

1



IBM Software Group | Tivoli Software

Monitoring: Become More Proactive Through Effective Historical Analysis

Ed Woods
Consulting IT Specialist
IBM Corporation

Session 9633
Tuesday, August 9th
1:30-2:30 PM

© 2011 IBM Corporation

Agenda

- How can you use history to improve your real-time monitoring strategy?
- Are you getting the most from your investment in monitoring and management solutions?
- How can you leverage history to improve your overall performance and availability?
- What are the most effective ways to use history to solve common problems?
- What are the optimal ways to collect historical information?
- How can you use history to become more proactive with real-time monitoring and management?

The Importance Of History Data

Why History Can Be Essential To Your Overall Strategy

- Not all problems or events can be seen and analyzed in real time
 - Inevitably some analysis will need to be done after the fact using such functions as Near Term History, snapshot history, or report analysis
- History provides an understanding of what happened in the past
 - History of application performance and response time
 - CICS/IMS response time, DB2 thread activity, z/OS batch activity
 - History of resource utilization and resource issues
 - DASD, memory/paging, CPU, pools
 - History of alerts and issues
 - What alerts fired and how often
- History can be used to help visualize what may happen in the future
 - Analysis of the past to help anticipate potential future issues/bottlenecks
- Use history to make real-time monitoring more effective and meaningful
 - Use historical information to make real time alerts more accurate and relevant
 - Include history in custom real time workspaces

Historical Data Collection Considerations



- Historical data collection varies in cost and quantity
 - CPU, memory, and software process cost of collection
 - Cost of data storage and retention
 - Cost of retrieval and post processing
 - Ease of review and analysis
- Some historical data will be more relevant and useful than other data
 - Consider the context, nature, and meaningfulness of the data

Types Of Historical Monitoring Data

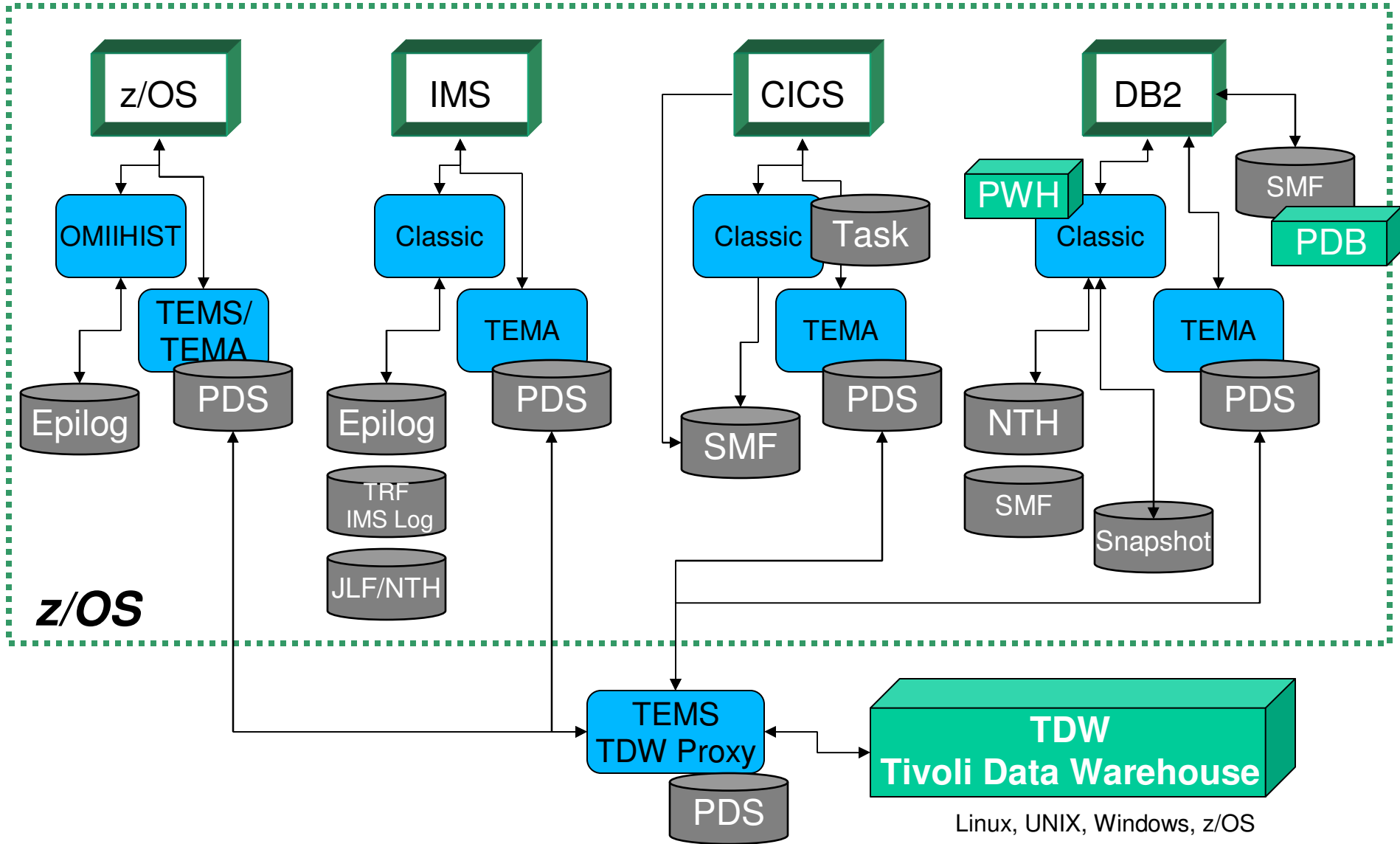
- Know the nature and characteristics of the history data being collected
- Detail data
 - Data that documents/measures detail of a specific event
 - Often high quantity data and the most detailed for analysis
 - May pose the greatest challenge in terms of cost, retention, post processing
 - Examples – DB2 Accounting records in Near Term History, CICS Task History, IMS Near Term History
- Summary data
 - Data that summarizes underlying detail data
 - Either an aggregation or an averaging of underlying detail records
 - May be useful for longer term trending and analysis
 - Reduces quantity of data and reduces cost of retention, post processing
 - Less detail may mean less diagnostic value
 - Examples – Summary data in Tivoli Data Warehouse, summary DB2 trace data

Types Of Historical Monitoring Data - continued

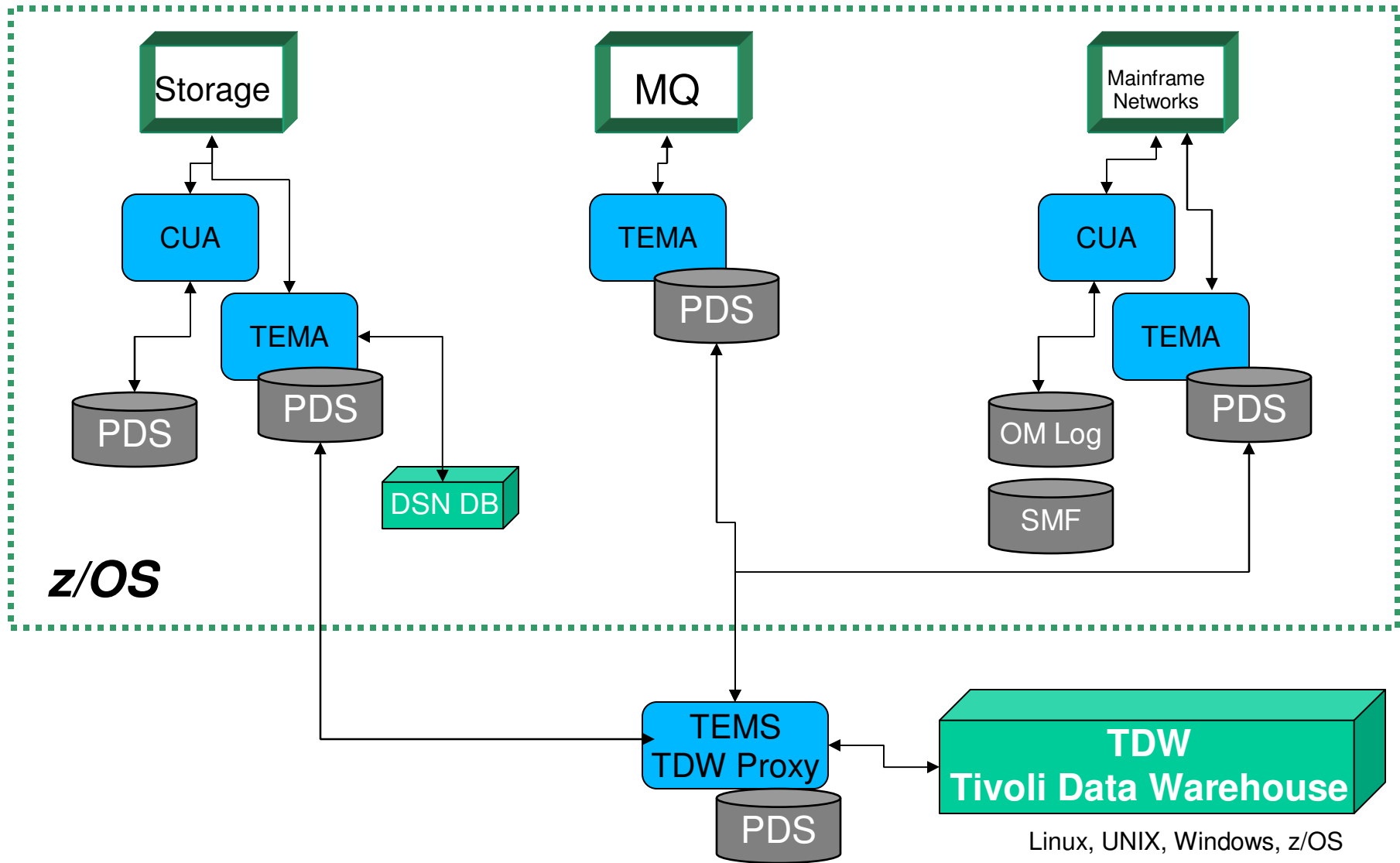
- Interval data
 - History data that includes an encapsulation of one or multiple events within a specified time interval
 - The data will include all activity within that given time interval
 - Useful for problem analysis and trending analysis
 - Examples – DB2 statistics records in Near Term History, Epilog IMS or z/OS history

- Snapshot data
 - Typically a point in time snapshot of activity
 - Snapshots are usually based on a specified time interval
 - Snapshots may be taken of types of history (detail, summary, or interval)
 - Snapshots will show activity at time of the snapshot, but may/may not reflect activity between snapshots
 - Useful for problem analysis and trending analysis
 - Useful as an aid in setting alert thresholds
 - Examples – OMEGAMON DB2 PE GUI snapshot history, Tivoli Data Warehouse snapshot history

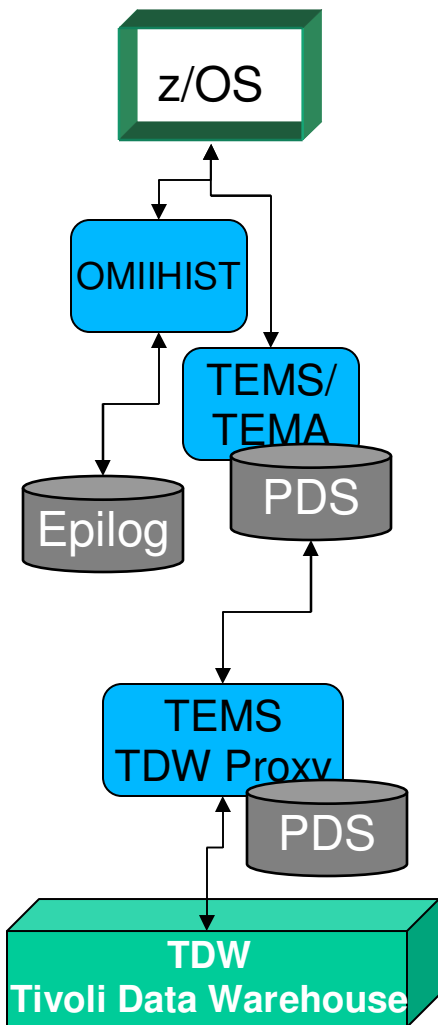
OMEGAMON History Collection Options



OMEGAMON History Collection Options - continued



OMEGAMON XE For z/OS History



Linux, UNIX, Windows, z/OS

- OMEGAMON XE on z/OS provides history data in the form of Epilog history
 - Service levels (elapsed times and response times)
 - Resource utilization data
 - Degradation data (bottleneck analysis of z/OS workload)
- Epilog history may be accessed via CUA interface, batch jobs, ISPF command interface
 - Sample batch reporter JCL is in *hilev.RKANSAMU(KEPPROC)*
- OMEGAMON XE on z/OS provides snapshot history data and supports the Tivoli Data Warehouse (TDW)
 - Data is stored in the TEMA/TEMS in the Persistent Data Store (PDS)
 - Data may be optionally sent to the TDW
 - Data may be summarized and pruned using the TDW
- Cost of collection relative to value

- Epilog – low cost	↔	Useful for problem analysis
- Snapshot – low cost	↔	Useful for trending

The Value Of OMEGAMON z/OS History

Epilog - historical problem analysis

```

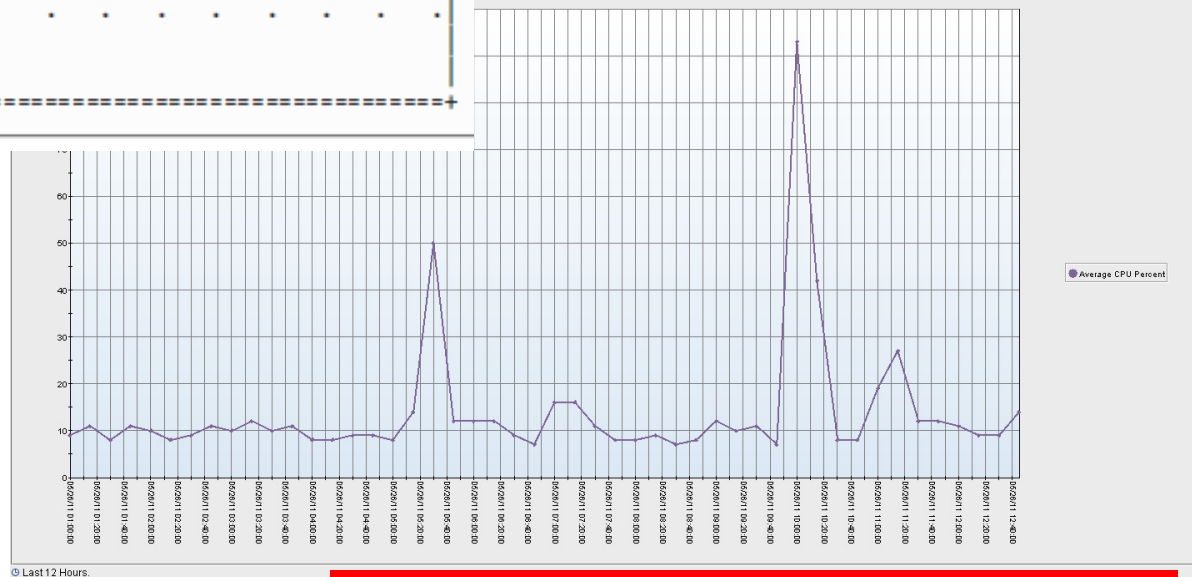
=====
Job = PAYROLL          JES Number = 2544          Job Steps = 3 / 6
Job Class = T          Account Number = 035G       Input Queue = 48.10 S
From 09:12 To 09:14 On 08/01/99                  Elap = 1:33 M SYSA
=====
WAIT Reason           Time           % | 0  1  2  3  4  5  6  7  8  9  0
-----
Using CPU              0.09 S         .1 | .  .  .  .  .  .  .  .  .  .  .
Swapped With WTOR     52.65 S       59.9 |----->
Tape Mount Pending    25.05 S       28.5 |----->
ECB Wait               5.01 S         5.7 |-->
Tape IEFU84 522 Que   5.01 S         5.7 |-->
Job Elapsed Time      1:27 M
=====
    
```

Epilog history provides historical bottleneck analysis data correlated with resource and usage data

Use Epilog for historical problem analysis

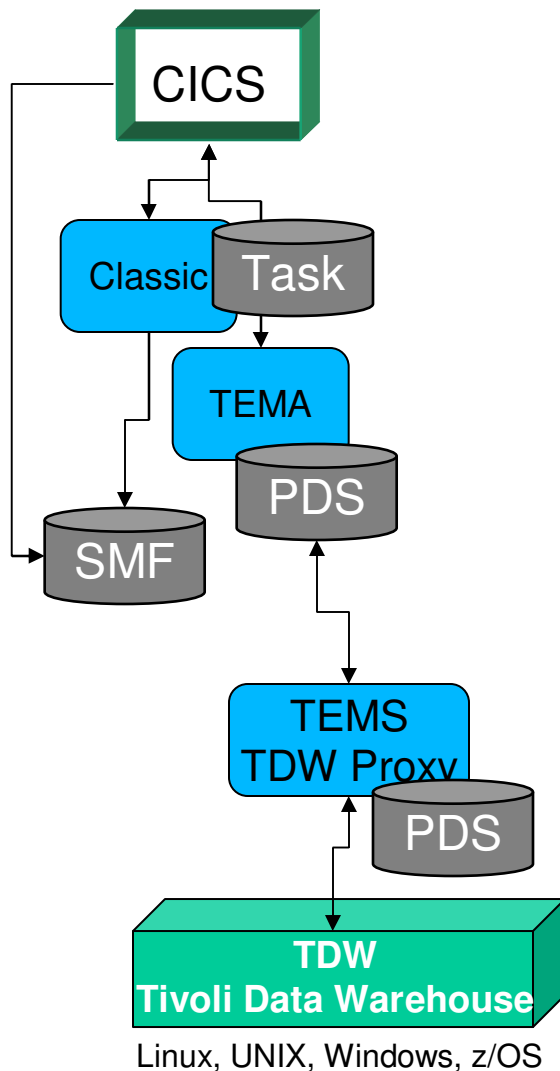
Use TDW for highlighting peaks and trending utilization (use as input for setting alert thresholds)

TDW - visualization



TDW is effective at providing snapshot data for trend analysis

OMEGAMON XE For CICS Provides History Options For History Detail, Near Term Detail And Trending



- Task history (also known as Online data viewing) provides detailed CICS transaction level history
 - Detailed transaction-level data stored in an ONDV task history file (wraparound VSAM file – one per CICS region)
 - Easy to access/filter – very good detail
- SMF 110 records
 - SMF 110 subtype 1 records - CICS task level data
 - CICS Statistics data (SMF 110, subtypes 2, 3, 4, 5)
 - Information collected on an interval basis and/or end of day
 - Note – OMEGAMON may optionally add additional detail to SMF 110 records
- OMEGAMON CICS provides snapshot history data and supports the Tivoli Data Warehouse (TDW)
 - Data is stored in the TEMA/TEMS in the PDS and optionally sent to the TDW
 - Data may be summarized and pruned using the TDW
- Cost of collection relative to value

- Task history – low to moderate	↔	Useful for problems
- SMF 110s – typically low	↔	Important for reporting
- Snapshot – typically low	↔	Useful for trending

OMEGAMON CICS Task History – Valuable For Problem Analysis Access Via 3270 Interface And The TEP

```

Actions  GoTo  View  Index  Options  Help
-----
KC2T01D          Task History          05/31/11
Fastpath: =H          Regi

Search range . . . 00/00/00  00:00:00  to  00/00/00  00:
Display range . . . 05/31/11  10:16:47  to  05/31/11  11:

T=Trace          More:  +
    
```

Task End Time	Tran ID	Task Number	Term ID	CPU Time	Resp Time	Storage HWM	File I/O	Abend Code
11:47:18	CWBG	219	n/a					
11:37:52	CSOL	4	n/a					
11:06:25	CSOL	4	n/a					
10:47:17	CWBG	218	n/a					
10:34:57	CSOL	4	n/a					
10:18:10	CSAC	217	Z006					
10:17:57	CSAC	216	Z006					
10:17:51	CEMT	215	Z006					
10:17:31	CSAC	214	Z006					
10:17:26	CSAC	213	Z006					
10:17:09	CSAC	212	Z006					
10:17:07	CSAC	211	Z006					
10:16:58	CSAC	210	Z006					

Even though history volume is often high, Task history is easy to navigate/filter to identify issues

Summary of tasks

```

GoTo  Index  Options  Help
-----
KC2T031D          Task History Timings          05/31/11
Re          Detail
Task number . . . . . : 4          Transaction ID . . . . . : CSOL

Overall Elapsed time: 0:31:27

Dispatch time . . . . . : 0.000085          Suspend time . . . . . : 31:27.43
QR TCB elapsed time : 0.000024          Total I/O wait times . . : 0.000s
Other TCBs elapsed : 0.000060          Total other wait times . : 0.000s
CPU time . . . . . : 0.000083          1st dispatch delay . . . : 0.000000
RLS CPU time . . . . . : 0.000000          Re-dispatch wait . . . . : 0.000057
RMI elapsed time . . . : 0.000000          Exception wait time . . . : 0.000000
JVM elapsed time . . . : 0.000000          Program load elapsed time: 0.000000
Syncpoint elapsed time . : 0.000000
    
```

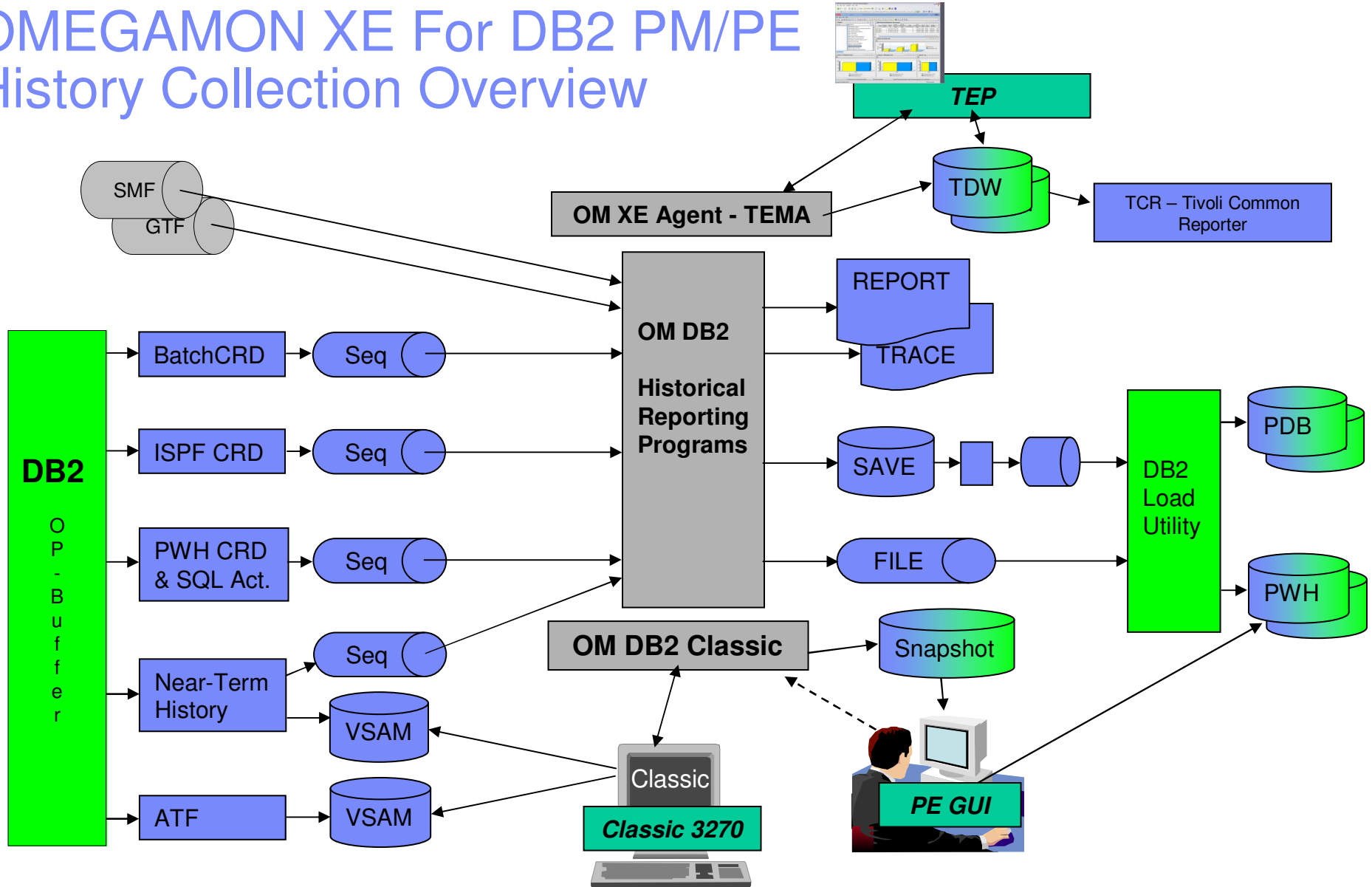
Navigate for detail

<Details>
<Storage>
(Timings)
<Statistics>
<Task Terminal>

CICS task history is easy to access via CUA/Classic 3270 and via the Tivoli portal

Task history provides important task level detail for problem analysis

OMEGAMON XE For DB2 PM/PE History Collection Overview



OMEGAMON DB2 Near Term Thread History

Easy Access To History Within OMEGAMON Classic Interface

View thread level data (Accounting detail or summary)

View DB2 Statistics data (interval data)

./C DSNCL 07/25/09 9:24:57 3
Down PF8 Zoom PF11

```

>
>          Enter a selection letter on the top line.
>
> *-BY PLAN          B-BY AUTHID          C-BY PLAN,AUTHID          D-BY AUTHID,PLAN
> O-OPTIONS
=====
>          THREAD HISTORY BY PLAN
HAGP
+ Report Interval:   15 mins
+ Report Filtered:   NO
plan
+          Thread Summary Not Available, Data Collected
+
+          DLk/   In-DB2   In-DB2   In-DB2           GetP/
+ Plan      Thrds Commit Abrt   DML   TOut   Elap Tm CPU Tm  Wait Tm  Getpage RIO
+ -----
+ ASNQA910    9     90    0  1486    0     .5     .15     .4     1225  1225
+ ASNQC910    1     10    0    20    0     .0     .00     .0      54  54.0
+ DISTSERV   15     49   30   117    0     .2     .04     .2     210  210.0
=====
    
```

Near Term History stored in VSAM files for easy access from Classic interface.

Many filter/view options.

F11 to see more detail on a specific thread

NTH is highly detailed, easy to filter and access via 3270 interface, and is highly useful for problem analysis

OMEGAMON DB2 Snapshot History Scrollable And Easy To Access Via PE GUI

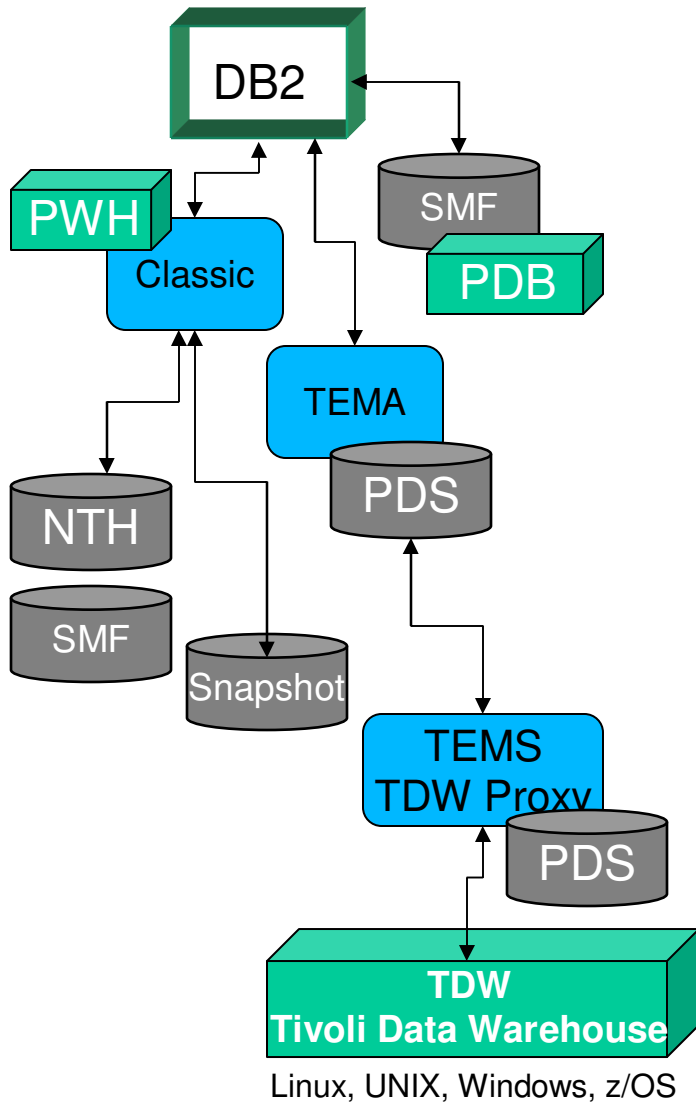
**Slide the scroll bar to
move history time frame**

The screenshot shows the 'DSNC - Thread Summary' window. At the top, there are menu options: Thread Summary, Selected, View, Tools, Window, Help. Below the menu is a toolbar with icons for home, print, and help. The main area contains a 'Data:' section with a dropdown set to 'History', a date field 'Sep 2, 2010', and a time field '14:45:19'. A 'Refresh:' section has a 'Manual' dropdown and a refresh icon. A time frame slider is visible, showing a range from '9/2/10 10:36:50 AM' to '9/2/10 4:31:16 PM', with a current selection at '9/2/10 2:45:19 PM'. Below the slider is a table of thread data. A red box highlights the first four rows of the table. A red callout box with white text is overlaid on the table, providing context and benefits of the snapshot history feature.

Primary Authorization	Total Getpage Request	Parallel threads	Member	Plan	CPU Class 2	Program Name	CPU Class 1	Elapsed Class 1	Elapsed Class 2	Total Class 3	Connection ID	Reque
DDS0510	28K	0	N/P	DI...	0.212428	SYSLH200	0.212587	0:55:18.412	0.623967	0.085851	SERVER	
DDS0510	24K	0	N/P	DI...	0.194603	SYSLH200	0.194760	0:56:07.961	0.596151	0.125646	SERVER	
DDS0510	28K	0	N/P	DI...	0.214595	SYSLH200	0.214922	0:58:44.492	0.598267	0.051097	SERVER	
DDS0510	28K	0	N/P	DI...	0.216882	SYSLH200	0.216255	0:58:51.288	0.654174	0.076507	SERVER	
DDS0510	28K	0	N/P	DI...	0.215566	SYSLH200	0.215751	0:59:31.495	0.898353	0.351939	SERVER	
DNET356	0K	0	N/P	DI...	0.009752	SYSLH200	0.010223	1:09:24.301	0.033706	0.011048	SERVER	
DNET356	0K	0	N/P	DI...	0.000844	SYSLH200	0.001036	1:09:25.686	0.002251	N/P	SERVER	
KLTAYLO	102008K	0	N/P	DI...	0:04:13.521	SQLC2FOA	0:04:30.456	13d 3:23:20.464	0:10:21.080	0.271977	SERVER	
DB2PM	N/P	0	N/P	KO...	0.665588	N/P	1.757291	1:25:48.501	0.701446	0.004503	RRSAF	
DB2PM	N/P	0	N/P	N/P	0.007812	N/P	0.029918	1:27:01.179	0.019389	0.008374	RRSAF	
DB:								1:27:03.080	24.097320	0.277831	RRSAF	
DB:								1:27:03.087	5.415562	4.901371	RRSAF	
DB:								1:27:03.424	0.040625	0.001323	RRSAF	
DB:								1:27:03.523	0.040050	0.001474	RRSAF	
DB:								0:19:19.426	0.287889	0.149888	RRSAF	
DB:								N/P	N/P	N/P	RRSAF	
SY:								N/P	N/P	N/P	CICSACB7	
SYSTC	N/P		N/P	N/P	N/P	N/P	N/P	N/P	N/P	N/P	CICACB10	



Snapshot history provides drill downs for detail
View history data in context
Excellent for analysis of issues such as thread conflicts

OMEGAMON XE For DB2 History Collection Summary

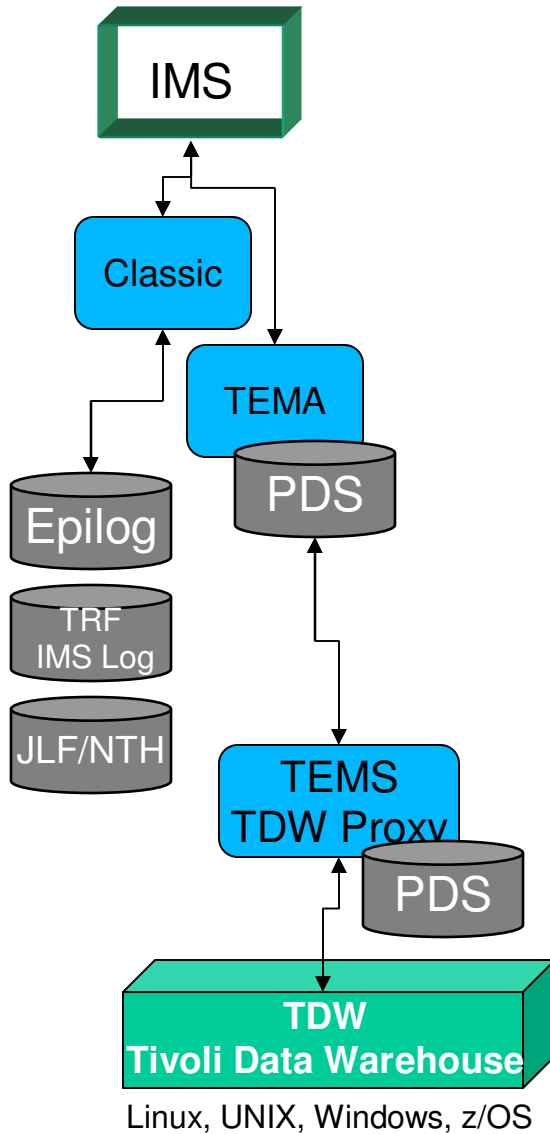


- **Near Term History**
 - Accounting (thread detail) and Statistics stored in a set of VSAM files – primary access via 3270 interface
 - Very detailed – useful for problem analysis
- **Performance Warehouse**
 - DB2 trace data (Accounting, Statistics, Performance) stored in DB2 tables
 - Collection and objects managed by OMEGAMON
 - Detail and quantity of data is variable
- **Performance Database**
 - DB2 trace data (Accounting, Statistics, Performance) stored in DB2 tables
 - Collection and objects managed by user
 - Detail and quantity of data is variable
- **Snapshot history – PE GUI**
 - Snapshots on a user defined interval
 - Easy to view and navigate via the PE GUI interface
- **TDW snapshot history (different from PE GUI)**
 - Use PDS and TDW infrastructure as other OMEGAMONs

OMEGAMON XE For DB2 History Collection Options Considerations And Recommendations

- **Near Term History (NTH)**
 - Detailed history data that is easily accessible
 - NTH is often the most costly to collect in most shops
 - Cost of collection – moderate to high  Value – usually high
- **Performance Warehouse**
 - Detail of data and cost of collection varies depending upon user requirements
 - General recommendation – use when desired for lower cost/quantity data
- **Performance Database**
 - For higher quantity/detail requirements
 - Provides more manual control for higher volumes of history data collection
- **Snapshot history – PE GUI**
 - Easy to access and low cost to collect – requires the PE GUI
 - A low cost alternative to NTH  limitations of snapshot data collection
- **TDW snapshot history**
 - Cost of collection - low
 - Useful for trending analysis, not as detailed NTH or PE GUI snapshot

OMEGAMON XE For IMS History



- **Epilog provides IMS history**
 - Service levels (response times), resource utilization data, and degradation data (bottleneck analysis of IMS workload)
 - Detail is limited - interval/group based
 - Cost – low ↔ Value – moderate
- **Near Term History (NTH)**
 - Transaction detail history (tran level/call level detail)
 - Easy to access via 3270 Classic interface
 - Data collected to Journal Logging Facility (JLF)
 - Cost – moderate ↔ Value – moderate to high
- **Transaction Reporting Facility (TRF)**
 - DB call level detail and summary data written to IMS log
 - Useful for chargeback and some performance analysis
 - Cost – high ↔ Value – limited use/requires batch
- **Snapshot history data and the Tivoli Data Warehouse (TDW)**
 - Data is stored in the TEMA/TEMS in the PDS and optionally sent to the TDW
 - Data may be summarized and pruned using the TDW
 - Cost of collection is low – data is useful for trending analysis

OMEGAMON IMS Historical Data Collection Alternatives

Epilog data is interval based

Useful for analysis of bottlenecks relative to overall system workload

```
EPILOG/IMS V420 09/29/09 7:13 Mode: PAGE
CMD==>
```

```
*****
```

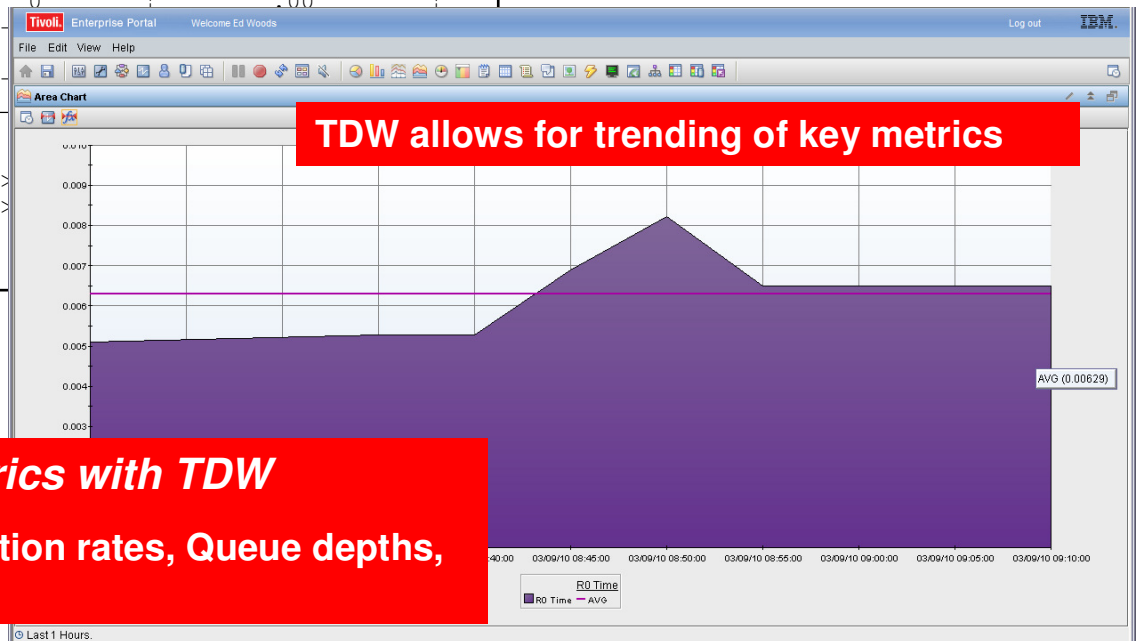
```
+-----+
| Transaction Group = 1      Symbolic Name =
| Period: 05:59 to 06:14 on 09/28/09      Elap = 14:00 M IVP1
```

RESPONSE TIME DATA

Response_Component	Avg. Rsp. Time	Trans. Count	Rate (per min.)
Input Queue	0.00 S	0	.00
Pgm Input Queue	0.05 S	56	3.81
Processing	0.73 S	55	3.74
Response time 0	0.77 S	55	3.74
Output Queue	0.00 S	0	.00
Response time 1	0.00 S	0	.00

DEGRADATION DATA

Competing_State	Time	%	0	1	2
MVS Waits	0.24 S	14.3	----->	.	
PGM Fetch I/O	(0.24) S	(14.3)	----->	.	
IMS Waits	0.49 S	28.6	----->		
Iswitched to CTL	(0.49) S	(28.6)	----->		



TDW allows for trending of key metrics

Trend key IMS performance metrics with TDW

Response time, Bottlenecks, Transaction rates, Queue depths, Buffer/pool performance

Near Term History Of IMS Transactions Useful For Problem Analysis

View a list of recent transactions. Transaction history data is recorded in the Journal Logging Facility (JLF), and viewable in Classic interface.

```

> Help PF1          KOINTVS VTM      01-11 V42
> Back PF3          Up PF7
>
> (H.B.B) View Near-Term History S
>
> A - Manage Trace  * - View Trace  C -
>
NTVS
+ Strt Date\Time  Trancode PSB Name  RGN Name  LTERM      R1 Time    CPU Time  Abend
+-----+-----+-----+-----+-----+-----+-----+-----+
+ 12/01 18:43:27  PART      DFSSAM02 IMS9AMS1  USER0014  00.004384  00.000000
+ 12/01 18:43:27  PART      DFSSAM02 IMS9AMS1  USER0013  00.004491  00.000000
+ 12/01 18:43:28  PART      DFSSAM02 IMS9AMS1  USER0003  00.004200  00.000000
+ 12/01 18:43:28  PART      DFSSAM02 IMS9AMS1  USER0002  00.003657  00.000000
+ 12/01 18:43:28  PART      DFSSAM02 IMS9AMS1  USER0008  00.007028  00.000000
+ 12/01 18:43:28  PART      DFSSAM02 IMS9AMS1  USER0008  00.007028  00.000000
+ 12/01 18:43:28  PART      DFSSAM02 IMS9AMS1  USER0008  00.007028  00.000000
+ 12/01 18:43:28  PART      DFSSAM02 IMS9AMS1  USER0008  00.007028  00.000000
+ 12/01 18:43:28  PART      DFSSAM02 IMS9AMS1  USER0008  00.007028  00.000000
+ 12/01 18:43:29  PART      DFSSAM02 IMS9AMS1  USER0008  00.007028  00.000000
+ 12/01 18:43:29  PART      DFSSAM02 IMS9AMS1  USER0008  00.007028  00.000000
+ 12/01 18:43:29  PART      DFSSAM02 IMS9AMS1  USER0008  00.007028  00.000000
    
```

F11 drill down for detail

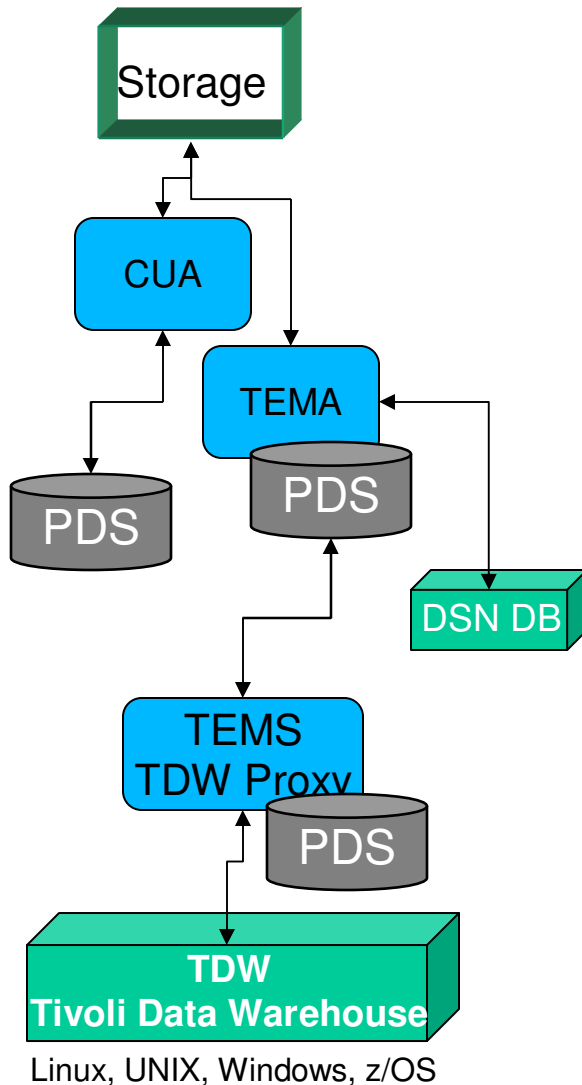
```

KOINTVW VTM      01-11 V420./C I91A 12/01/08 18:53:12 B
Back PF3      Up PF7      Down PF8      Zoom PF11
>
> (H.B.B) View Near-Term History Overview
>
> A - Application Trace Facility
>
NTVW
+ Transaction . . . . . PART      PSB . . . . . DFSSAM02
+ Logical Terminal . . . . . USER0008  Transaction Class . . . . . 001
+ Region Type . . . . . MPP      Message Source . . . . . TERM
+ Region ID . . . . . 4      Primed Message . . . . . NO
+ Jobname . . . . . IMS9AMS1  Step Name . . . . . REGION
+ UserID . . . . . USER0008  Quick Schedule . . . . . NO
+ Abend Code . . . . .      Current SPA Size . . . . . N/A
+ Start Date . . . . . 12/01/08  Start Time . . . . . 18:43:28.202
+ Total Elapsed Time . . . . . 00:00:00.003.220  Total CPU Time . . . . . 00:00:000.000
+ Response Time (R0) . . . . . 00:00:00.007.175  Storage Used <16mb . . . . . 152K
+ Response Time (R1) . . . . . 00:00:00.007.028  Storage Used >16mb . . . . . 1184K
    
```

View call level detail for specific transactions

Type	Count	Total Elapsed Time (mm:ss.ttt.iii)	Average Elapsed Time (mm:ss.ttt.iii)
DLI DB GU	1	00:00.000.115	00:00.000.115

OMEGAMON XE For Storage History



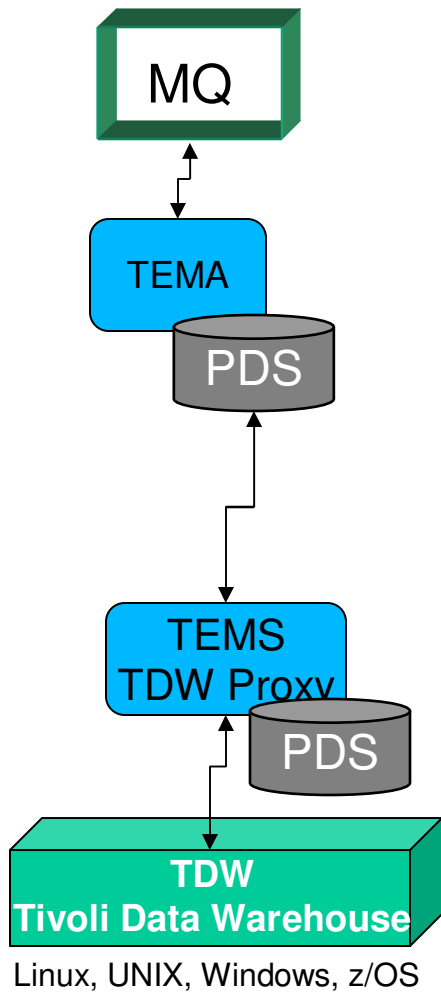
- OMEGAMON XE For Storage makes extensive use of the Persistent Data Store (PDS) for data collection
- PDS data may be accessed by both the CUA 3270 and Tivoli Enterprise Portal interfaces
- OMEGAMON Storage provides numerous product provided Tivoli Enterprise Portal history workspaces
- Cost of collection
 - Potentially high since many shops may have thousands of devices to gather information about
 - Observe best practices for OMEGAMON Storage monitoring
 - Avoid redundant monitoring of devices
 - Group related devices and use wild cards to set options
 - Consider options carefully when monitoring at the application and data set level
 - When defining history in the TEP and TDW consider quantity of data being collected
 - Number of devices, controllers, data sets, applications
- Value can be high, but so can cost

OMEGAMON XE For Storage Provides Trending/History Information At Several Levels

The screenshot displays the OMEGAMON XE For Storage interface. A central menu lists various trending and history options, with red arrows pointing to specific items. The interface includes several charts: 'Write Hit Percent' (bar chart), 'DFW Hit Percent' (bar chart), 'Track Destaging Rate' (bar chart), and 'Cache CU Performance Report' (table). The table at the bottom provides detailed performance metrics for Subsystem ID 2585.

Subsystem ID	Control Unit Type	Read Hit Percent	Write Hit Percent	Read I/O Percent	Bypass Cache Percent	Inhibit Cache Percent	DFW Hit Percent	DFW Retry Percent	CFW Read Percent	CFW Write Percent	Track Destaging Rate	Track Staging Rate	Storage Facility Series
2585	2105	99.2	99.6	93.7	0.0	0.0	100.0	0.0	n/a	n/a	71.5	43.5	Shark

OMEGAMON XE For Messaging History



- OMEGAMON XE For Messaging provides snapshot history data and supports the Tivoli Data Warehouse (TDW)
 - Data is stored in the TEMA/TEMS in the Persistent Data Store (PDS)
 - Data may be optionally sent to the TDW
 - Data may be summarized and pruned using the TDW
- OMEGAMON XE For Messaging provides many history workspaces out of the box
 - Examples of product provided workspaces include
 - Queue statistics, tran/program statistics by queue, Message statistics, Page set statistics, Message manager performance, Log manager performance, Channel performance
- Snapshot data is easy to access within the Tivoli Portal
 - Cost of collection is low ↔ value is moderate to high

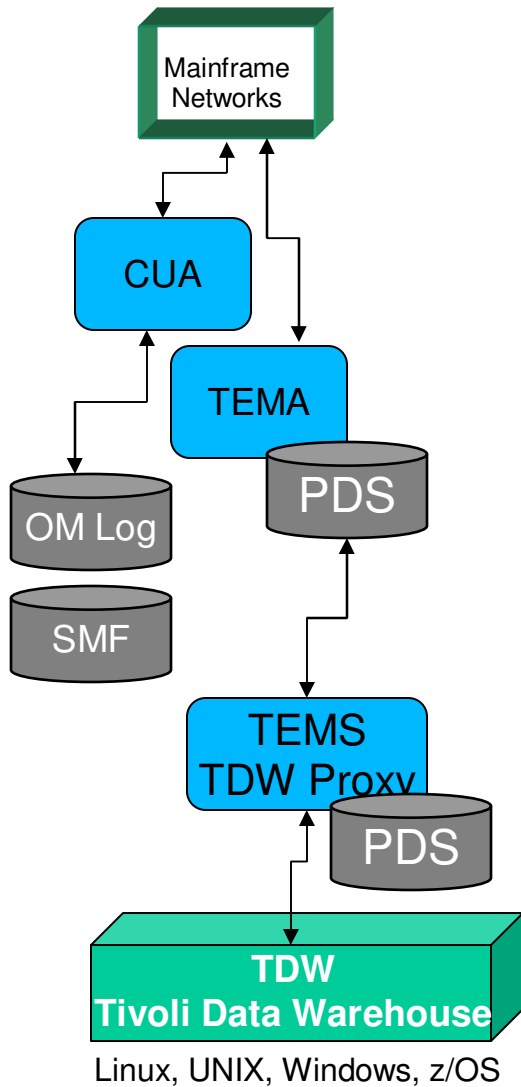
Accessing OMEGAMON Messaging History Data

OMEGAMON Messaging product provided real time and historical workspaces

Drill down for history within the TEP

Queue Name	Queue Usage	Definition Type	Ret Inw Exceeded	Get Status	Put Status	Cu De	Creation Date & Time	Storage Class
SYST...	Normal	Predefin...	No	Enabl...	Enabl...	Yes	09/12/06 23...	DEFAU...
m60n...	Normal	Predefin...	No	Enabl...	Enabl...	Yes	09/19/06 23...	DEFAU...
KMQ...	Normal	PermDyr	No	Enabl...	Enabl...	Yes	09/19/06 21...	DEFAU...
KMC.L...	Normal	PermDyr	No	Enabl...	Enabl...	Yes	09/19/06 21...	DEFAU...
KMQ...	Normal	PermDyr	No	Enabl...	Enabl...	Yes	09/20/06 18...	DEFAU...
CICS...	Normal	Predefin...	No	Enabl...	Enabl...	Yes	09/20/06 18...	DEFAU...
KMQ...	Normal	PermDyr	No	Enabl...	Enabl...	Yes	09/20/06 18...	DEFAU...
CSQ4...	Normal	Predefin...	No	Enabl...	Enabl...	Yes	09/24/06 19...	DEFAU...
CSQ4...	Normal	Predefin...	No	Enabl...	Enabl...	Yes	09/24/06 19...	DEFAU...

OMEGAMON XE For Mainframe Networks History



- OMEGAMON XE for Mainframe Networks provides snapshot history data and supports the Tivoli Data Warehouse (TDW)
 - Data is stored in the TEMA/TEMS in the Persistent Data Store (PDS)
 - Data may be optionally sent to the TDW
 - Data may be summarized and pruned using the TDW
- When configuring history in the TEP/TDW
 - Be aware of relative number of rows per snapshot and snapshot frequency when specifying collection
 - Example- application level versus connection level history
- OMEGAMON XE For Mainframe Networks provides trending history log in the CUA 3270 interface
 - Data is logged and viewable in CUA
 - Recommendation - the most current information is in the Tivoli Portal, therefore focus history efforts in the TEP
- Cost of collection relative to value

- CUA log – typically low	↔	Limited data – use TEP
- Snapshot – typically low	↔	Useful for trending/analysis

OMEGAMON Mainframe Networks Example A Custom Workspace Showing Network Problem Indicators

Create a custom workspace tracking metrics that may indicate potential network issues

Provide links to related workspaces for further analysis

Link to TCPIP application detail

Plot potential TCPIP stack problem indicators

Tabular performance history

Collection Time	Receive Datagram Rate	Input Packet Count (in G)	Input Packet Count	Output Packet	Output Packet	Transmit Datagram	Input Datagrams	Input Datagram	Input Datagrams Error	Input Datagrams Delivered	Input Datagram Delivery Rate	Input Discards	Input Discard Percentage	Output Discards	Output Discard Percentage
02/09/10 11:02:20	41977	0	6205						0	264498	52900	0	0	0	0

Using History To Become More Proactive

- A strategy to be more proactive
 - ***Visualize - Control - Automate***
- Use history data to improve the visualization of system activity and resource utilization
 - Use history data to identify peaks/valleys/bottlenecks
 - Use trending and visualization to identify potential issues
- Use history to improve control
 - Customize workspaces, views and navigation
 - Identify and isolate issues and take corrective actions
- Use history to improve automation
 - Improve alerts by making situation thresholds more accurate and relevant
 - History data can be used as a reference point to make sure threshold levels in situations reflect real problems

Considerations For Collecting Tivoli Data Warehouse Snapshot History Data

- Avoid the “turn on everything” method
 - Turning on everything will result in a fire-hose of information that will potentially obscure useful information, waste space, increase cost of collection, and slow down data recall

- Project potential volume of history being collected
 - Warehouse projection worksheet provides a means to calculate
 - Here is a link to documentation for the tool:
 - http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/index.jsp?topic=/com.ibm.itm.doc_6.2.2fp1/ch2.3warehousecon.htm

- Consider options for history data retention
 - Many deploy TDW with DB2 on a Linux/Unix/Windows type platform to collect and house data
 - You may optionally store your history data on DB2 on z/OS
 - Requires DB2 on z/OS at the V9 level, or above
 - Here is a link to a white paper that goes through the setup of TDW on DB2 on z/OS:
 - [http://www-03.ibm.com/support/techdocs/atmastr.nsf/5cb5ed706d254a8186256c71006d2e0a/b327c2b1683071e28625786400634a7f/\\$FILE/TDW_DB2_ZOS_Considerations.pdf](http://www-03.ibm.com/support/techdocs/atmastr.nsf/5cb5ed706d254a8186256c71006d2e0a/b327c2b1683071e28625786400634a7f/$FILE/TDW_DB2_ZOS_Considerations.pdf)

Enabling TDW History Collection

Group	Prune Detailed	Summarize Hourly	Prune Hourly	Summariz Daily
System CPU Utilization	30 Days			On
System Paging Activity	30 Days			On

Configuration Controls

Summarization

- Yearly
- Quarterly
- Monthly
- Weekly
- Daily
- Hourly

Pruning

- Yearly keep [] Years
- Quarterly keep [] Years
- Monthly keep [] Months
- Weekly keep [] Months
- Daily keep [1] Years
- Hourly keep [] Days
- Detailed data keep [30] Days

Example – specify System CPU Utilization history collection

Summarization and retention options

Cost of collection/retention is usually a function of frequency of collection and number of rows per snapshot

Consider warehouse interval to avoid surges of data to TDW

Basic | Distribution

Attribute Group: System CPU Utilization

Name: System CPU

Description:

Configuration

- Collection Interval: 15 minutes
- Collection Location: TEMS
- Warehouse Interval: 1 hour

Specify snapshot interval and frequency of sending data to TDW

Example - Use The TEP To Create A Custom Workspace As A Starting Point For Historical Data Analysis

System CPU Utilization

	Average CPU Percent	RMF MVS CPU Percent	RMF LPAR CPU Percent	Total TCB%	Total SRB%	Average IFA Percent	Average IFA on CP Percent	Average zIIP Percent	Average zIIP on CP Percent	MVS Overhead	4 Hour MSUs	HiperDispatch Management	Partition LCPD%	Partition PCPD%
Real time	38	11.5	32,767.0	12	2	0	0	0	0	3	Unavailable	Unavailable	17	17

System CPU Utilization Interval History

Recording Time	Average CPU Percent	RMF MVS CPU Percent	RMF LPAR CPU Percent	Total TCB%	Total SRB%	Average IFA Percent	Average IFA on CP Percent	Average zIIP Percent	Average zIIP on CP Percent	MVS Overhead	4 Hour MSUs	HiperDispatch Management
05/26/11 01:00:00	9	11.5	32,767.0	9	2	0	0	0	0	3	Unavailable	Unavailable
05/26/11 01:15:00	0	0	0	0	0	0	0	0	0	3	Unavailable	Unavailable
05/26/11 01:30:00	0	0	0	0	0	0	0	0	0	3	Unavailable	Unavailable
05/26/11 01:45:00	0	0	0	0	0	0	0	0	0	3	Unavailable	Unavailable

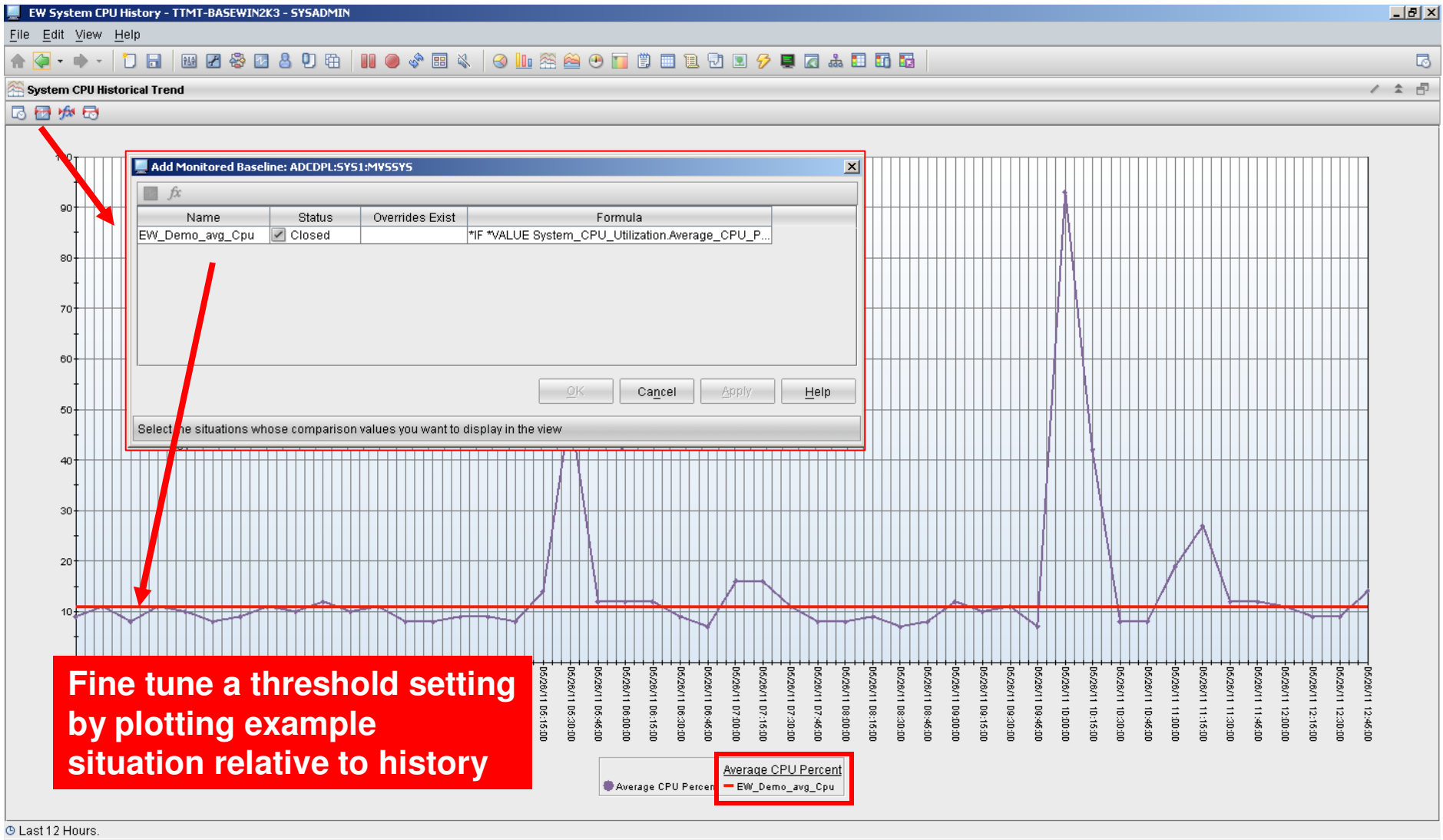
System CPU Historical Trend

History plot chart
Average CPU over the past 12 hours

Legend: Average CPU Percent

Last 12 Hours.

Use A Situation To Track A Monitored Baseline Help Determine Where To Set A Threshold Level



Fine tune a threshold setting by plotting example situation relative to history

Use Arithmetic Functions To Trend History

Show arithmetic data such as Average or Min/Max relative to trend over time

Average CPU Percent	RMF MVS CPU Percent	RMF LPAR CPU Percent	Total TCB%	Total SRB%	Average IFA Percent
8	11.5	32,767.0	18	3	0

Add Statistical Baseline

Name	Argument	Result
<input type="checkbox"/> RANGE - MIN/MAX		
<input checked="" type="checkbox"/> AVG	+/- 0 standard deviation	
<input type="checkbox"/> MIN	+/- 0 percent	
<input type="checkbox"/> MAX	+/- 0 percent	
<input type="checkbox"/> PERCENTILE	50	
<input type="checkbox"/> MODE		

Attribute: Average CPU Percent

Time Span: Last 24 Hours

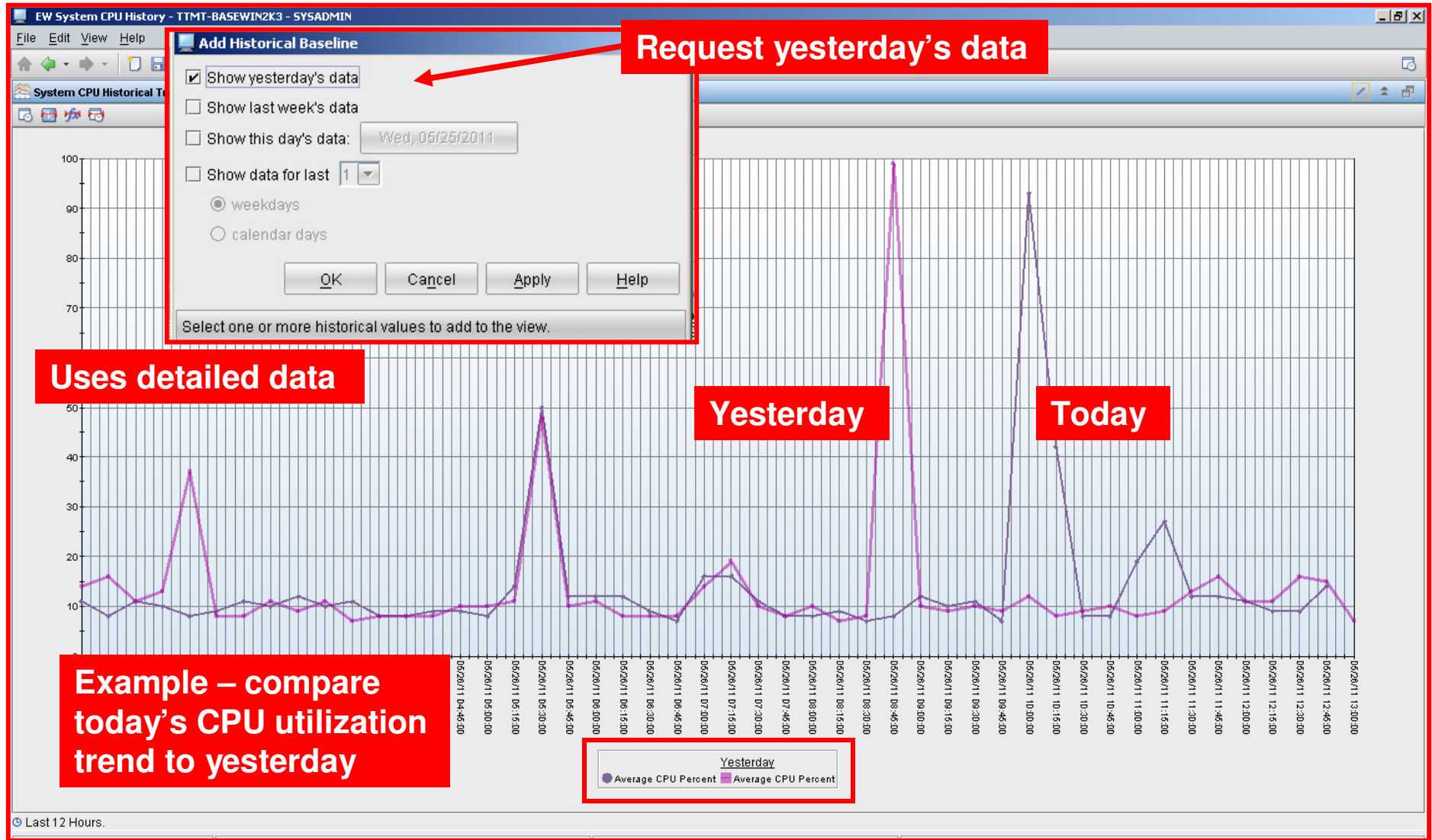
Specify time span for calculation

Select one or more statistical values to add to the view.

Example – how often is CPU utilization above average?

Legend: Average CPU Percent (blue line), AVG (purple line)

Use Historical Baseline To Compare Past Trends To Current Trends



Create Model Situations Using History Data

Model Situation

Time Span: 12 Hours, Time Column: Recording Time

Name	Argument	Result
AVG	+/- 0 standard deviation	
MIN	+/- 0 percent	
<input checked="" type="checkbox"/> MAX	+/- 0 percent	93
PERCENTILE	50	
MODE		

Buttons: Recalculate, Create Situation..., Cancel, Help

Situations for - System CPU Utilization

Name: EW_Demo_Alert

Description:

Formula:

Formula
Average CPU Percent
1 > 93
2
3

Situation Formula Capacity: 5%

Sampling interval: 0:0:15:0

Sound: Enable critical.wav

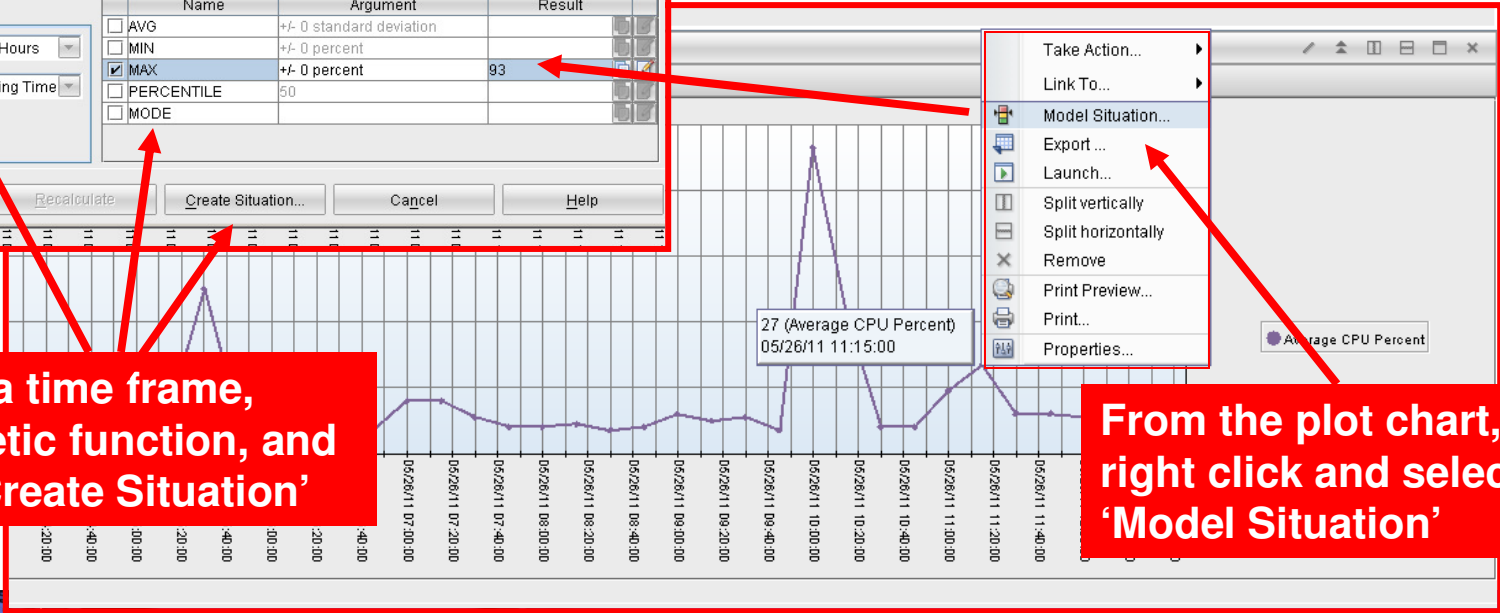
State: Critical

Buttons: Play, Edit..., Run at startup

Situation editor

Select a time frame, arithmetic function, and click 'Create Situation'

From the plot chart, right click and select 'Model Situation'



History Links And Drill Down

Select the Time Span

Real time
 Real time plus Last Hours
 Last Hours

Last parameters

Use detailed data
 Time Column:
 Use summarized data
 Shift:
 Days:

Custom

Custom parameters

Use detailed data
 Time Column:
 Use summarized data
 Interval:
 Shift:
 Days:
 Start Time: End Time:

Apply to all views associated with this view's query Lock time span for Historical Navigation
 Use Hub time

OK Cancel Help

Specify history time frame

EW Addr Space CPU - TTMT-BASEWIN2K3 - SYSADMIN

File Edit View Help

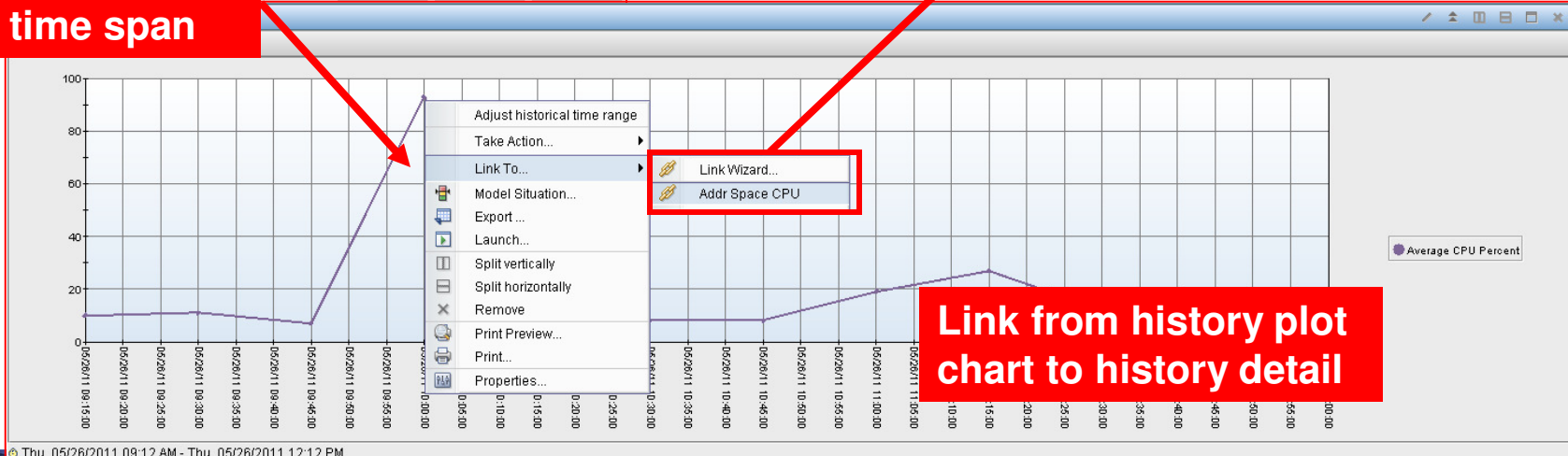
Navigator View: Physical

Address Space CPU Utilization Summary

Recording Time	Job Name	Step Name	Proc Step	Type
05/26/11 10:00:01	SMFDMP	SMFDMP	IFASMPDP	STC
05/26/11 10:00:01	CXEGA01	CXEGA01	TEMS	STC
05/26/11 10:00:01	XCFAS	XCFAS	IEFFPROC	STC
05/26/11 10:00:01	CATALOG	CATALOG	IEFFPROC	STC
05/26/11 10:00:01	WLM	WLM	IEFFPROC	STC
05/26/11 10:00:01	TCPIP	TCPIP	TCPIP	STC
05/26/11 10:00:01	SMSPDSE	SMSPDSE	SMSPDSE	STC
05/26/11 10:00:01	JES2MON	JES2MON	IEFFPROC	STC
05/26/11 10:00:01	*MASTER*			STC
05/26/11 10:00:01	JES2	JES2	IEFFPROC	STC
05/26/11 10:00:01	TN3270	TN3270	TN3270	STC
05/26/11 10:00:01	HTTPD1	HTTPD1	WEBSRV1	STC
05/26/11 10:00:01	PCAUTH	PCAUTH		STC
05/26/11 10:00:01	RASP	RASP		STC
05/26/11 10:00:01	TRACE	TRACE		STC
05/26/11 10:00:01	DUMPSRV	DUMPSRV	DUMPSRV	STC
05/26/11 10:00:01	GRS	GRS		STC
05/26/11 10:00:01	CONSOLE	CONSOLE		STC
05/26/11 10:00:01	ANTMAIN	ANTMAIN	IEFFPROC	STC
05/26/11 10:00:01	ANTAS000	ANTAS000	IEFFPROC	STC
05/26/11 10:00:01	DEVMAN	DEVMAN	IEFFPROC	STC
05/26/11 10:00:01	OMVS	OMVS	OMVS	STC

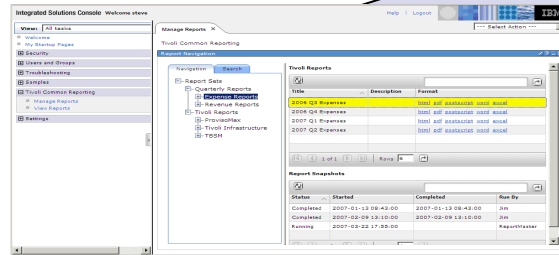
Address space history

Select lock time span



Historical Reporting Options

- Tivoli Common Reporting (TCR) provides a consistent reporting solution shared across the Tivoli Portfolio
 - TCR included as part of the OMEGAMON suite



- OMEGAMON z/OS provides batch Epilog reporting
- OMEGAMON IMS provides batch Epilog reporting
- OMEGAMON CICS provides some basic reports
- OMEGAMON DB2 provides a detailed/robust best-of-breed historical reporting suite

Improve Your Analysis

Additional Components To Consider For Historical Data Collection And Analysis

- **Tivoli Decision Support For z/OS**
 - Generate customized reports to communicate system performance, capacity management, resource availability and cost allocation information
 - Collects data, such as SMF, CICS, IMS performance data
 - Provides a central data repository (DB2) and integrates with the Tivoli Portal
 - Integrates with a variety of Tivoli solutions
- **IMS Performance Analyzer**
 - Provides robust reporting and information on IMS system performance for monitoring, tuning, managing service levels, analyzing trends, and capacity planning
 - Expands the reporting options beyond what's available with OMEGAMON IMS
- **CICS Performance Analyzer**
 - Comprehensive performance reporting and analysis for CICS, including use of DB2, WebSphere MQ, IMS, and z/OS System Logger
 - Evaluate CICS system efficiency, eliminate system bottlenecks and proactively tune system performance
 - Expands the reporting options beyond what's available with OMEGAMON CICS

Summary And Conclusions

- Each OMEGAMON monitoring solution offers history along with real time data collection
- Each OMEGAMON has it's own unique considerations specific to history collection and the data that is available
- History data collection is a classic trade-off of cost versus benefit
 - In some scenarios history data collection can be costly
 - Understand the costs versus the benefits
- History is essential to solve problems after the fact
- History is useful to make monitoring more proactive
 - Historical trending to identify peaks/valleys/issues
 - Historical data analysis to optimize alerts and thresholds

Check Out My Blog

http://tivoliwithaz.blogspot.com

The screenshot shows a browser window titled "Tivoli With A z - Microsoft Internet Explorer" displaying a blog post. The browser's address bar shows "http://tivoliwithaz.blogspot.com/". The blog header features the title "Tivoli With A z" and a description: "This is a blog to discuss what is happening in the area of IBM z/Series, Tivoli, OMEGAMON monitoring, System Automation, and other relevant IBM Tivoli technology for z/OS performance and availability management." The author is identified as Ed Woods, IBM Corporation.

The main content of the post is dated "Friday, February 5, 2010" and titled "OMEGAMON DB2 Near Term History". It includes two screenshots of OMEGAMON DB2 terminal output. The first screenshot shows the "NEAR-TERM HISTORY" menu with options for "COLLECTION OPTIONS", "RECORD INFORMATION", and "CURRENT STATUS". The second screenshot shows the "DB2 Record Information" table, which lists record counts for various types such as Accounting, DB2 Section, Perf Section, and Perf-Dyn SQL.

The text of the post explains that OMEGAMON DB2 has a useful Near Term History (NTH) function. It states that NTH provides an easy way to retrieve and review DB2 Accounting and Statistics records from the past few hours of DB2 processing. The data is stored in a set of VSAM files allocated to the OMEGAMON collection task. The amount of data being written to these files is driven by the DB2 workload activity. Accounting records are typically written when a DB2 thread terminates processing, and it is the Accounting data that is often looked at by the analyst when studying what DB2 applications have been doing. Statistics records are created on a time interval basis. Usually, you will have much more accounting data than statistics data. Also, OMEGAMON has the ability to pull in additional trace IFCIDs to get information on things such as dynamic SQL activity.

To understand the amount of data being gathered by NTH, there are displays that show the number of records written to the NTH files, by type. In the example I show, you see an example of common NTH settings/options, and then you see the record count in the NTH record information display. If you look carefully you see that 'Perf-Dyn SQL' has a lot of records written relative to the other record types. This is a good way to understand the impact of enabling certain collection options, such as dynamic SQL collection, and see how many trace records are being gathered, as a result.

The post is signed "Posted by Ed Woods at 3:13 PM 0 comments".

On the right side of the blog, there is a bio for Ed Woods: "ED WOODS I'm an IT Specialist with IBM Corporation supporting Tivoli Performance solutions on z/OS. Please note that comments made on this blog are my own, and do not necessarily reflect the position of IBM Corporation. View my complete profile". Below the bio are sections for "Links To My Articles" and "Useful Links", each containing several hyperlinks to related content.

Thank You!