#SHAREorg

SHARE
Technology · Connections · Results

DB2 Performance Tuning Using Omegamon DB2 Performance Expert

- Use Case Examples and Practical Applications

Akiko Hoshikawa
IBM Silicon Valley Lab.

2013 Feb. 5

Session Number 12693

SHARE in San Francisco 2013

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.

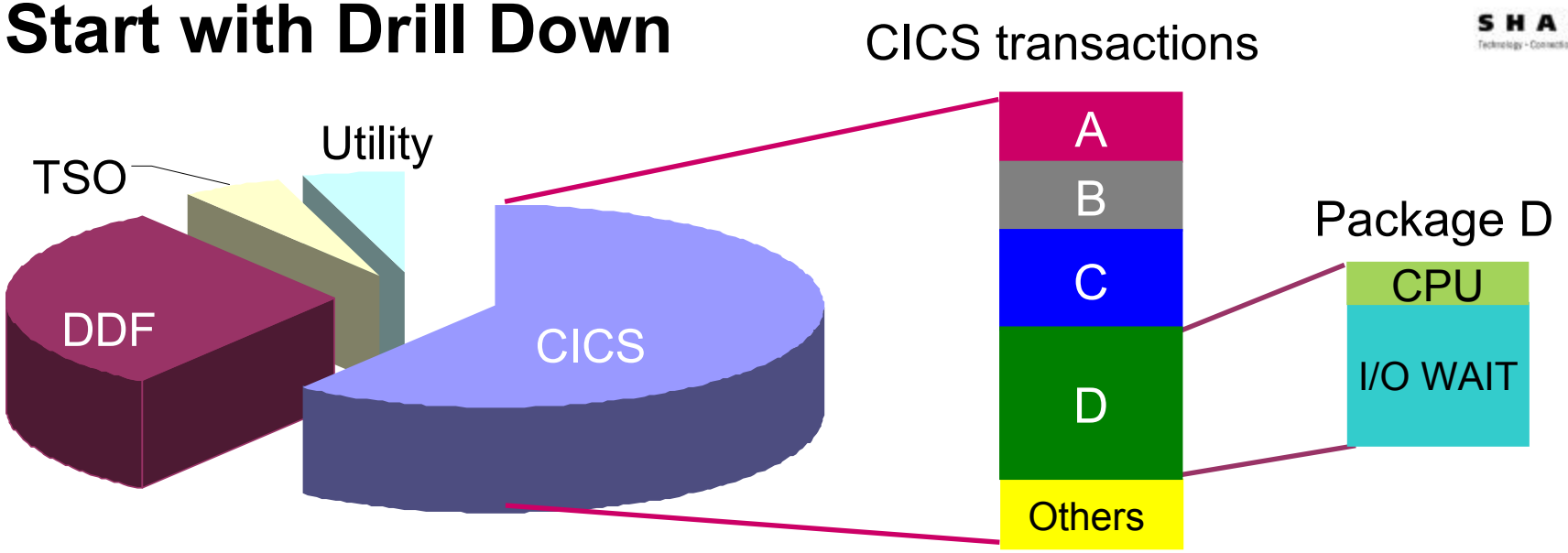
The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features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

Session Objectives and Agenda

- Objectives:
  - Provides the first step of understanding the most important DB2 traces - accounting and statistics
  - Using batch report from **IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS** (OMPE), examine real case studies
- Agenda
  - Part I : DB2 trace basics
    - Statistics and Accounting traces
    - From workload to thread level
  - Part II : Case studies
    - Examples : How to use DB2 traces to tune the DB2 applications

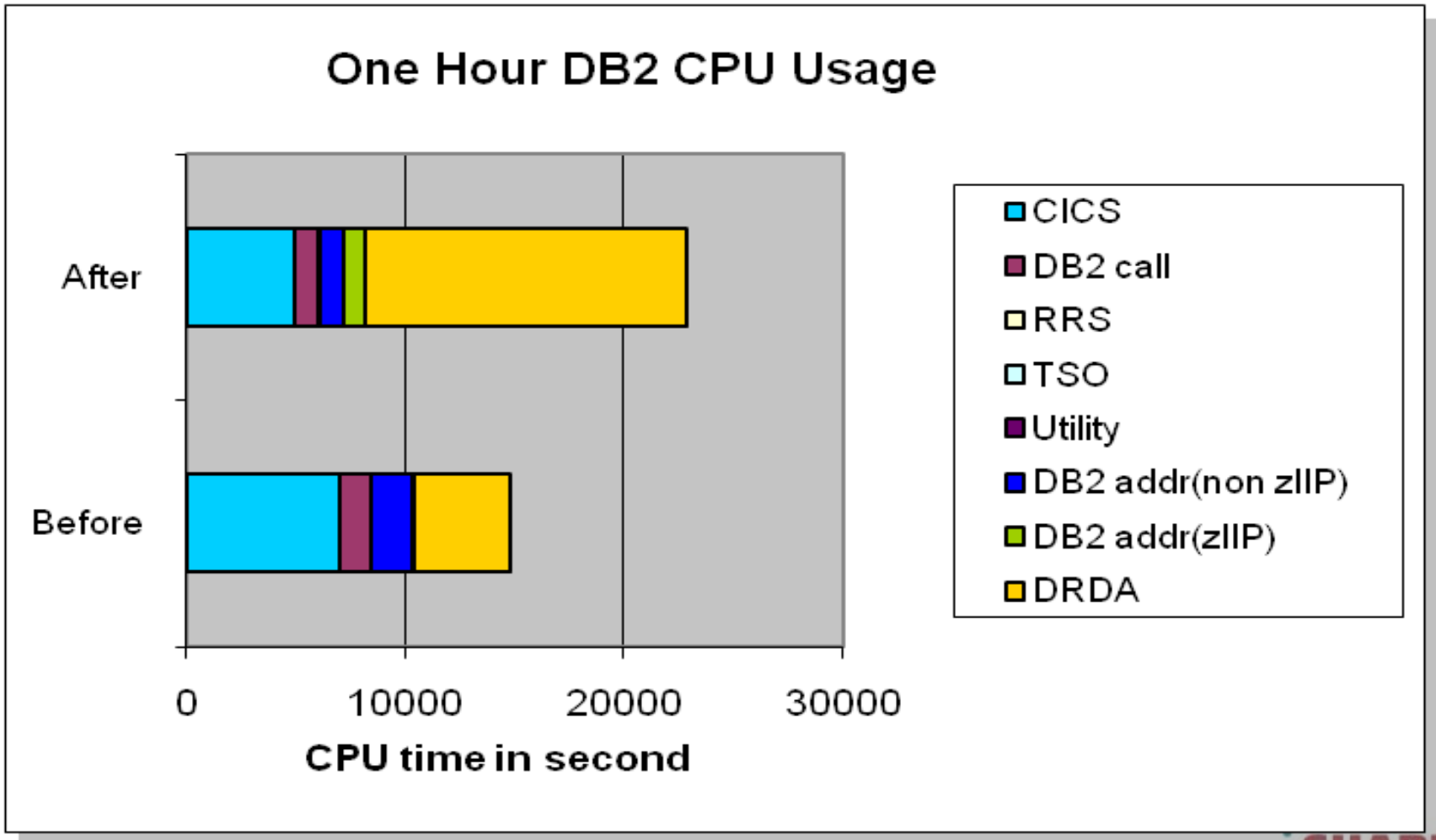
DB2 Application Monitoring and Tuning - Start with Drill Down



- Which DB2 applications that you want to focus?
  - RMF Workload activity report
  - DB2 accounting trace (connection type)
- Within the workload, which packages spend most CPU or elapsed time?

- **PART I**
  - Most Important DB2 Traces - Accounting and Statistics
  - OMPE support on Aggregated Accounting in Statistics
  - Creating Effective Accounting Reports
  - Case Study (1) – DB2 CPU Increase at Migration
- **PART II**
  - Case Study (2) – Sync I/O Wait
  - Case Study (3) – DB2 CPU Increase
  - Case Study (4) – Page Latch Contention
  - Case Study (5) – Page Latch Contention
  - Case Study (6) – Monitoring accelerated queries using IDAA

Case Study (1) Top Down Analysis -1



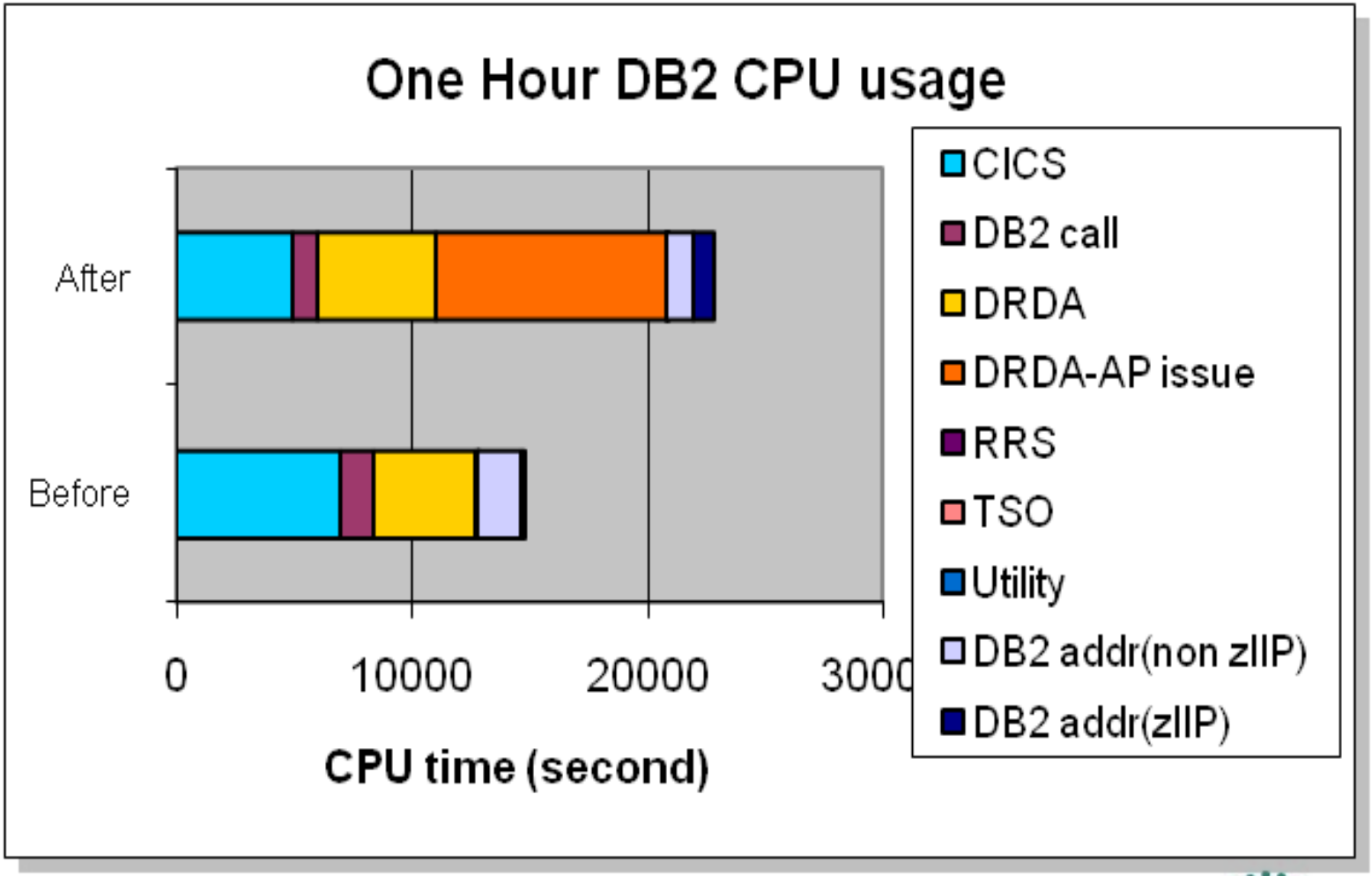
Case Study (1) Top Down Analysis -2

- OMPE :

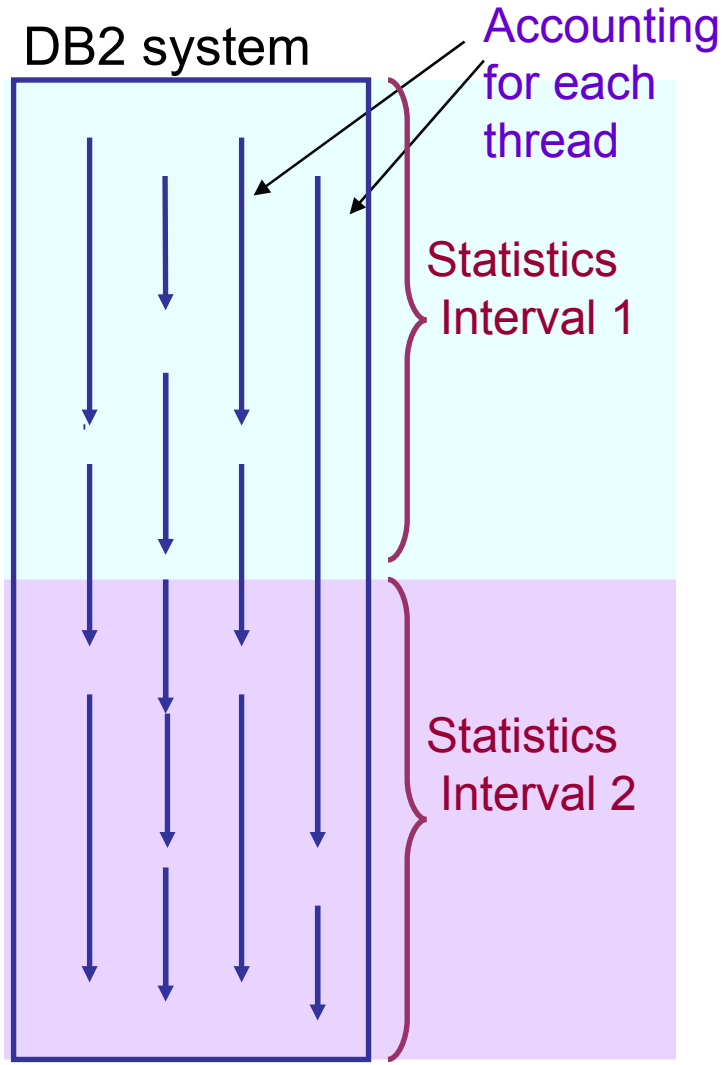
  ACCOUNTING REPORT REPORT LAYOUT(SHORT) ORDER(PROGRAM)

Package	Occurrence	CPU	Total CPU
DB2PROD.aaaa.package1	59	163.000	9617.000
DB2PROD.aaaa.package2	15464	0.020808	321.775
DB2PROD.aaaa.package3	37263	0.004912	183.036
DB2PROD.aaaa.package4	96409	0.00189	182.213
DB2PROD.aaaa.package5	164	0.996136	163.366
DB2PROD.aaaa.package6	96	1.660326	159.391
DB2PROD.aaaa.package7	334	0.426137	142.330
DB2PROD.aaaa.package8	61	2.231446	136.118
DB2PROD.aaaa.package9	51	1.738584	88.668
DB2PROD.aaaa.package10	11	7.44109	81.859
DB2PROD.aaaa.package11	208	0.359636	74.804
DB2PROD.aaaa.package12	204	0.348851	71.165

Case Study (1) Top Down Analysis -3



DB2 Accounting and Statistics



- **Statistics (SMF 100, 102)**
  - DB2 Subsystem level
    - How many DML, getpage, lock, etc per second in the DB2 member
    - Interval base
  - CLASS 1,2,3,4,5,6,8,9
    - Recommend CLASS(*)
- **Accounting (SMF 101)**
  - DB2 Thread level
    - How much CPU the thread consumed
  - CLASS 1,2,3,4,5,6,7,8,10
    - Recommend Class(1,2,3,7,8)

Statistics Trace & Report

- Statistics trace is a prime indicator of system status and performance
  - Address space CPU time
    - TCB and SRB time for
      - MSTR, DBM1, IRLM, DDF
    - zIIP time under address space
  - Workload at system level
    - DML/DDL activities
    - Locking activities
    - DB2 latches
    - Log activities
    - Plan/Package
    - DRDA activities
    - Open/Close
  - Pools/Storage usage
    - EDM pools
    - RID pool
    - Statement cache
    - Local Buffer pools
    - Group buffer pools
    - Workfile usage
    - Authorization cache
    - Virtual and real storage usage
    - zOS metrics

Statistics Monitoring - Best Practices

- Statistics Interval – STATIME
  - DB2 9 default 5 min
  - DB2 10 1 minute for basic statistics
- Statistics overhead
  - Negligible cpu cost
- Best practices and Rules of thumbs based on DB2 statistics can be found…
  - https://www.ibm.com/developerworks/data/bestpractices/db2zos/

What is recorded in DB2 Accounting Trace ?

Almost everything you want to know about the DB2 thread

- Elapsed time
  - Nested and non nested
- CPU time
  - GP and zIIP/zAAP
- Wait time
- DML activities
- DDL activities
- Local and Global Lock activities
- Local buffer pools
- Group buffer pools
- DRDA information
- LOG activities
- Data Capture
- Dynamic Statements
- Query Parallelism
- Sproc, Triggers, UDF
- LOB and XML

- Note: zIIP CPU times reported in DB2 traces are normalized to the speed of the general purpose processors

DB2 Accounting Trace

- To Start
  - -STA TRA(ACCTG) …. Or at DB2 startup SMFACCT in ZPARM
- IFCIDs
  - IFCID 3 – plan level info
  - IFCID 239 – package level info
- Accounting Trace classes
  - Class 1 – total time (elapsed and CPU) – IFCID 3
  - Class 2 – time in DB2 (elapsed and CPU) – adds info to IFCID3
  - Class 3 – suspension time in DB2 – adds info to IFCID 3
  - Class 7 – package info (similar to class 2)
  - Class 8 – package info (similar to class 3)
  - Class 10 – package detail SQL/locking/BP info at package level

Notes : DB2 Accounting data is written at

- Thread deallocation (including abends)
- (re)signon in case of thread reuse by IMS/CICS
  - ACCOUNTREC (UOW / TASK) in DB2ENTRY for CICS
- At commit for RRS threads using accounting-interval COMMIT on signon/ auth signon / context signon
- At commit when CMTSTAT=INACTIVE and provided connection can go inactive
  - JCC applications : same packages (i.e SYSLN200)
    - to identify a specific JDBC transaction/application, accounting strings need to be set.
    - setDB2ClientAccountingInformation
    - db2.jcc.accountingInterval=COMMIT in DB2JccConfiguration.properties
  - Things that prevent a connection from going inactive:
    - Touched a package that uses Keepdynamic(yes)
      - *If only reason for not going inactive is the use of keepdynamic(yes), still cut acctg (and reset enclave)*
    - Active DGTT (not explicitly or implicitly dropped)
    - Open cursor with hold
    - Held LOB locator

Notes: How To Set Client Information

- Ease of analysis
  - PLANNAME: db2jcc_a if you do not set client info set
- Java methods for existing Set Client Information API
  - setClientUser(zheng)
  - setClientWorkStation(CL01)
  - setClientApplicationInformation(payment)
  - setClientAccountingInformation(String)
- WebSphere Application Server before V6.0
  - Only settable as DB2 DataSource property
  - limited because information can not be changed dynamically
- WebSphere Application Server Version 6.0 supports explicite and implicite setting of client information
  - Example how to call explicitely
    - WSConnection conn = (WSConnection) ds.getConnection();
    - props.setProperty(WSConnection.CLIENT_ID, "user123");
    - conn.setClientInformation(props);
  - Example how to call implicitely by turning on WebSphere Trace Group
    - WAS.clientinfo=all=enabled or
    - WAS.clientinfopluslogging=all=enabled

DB2 Traces and Overhead

- DB2 Statistics : Negligible
- DB2 Accounting : CPU Typically less than 5%
  - Class 1: less than 5% CPU overhead
  - Class 2: 1 to 10% CPU overhead, typically less than 5%
    - Class 2 accounting (SQL statement level)
      - *Can be higher for fetch-intensive applications, up to 20%*
      - *V8 multi-row Fetch can make this overhead negligible*
  - Class 3: less than 1% CPU overhead
    - Much higher than 1% cpu overhead for class 3 acctg has been observed in a rare situation of very high internal DB2 latch contention rate, eg over 10000/sec.
  - Class 7 and 8 (Package): less than 1-3% cpu overhead,
  - Class 10 (Package Detail) : 2-5% overhead

CPU time and Trace / Monitor Products

- One of the most common reason of CPU degradation in many DB2 performance PMRs
- Suspect trace / Monitoring overhead if you see 2-3x CPU increase without access path change
- Be aware what traces are on in your system by display trace command
- Minimize orphaned trace records
  - Orphaned traces because monitoring (eg vendor tool) stopped but not DB2 trace. The same CPU overhead as monitoring is on.
  - DB2 tries to eliminate orphaned trace records
- How can you tell CPU usage without your online monitoring?
  - Batch reporting
  - RMF Workload activity report

Statistics - from a customer

IFC DEST	Written	Not written, Not accepted
SMF	6306K	0
OP1	7851K	11199
OP6	0	0
OP7	0	6304K
OP8	0	0
Others	0	0

How to connect RMF and DB2 accounting-1?

RMF Workload Activity Report for 6 mins

```
-TRANSACTIONS-   TRANS-TIME HHH.MM.SS.TTT   --DASD I/O--
AVG      413.87  ACTUAL                 69  SSCHRT 54643
MPL      413.87  EXECUTION              69  RESP     7.0
ENDED   2222178  QUEUED                  0  CONN     0.1
END/S   5995.48  R/S AFFIN               0  DISC     6.7
#SWAPS         0  INELIGIBLE              0  Q+PEND   0.1
EXCTD          0  CONVERSION              0  IOSQ     0.0
AVG ENC  413.87  STD DEV               226  -
```

Transaction Rate 5995 trx/sec

Transaction response time 69 ms

Accounting Report for 6 mins

```
HIGHLIGHTS                    AVERAGE        APPL(CL.1)   DB2 (CL.2)
--------------------------    -----------    ----------   ----------
#OCCURRENCES     :   2107173  ELAPSED TIME     0.068947     0.067564
#NORMAL TERMINAT:   2107173   NONNESTED       0.001749     0.000700
#COMMITS         :   2105049   STORED PROC     0.066852     0.066852
#ROLLBACKS       :      2158   UDF             0.000346     0.000012

SYNCH I/O AVG.   : 0.008253
```

How to connect RMF and DB2 accounting -2?

RMF Workload Activity Report for 6 mins

CPU / #TRAN
3630/2222178
=1.6ms

```
SERVICE TIME   ---APPL %---
CPU  3630.475  CP     979.51
SRB     0.000  AAPCP    0.00
RCT     0.000  IIPCP 578.75
IIT     0.000
HST     0.000  AAP       N/A
AAP       N/A  IIP       N/A
IIP       N/A
```

IIPCP/CP
579 / 979.5
=59%

```
-TRANSACTIONS-
AVG       413.87
MPL       413.87
ENDED   2222178
END/S   5995.48
```

Transaction CPU time 1.6 ms

# of Concurrent Threads 414

Accounting and Statistics Report for 6 mins

```
AVERAGE          APPL(CL.1)
-----------      ----------
CP CPU TIME        0.001598
 AGENT             0.001598
  NONNESTED        0.000240
  STORED PRC       0.001342
  UDF              0.000016
  TRIGGER          0.000000
 PAR.TASKS         0.000000

 SECP CPU          0.000956
```

Eligible zIIP offload rate 59%

```
Statistics - DBM1
-----------------------------
NUMBER OF ACTIVE DBATS    414.26
```

SECP / CP CPU
0.000956 / 0.001598
=59%

Reducing Volumes of Accounting Traces

- Accounting roll-up for RRS and DDF
  - Aggregating accounting data for the same values of the client-side identifiers (end user ID, end user transaction/application name, end user workstation name)
    - zparm ACCUMUID controls which combination of IDs is used
    - zparm ACCUMACC controls when the aggregated accounting records are externalized – default 10
  - DB2 10 add roll up support for package Accounting
- DB2 10 Compression of SMF records
  - zparm SMFCOMP ON (default OFF)
  - Requires decompression support from monitoring products
  - DB2 supplies DSNTSMFD to decompress DB2 traces
  - Small overhead (1%) with great compression ratio(60-80%)

New Accounting in Statistics Support

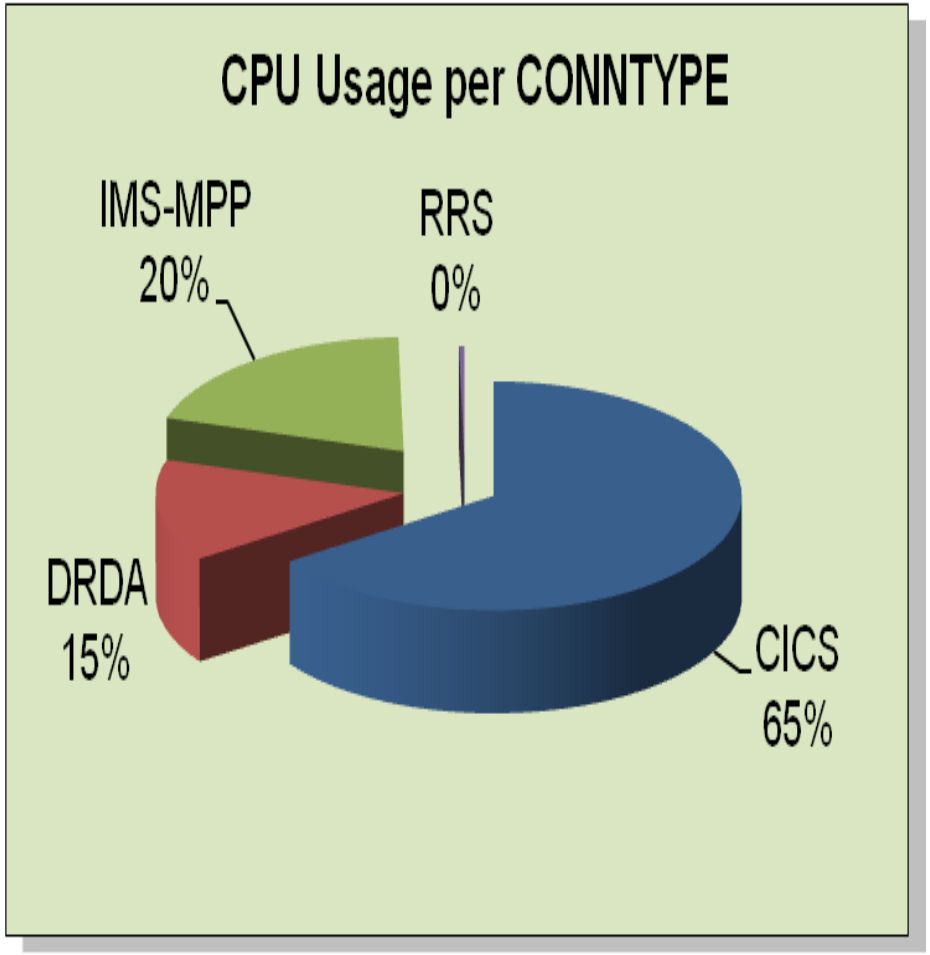
- DB2 APAR PM62797
  - Supports aggregation of accounting data into statistics report
  - Collect IFCID3 and IFCID369 (or statistics class 9)
- OMPE V511 APAR PM72949
  - Statistics formatter can publish aggregated accounting data
  - Statistics aggregate CPU/Elapsed time info into statistics trace

Aggregated Accounting in Statistics (OMPE example)

CPU TIMES	TCB TIME	PREEMPT SRB	NONPREEMPT SRB	TOTAL TIME	PREEMPT IIP SRB	/COMMIT
SYSTEM SERVICES ADDRESS SPACE	0.311114	5.912448	0.020002	6.243564	N/A	0.000019
DATABASE SERVICES ADDRESS SPACE	6.097576	48.396869	0.697035	55.191480	0.000000	0.000166
IRLM	0.000108	0.000000	0.296435	0.296543	N/A	0.000001
DDF ADDRESS SPACE	0.056293	4:15.988350	11.350365	4:27.395008	0.000000	0.000803
TOTAL	6.465091	5:10.297667	12.363837	5:29.126594	0.000000	0.000988

CONNTYPE	CL1 ELAPSED	CL1 CPU	CL1 SE CPU	CL2 ELAPSED	CL2 CPU	CL2 SE CPU	CL3 SUSP	THREADS
BATCH	1:14.326062	8.882753	0.000000	1:14.234073	8.879979	0.000000	1:05.997069	3.00
CICS	N/P	N/P	N/P	N/P	N/P	N/P	N/P	0.00
DDF	1:23:47.54247	4:02.519462	0.000000	1:11:52.05797	2:33.206501	0.000000	1:07:58.67980	221.0K
IMS	N/P	N/P	N/P	N/P	N/P	N/P	N/P	0.00
RRSAF	1:40:13.74934	1:46.557250	0.000000	1:11:53.96995	1:09.962190	0.000000	1:10:43.87768	110.1K
UTILITY	27:27.200715	9:09.367476	0.000000	9:54.465831	6:40.422900	0.000000	1:30.511908	16.00

Find Focus Area – Top Consumers



- This gives you a big picture of your DB2 activities
- To produce this information :
  - Aggregated accounting in statistics by CONNTYPE, or
  - OMPE batch report
    ACCOUNTING REPORT
    ORDER(CONNTYPE)

How To Narrow Down Plan/Package Level?

- OMPE ACCOUNTING FILE or SAVE-FILE to create input data for,
  - Performance Warehouse
    - ACCOUNTING FILE or Save-File
    - LOAD into DB2 Performance table
    - Query from the tables
  - Spreadsheet input generator : PM73732 (Dec. 2012)
    - ACCOUNTING FILE or Save-File
    - Spreadsheet Input Data Generator Utility to create CSV format
    - Imported into spreadsheets
- OMPE batch ACCOUNTING report generation

Creating Effective Accounting Report

- Filter input data using INCLUDE, EXCLUDE
  - By connection INCLUDE(CONNTYPE(DRDA))
  - By authid INCLUDE(AUTHID(DB2USR1))
- Reduce to cover the specific time using REDUCE
  - Example : Reports for every hour from 8:30 to 12:00

```
REDUCE FROM (,08:30) TO (,12:00)
     INTERVAL (60) BOUNDARY (60)
REPORT FROM (,08:30) TO (,12:00)
```

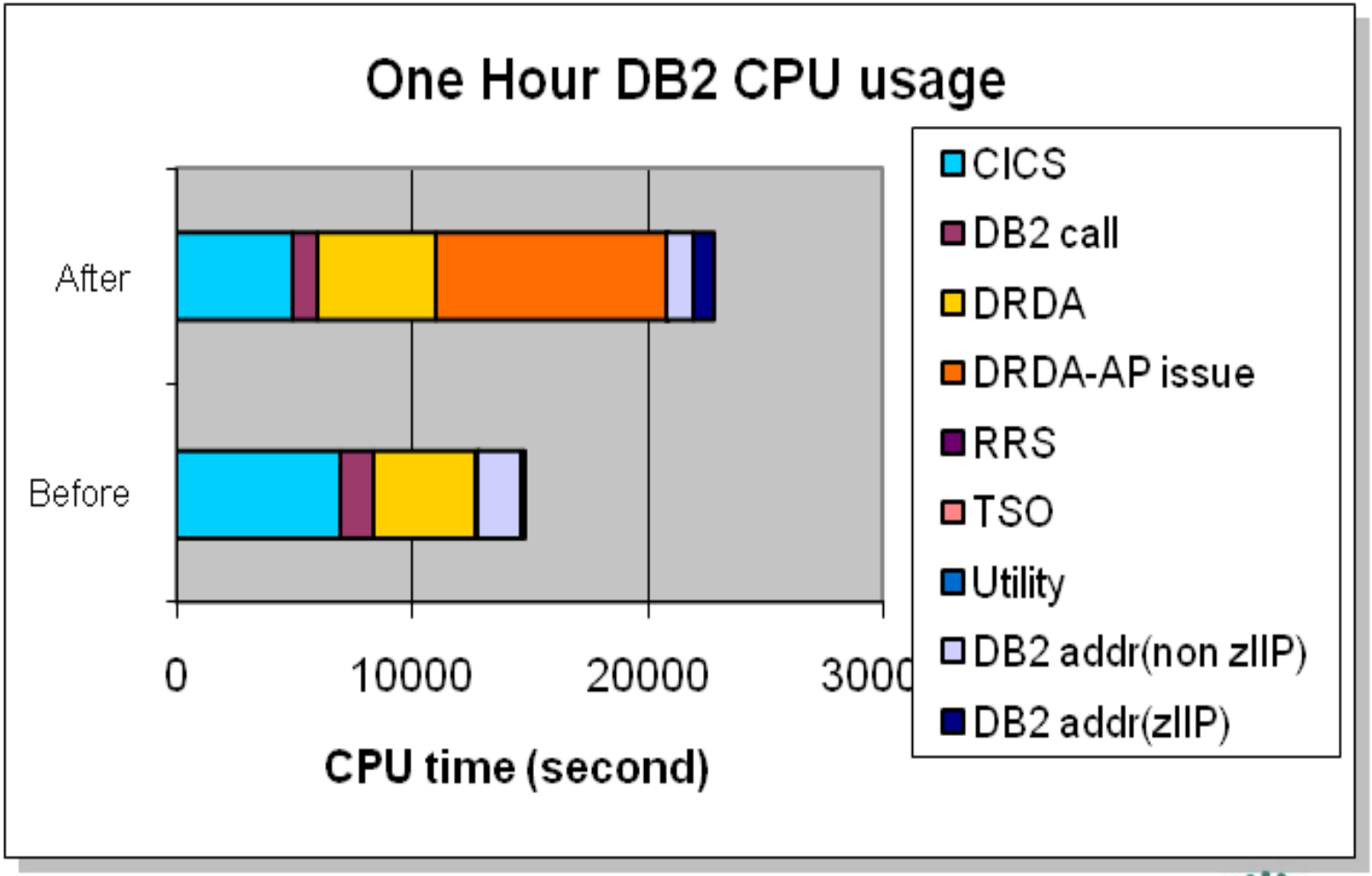
- TOP processing
  - Example : Top 10 occurrences of highest DB2 CPU usage

```
ACCOUNTING TRACE TOP (10 ONLY INDB2PT)
```

Notes : TOP Processing

- Useful TOP keywords
  - INAPPLET(cl1 elapsed)
  - INDB2ET (cl2 elapsed)
  - INAPPLPT(cl1 CPU)
  - INDB2PT(cl2 CPU)
  - INDB2WT(wait in DB2)
  - TOTSUSTM(class3)
  - DMLSTATS(DML)
  - UPDPERCM (Update/Insert/Del)
  - TOTSUSP(class3 events)
  - NOTACCT(not account)
  - GETPATES(getpages)
  - BUFUPDTS(buffer updates)
  - SYNCREAD(sync read events)

Case Study (1) Top Down Analysis -3



How Do You Determine Access Path Issue or Not from Accounting Trace ?

- Primary indicators in accounting traces
  - Number of SQL statements and rows fetched
  - Number of get pages
- Verify access path information
- Example..

	Package A – Bad	Package A - Good
CL7 Elapsed	247 sec	20 sec
CL7 CPU	163 sec	6 sec
SQL STMT	120	120
DB2 Entry/Exit	360	360
GETPAGES	1636K	1453

Notes: Once Access Path Issue Confirmed

- Make sure optimizer does have enough information – RUNSTATS should be up to date to find the right filter factors
- Collect information
  - Service SQL information (DDL, Catalog statistics information, explain tables)
- Use plan management (PLANMGMT) for static applications, hint (static/dynamic) to correct access path.
- Consider DB2 10 APPREUSE(ERROR)

Notes: Capturing documentation for IBM

ethods for capturing documentation for all releases is documented here

- https://www.ibm.com/support/docview.wss?uid=swg21206998
- OSC and DB2PLI8 do not support DB2 10

YSPROC.ADMIN_INFO_SQL supports V8 -> V10 (Required)

- Excellent developerWorks article here:
  - http://www.ibm.com/developerworks/data/library/techarticle/dm-1012capturequery/index.html
- It is installed in V10 base and is subject to the installation verification process
  - DB2HLQ.SDSNSAMP(DSNTESR) will create and bind it
  - calling program is DSNADMSB, and sample JCL in DSNTEJ6I
- Ensure DB2 9 and DB2 10 have APAR PM39871 applied

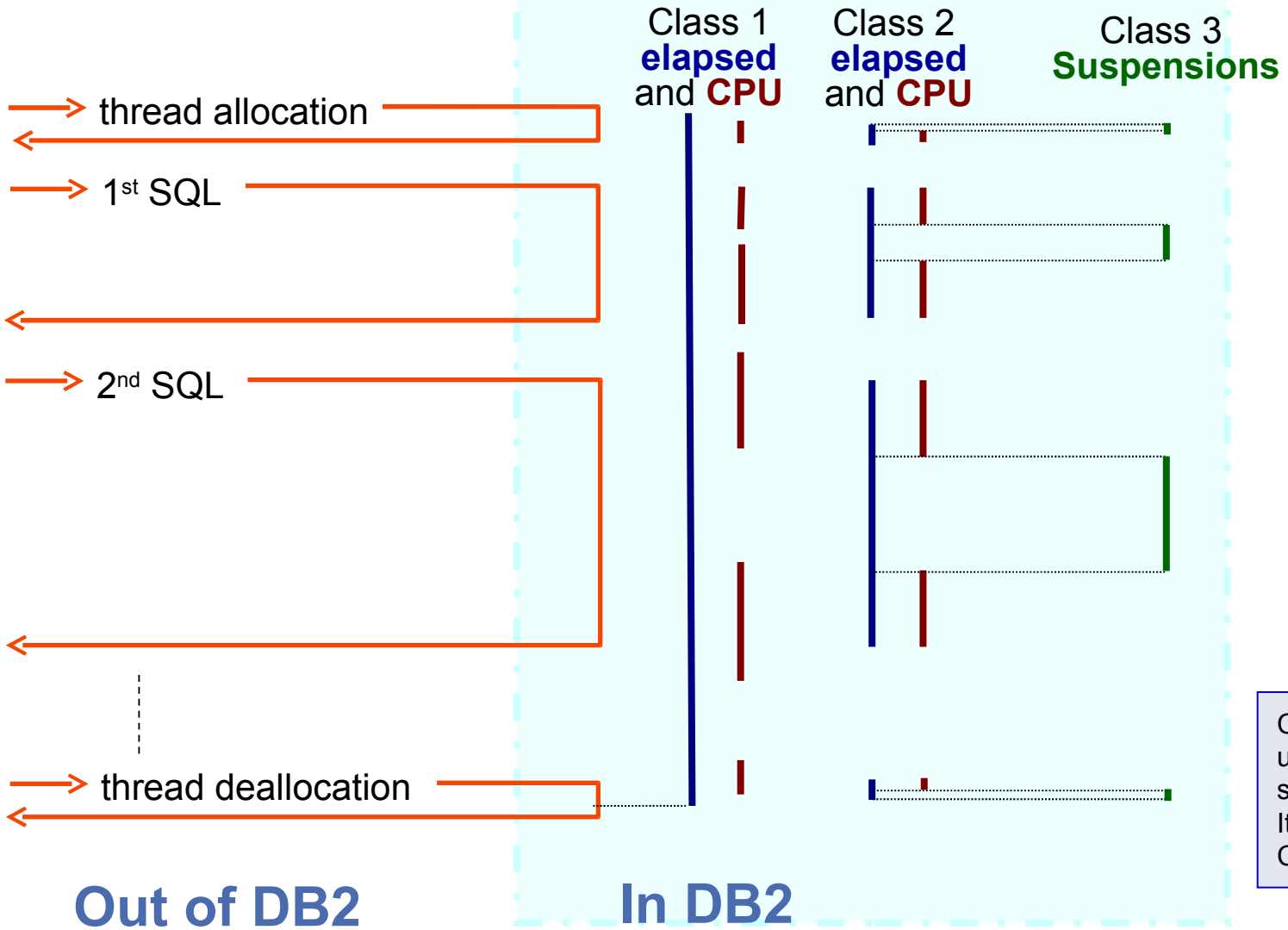
ata Studio V3.1 incorporates this procedure into a GUI (Best Practice)

- http://www.ibm.com/developerworks/downloads/im/da
  - No charge product, replacement for OSC and Visual Explain
  - Several versions:
    - *DBA's should download the Administration Client*
  - Incorporates Statistics Advisor
  - FTP doc directly to DB2 Level 2
  - Can be used to duplicate stats in TEST environment

Component /Operating system	Version	Size
Administration client	V3.1	359MB - 362MB
Red Hat Linux, SUSE Linux, Windows		

- PART I
  - Most Important DB2 Traces - Accounting and Statistics
  - OMPE support on Aggregated Accounting in Statistics
  - Creating Effective Accounting Reports
  - Case Study (1) – DB2 CPU Increase at Migration
- **PART II**
  - Case Study (2) – Sync I/O Wait
  - Case Study (3) – DB2 CPU Increase
  - Case Study (4) – Page Latch Contention
  - Case Study (5) – Page Latch Contention
  - Case Study (6) – Monitoring accelerated queries using IDAA

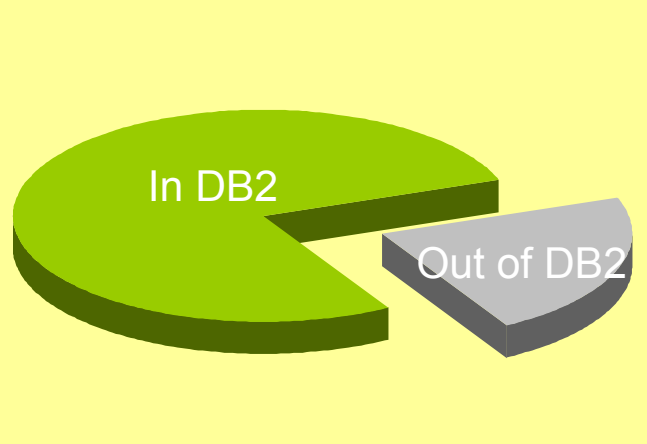
Accounting Class 1,2,3 and 7,8



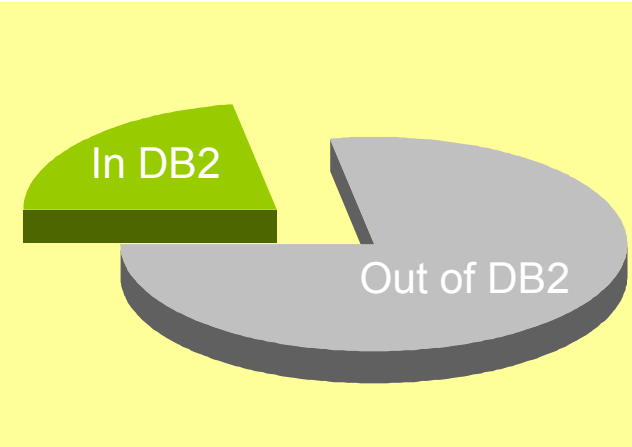
Class 7 and 8 is used for package scope reporting. It is equivalent to Class 2 and 3.

What If Activity Time Is Split as …

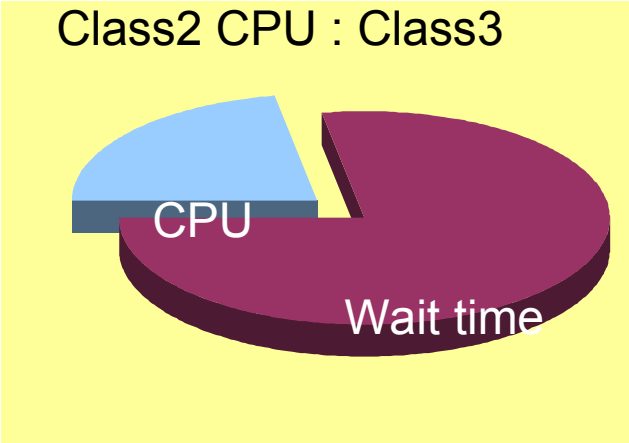
Class1 : Class2



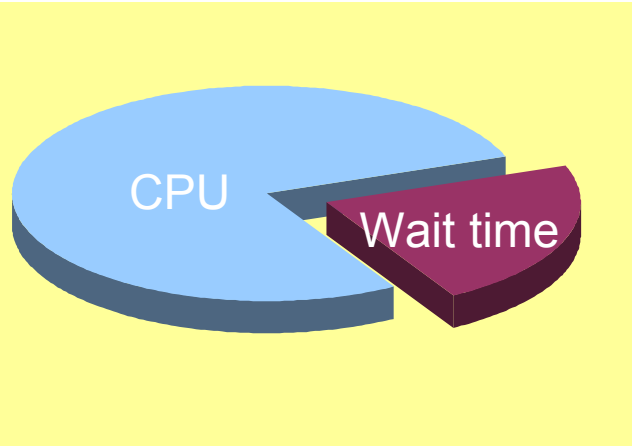
Class1 : Class2



Class2 CPU : Class3



Class2 CPU : Class3



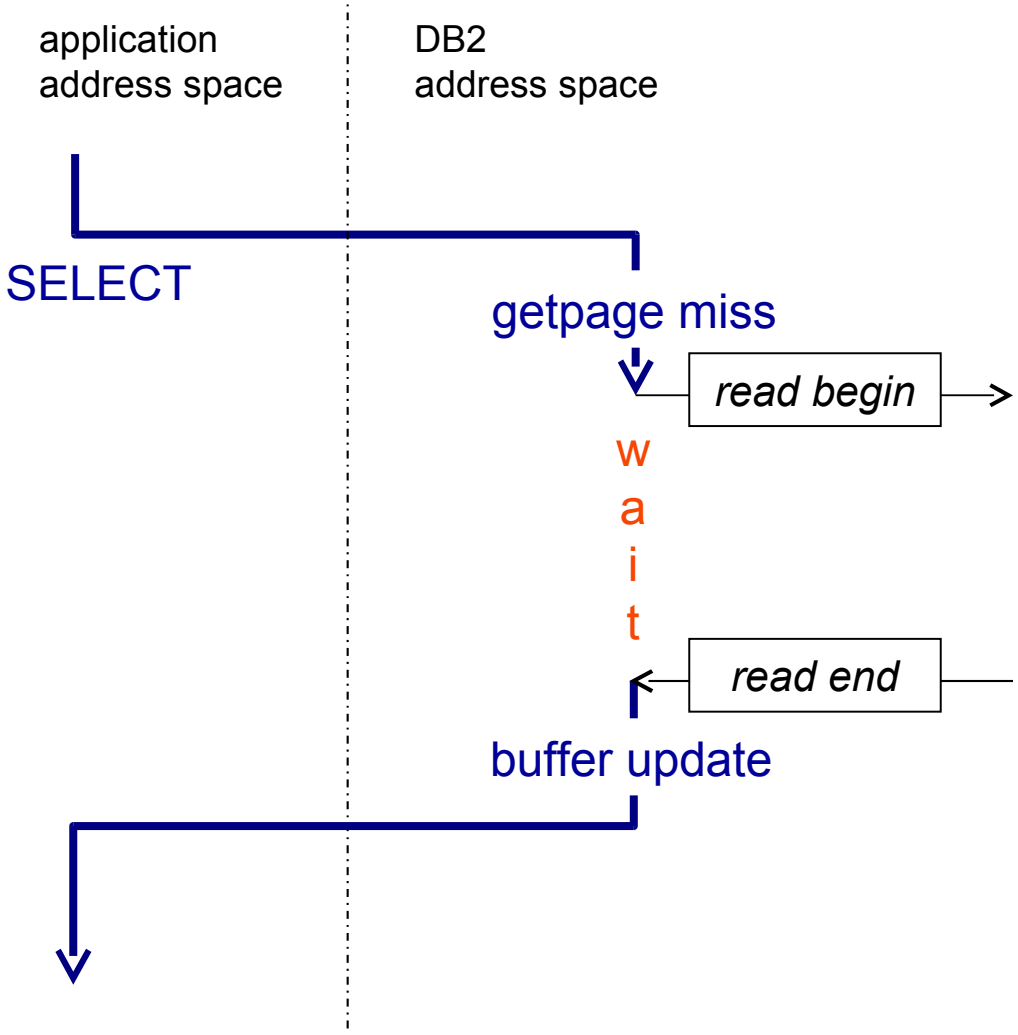
Case Study (2) Sync I/O Wait -1

MARKET#0	AVERAGE TIME	AVG.EV
LOCK/LATCH	0.001157	2.05
SYNCHRONOUS I/O	0.117791	81.96
OTHER READ I/O	0.002557	1.26
OTHER WRITE I/O	0.000000	0.00
TOTAL CL8 SUSPENS.	0.121505	85.26

Avg 1.4 ms per I/O req

MARKET#0	AVERAGE	TOTAL
BPOOL HIT RATIO (%)	13.55	N/A
GETPAGES	605.21	2126103
BUFFER UPDATES	0.01	51
SYNCHRONOUS WRITE	0.00	0
SYNCHRONOUS READ	81.96	287925

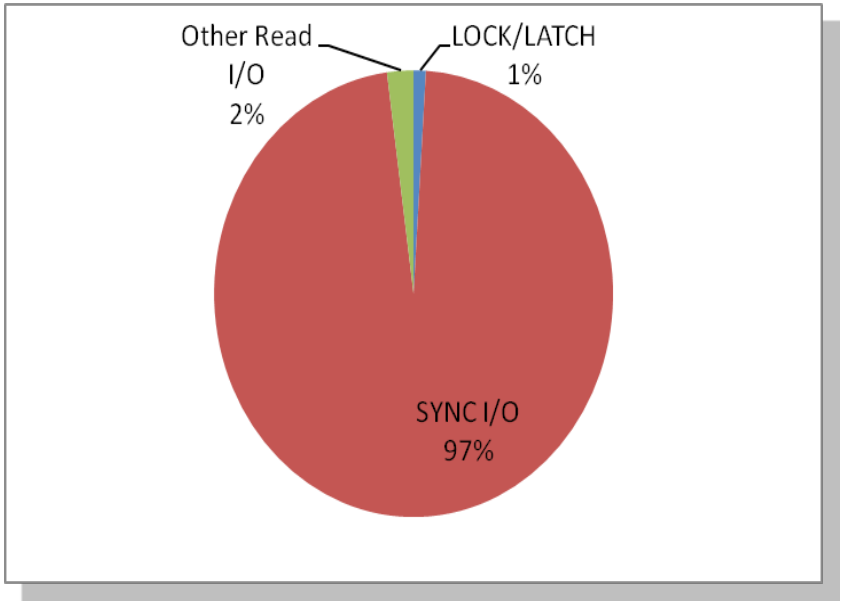
Synchronous Database I/O Suspensions



- A common timer for:
  - Synchronous database reads
  - Synchronous database writes
  - Delay due CPU constraint
- A separate timer for synchronous log writes
- Use BP statistics to find out how many of the suspensions should be attributed to synchronous reads vs. writes
- Avg I/O wait time in Acctg rpt = 0.028539 ms/12.41 events = 0.002 msec / sync DB IO

When You See High I/O Wait ..

- Verify if access path is optimal
  - Excessive Getpage
  - Identify which objects from BPOOL section
- Verify average I/O resp time
  - If significantly larger than RMF
    - CPU delay instead of I/O delay
- Buffer Pool Tuning
- REORG : Data/Index
- I/O configuration tuning using RMF data
    - DISK CACHE hit
    - DASD Acitivity report
  - Make sure of sufficient I/O resources
    - Hiper PAV (Parallel Access Volume) for concurrent I/O
    - SSD hot data which cannot be fit in DB2 buffer pools

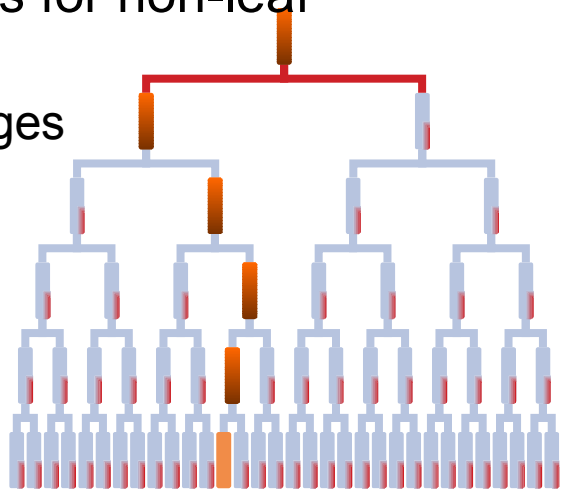


Notes: Buffer Pool Guideline

- Buffer pool hit ratio = GETPAGE req/ Sync Read
- Buffer pool separation
  - For example,
    - Index, Data, Workfile
    - Write intensive data and read intensive data
  - In memory buffer pool
    - Page steal - FIFO
    - Disable prefetch - avoid unnecessary prefetch
    - High VDWQT threshold - keep pages in buffer pools
  - Compressed data page stays compressed in buffer pool, expands when rows are evaluated
  - Compressed index (V9 NFM) is expanded in buffer pool
  - Control unit caching
    - Control unit cache is typically larger than buffer pools (= real storage)
    - Compression can raise control unit cache if not buffer pool hit
- Workfile bpool
  - Set Sequential Prefetch threshold = 99 % as all workfile read are sequential
  - Assign enough 32K workfile buffers for DB2 9.
    - Before DB2 9 record size > 4K uses 32K workfile
    - DB2 9 record ize > 100bytes uses 32K workfile

Notes : Buffer Pool Guideline – Index buffer pools

- Generally, assign more buffers than data
  - Better chance to be utilized than Table
- For large indexes, at least assign enough buffers for non-leaf pages
  - 1billion rows table with 200 index entries per 4K pages
  - 1,000 M / 200 = 5,000,000 Leaf Pages
  - 5,000,000 / 200 = 25,000 Level 2 (Non Leaf)
  - 25,000 / 200 = 125 Level 3 (Non Leaf)
  - 125 / 200 = 1 Level 4 (Non leaf)
  - Total Non Leaf page = 25126



- Consider to use Large Index pages (V9 NFM) <u>if the index is accessed key order</u>
  - Getpage and split reduction
  - Compressed Index page is expanded at I/O time, stays uncompressed in buffer pool.

Case Study (2) Sync I/O Wait -2

MARKET#0	AVERAGE TIME	AVG.EV
LOCK/LATCH	0.001157	2.05
SYNCHRONOUS I/O	0.117791	81.96
OTHER READ I/O	0.002557	1.26
OTHER WRITE I/O	0.000000	0.00
TOTAL CL8 SUSPENS.	0.121505	85.26

MARKET#0	AVERAGE TIME	AVG.EV
LOCK/LATCH	0.000569	1.57
SYNCHRONOUS I/O	0.000495	0.62
OTHER READ I/O	0.000000	0.00
OTHER WRITE I/O	0.000000	0.00
TOTAL CL8 SUSPENS.	0.001064	2.19

Sync I/O reduction by
1. Separate BPOOL for better hit ratio
2. Increase BPOOL size
3. Compress the data page

Case Study (3) DB2 CPU Increase -1

BEFORE

TIMES/EVENTS	APPL(CL.1)	DB2 (CL.2)	CLASS 3 SUSPENSIONS	ELAPSED TIME	EVENTS	HIGHLIGHTS
ELAPSED TIME	5:42.43304	5:41.25154	LOCK/LATCH(DB2+IRLM)	0.011943	899	THREAD TYPE : ALLIED
NONNESTED	5:42.43304	5:41.25154	IRLM LOCK+LATCH	N/A	N/A	TERM.CONDITION: NORMAL
STORED PROC	0.000000	0.000000	DB2 LATCH	N/A	N/A	INVOKE REASON : DEALLOC
UDF	0.000000	0.000000	SYNCHRON. I/O	1:29.529632	140511	PARALLELISM : NO
TRIGGER	0.000000	0.000000	DATABASE I/O	1:29.529632	140511	PCA RUP COUNT : N/A
			LOG WRITE I/O	0.000000	0	RUP AUTONOM.TX: N/A
CP CPU TIME	2:49.02071	2:48.90302	OTHER READ I/O	1:17.309059	19145	AUTONOMOUS TX : N/A
AGENT	2:49.02071	2:48.90302	OTHER WRTE I/O	0.528574	964	QUANTITY : 0
NONNESTED	2:49.02071	2:48.90302	SER.TASK SWTCH	0.005164	2	COMMITS : 1
STORED PRC	0.000000	0.000000	UPDATE COMMIT	0.000760	1	ROLLBACK : 0

AFTER

TIMES/EVENTS	APPL(CL.1)	DB2 (CL.2)	CLASS 3 SUSPENSIONS	ELAPSED TIME	EVENTS	HIGHLIGHTS
ELAPSED TIME	1:30:35.31	1:30:33.65	LOCK/LATCH(DB2+IRLM)	0.005413	865	THREAD TYPE : ALLIED
NONNESTED	1:30:35.31	1:30:33.65	IRLM LOCK+LATCH	0.002994	8	TERM.CONDITION: NORMAL
STORED PROC	0.000000	0.000000	DB2 LATCH	0.002419	857	INVOKE REASON : DEALLOC
UDF	0.000000	0.000000	SYNCHRON. I/O	1:21.613858	149888	PARALLELISM : NO
TRIGGER	0.000000	0.000000	DATABASE I/O	1:21.613858	149888	PCA RUP COUNT : N/A
			LOG WRITE I/O	0.000000	0	RUP AUTONOM.TX: N/A
CP CPU TIME	1:26:14.02	1:26:13.91	OTHER READ I/O	1:07.245210	14135	AUTONOMOUS TX : N/A
AGENT	1:26:14.02	1:26:13.91	OTHER WRTE I/O	0.421171	621	QUANTITY : 0
NONNESTED	1:26:14.02	1:26:13.91	SER.TASK SWTCH	2.682449	163	COMMITS : 1
STORED PRC	0.000000	0.000000	UPDATE COMM			ACK : 0

DB2 CPU time
Before : 2:48.90
After : 1:26:13.91

Case Study(2) – DB2 CPU Increase -2

Key Counters In Accounting

	Good case (V9)	Bad case (V10)
CL2 Elapsed Time	5:41.251	1:30:33.650
CL2 CPU Time	2:48.903	1:26:13.912
CL3 Suspension Time	2:37.384	2:31.968
Commit	1	1
SELECT	2	2
UPDATE / ROWs	1 / 1	1 / 1
OPEN / CLOSE	3 / 3	3 / 3
FETCH / ROWS	74 / 71	74 / 71
GETPAGES	6143825	6135464
Sync Read	140518	149888
Dyn. Prefetch	104050	440213
Pages Read Async	1607301	1610632

Case Study(2) CPU Increase -3

- Same DML activities, approx. same getpage
  - Indication of NO Access Path change
  - Plan table confirmed they are taking same access path
  - Dynamic prefetch change does not impact Class 2 CPU time
- Significant CPU increase without access path change
  - First, suspect monitoring issue
    - Try without monitoring product
    - This particular example was confirmed there was no CPU increase without monitoring product
  - If not, get a help from IBM

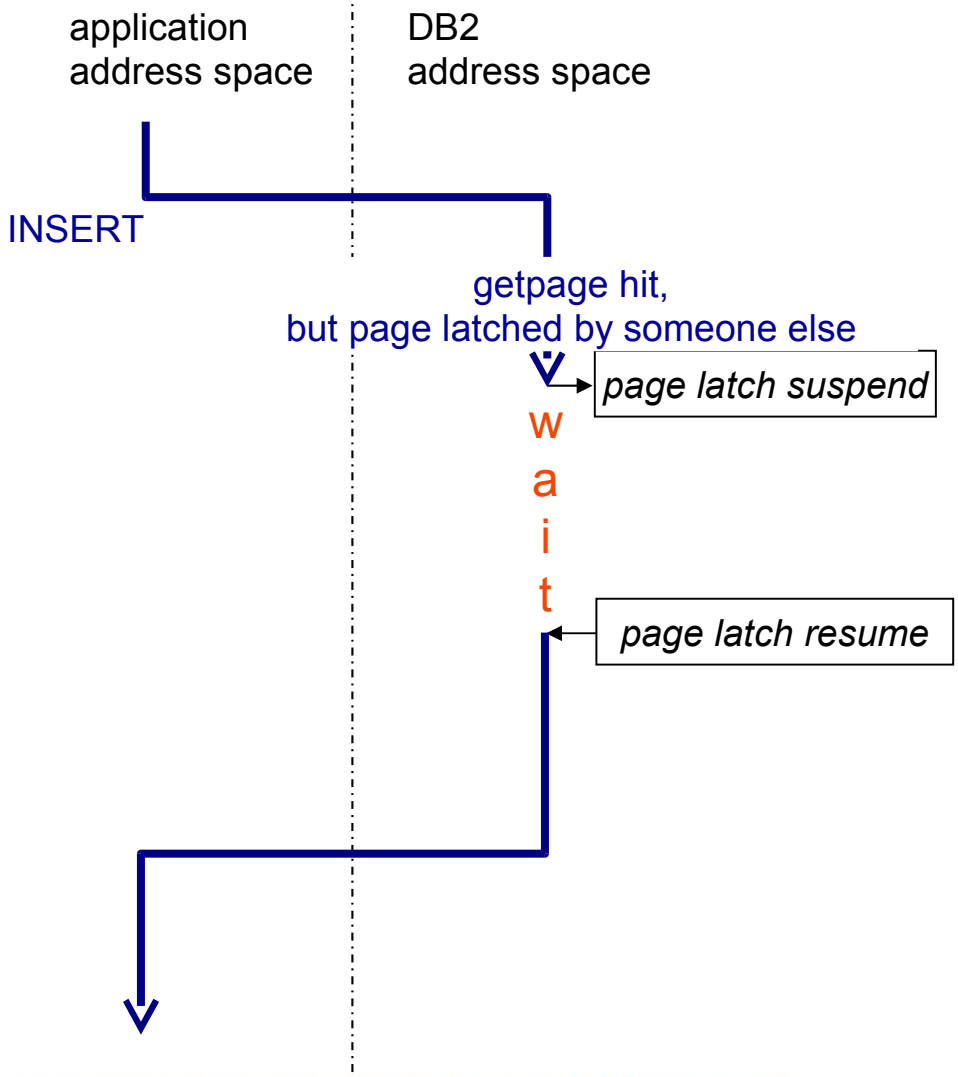
Case Study(4) High Page Latch Wait in Read Only Concurrent Batch Programs

Accounting Trace	Time in second
Class 1 Elapsed	3954
Class 2 Elapsed	3595
Class 2 CPU	1140
Class 3 Suspension	2283
DB2 Latch	94
Page Latch	2188

Accounting Trace	Count
SELECT	16327K
FETCH	25318K
COMMIT	10981
Getpages	125847K
Buffer Update	46633K

- What is page latch wait?
- Why are there Buffer Update with Read only transaction?

Page Latch Suspensions



What if …

Most often during high insert activity

- Space map page
  - Partitioning
  - MEMBER CLUSTER
- Data page
  - Randomize inserts
- Use IFCID 226, 227 for detail analysis

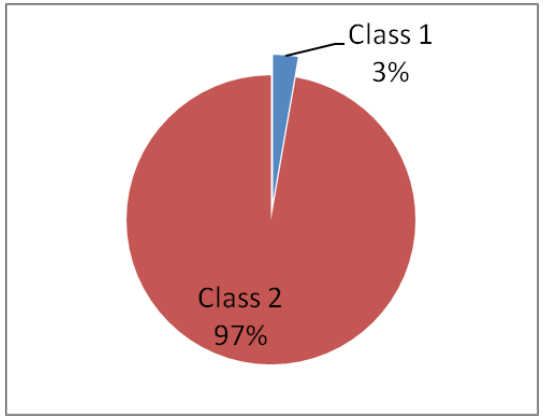
Case Study(4) High Page Latch with Read Only Concurrent Batches -2

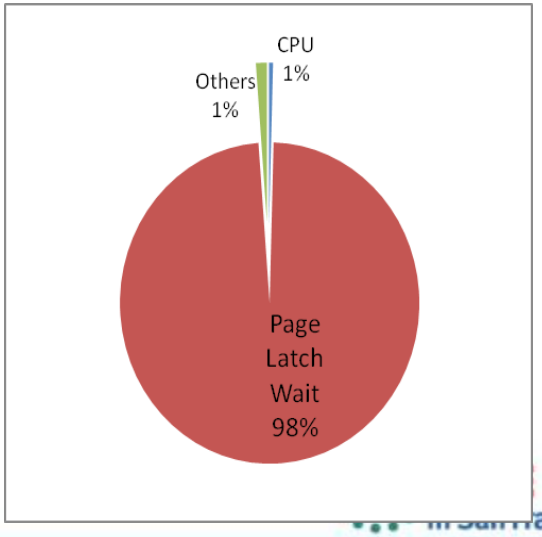
- Why are there buffer update in read-only plan?
  - Workfile activities due to sort operation
- Concurrent insert into sort workfiles
  - Space map contention if there are not enough workfile datasets
  - Many 32K workfiles but not enough 4K workfiles
    - Additional workfile table spaces to distribute workfile activities
    - Significant improvement once additional 4K workfiles are added

Case Study(5) Concurrent Insert Transaction -1

- Problem : 600 concurrent sequential inserts running slow
- Accounting shows,

Per transaction	Time in second
#of transaction	65651
Class 1 Elapsed	1.748
Class 2 Elapsed	1.700
Class 2 CPU	0.007
Class 3 Suspension	1.666
Page Latch	1.646
INSERT	1
Getpage	45





Case Study(5) Concurrent Insert Transaction -2

- High page latch contention using concurrent sequential insert
  - Detail page latch trace (IFCID 226/227) points out the table space's space map pages are suffering contention
    DB2PM LOCKING REPORT LEVEL(SUSPENSION)
    ORDER (DATABASE-PAGESET)
  - Large record size -> only 1-2 rows fit in the page
  - Contention for searching space vs. marking space map full

Page Latch Contention

		--SUSPEND REASONS--				---------- RESUME REASONS	
-- L O C K R E S O U R C E -		TOTAL	LOCAL	GLOB.	S.NFY	---- NORMAL ----	
TYPE	NAME	SUSPENDS	LATCH	IRLMQ	OTHER	NMBR	AET NMBR
PAGE	PAGE=X'000C4D05'	92055	0	0	0	92K	0.044785
	BPID=BP10		0	0	92055		
PAGE	PAGE=X'000C4223'	58299	0	0	0	58K	0.058537
	BPID=BP10		0	0	58299		

Space Map pages

- Larger page size (32K) should reduce frequency of space map update

Case Study(5) Concurrent Insert Transaction -3

- Changed the page size from 4K to 32K
- Accounting shows, significant performance improvement

Per transaction	4K page	32K page
#of transaction	65651	51833
Class 1 Elapsed	1.748	0.582
Class 2 Elapsed	1.700	0.026
Class 2 CPU	0.007	0.0009
Class 3 Suspension	1.666	0.024
Page Latch	1.646	0.020
INSERT	1	1
Getpage	45	5

Case Study(6) IDAA Queries Executed in Netezza

Accounting Trace	Time in second
Class 1 Elapsed	1399
Class 2 Elapsed	1396
Class 2 CPU	6.776
Class 3 Suspension	1389
Other Service Task Switch	1389
# Rows Fetched	1708862

Time spent in Accelerator

OMPE Support For Accelerator - Accounting

New ACCELERATOR section in Accounting

ACCELERATOR	TOTAL
OCCURRENCES	1
CONNECTS	1
REQUESTS	2
TIMED OUT	0
FAILED	0
SENT	
BYTES	4003
MESSAGES	11
BLOCKS	0
ROWS	0
RECEIVED	
BYTES	54684708
MESSAGES	24
BLOCKS	14
ROWS	1708861

ACCELERATOR	TOTAL
ELAPSED TIME	
SVCS TCP/IP	23:09.097443
ACCUM ACCEL	23:09.004780
CPU TIME	
SVCS TCP/IP	0.035621
ACCUM ACCEL	3:47.718000
WAIT TIME	
ACCUM ACCEL	0.040455
DB2 THREAD	
CLASS 1	
ELAPSED	23:19.330365
CP CPU	10.117196
SE CPU	0.000000
CLASS 2	
ELAPSED	23:16.087773
CP CPU	6.776217
SE CPU	0.000000

OMPE Support For Accelerator – Statistics

New ACCELERATOR section in Statistics

TEST1 ACCELERATION	QUANTITY
QUERIES SUCCESSFULLY EXECUTED	27.00
QUERIES FAILED TO EXECUTE	0.00
ACCELERATOR IN INVALID STATE	0.00
CURRENTLY EXECUTING QUERIES	0.99
MAXIMUM EXECUTING QUERIES	61.00
CONNECTS TO ACCELERATOR	28.00
REQUESTS SENT TO ACCELERATOR	55.00
TIMED OUT	0.00
FAILED	0.00
BYTES SENT TO ACCELERATOR	82902.00
BYTES RECEIVED FROM ACCELERATOR	21230912161.00
MESSAGES SENT TO ACCELERATOR	308.00
MESSAGES RECEIVED FROM ACCEL	5355.00
BLOCKS SENT TO ACCELERATOR	0.00
BLOCKS RECEIVED FROM ACCELERATOR	5085.00
ROWS SENT TO ACCELERATOR	0.00
ROWS RECEIVED FROM ACCELERATOR	174039273.00

TEST1 CONTINUED	QUANTITY
AVG QUEUE LENGTH (LAST 3 HRS)	0.00
AVG QUEUE LENGTH (LAST 24 HRS)	0.00
MAXIMUM QUEUE LENGTH	2.00
AVG QUEUE WAIT ELAPSED TIME	0.071908
MAX QUEUE WAIT ELAPSED TIME	2.570000
WORKER NODES	12.00
WORKER NODES AVG CPU UTILIZATION (%)	29.40
COORDINATOR AVG CPU UTILIZATION (%)	3.04
DISK STORAGE AVAILABLE (MB)	33557183.98
IN USE (%)	34.88
IN USE FOR DATABASE (MB)	2568298.00
DATA SLICES	92.00
DATA SKEW (%)	1.68
PROCESSORS	96.00

Summary

- Sources of key performance indicators
  - DB2 Statistics and Accounting
  - RMF Reports
- Use DB2 traces to make an informed tuning
  - Top down analysis
  - Where does problem transaction spend time?
    - CPU vs. suspension time
- Case Studies using OMPE batch reporting
  - Problem diagnosis
  - Performance tuning
  - Performance monitoring

Thank you !

- Questions?
- email : Akiko Hoshikawa, akiko@us.ibm.com

Disclaimers:

Availability. References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates.

The workshops, sessions and materials have been prepared by IBM or the session speakers and reflect their own views. They are provided for informational purposes only, and are neither intended to, nor shall have the effect of being, legal or other guidance or advice to any participant. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it is provided AS-IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other materials. Nothing contained in this presentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.