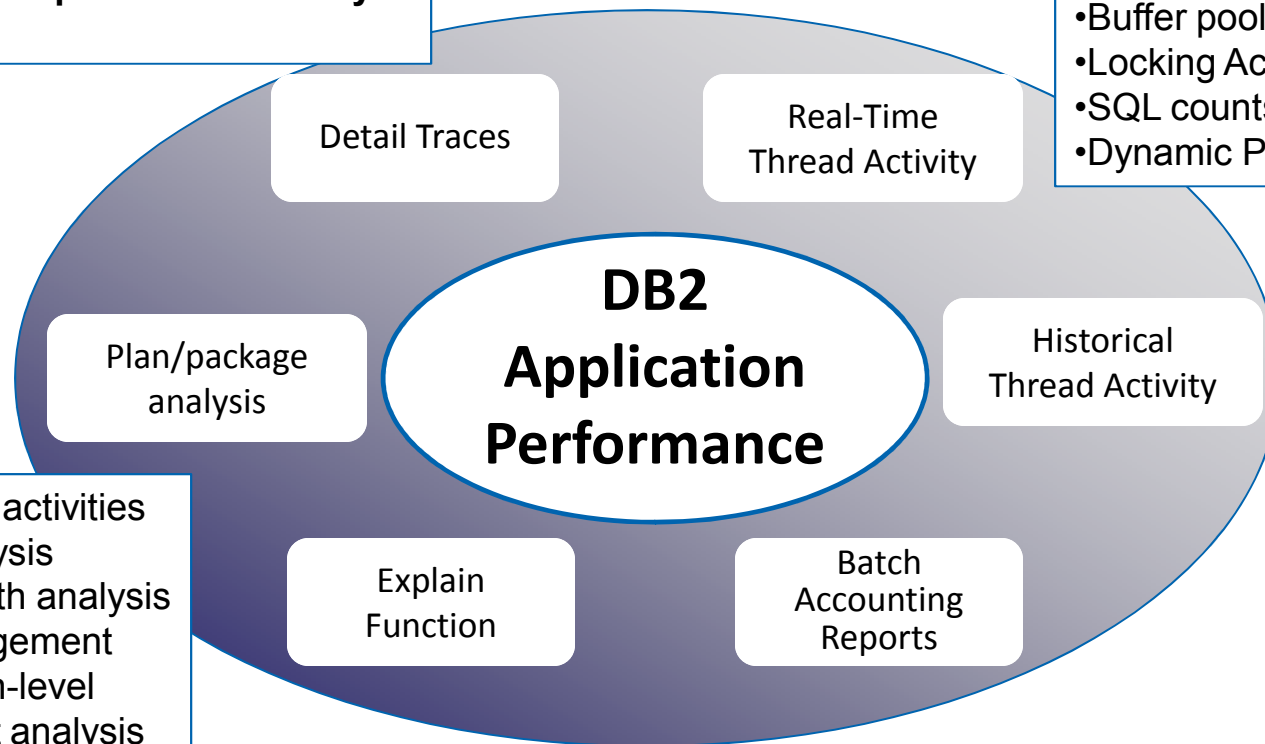
- Overall application throughput
- Transaction response times
- Application-specific availability objectives

Thread activity

- CPU & elapsed metrics
- Buffer pool usage
- Locking Activity
- SQL counts
- Dynamic Prepares

Application activities

- SQL Analysis
- Access path analysis and management
- Application-level throughput analysis



DB2 Performance Management

How is Application Performance Data Collected

IFI (Instrumentation Facility Interface) is used to look at real-time performance data and can be leveraged to capture timeouts, deadlocks, and similar events required for application analysis

- READS to retrieve trace records synchronously and cause them to be returned immediately
- READA to retrieve trace records asynchronously; trace records accumulate and is then moved to the requesting monitor program

Some monitor programs collect DB2 101 and 102 records and store them in history for application level analysis (100 records also)

- Accounting (102) records provide detail for thread executions that can then be rolled up for application level analysis
 - Important to use consistent identifiers so it's easy to group and analyze like records

Monitors and other products provide capabilities for reporting and analyzing SMF 102 records

Detailed SQL analysis along with plan analysis is accomplished using a combination of SMF data and DB2 catalog information

- Similar to system performance tools, performance products often read detailed application and SQL performance metrics using other techniques

DB2 Performance Management Accounting - Important Metrics

Workload Metrics are similar; however, we have the complete accounting records so we see details, not summarization as in the 100 records

- # of completed accounting records for a given interval
- GETPAGE requests – Getpage detail is still a good metric for overall workload but now we are able to focus at the application level
- Biggest resource consumers
- Overall response time can be evaluated since we know in-application and in-DB2 times
 - These response times aren't end-to-end; We just don't have enough information for that

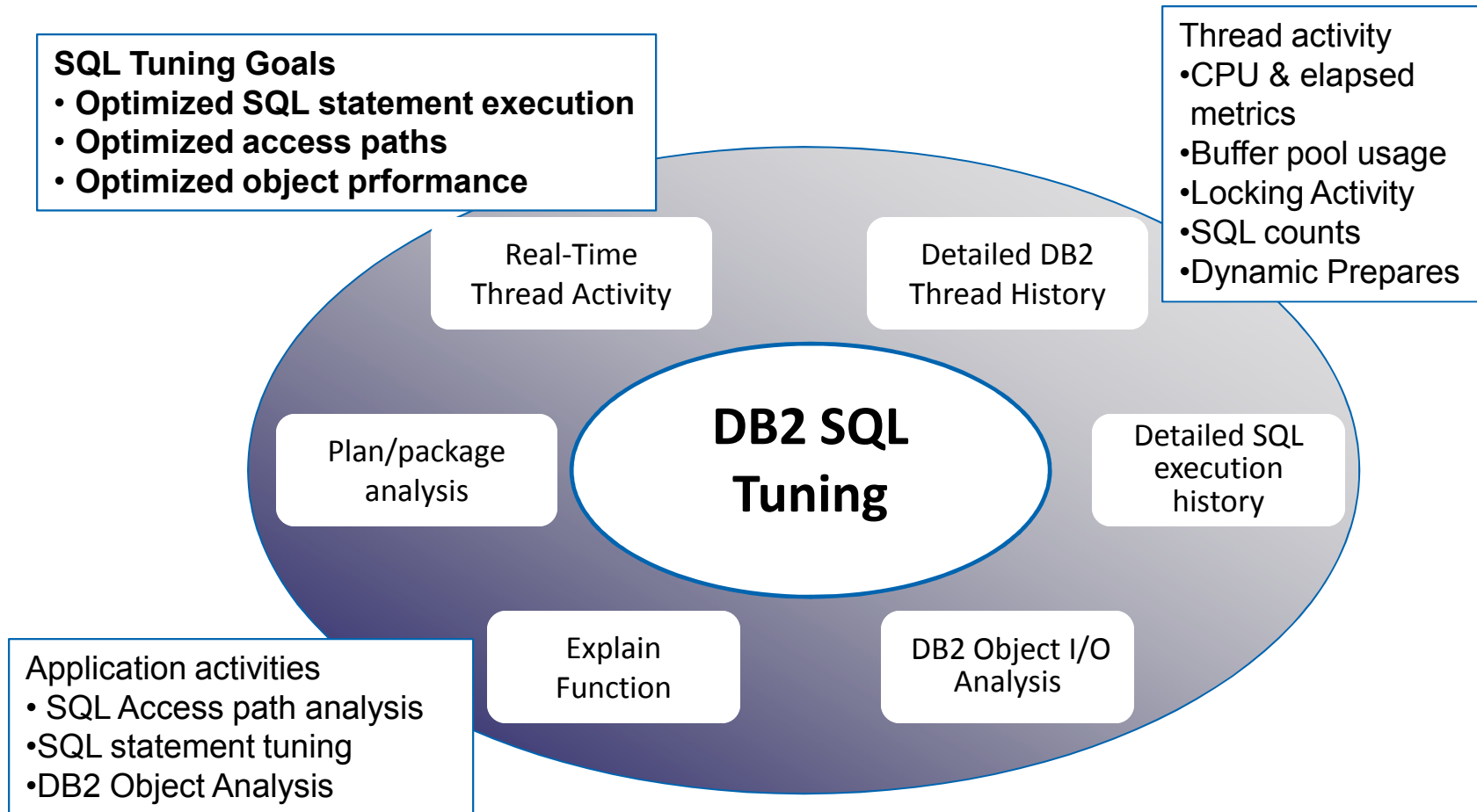
Problems to watch for (not nearly a complete list)

- Thread issues – We are now more concerned about concurrent threads. Can we support the required number of concurrent threads for our application. This can be impacted by CPU consumption and locking problems, among others.
- Buffer pool problems – We are now concerned about an individual application's usage of DB2 buffer pools. Hit ratios are an important metric here. It's often a good idea to isolate these objects in separate buffer pools to get accurate measurements for the application.
- Locking problems – Can be a major issue at the application level. Timeouts and deadlocks can seriously impact overall throughput
- Dynamic Statement Cache problems – For applications from a distributed front-end the dynamic statement cache is critical. Poor re-use of cached statements will drive up CPU cost for the application and impact overall throughput

Automation – Like system performance, automation can help identify and resolve problems without manual effort

DB2 Performance Management Disciplines

SQL Tuning



DB2 Performance Management

Collecting SQL Tuning Data

IFC detail traces can be used to collect SQL level detail

- Complete trace of the statement including:
 - SQL statement text
 - Details on all events (cursor creation, reads, writes, fetches)
 - Index accesses
 - SQL error conditions
 - Referential data accesses
- I/O operations
- High cost (20 – 100% overhead depending on IFCIDs being collected)
- Externalized to SMF type 102 records (recommended target is GTF)

Several vendors offer solutions that collect similar data at a lower cost using other techniques for collecting SQL metrics

- Some overhead is still incurred; however, it's smaller than the DB2 detailed trace

Access path information is also provided by the DB2 Explain feature

Additional SQL detailed data can be retrieved from the catalog

DB2 Performance Management

SQL Tuning - Important Metrics

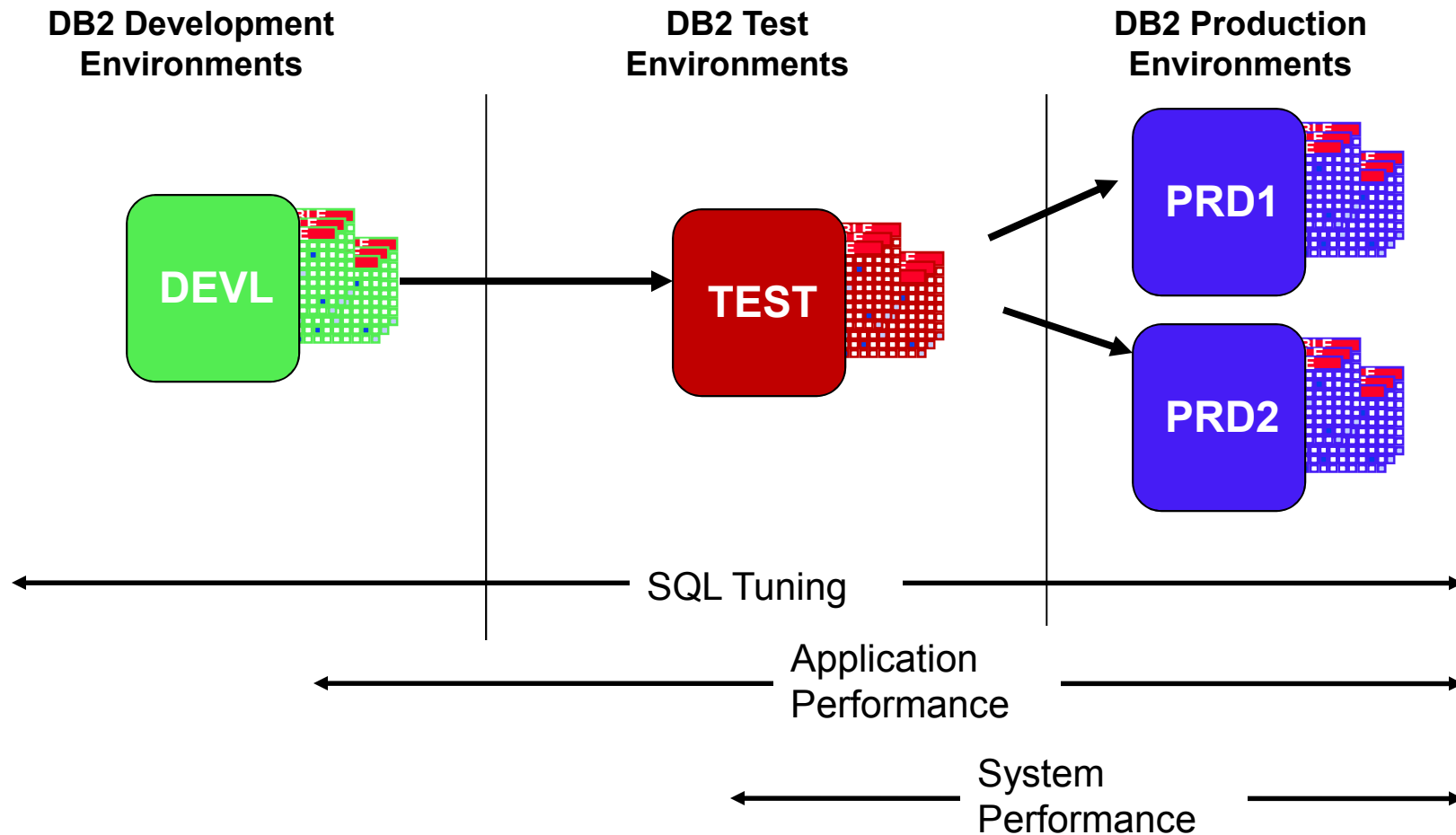
Workload Metrics are similar at the high level but we now have individual statement detail

- SQL elapsed and CPU time
- GETPAGE and I/O detail
- Wait events
- Dynamic SQL statement text

Problems that are more easily identified

- Bad access paths – SQL statement analysis helps identify bad access paths and can identify other options
- Negative SQL Return Codes – frequently occurring SQL return codes can be trapped
- And others

DB2 Performance Management Tuning Across the Application Lifecycle



thank you

legal notice

© Copyright CA 2011. All rights reserved. All trademarks, trade names, service marks and logos referenced herein belong to their respective companies. No unauthorized use, copying or distribution permitted.

THIS PRESENTATION IS FOR YOUR INFORMATIONAL PURPOSES ONLY. CA assumes no responsibility for the accuracy or completeness of the information. TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENT “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. In no event will CA be liable for any loss or damage, direct or indirect, in connection with this presentation, including, without limitation, lost profits, lost investment, business interruption, goodwill, or lost data, even if CA is expressly advised of the possibility of such damages.

Certain information in this presentation may outline CA’s general product direction. This presentation shall not serve to (i) affect the rights and/or obligations of CA or its licensees under any existing or future written license agreement or services agreement relating to any CA software product; or (ii) amend any product documentation or specifications for any CA software product. The development, release and timing of any features or functionality described in this presentation remain at CA’s sole discretion.

Notwithstanding anything in this presentation to the contrary, upon the general availability of any future CA product release referenced in this presentation, CA may make such release available (i) for sale to new licensees of such product; and (ii) in the form of a regularly scheduled major product release. Such releases may be made available to current licensees of such product who are current subscribers to CA maintenance and support on a when and if-available basis.