Performance is key to user satisfaction….What do you do to ensure that IMS meets it's service level goals for performance ?

Ed Woods

Consulting IT Specialist

David Viguers

IMS Performance Specialist

@.business on demand.

Agenda

- Introduction to the facilities and capabilities of OMEGAMON XE for IMS

- Introduction to the facilities and capabilities of IMS Performance Analyzer

- Analyzing a performance issue using OMEGAMON XE For IMS

- Analyzing the performance issue using IMS Performance Analyzer

OMEGAMON XE For IMS V3.1
Real Time Components And Facilities

- **Real Time Monitor**
  - ▸ *Subsystems, regions, resources, pools, DBs, Fast path*
  - ▸ *IMS Connect, OTMA*

- **Response Time Analysis**
  - ▸ *Transaction Response time by user defined groups*

- **Bottleneck Analysis**
  - ▸ *Workload performance and task analysis*

- **Operator Assist & Integrated Console Facility**
  - ▸ *Consolidation of IMS MTO consoles*

- **Online TRF**

- **Trace Facilities**

- **Exceptions & Alerts**

- **Plex level information**
 - ▸ *Integrated alert/automation*
  - ▸ *N-way, MSC*

OMEGAMON XE For IMS V3.1
Historical Facilities

- **EPILOG Historical**

  - Historical analysis of transaction response, bottlenecks and IMS resources

  - Stored in VSAM Epilog Data Store (EDS) by group and time interval

- **TRF Historical**

  - Detailed transaction & database data – individual transactions

  - Suitable for performance analysis & chargeback

- **XE Historical**

  - Snapshot historical
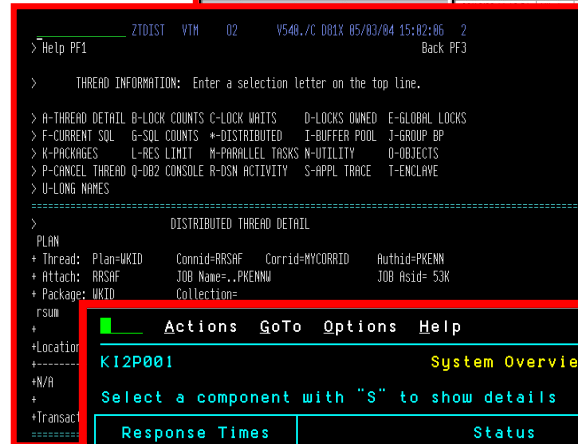
OMEGAMON
Choice Of Interfaces – Unique Capabilities

- **OMEGAMON XE GUI Interface**
  - ▸ **Java client or web browser – Tivoli Portal**
  - ▸ **Real time and historical**
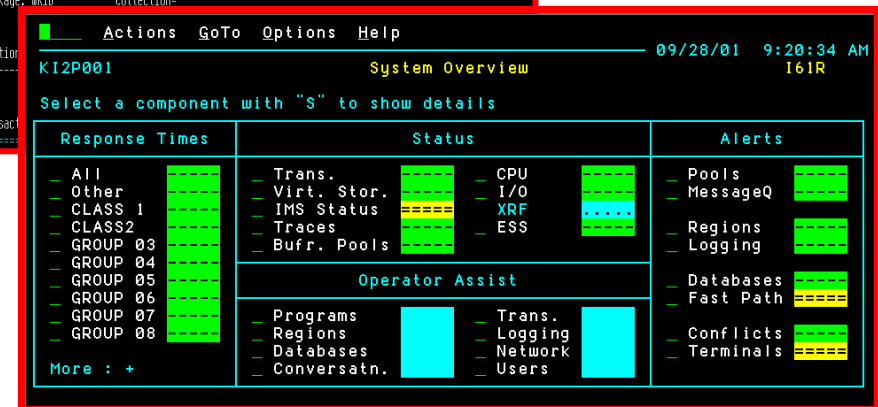 - ▸ **Automation & alerts**
  - ▸ **Plex level information (CF, n-way)**
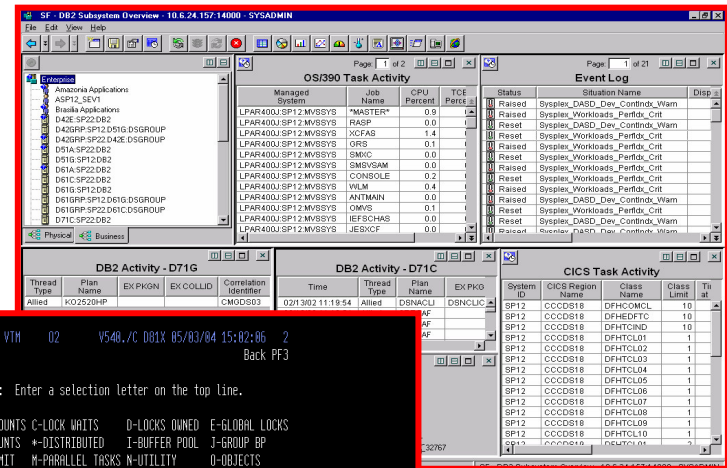
  OMEGAMON Classic
  - ▸ **3270 Interface command interface**
  - ▸ **Real Time & Historical**
  - ▸ **RTA, Dexan**
  - ▸ **Exceptions**

- **OMEGAMON CUA**
  - ▸ **3270 interface**
  - ▸ **ICF & Operator Assist run in CUA**
  - ▸ **Different views from Classic**
  - ▸ **Warning & Critical exception alerts**

IMS PA – Functions

- IMS PA can report on both log and monitor data
  - ▶ PA Log
    - Uses IMS logs (OLDS or SLDS) to create reports on various IMS resources
    - Can merge logs from multiple systems
    - Can include/exclude any or all reports
    - Can include/exclude any or all resources (Trans, Pgms, etc.)
  - ▶ PA Mon
    - Uses IMS monitor file as input
      - Could use Omegamon to trigger running monitor based on various criteria
    - Typically more detailed information than what is available on the log
      - Allows drilling down to specific areas or calls
    - Application trace function
      - Can see call and iwait flow and times

IMS PA - Interface

- **Good ole JCL and control card interface**
  - ▸ Okay when you already have a job setup and typically run the same reports
  - ▸ Not okay to try to do from scratch

- **ISPF menu driven interface**
  - ▸ Much easier to use
    - No need to figure out syntax of control statements
    - Allows use of DBRC to figure out which logs to select
    - Easy to select/deselect various reports based on needs
    - Creates all JCL and control statements

An IMS Performance Analysis Scenario Using OMEGAMON XE And IMS Performance Analyzer

- Application users are reporting slower than expected response time

- The application is a complex composite application incorporating several technology components

- IMS is a central part of the application

- OMEGAMON XE For IMS is installed and integrated with OMEGAMON DE and other IBM monitoring technologies

- IMS Performance Analyzer is available

- The objectives

  ▸ How to identify and isolate the performance issue

  ▸ How to analyze the performance issue

  ▸ How to do a detailed analysis of the issue with root cause analysis

OMEGAMON XE For IMS And OMEGAMON DE

Tivoli Enterprise Portal With OMEGAMON DE (Dashboard Edition) enables integrated multi-component views

Customizable graphic overview

User-definable drill downs for detail

Combine information from multiple sources

OMEGAMON has detected an issue in IMS

Problem Isolation – A Situation Highlights The Issue

Isolate The Problem – Situation Detail

What are the details?

What is the problem?

Any Predefined Actions?

Any expert advice?

Situation Definition

Situations for - IMS Transaction Summary

Formula | Distribution | Expert Advice | Action

IMS Transaction Summary
IMS
EW_IMS_Tran_Queued

Description

Formula

	Count of Transactions	Description
1	> 25	== Transactions_In_Queued_St
2		
3		

Specify Situation alert criteria

In this example alert if transaction queue beyond 25 transactions

Allows the use of And/Or logic

the formula editor to see a description of the attribute for that column expression.

licking **Add conditions** and selecting the situations to embed or o include.

Situation Formula Capacity   13%   Add conditions...   Advanced...

Specify multiple attributes & sampling interval

Sampling interval
0 / 0 : 0 : 30
ddd hh mm ss

Sound
☐ Enable critical.wav
Play   Edit...

State
Critical
☑ Run at startup

Specify alert level and whether to run at startup

Cancel   Apply   Help

Problem Isolation – Next Steps
Look For Stopped And Bottlenecked Resources

- Use OMEGAMON XE to determine the likely cause of the transaction queue

- Check for stopped resources that may be needed by the transactions

  ▸ Example – transactions may be abending and stopping needed resources

- Use OMEGAMON XE For IMS DEXAN (Bottleneck Analysis) to analyze the workload

Leverage The Power Of OMEGAMON XE
Create A 'Stopped Resource' Display

Create a 'Stopped Resource' workspace to highlight if any key resources are being stopped

Easy to view from a single screen

Note Region Occupancy %

Regions

Transactions

Databases

PSBs

How To Create A Custom Workspace

Go to 'Properties' and select the Filters Tab

Select which columns are to appear in the workspace.

Query controls what data appears on the screen

Specify which conditions will be included

Save The New Workspace

Select 'File' and 'Save As' to save the newly created workspace in the Tivoli Enterprise Portal Server

Problem Analysis – Next Steps

- Use XE Interface for 'Big Picture'

  ▶ Problem identification and problem isolation

- Use XE Interface for the most flexible alerts

- Use XE Interface for the integrated views of the enterprise and applications

- Use 3270 Classic & CUA for deep dive analysis

  ▶ Response Time Analysis And Bottleneck Analysis

- Don't forget about history

  ▶ TEP history, EPILOG history, TRF history

- Deep dive analysis using IMS Performance Analyzer

Response Time Analyzer - RTA

Captures detailed response time data from IMS

Works in conjunction with DEXAN

Places data into summary buckets

Reports data in user specified groups
Groups defined with KOIGBLxx macro

RTA measures
Queuing and service times within IMS

Recommendation – Customize RTA groups (benefits both RTA and DEXAN)

OMEGAMON IMS
Response Time Buckets

MSG RECEIVED

MSG ENQUEUED

I

PI

R0

PGM SCHEDULE

MESSAGE GU

R1

P

MESSAGE ISRT

PGM END

O

OUTPUT ENQUEUE

IMS MSG SENT

IMS
CTL
Region

MPP
Region

IMS
CTL
Region
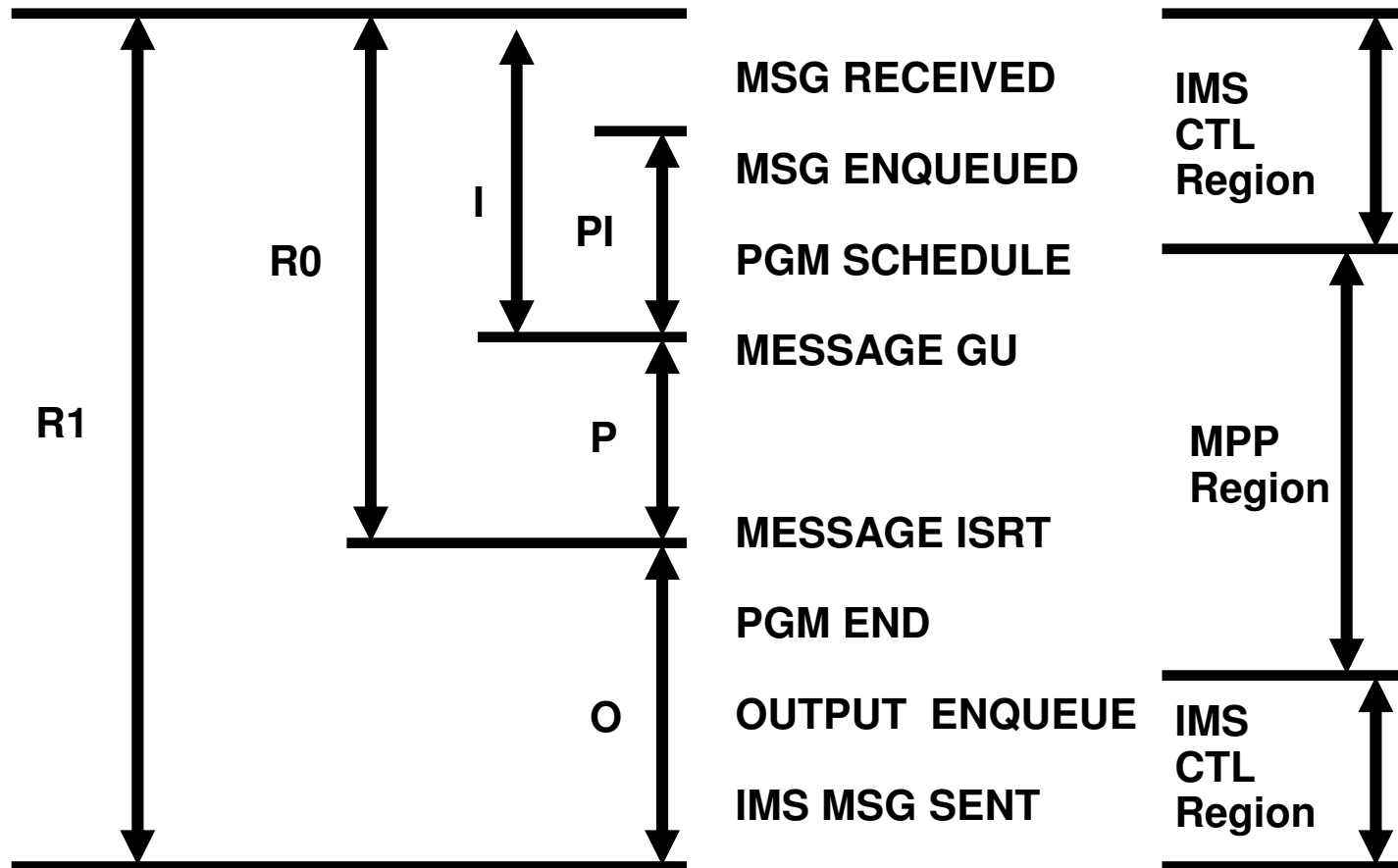
OMEGAMON RTA
Customized RTA Group Targets Key Applications

```
Actions   GoTo   View   Options   Help
--------------------------------------------------------- 04/09/06 13:40:33
KI2PRT              Response Time 1 Analysis for All Groups     IMS: IMS    +
                                                     Lines  1 to 17 of 17

-------------ç-------------------ç-------------------ç-------------------
:           : Interval 00:05 hrs : Interval 00:30 hrs : Interval 01:00 hrs :
: Group Name : Elapsed 00:00:26  : Elapsed 00:05:26  : Elapsed 00:35:26  :
-------------º-------------------º-------------------º-------------------o
:   ALL      :(     542)   0.1s :(     6888)    0.1s :(    47196)    0.1s :
:   OTHER    :(       7)   0.0s :(      189)    0.1s :(     1180)    .1s :
:   TRN      :(     188)  18.1s :(     1241)   19.1s :(     9104)   15.1s :
:   ABCXX    :                  :                    :                    :
:   REGIO    :(      72)   0.1s :(     1177)    0.1s :(     7686)    0.1s :
:   ABC      :                  :                    :                    :
:   LEDGER   :(      16)   0.2s :(      277)    0.2s :(     1669)    0.2s :
:   FINANCE  :                  :                    :                    :
:   DBA      :(       3)   0.0s :(        9)    0.0s :(      351)    0.0s :
:   AA       :                  :                    :                    :
:   BB       :(     107)        :                    :                    :
:   EMAIL    :                  :
1---------------------------------

 (Response Time)  <Response
 F1=Help  F2=Keys  F3=Exit
```

```
Actions   GoTo   View   Options   Help
--------------------------------------------------------- 04/09/06 13:41:01
KI2PRG01              Response Time 1 Analysis for Group: TRN        IMS
                                                     Lines 1 to 2 of 2

-------------ç-------------------ç-------------------ç-------------------
:           : Interval 00:05 hrs : Interval 00:30 hrs : Interval 01:00 hrs :
:  Item ID   : Elapsed 00:00:54  : Elapsed 00:05:54  : Elapsed 00:35:54  :
-------------º-------------------º-------------------º-------------------
: T=TRN*     :(     200)  19.1s :(     1344)   18.1s :(     9207)   14.1s :
```

The TRN application transaction codes are in a group and are having poor response time

DEXAN
Degradation Exception Analysis

Degradation Analysis – Sampling Algorithm

> Breaks down workload into components

> Concentrate on significant component

- Competing - eligible to be processed

- Not Competing - cannot be processed

- Executing – currently being processed by an IMS ITASK

- Group definitions apply to DEXAN as well as RTA

Use DEXAN (Bottleneck Analysis) To Determine Where The Work Is Spending Time

```
GoTo   Options   Help
------------------------------------------------------------- 10/09/05 13:31:20

KI2PSDX2                    Bottlenecks Analysis for Group TRN           IMSA

----------------------------------------------------------
: Elapsed time . . . : 17:24 MN        Sampl
: Suppress states . . <  0 %           Sampl
: Display COMPETING TRANSACTIONS +     Sampl
----------------------------------------------------------

-----------------------ç--------------------
:      Wait Reason       :        Short Term
:                        : %  0------- 50---
-----------------------º--ç-----------------
: Using CPU:             : 15.0:-->  .     .
:  Using CPU in Appl     :10.70:->   .     .
:  Using CPU in IMS      : 4.20:>    .     .
: Scheduling Waits:      :  7.9:>    .     .
:  Wait for MPP          : 7.70:>    .     .
:  Intent Conflict       :  .10:>    .     .
:  TM Schedule Latch     :   0:      .     .
: IMS Activity:          : 10.0:->   .     .
:  Other DL/I IWAIT      : 5.60:>    .     .
:  IWAIT in IMS Disp     : 1.20:>    .     .
:  IWAIT in Term         :   0:      .     .
:  LOGL Latch            : .50:>     .     .
:  DBBP Latch            : .10:>     .     .
:  ISWITCHed to CTL      : 2.40:>    .     .
----------------------------------------------

 <Response Time>   <Response Time Components
```

Analyze where IMS workload is spending its time. Provide analysis of where to tune.

```
GoTo   Options   Help
--------------------------------------------------------------- 10/09/05 13:31:28

KI2PSDX2                    Bottlenecks Analysis for Group TRN              IMSA

-------------------------------------------------------------------------
: Elapsed time . . . : 17:24 MN     Samples taken (short) . . : 281        :
: Suppress states . . <  0 %        Samples taken (long)  . . : 2026       :
: Display COMPETING TRANSACTIONS +  Sampling interval . . . . : 5 tenths-sec :
-------------------------------------------------------------------------

                                                     Lines 15 to 28 of 29

-------------------------------------------------------------------------
:    Wait Reason     :       Short Term %      :       Long Term %       :
:                    : % 0------- 50-------100 : % 0------- 50-------100 :
-------------------------------------------------------------------------
:  DC Sys Ckpt Latch :    0:  .    .    .  . : .20:>  .    .    .  . . :
: Database I/O Waits :  .3:>  .    .    .  . : .2:>   .    .    .  . . :
:  HISTDB            : 30.0:----->  .    .  .  . :30.0:----->       :
:  D1B80002          : .30:----->  .    .  .  . :  . .:.30           :
: MVS Waits:         :  3.2:>  .    .    .  . : 2.0:  .    .    .  . . :
:  CPU Wait (DEP)    : 3.20:>  .    .    .  . : 2.00:----->   .  . . :
:  Program Fetch I/O :   0:  .    .    .  . :  0:  .    .    .  . . :
: ESS Waits:         : 26.5:---->  .    .  . : 23.8:--->.   .  . . :
:  Commit (Phase 2)  : 2.80:>  .    .    .  . : 2.30:>  .    .  . . :
:  Prepare to Commit : 4.70:>  .    .    .  . : 5.60:>  .    .  . . :
:  User Sign on DB2  : .10:>  .    .    .  . : .30:>   .    .  . . :
:  Terminate Thread  :   0:  .    .    .  . :  0:  .    .    .  . . :
:  SQL Call          :18.70:-->  .    .  . :15.30:-->  .    .  . . :
: Other Waits:       :      :    :         :     :                 ø :
-------------------------------------------------------------------------

 <Response Time>   <Response Time Components>   (Bottlenecks)
```

DEXAN data based on RTA groups defined in KOIGBL

Getting To The Root Cause Analysis

- **Exploit TEP historical capabilities**
  - ▶ Snapshot history
  - ▶ History of alerts to show how often the workload is queued

- **EPILOG history**
  - ▶ Historical RTA, DEXAN and resource information
  - ▶ Stored based on 15 minute time intervals

- **Transaction Reporting Facility**
  - ▶ Performance report analysis from TRF records stored in the IMS log

- **IMS Performance Analyzer**
  - ▶ Detailed IMS performance analysis

IMS PA

- As mentioned before we have both PA Log and PA Mon functions

- For today let's assume we need detailed application analysis
  - ▶ Therefore we will use the PA Monitor function
  - ▶ Remember we could have triggered the IMS monitor to be run based on Omegamon alerts
    - While TRF might give us some of the same information we might not choose to run TRF all the time

- So let's look at some scenarios and how we might use PA MON to further analyze

- One key thing to remember with almost all performance problems is:
  - ▶ Have comparison data available
    - Sometimes a single sample is enough when something is REALLY bad
    - Many times there is some subtle difference causing the problem

IMS PA

- As mention before we have both PA Log and PA Mon functions

- For today let's assume we need detailed application analysis

  ▶ Therefore we will use the PA Monitor function

  ▶ Remember we could have triggered the IMS monitor to be run based on Omegamon alerts

    - While TRF might give us some of the same information we might not choose to run TRF all the time

- So let's look at some scenarios and how we might use PA MON to further analyze

- One key thing to remember with almost all performance problems is:

  ▶ Have comparison data available

    - Sometimes a single sample is enough when something is REALLY bad

    - Many times there is some subtle difference causing the problem

IMS PA – scenario #1

- Scenario #1
  - ▶ Tran TRN030 running very slow (over 20 seconds) all of the sudden
  - ▶ Omegamon showed lots of contention for DB HISTDB
  - ▶ Holder was many times TRN020 but not always
  - ▶ Ran IMS PA Monitor report with many reports selected
    - Better to run with too many because
      - – If you try to select what you think you need you will almost always realize you missed something and have to rerun (and rerun) to get what you need.
  - ▶ Looked at Program Summary report and found TRN020 is indeed running a long time

IMS PA – scenario #1

```
                                    P R O G R A M   S U M M A R Y


                                    _____

Region Totals       From 25Apr2006 11.06.41.15 To 25Apr2006  11.12.14.77   Elapsed=   0 Hrs   5 Mins  33.615.197 Secs
** Program Analysis **

                               -------- Per  Schedule -------- |---------- P e r   T r a n s a c t i o n ----------
                              Trans  Scd-DLI   DLI-Term Schd                    Elapsed  CPUtime  Pct of Tran Elap
PSBname  TranCode  Scheds   Trans  /Schd  Mil.Mic  Sc.Mil.Mic on Q  Calls    IWTs  Sc.Mil.Mic  Mil.Mic  CPU  Call  IWT
PSB0001  TRN0001      26      26   1.00   0.288      6.833   .0    6.4      .6      6.833    9.369  137% 89.9% 42.0%
PSB005A  TRN005A       2       2   1.00   0.000      0.128   .0    1.0      .0      0.128   10.010  851% 78.4%  .0%
PSB008C  TRN008C       1       1   1.00   0.000      0.118   .0    1.0      .0      0.118   10.010  483% 83.9%  .0%
PSB013A  TRN013A       1       1   1.00   0.000      0.134   .0    1.0      .0      0.134   10.010  470% 85.8%  .0%
PSB014B  TRN014B       6       6   1.00   2.137    126.619   .0   68.3     9.7    126.619   15.509 12.2% 75.8% 56.8%
PSB015A  TRN015A      17      17   1.00   1.896     49.939   .0  103.1     9.8     49.939    9.882 19.8% 93.8% 77.0%
PSB020N  TRN020N       5       5   1.00   2.154  21.578.652   .0   36.6     8.4  21.578.652   12.917   .8%  8.8%  5.9%
```

- Look for a TRAN/PSB with similar elapsed time
  ▶ Found PSB020N
- Note that very little time is CPU/Call/IWT
  ▶ Decided to run the PA Mon application trace for PSB020N

IMS PA – scenario #1

```
                              P R O G R A M   T R A C E



                              _____

Relative Time   Pgm Time    Call              ST PCB  Feedbck  Mod IWT  DDname   IWT Elapsed  Breakdown of Call Time   Call Elap
Secs.Mil.Mic    Sc.Mil.Mic  No. PCBname   Func Cd Segname  Lvl  ULE  (# IWAITs)  Sc.Mil.Mic  Pct CPY Pct DLA Pct IWT  Sc.Mil.Mic


       3.238       0.037     10  HISTDB    GN   GE MYROOT   01                                 .00% 100.00%    .00%       0.007

       4.470       1.225     11  HISTDB    ISRT    HIST     02            1        1.112        .00% 100.00%  86.34%      1.288
       4.603                                          FPD DEDB  HISTDD    1.112
   24.009.143  24.003.385    12  HISTDB    GU      DEP02    02            1        0.958        .00% 100.00%  81.67%      1.173
   24.009.313                                         FPD DEDB  HISTDD    0.958
   24.010.332      0.016     13  HISTDB    GN   GE MYROOT   01                                 .00% 100.00%    .00%       0.009
   24.010.374      0.033     14  HISTDB    GHU     DEP02    02                                 .00% 100.00%    .00%       0.005
```

▪ Note the relative time difference between call 11 and 12

  ▸ But call elapsed is small

  ▸ So what is happening between call 11 and 12?

    ▪ Good question and not one that could be answered from IMS data

    ▪ However a call to the application owner found that a one line change had been made over
      the weekend – a simple native APPC call.

  ▸ IMS was the victim

IMS PA – scenario #2

- Scenario #1
  - ▶ Many Transactions running very slow at peak times
    - Low to moderate loads are fine
  - ▶ Since the problem seems to be most trans decided to go directly to IWT summary

IMS PA – scenario #2

```
Report from 25Apr2006  11.06.41.15          IMS 8.1.0      IMS Performance Analyzer 3.3       Report to 25Apr2006  11.12.14.77

                                                Total System IWAIT Summary
                                            _____


                    --------------- I / O   I W A I T S -----------------|------------- N O N  I / O   I W A I T S -------------
                              Elap/IWAIT  StdDv   Max value  Pct of  Pct of          Elap/IWAIT  StdDv   Max value  Pct of  Pct of
                      Count   Sc.Mil.Mic  X Avg   Sc.Mil.Mic IWAITs  IWT Elp |  Count  Sc.Mil.Mic  X Avg   Sc.Mil.Mic IWAITs  IWT Elp


Database IWAITs                                                            |
     OSAM          299       7.217  1.226     68.162   .08%    .19% |
     VSAM        1,832      13.544  1.349    623.641   .50%   2.14% |
     VSAM                                                          |  3,856      1.334   .744      4.996   3.06%    .01%
     DEDB(Read)  355,396     3.081  2.208    634.580  96.12%  94.27% |
     DEDB(Write)  12,210     3.239  1.359    397.100   3.30%   3.41% |
     DEDB(Lock)                                                    |  1,613      9.345  7.219   2.532.431   1.28%    .03%


  Report from 25Apr2006  11.06.41.15          IMS 8.1.0      IMS Performance Analyzer 3.3       Report to 25Apr2006  11.12.14.77

                                                  Data Base IWAITs
                                            _____


                    -------------- I / O   I W A I T S ----------------|                -------------- I / O   I W A I T S ----------------
                              Elap/IWAIT  StdDv   Max value  Pct Tot Pct Tot|                    Elap/IWAIT  StdDv   Max value Pct Tot Pct Tot
OSAM DDn    IWAITs  Sc.Mil.Mic  X Avg   Sc.Mil.Mic IWAITs IWT Elp| VSAM DDn   IWAITs  Sc.Mil.Mic  X Avg   Sc.Mil.Mic IWAITs IWT Elp
ODDNAM01        1      25.584   .000      25.584   .33%   1.19%| VDDNAM10       6      12.461   .704      31.171   .33%    .30%
ODDNAM02        1      14.984   .000      14.984   .33%    .69%| VDDNAM20       2       8.956   .041       9.324   .11%    .07%
ODDNAM03        1       1.464   .000       1.464   .33%    .07%| VDDNAM30       5       9.168   .198      11.941   .27%    .18%
ODDNAM04        1       0.596   .000       0.596   .33%    .03%| VDDNAM40       1       9.852   .000       9.852   .05%    .04%
ODDNAM05       47       7.834   .704      20.897  15.72%  17.06%| VDDNAM50       3      10.153   .147      12.172   .16%    .12%
ODDNAM06       12       4.836  1.229      17.082   4.01%   2.69%| VDDNAM60      23      13.134   .173      19.316   1.26%   1.22%
```

IMS PA – scenario #2

- Any problems in previous report?
  - On the surface it doesn't look too bad (and it's not)
  - HOWEVER
    - This is a moderately low load time and comparing to a peak time report the I/O IWAIT times are 2-3 times higher
    - And since they are ALL (or at least most) are higher a closer look at RMF reports revealed that the DB dasd paths were just plain old overloaded
  - So what can be done?
    - Some tuning of DB buffer pools helped reduce the I/O a bit but
    - As the volume increases then the problem comes back
    - Time for more hardware

IMS PA – Summary

- Omegamon and IMS PA have many overlapping reports but
    - Each is suited best for different types of monitoring

- Many functions and features
    - Too many to cover here and some are more commonly used than others

- Deciding which functions to use will be best determined by need
    - Some folks will need many functions – others don't need the entire suite

- Don't be afraid to run with all or most PA reports
    - Use (default) to separate DD names to select what you want to look at

Thank You!!