

IDUG® 2003 – Europe



DB2 Insights for
Industries in Motion



Finance • Telecommunications • Medical
Manufacturing • Transportation • Distribution
Insurance • Government • Commerce

DB2 Universal Database for
z/OS and OS/390
Intermediate Level

**I Know How to Tune
SQL, What SQL Do I
Tune?**

October 9, 2003 13:30 – 14:30
Session F11

Tom Moulder
TREX Associates, Inc.

INTERNATIONAL
DB2 USERS GROUP



Independent • Not-for-Profit • User Run

What will we discuss

- Introduction
 - Pareto Principle
- Data Sources
 - z/OS
 - IMS or CICS
 - Native DB2
 - Third Party
- Data Analysis
 - Correlation
 - Drill Down
- Tuning SQL
 - Static
 - Dynamic
- Conclusion

Introduction

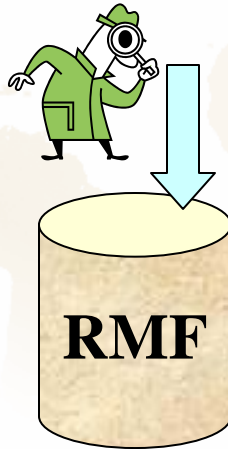
- Pareto Principle Applied to a Tuning Methodology
 - What is the Pareto Principle?
 - The 80/20 Rule
 - 80% of the benefit will probably come from 20% of the tuning
 - Its Application to SQL Tuning
 - Small Number of SQL Statements
 - Large Amount of Resource Savings from Tuning
 - The Dilemma
 - How do I Find the Small Number of SQL Statements That will Yield Large Savings

Introduction

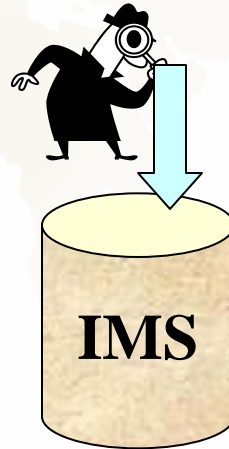
- Searching for Answers
 - Why?
 - Tight Budgets
 - Fewer Hardware Upgrades
 - Service Level Agreements
 - When
 - Reactively, After a Problem Incident
 - Proactively, During a Version Change or a New Application Install
 - Proactively as a planned project to search for cost savings
 - Who and How ...

Introduction - Who

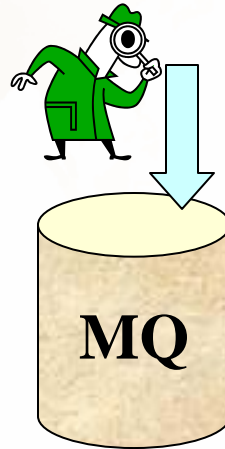
MVS
Tuner



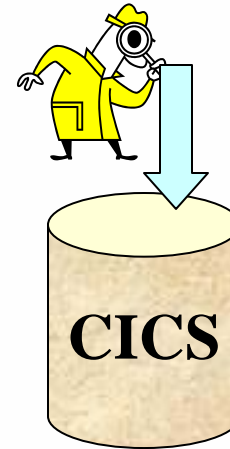
IMS
Sys Prog



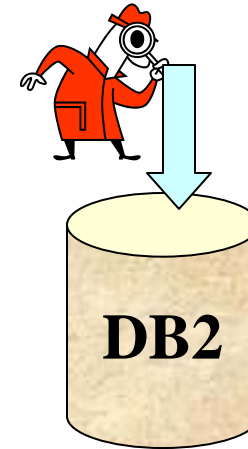
Websphere/MQ
Series Sys Prog



CICS
Sys Prog



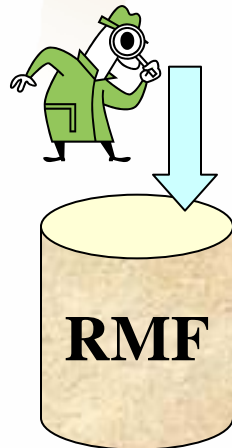
DBA



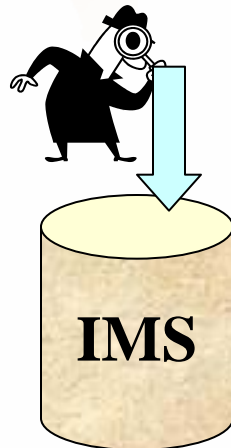
Introduction - How

Vertical Analysis

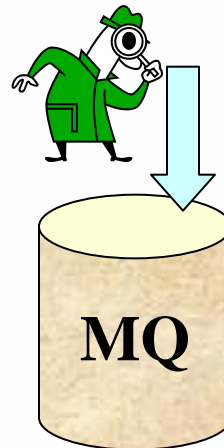
MVS
Tuner



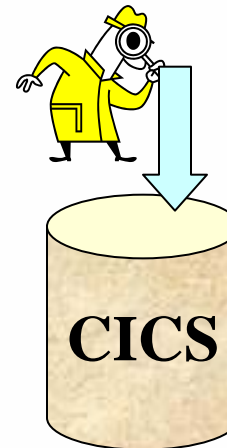
IMS
Sys Prog



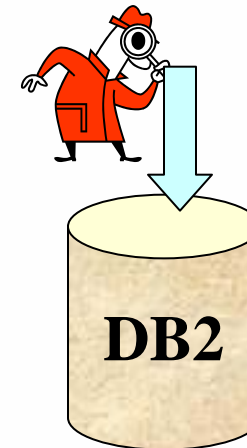
Websphere/MQ
Series Sys Prog



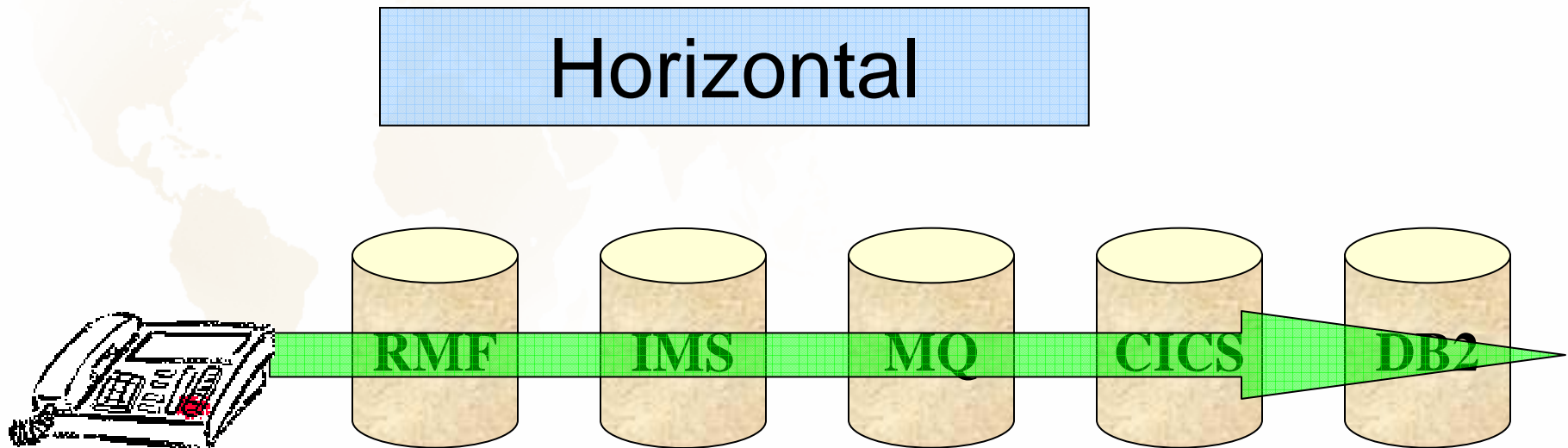
CICS
Sys Prog



DBA



Introduction – User’s Perspective



Users do not see silos, they see functionality and response time. To support their application our view must be horizontal and encompass whatever silos make up the application they are using.

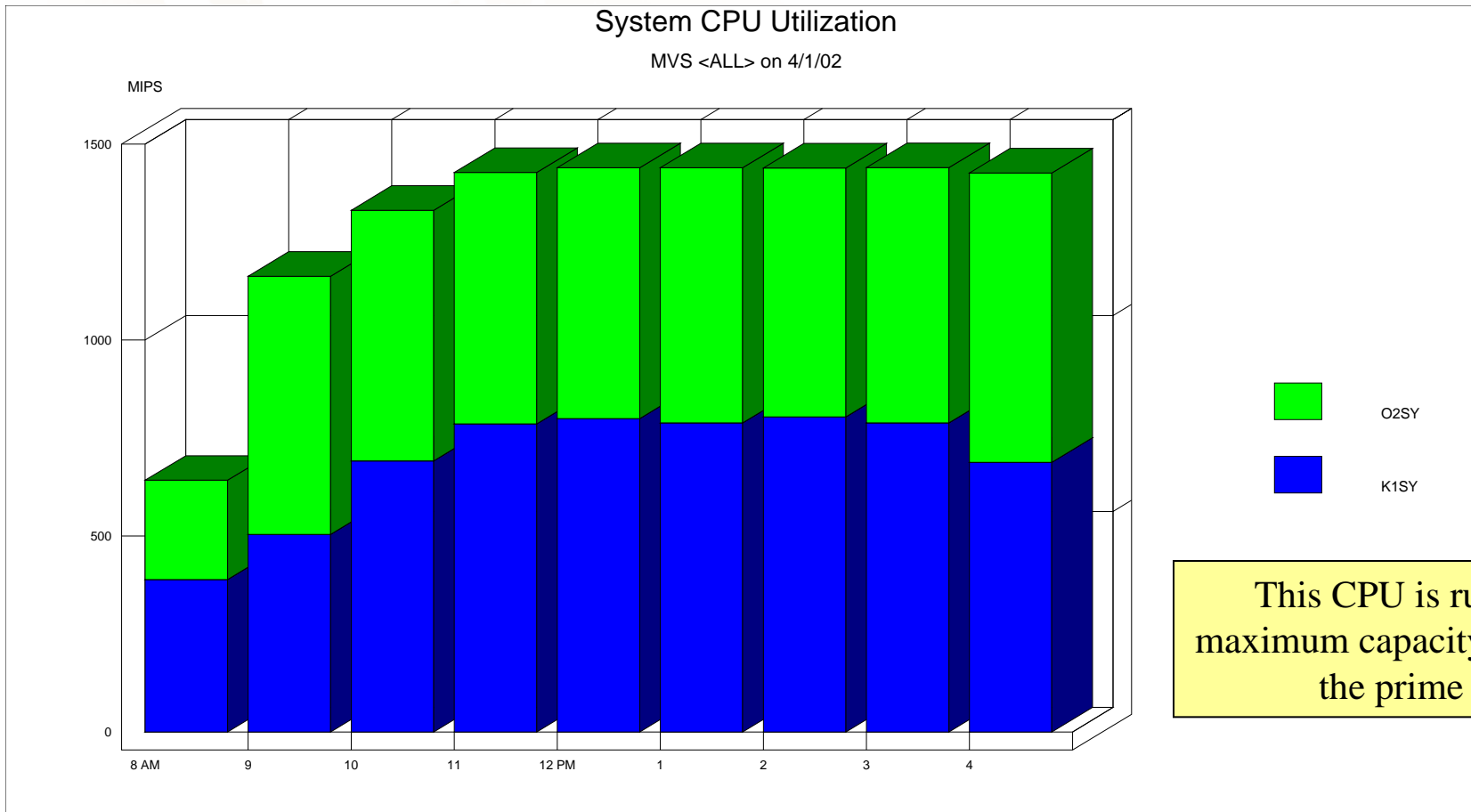
Introduction

- Tuning Perspective
 - Not Just DB2
 - Affects Many Others
- Tuning Justification
 - Business Leverage
 - Competitive Advantage
 - Through Better Response Times
 - Fewer Resources to meet Business Needs

Data Sources

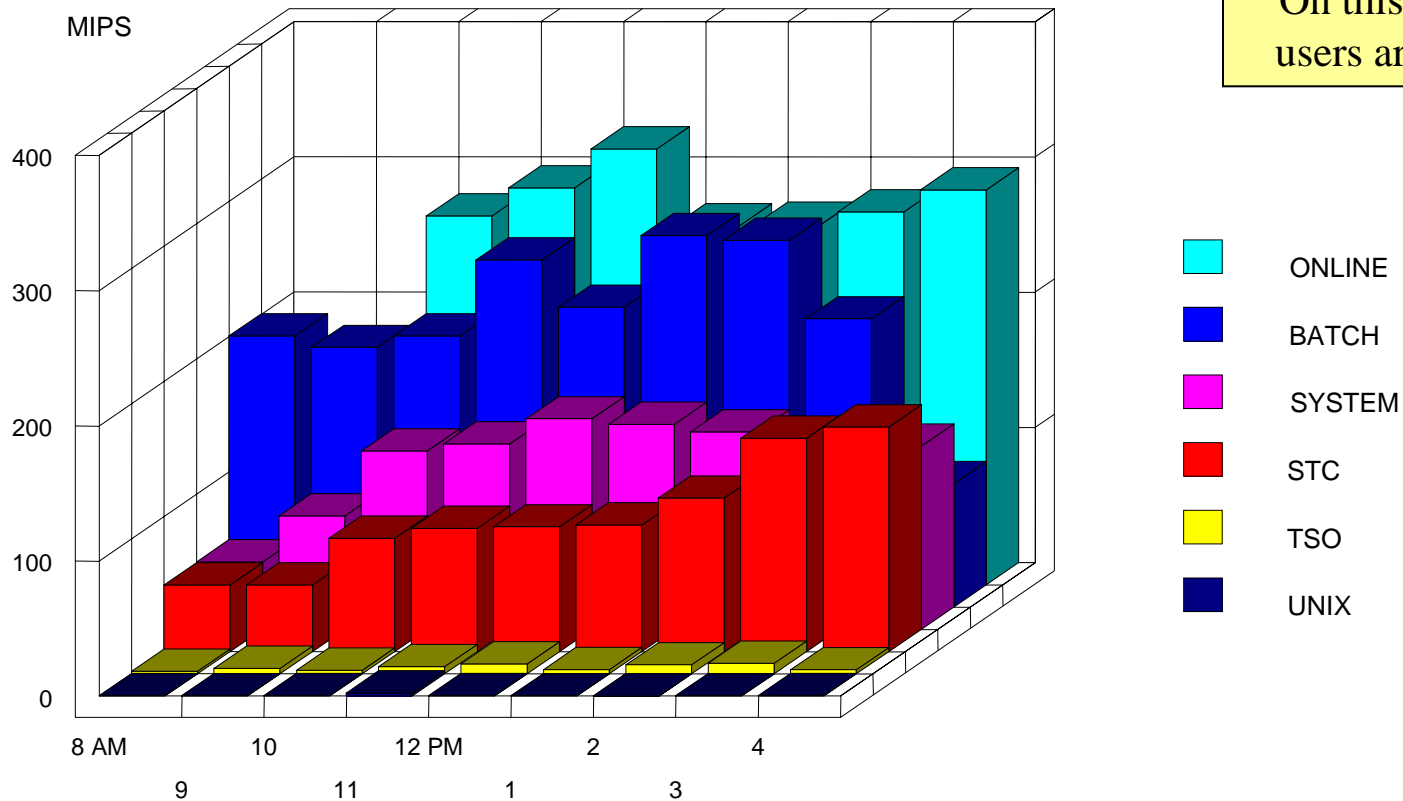
- z/OS
 - RMF
 - Type 70 Records
 - SMF
 - Type 30 Records
- IMS or CICS
 - CICS Type 110
 - IMS - Mainview FA/F9 Records
- Native DB2
 - Accounting Records (100, 101)
 - Thread
 - Package
- Third Party
 - SQL statement Accounting

Data Sources - RMF



Data Sources - SMF

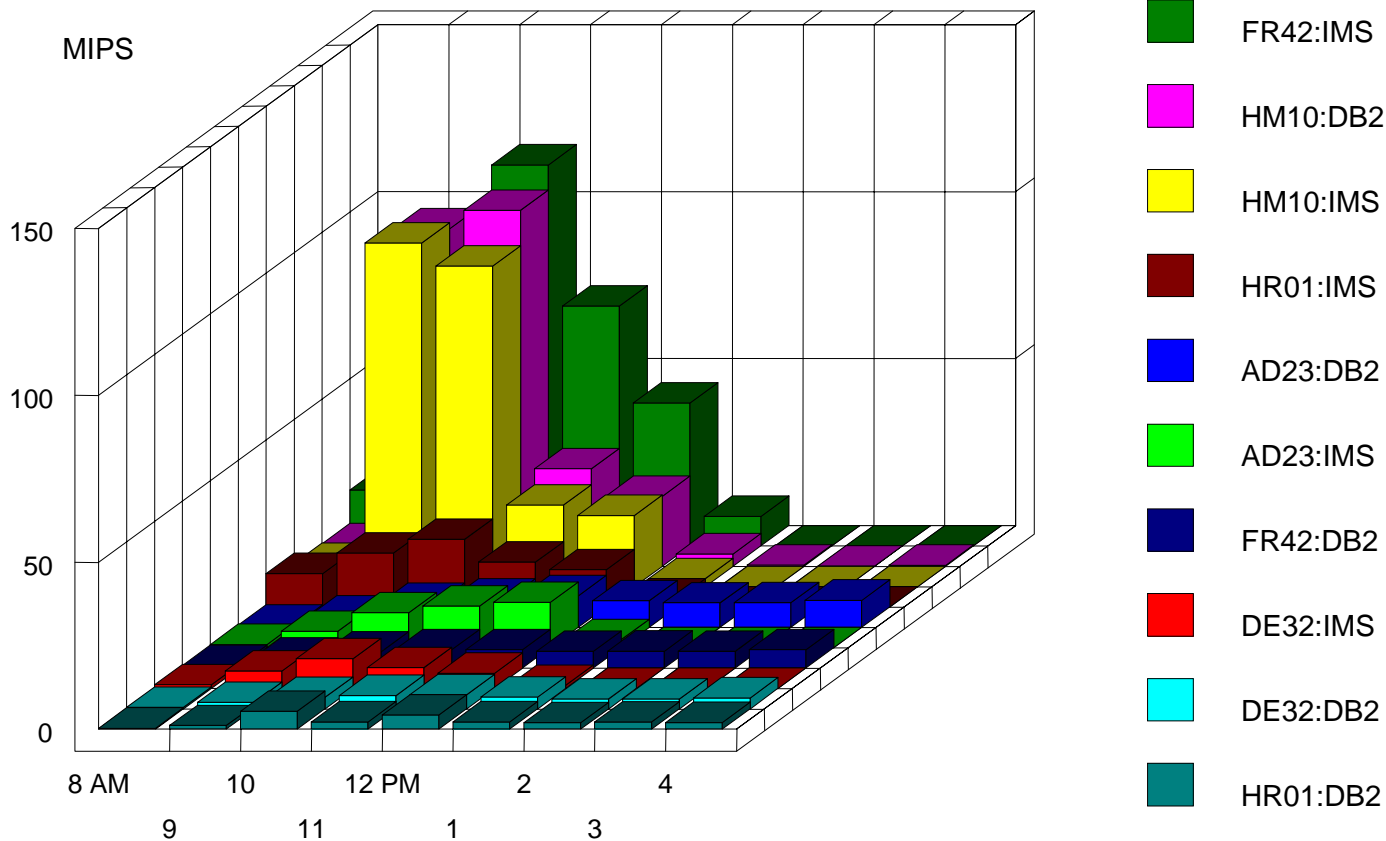
Workload Category CPU Utilization
MVS S1KY Workload Category <ALL> on 4/1/2002



On this system the primary users are Online and Batch.

Data Sources - IMS

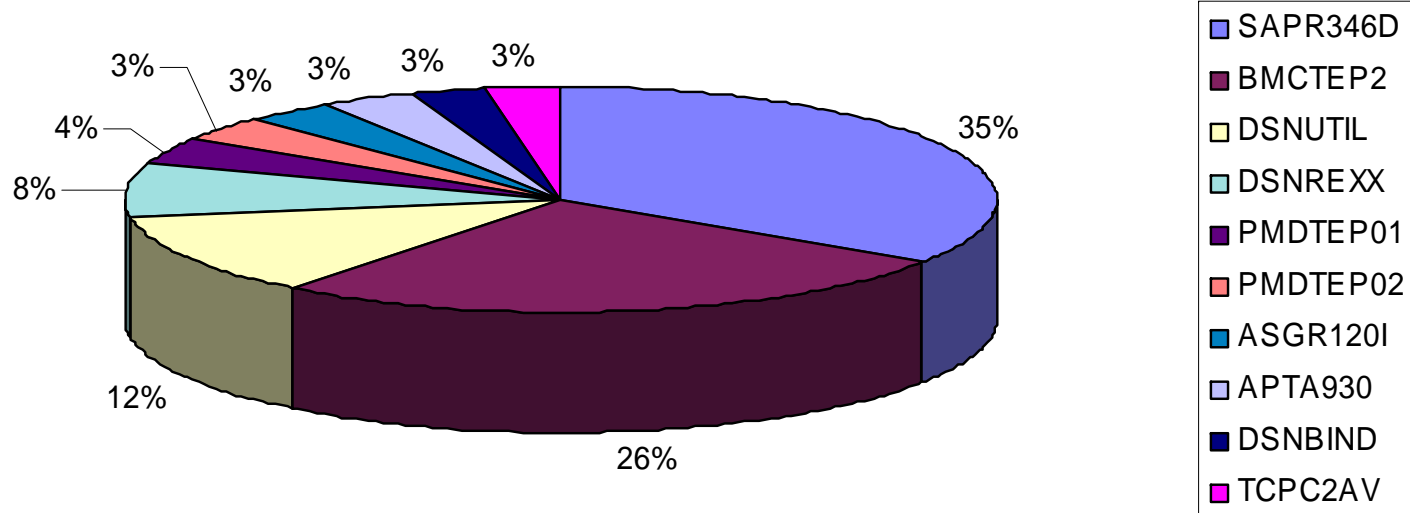
CPU Utilization by Application/Type
for [GROUP_7] on 4/1/2002



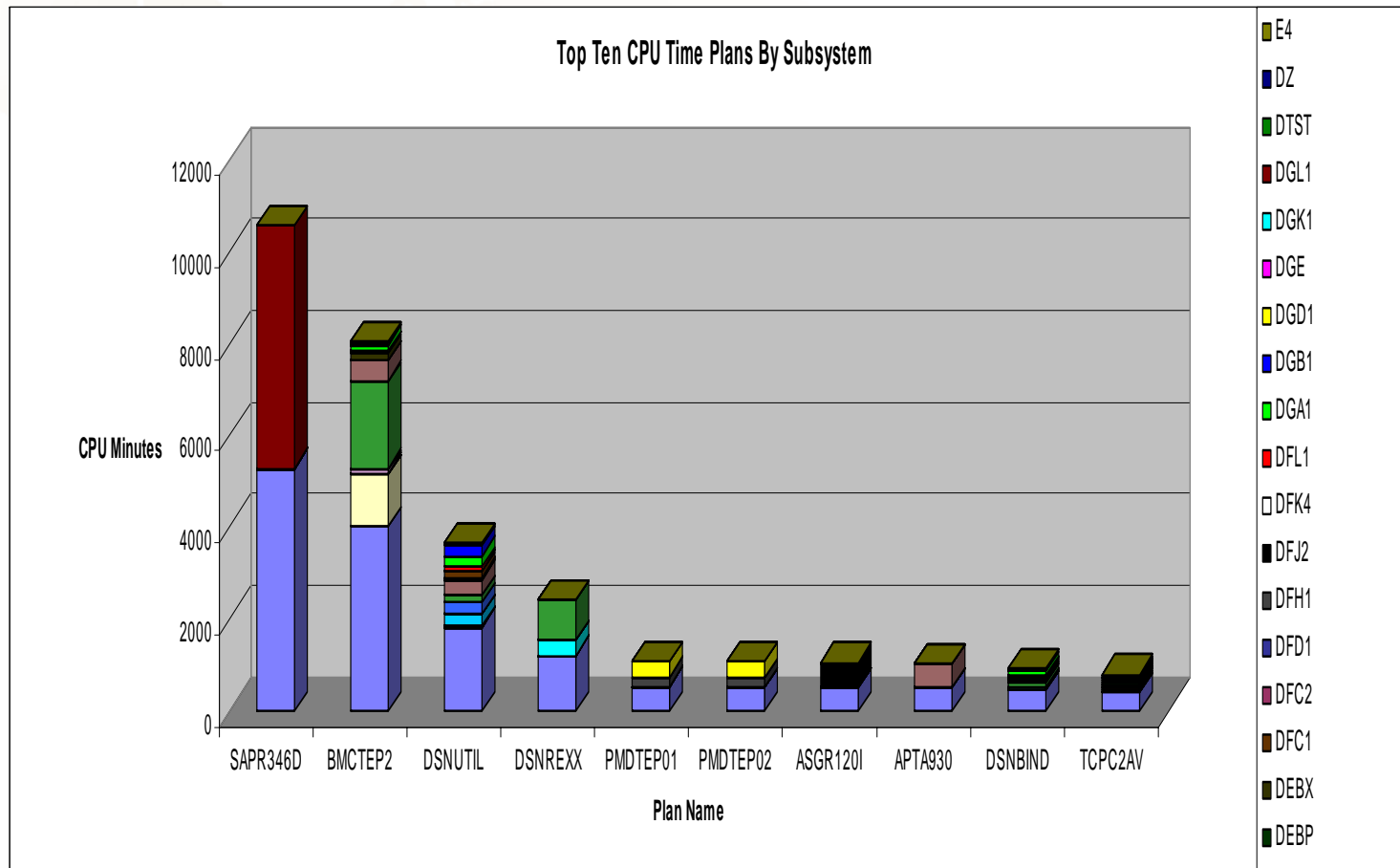
This represents the IMS applications that run within the Online workload from the previous graph. The utilization is broken down by the IMS component and the DB2 component.

Data Sources – Native DB2

Top Ten Plans by CPU Time



Data Sources – Native DB2



Data Sources – Third Party

1 - Default (sysp.bmc.com)

File Edit Transfer Fonts Options Macro View Window Help

PR1 PA2 PA3 ENT CLR

DOMEVLGW/P LINE 1 OF 132
 Command =====> Scroll ==> PAGE

SQMCACTX SQL STATEMENT ANALYSIS (DATA) Logged : 04/05/2002 15:26
 =====> More: + >

BMCSftwr.SQMCACTX -- SQL STATEMENT ANALYSIS (DATA) -- 04/05 15:26:36
 Source : DC01-ACTIVE Intvl : 04/05 00:00 - UNLIMITED More: + >

----->

Actions: D-DB2 P-Plan C-ConnID S-SQL Text H-Header L-CorrID R-Recommend
 G-AppGrp U-User T-Detail E-SQL Error Q-CatSQL O-Object X-Exception
 A-Average

Subsys: CBOA AppGrp: User: ConnID:
 Plan: BM10D17 CorrID:

Program	Stmt Type	Call Type	Sect No.	Stmt No.	SQL Calls	+----- Elapsed	Total	IN-SQL % CPU	Time	----- %
+ BM10D143	STATIC	OPEN	3	639	1416	40:54.34637	95.5%	26:20.16461	98.	
+ BM10D17S	STATIC	FETCH	42	7803	1624	00:18.14590	0.7%	00:00.43710	0.	
+ BM10D17S	STATIC	FETCH	5	3428	1628	00:14.33953	0.6%	00:01.77618	0.	
+ BM10D17S	STATIC	FETCH	7	3677	2223	00:06.55990	0.3%	00:00.47573	0.	
+ BM10D17S	STATIC	OPEN	32	6546	1628	00:06.51832	0.3%	00:00.74842	0.	
+ BM10D17S	STATIC	SELECT	2	3213	3023	00:06.50099	0.3%	00:02.17266	0.	
+ BM10A183	STATIC	INSERT	3	912	335	00:06.21319	0.2%	00:00.21632	0.	
+ TIRPROFD	STATIC	OPEN	2	447	2898	00:06.09540	0.2%	00:02.00241	0.	
+ BM10A251	STATIC	INSERT	3	838	733	00:05.44824	0.2%	00:00.52666	0.	
+ BM10D17S	STATIC	FETCH	13	4342	1630	00:05.10277	0.2%	00:00.38603	0.	
+ BM10D141	STATIC	INSERT	2	649	1416	00:04.36032	0.2%	00:00.35829	0.	

F1=Help F2=Keys F3=Exit F4=Terse F5=RFind F6=Filter
 F7=Backward F8=Forward F9=Retrieve F10=Left F11=Right F12=Cancel

4-© | 1 Sess-2 172.17.4.175 | SYSP0007 | 11/10

Start | Microsoft PowerPoint - [Lo... | Inbox - Microsoft Outlook | Microsoft Access - [CPAP... | 1 - Default (sysp.bmc... | 4:29 PM

Data Analysis – Correlation

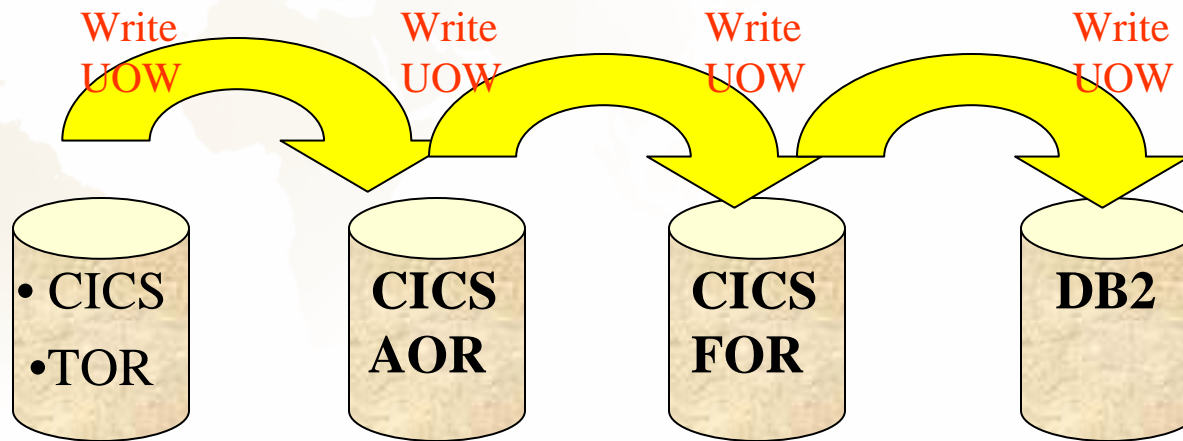
- Know your applications
 - Payroll uses
 - Windows Client
 - Websphere Application Server
 - CICS Transactions that start with PR*
 - DB2 plans and tables owned by PR
 - Billing uses
 - IMS transactions and databases that start with BI
 - DB2 plans and tables that are owned by BI
- Connect all the pieces of technology that affect each application
- Some are simple, some very complex

Data Analysis – Correlation Joining the Silos Together

- The Unit of Work timestamp (UOW)
- Unique descriptor originally used to link CICS transactions from the TOR to the AOR and from the AOR to the FOR in the SMF 110 records created by the CICS Monitor Program (CMP)
- Now, if turned on, can pass the unique identifier to DB2 for each use of a thread.
 - CICS Rel 4.1 RCT -- Tokeni
 - CICS Rel 5.1+ RDO -- AccountRec
- For some, it may make sense to use the DB2 correlation ID to link back to the transaction. However, this will take additional work.

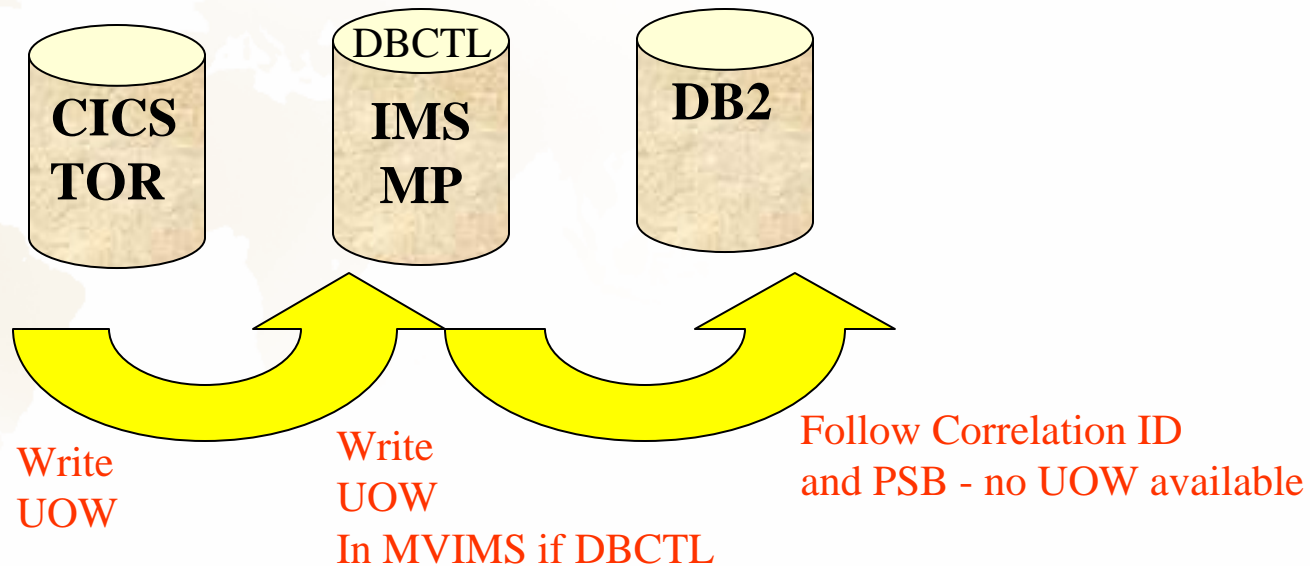
Data Analysis – Correlation

Joining the Silos Together



CICS to DB2 Applications

Data Analysis – Correlation Joining the Silos Together



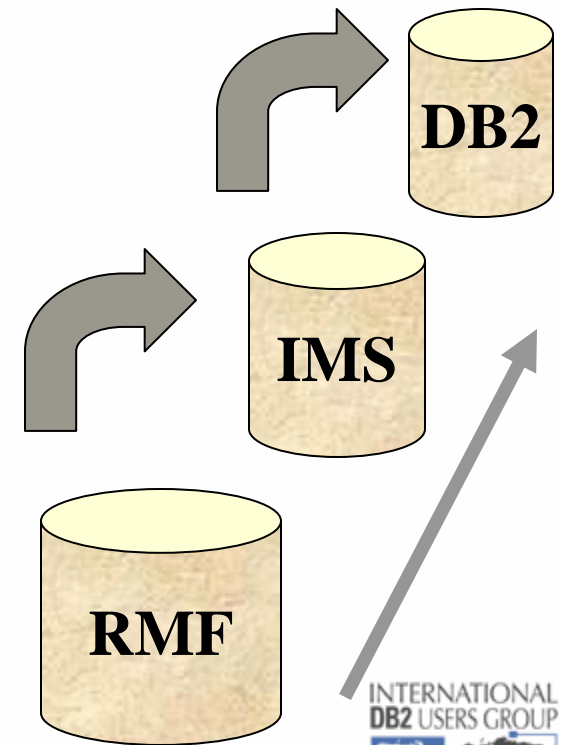
Applications with IMS Involved

Data Analysis – Drill Down

- Overview of the Drill Down Process
 - Start at the Service Class or Report Class level
 - Use performance groups if you have not gone to goal mode
 - Go to the “Biggest” Subsystem
 - Go to the “Biggest” Application
 - Go to the DB2 SQL for that Application
 - Tune it and return to the top
- Look upon this as a program loop
- Proactive approach
- Uses the Pareto Principle

Data Analysis – Drill Down

- An Analogy from World War II
- Island Hopping
 - Purpose was to fight only the battles needed to insure victory
 - Why? To save lives and resources
 - It was focused on the critical path
 - It resulted in Victory
- Apply this to Performance Tuning
- SILO Hopping



Data Analysis – Drill Down

An Example

Q: What Service Class/Performance Group uses the most CPU?

A: Onlines

Q: What Online Regions use the most CPU?

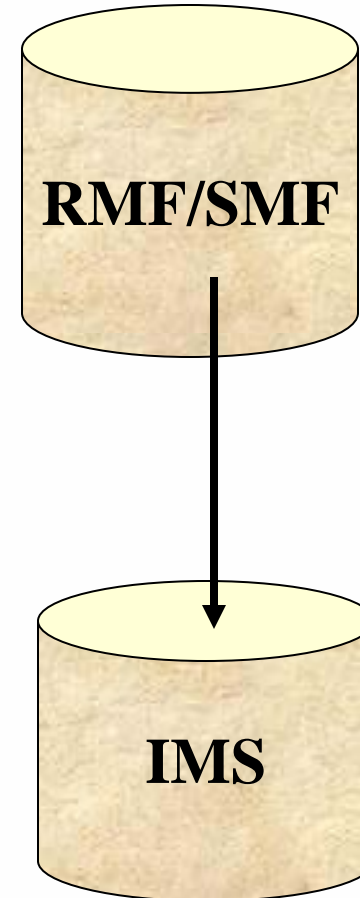
A: IMSxxxxx

Q: What transactions in IMSxxxxx use the most CPU?

A: BM10

Q: Why does BM10 use this much CPU

A: DB2



Data Analysis – Drill Down

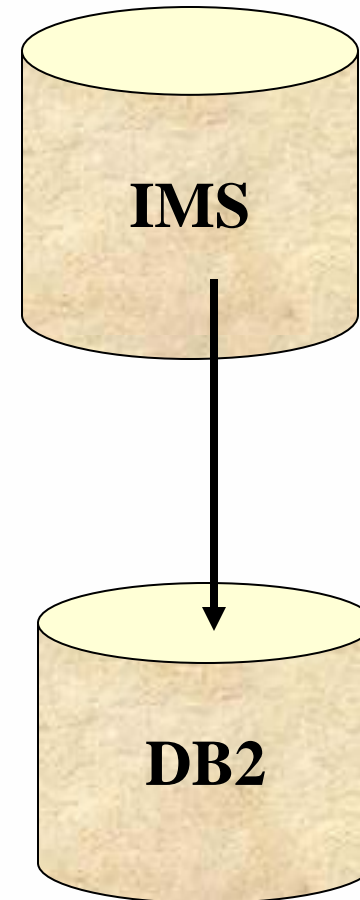
An Example

Q: What plan ID does BM10 use?

A: BM10D17

Q: What SQL does BM10D17 use?

A: See next page.....



Data Analysis – Drill Down

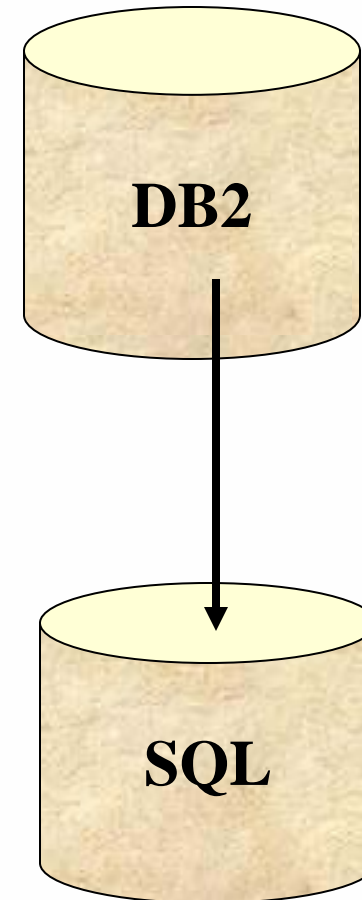
An Example

Q: What SQL statement uses the most CPU?

A: “Open”

Q: What is the “Open” SQL doing?

A: Joining two tables using one index and creating a Cartesian product.



Data Analysis – Drill Down

An Example

Q: Does the access match the index?

A: No

Q: Can the access use another index?

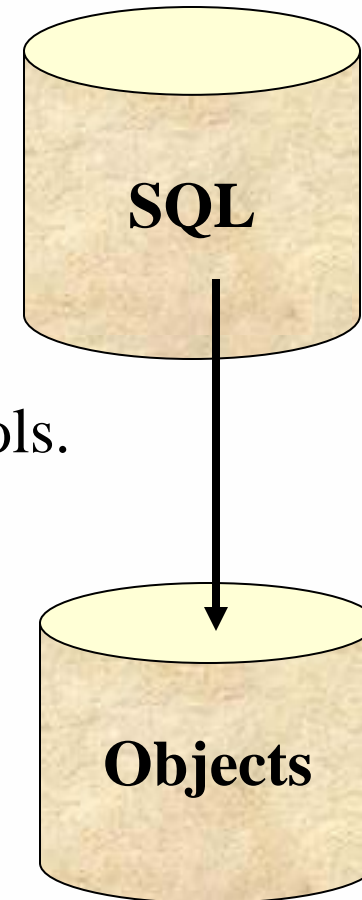
A: Yes, one that matches the requested cols.

Q: Can the DBA change the index?

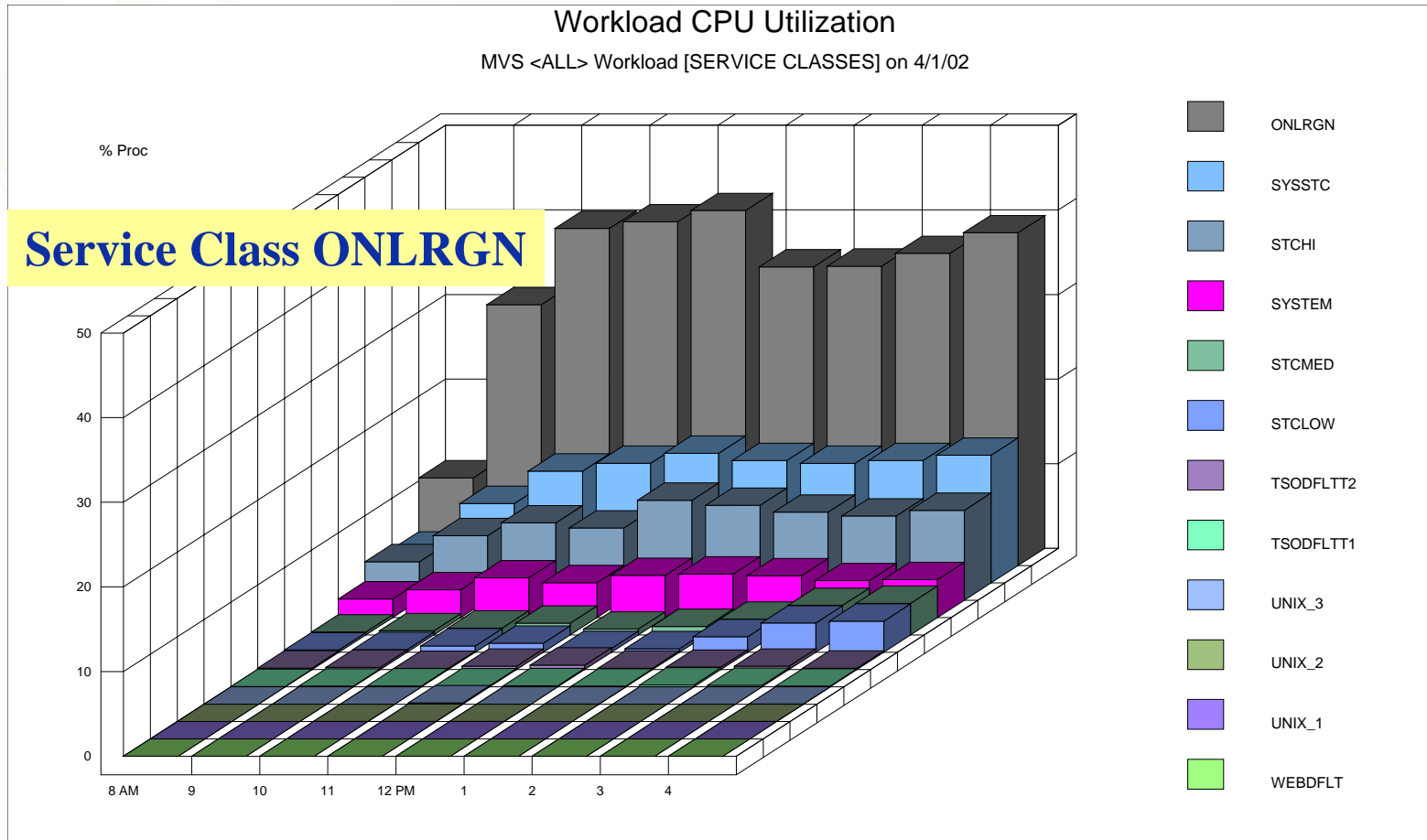
A: Yes

Q: What was the impact?

A: 30% Savings

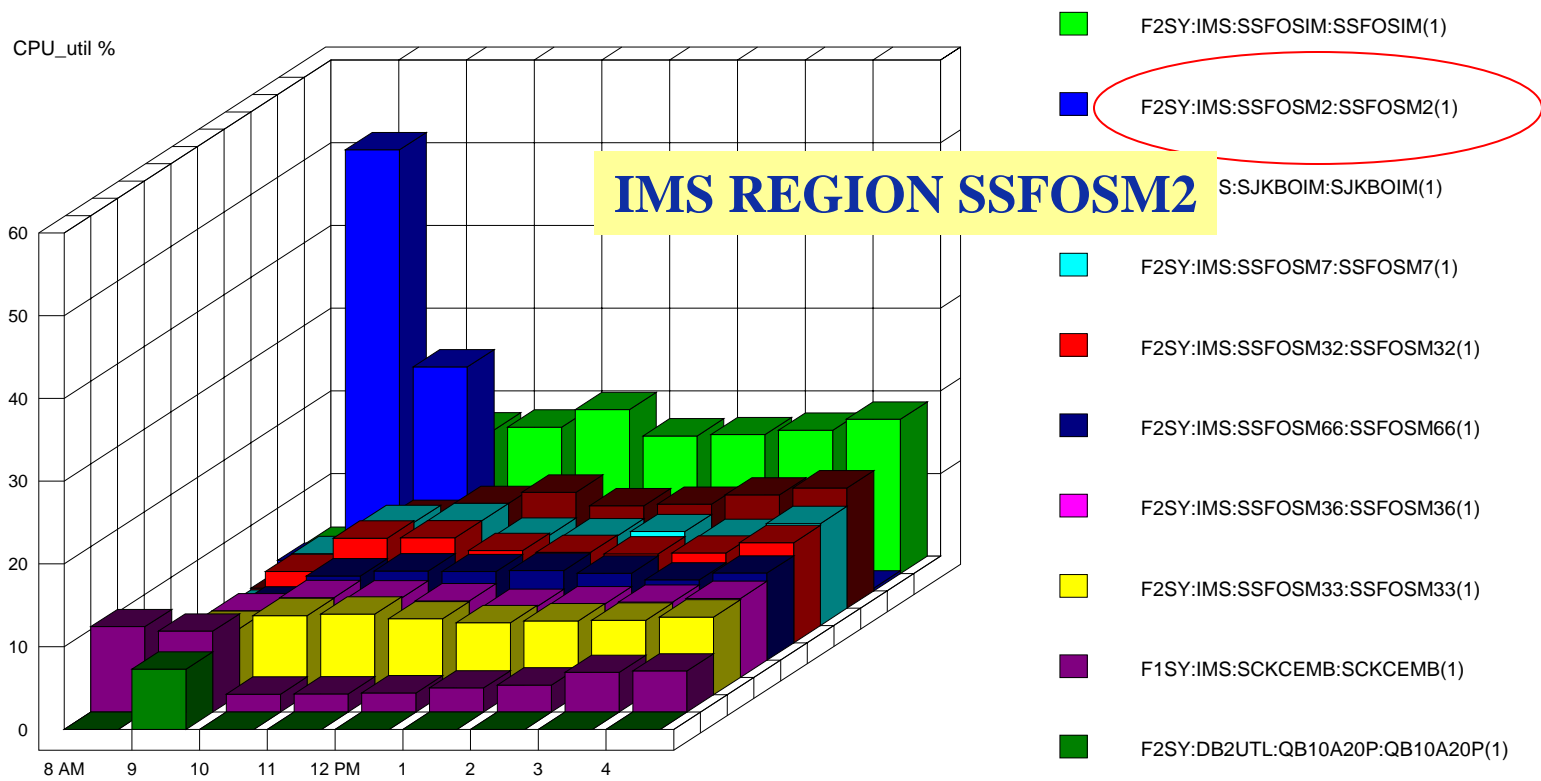


Data Analysis – Drill Down



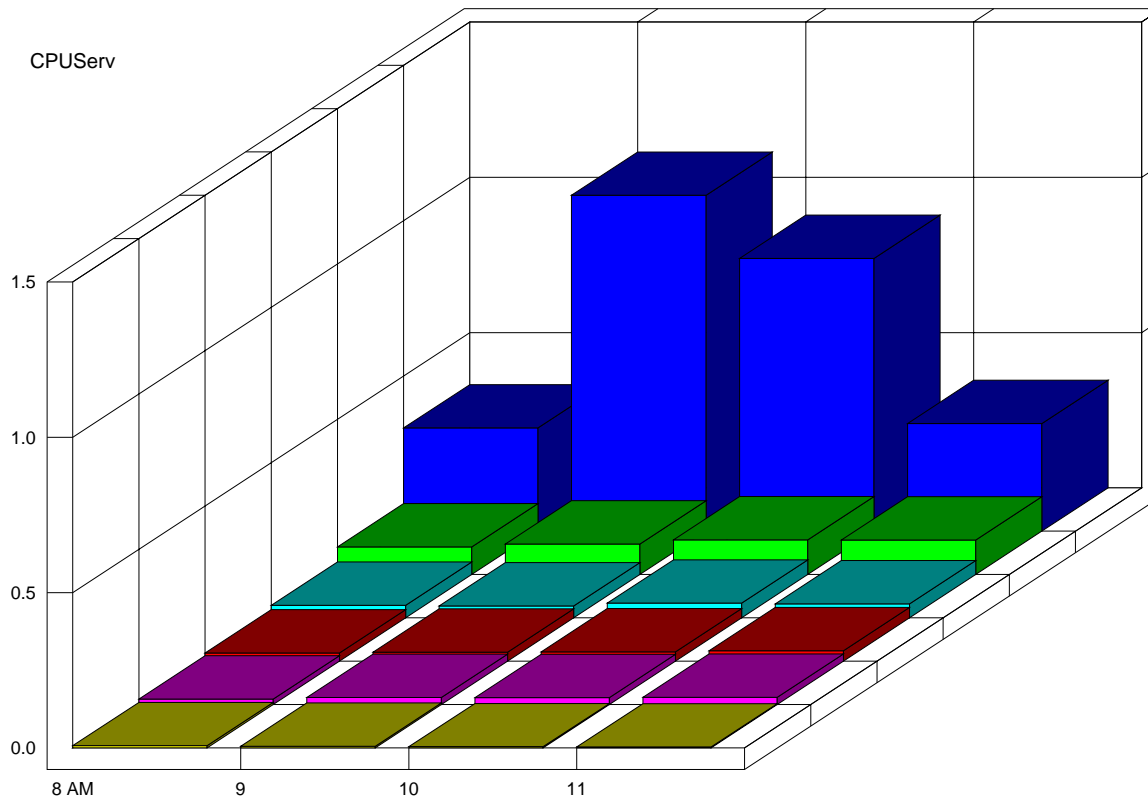
Data Analysis – Drill Down

Selected Subsystem Address Space Data
for <ALL> on 4/1/02



Data Analysis – Drill Down

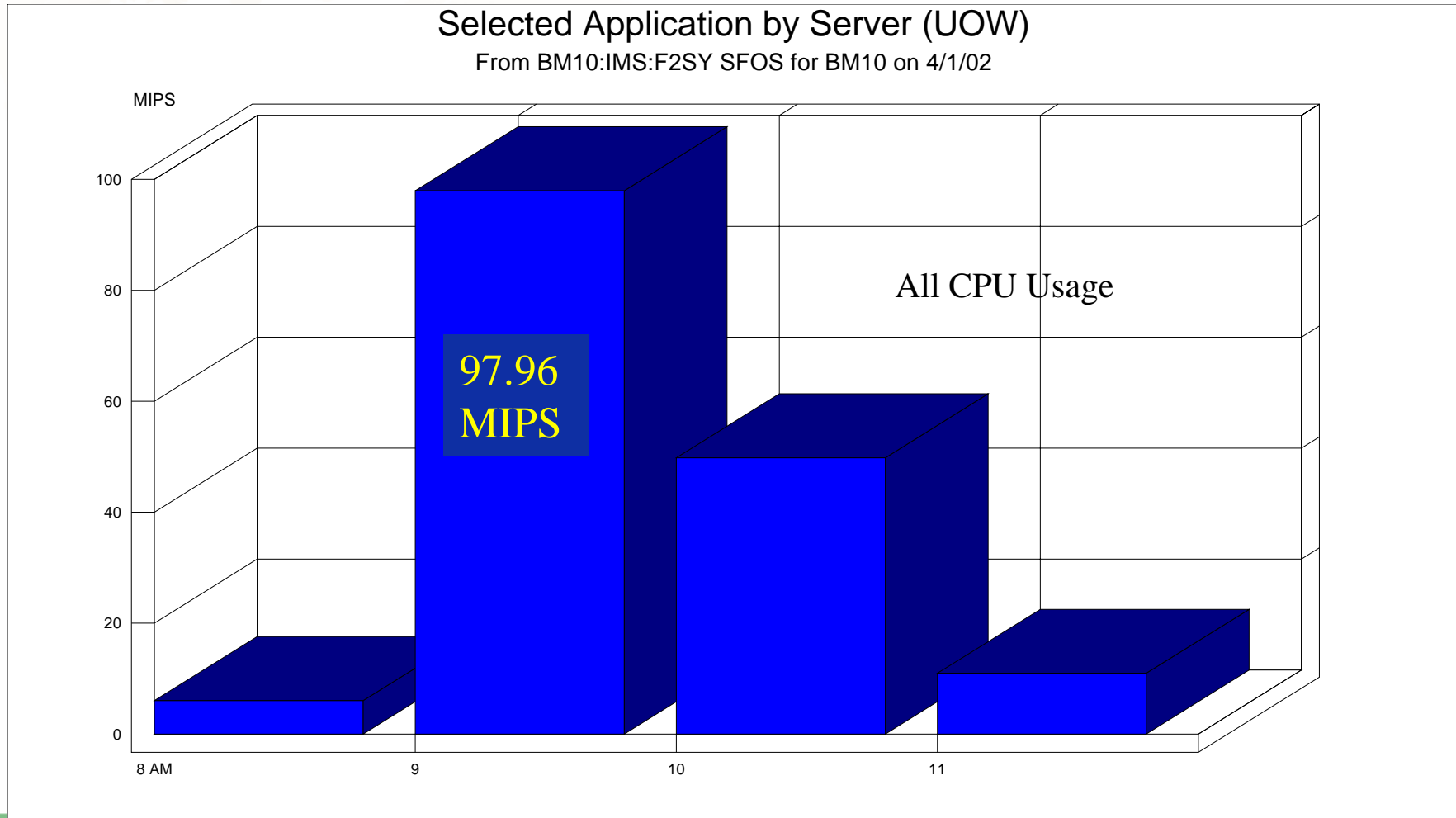
Selected IMS Application by Region
From F2SY SFOS Tran CPU service



IMS Tran BM10

- BM10
- AF42
- LQBMP
- CS32
- SFOSMPP
- RH01

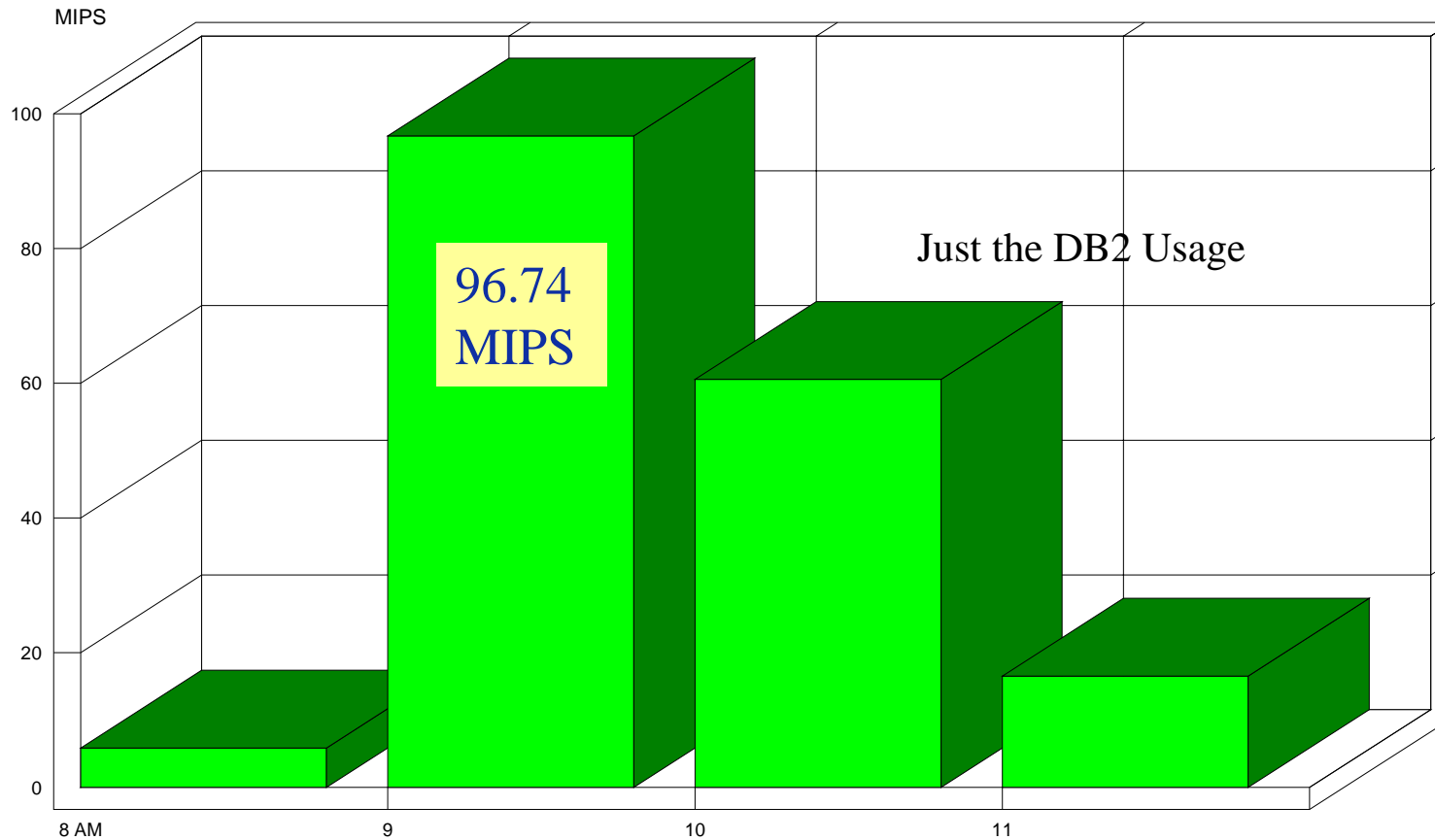
Data Analysis – Drill Down



Data Analysis – Drill Down

Selected Application by Server (UOW)

From F2SY CBOA -- BM10 usage



Tuning the SQL - Static

- Drilling Down Involves Coordination and Multiple People
 - RMF analysis from the Systems Programmer
 - CICS/IMS from a Systems Administrator
 - DB2 from Systems Administrator and DBA
- Previous Drill Down provides the DB2 DBA with
 - Plan that uses the most resources
 - Focused place to start tuning
- Still work to do
 - Plan could have lots of statements
 - How to handle Static as well as Dynamic statements

Tuning the SQL - Static

1 - Default (sysp.bmc.com)

File Edit Transfer Fonts Options Macro View Window Help

PA1 PA2 PA3 ENT CLR

DOMEVLGW/P LINE 1 OF 132
 Command =====> Scroll ==> PAGE

SQMCACTX SQL STATEMENT ANALYSIS (DATA) Logged : 04/05/2002 15:26
 ----- More: + >

NT ANALYSIS (DATA) -- 04/05 15:26:36
 5 00:00 - UNLIMITED More: + >

D S-SQL Text H-Header L-CorrID R-Recommend
 l E-SQL Error Q-CatSQL O-Object X-Exception

User: ConnID:
 CorrID:

tmt SQL +----- Total IN-SQL Time -----
 No. Calls Elapsed % CPU %

+	BM10D143	STATIC	OPEN	3	639	1416	40:54.34637	95.5%	26:20.16461	98.
+	BM10D17S	STATIC	FETCH	42	7803	1624	00:18.14590	0.7%	00:00.43710	0.
+	BM10D17S	STATIC	FETCH	5	3428	1628	00:14.33953	0.6%	00:01.77618	0.
+	BM10D17S	STATIC	FETCH	7	3677	2223	00:06.55990	0.3%	00:00.47573	0.
+	BM10D17S	STATIC	OPEN	32	6546	1628	00:06.51832	0.3%	00:00.74842	0.
+	BM10D17S	STATIC	SELECT	2	3213	3023	00:06.50099	0.3%	00:02.17266	0.
+	BM10A183	STATIC	INSERT	3	912	335	00:06.21319	0.2%	00:00.21632	0.
+	TIRPROFD	STATIC	OPEN	2	447	2898	00:06.09540	0.2%	00:02.00241	0.
+	BM10A251	STATIC	INSERT	3	838	733	00:05.44824	0.2%	00:00.52666	0.
+	BM10D17S	STATIC	FETCH	13	4342	1630	00:05.10277	0.2%	00:00.38603	0.
+	BM10D141	STATIC	INSERT	2	649	1416	00:04.36032	0.2%	00:00.35829	0.

F1=Help F2=Keys F3=Exit F4=Terse F5=RFind F6=Filter
 F7=Backward F8=Forward F9=Retrieve F10=Left F11=Right F12=Cancel

4-© | 1 Sess-2 172.17.4.175 | SYSP0007 | 11/10

Start | Microsoft PowerPoint - [Lo... | Inbox - Microsoft Outlook | Microsoft Access - [CPAP... | 1 - Default (sysp.bmc... | 4:29 PM

Program BM10D143 is by far.
With 95% of the total SQL CPU time



Tuning the SQL - Static

And the Text is ...

```
DECLARE CUR_1234567890_1 CURSOR FOR
SELECT BM1028TB01 . "LINE_NUMBER" ,
       BM1028TB01 . "FK_BM1020TBNUMBER" ,
       BM1020TB02 . "NUMBER"
FROM   "BM1028TB" BM1028TB01 ,
       "BM1020TB" BM1020TB02
WHERE  BM1028TB01 . "FK_BM1020TBNUMBER" = : H
ORDER BY 1 DESC
```

Table BM1028TB01 has an index on column
FK_BM1020TBNUMBER

This table contains 300,000 Rows approximately

Table BM1020TB02 has no index on column NUMBER

This table contains 150,000 Rows approximately

Tuning the SQL - Static

- Index added for BM1020TB02 Column NUMBER
- CPU Time Savings of 30%
- No Other SQL Negatively Impacted by the Index
- Index Required no SQL Changes
- Simple Change Management process
- Application Programmers were not Involved

Tuning the SQL - Static

- Summary of the Process
 - Systems Programmers found the Service Class using the most CPU → Online Regions
 - IMS Systems Programmer Identified the Message Processing Region using the most CPU → SSFOSM2
 - IMS Systems programmer Identified the Transaction in that Region using the most CPU → BM10
 - We Looked through the plan for BM10 and found the SQL statement using the most CPU → Open in Program BM10D143
 - We Tuned the SQL statement and measured the results

Tuning the SQL - Dynamic

- Static is Easy to Find and Tune, how about Dynamic?
- To use a Product or Not, that is the Question
 - Without a Product ...
 - Use IFCID 22 and 63 to Capture the SQL Text and the Mini-Bind information
 - With a Product
 - Read the Manual and then find the offenders
- Then, However, After you've Located it, How do you Tune it?

Tuning the SQL - Dynamic

- Ask yourself, Can I Change this SQL?
 - Not Possible if it came from
 - Purchased Application
 - Generated by Query Tool
 - Created by a JAVA Applet Connecting to your DB2
 - Possible if
 - You Wrote the Application
 - Your Users Wrote the Query after one Class on DB2 SQL

Tuning the SQL - Dynamic

- If You can't Change the SQL
 - Tune the Index Usage Where Possible
 - Tune Buffer Pool Usage
 - Isolate the Objects into a Unique Buffer Pool based on Access Patterns
- If You Can Change the SQL
 - Tune it
 - Give Your Users more SQL Education

Tuning the SQL - Dynamic

- Just a Product Note Here ...
 - Products that Provide measurement of Dynamic SQL
 - APPTUNE from BMC Software, Inc.
 - DETECTOR from Computer Associates
 - Query Monitor from IBM/Rocket Software
- Check with them for Product Specifics

Conclusion

- It is possible to find the worst SQL offenders
- You already know how to tune them
- You can quantify the savings associated with tuning
- Tuning does translate into more efficient applications and a better run company
- The process will lead you to the statements that offer the greatest return on your tuning time

Session F11

Tom Moulder

TREX Associates, Inc.

Tom.moulder@trexassociates.com

www.t-rex-associates.com