Understanding The Impact Of The Network On z/OS Performance

Ed Woods - IBM Corporation
Session #16737
Wednesday, March 4th: 11:15 AM - 12:15 PM
Agenda

- Introduction
- Looking at the application time line
- Why monitor mainframe networks
  - What are the typical concerns
- Examples of mainframe/network interaction
  - Examples for various subsystems
- Defining a consistent monitoring strategy
The Challenges Of Performance And Availability Management Of Complex Systems

- Most new applications are composite by design
  - Applications cross multiple subsystems and platforms
  - Integration and utilization of multiple core technologies
  - Pose challenges from a management and monitoring perspective

- Common Technical Challenges
  - Multiple platforms
  - Potentially multiple DB systems
  - Middleware considerations
  - One or multiple network hops

- *Is the problem the network, the host, the DB, the client, or somewhere in between?*
Portions of response time may reside in any of the following:
- End user client processing, the application server or middleware level, the database, or other aspects of host z/OS application processing
- Potential for bottlenecks at multiple points

The network will impact the overall application time line:
- Time is required to send messages across the network
- Overhead processing, including communication subsystem session management
- Network hardware, traffic, connections, connection pools
The Impact Of The Network On Critical z/OS Components

- The network has impact on z/OS workload in many ways
- Speed of the network – network congestion and bottlenecks
- Each z/OS application or component subsystem has unique network considerations
  - IMS, DB2, CICS, MQSeries, WebSphere, FTP
- Keep in mind that z/OS application/subsystem configuration and logic may also impact the network
  - Subsystem configuration options and settings impact network interaction
  - Application logic potentially impacts network usage and performance
Categories Of z/OS Network Concerns
Why Monitor z/OS Network Activity?

- It’s important to monitor and see the full picture
  - Monitor the z/OS host and its subsystems
  - Monitor network activity from the z/OS perspective
    - Don’t rely on another group to monitor network activity

- Categories of issues
  - Application logic and design issues
    - How (and how efficiently) does the application interact?
  - Subsystem and configuration issues
    - Are optimal subsystem options being used?
  - Network congestion issues
    - Is the issue inside the network itself?
Example Scenarios - DB2 Has Several Potential Bottlenecks

Many methods of connection to DB2

z/OS

IMS
CICS
Batch
TSO
WebSphere

Network delays
Connection bottlenecks
Connection bottlenecks
Network delays
Network delays
Network delays

MSTR
Connections Threads
Logging

DBM1
SQL

EDM
PTs, SKPTs, CTs, SKCTs
DSC – Dynamic SQL

Sort Pool
RID Pool

DB, BP I/O delays
Lock Conflicts

DB2 Connect

Network delays

Application

Network delays

Application

DDF
Distributed threads

SP Sched Delays

SP Addr Space(s)
Stored Procedures
UDFs
The trade-off – the cost of network interaction versus quantity of data
- Do more with SQL to eliminate redundant back/forth activity
- Crossing more layers will mean more overhead
- However – larger results will mean more network time
## An Application Logic Example

**DB2 Statistics Trace Data For The DB2 Subsystem**

**How Much Data is Sent And Received?**

<table>
<thead>
<tr>
<th>Thread creation queues?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Is buffering occurring?</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HWM of DBAT usage</th>
</tr>
</thead>
</table>

### DISTRIBUTED DATA FACILITY STATISTICS

```
> DFST
+ Collection Interval: REALTIME                    Start: 06/04 13:40:46
+ Report Interval: 4 sec                           End: 06/04 13:40:49
+ Location Name = NDCDB203  DDF Status = ACTIVE  
+ Active DBATs = 3  Inactive DBATs = 0             
+ DDF Send Rate = 0K/sec  DDF Receive Rate = 0K/sec
+ Resync Attempts = 0  Resync Successes = 0       
+ Cold Start Connections = 0  Warm Start Connections = 0
+ DBAT Queued = 0  Conversations Dealloc = 0      
+ HWM All DBATs = 5  HWM Active DBATs = 5         
+ Max DB Access (MAXDBAT) = 500  HWM Inactive DBATs = 0

+ Remote Location Name = DRDA REMOTE LOCS

+ Conversations Queued = 0  Binds for Remote Access = 0
+ Message Buffer Rows = 174874  Block Mode Switches = 0
+ Commits/Remote = 0  Rollbacks/Remote = 0
+ Indoubts/Remote = 0

+ Tran  SQL  Row Message  Byte  Commit  Abort  Conv  Blocks
+ Sent 0 0 174927 2486 43164569 0 0 0 1746
+ Recv 55 1714 0 2285 238429 363 8 55 0
```

- **Recv** – received into DB2
- **Sent** – out to client/apps

---

**Bytes and messages sent and received**

**Thread creation queues?**

**Is buffering occurring?**

**HWM of DBAT usage**

**Recv – received into DB2**

**Sent – out to client/apps**
Looking At The DB2 Application Thread Level DB2 Accounting Information Analysis

<table>
<thead>
<tr>
<th>PLAN</th>
<th>Thread: Plan=DISTSERV Connid=SERVER Corrid=db2bp.exe Authid=DNET581</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dist</td>
<td>Type=DATABASE ACCESS, Luwid=G941491B.FC10.090604182432=169</td>
</tr>
<tr>
<td>Location</td>
<td>9.65.73.27</td>
</tr>
</tbody>
</table>

Note row versus message/block counts to determine blocking

<table>
<thead>
<tr>
<th>Distributed TCP/IP Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>+9.65.73.27</td>
</tr>
</tbody>
</table>

Note the quantity of data being sent

- Recv – received into DB2
- Sent – out to client/app

# SQL calls
# of messages sent
DRDA blocks for queries
DB2 Subsystem Configuration Example
How Subsystem (zparm) Options May Impact The Workload

New Connection

- Reject
- CONDBAT?
  - Yes
  - Resumed Connection
    - Pooled DBAT Avail?
      - Yes
      - Reuse DBAT
        - Pool DBAT / Inactv. Conn.
      - No
        - MAXDBAT Reached?
          - Yes
            - Queue
          - No
            - Create DBAT
              - Reply “ready” to client
              - Process SQL
              - End processing

- No
  - Pool DBAT / Inactv. Conn.

Note – DB2 has changed the settings with newer releases
Increases in MAXDBAT, IDFORE, IDBACK, and CTHREAD.

zparm for queue wait time
## Monitoring For Subsystem Configuration Issues

Are There Connections In Backlog Or Rejected?

- Connection activity, connection counts, connection backlogs
  - Look for applications with connection failures and backlogs
What About Potential Network Issues?

- Look for indicators, such as Retransmission, Discard, and Fragmentation counts
  - Note – Example from OMEGAMON for Mainframe Networks
**NETSTATAT Connection Detail**

### netstat all (port 448)

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>EZZ2350I</td>
<td>MVS TCP/IP NETSTAT CS V1R10</td>
<td>TCPIP Name: TCPIP</td>
</tr>
<tr>
<td>EZZ2550I</td>
<td>Client Name: DSNCDIST</td>
<td>Client Id: 0000C90E</td>
</tr>
<tr>
<td>EZZ2551I</td>
<td>Local Socket: 9.39.68.147..448</td>
<td>Foreign Socket: 9.65.73.27..4255</td>
</tr>
<tr>
<td>EZZ2552I</td>
<td>Last Touched: 19:14:58</td>
<td>State: Establish</td>
</tr>
<tr>
<td>EZZ2577I</td>
<td>BytesIn: 00000006973</td>
<td>BytesOut: 00086457981</td>
</tr>
<tr>
<td>EZZ2574I</td>
<td>SegmentsIn: 0000003423</td>
<td>SegmentsOut: 00000006614</td>
</tr>
<tr>
<td>EZZ2553I</td>
<td>RcvNxt: 3808791478</td>
<td>SndNxt: 2538223807</td>
</tr>
<tr>
<td>EZZ2554I</td>
<td>ClientRcvNxt: 3808791478</td>
<td>ClientSndNxt: 2538223807</td>
</tr>
<tr>
<td>EZZ2555I</td>
<td>InitRcvSeqNum: 3808784504</td>
<td>InitSndSeqNum: 2529765825</td>
</tr>
<tr>
<td>EZZ2556I</td>
<td>CongestionWindow: 0000017349</td>
<td>SlowStartThreshold: 0000002620</td>
</tr>
<tr>
<td>EZZ2557I</td>
<td>IncomingWindowNum: 3808824236</td>
<td>OutgoingWindowNum: 2538289289</td>
</tr>
<tr>
<td>EZZ2558I</td>
<td>SndW11: 3808791478</td>
<td>SndW12: 2538223807</td>
</tr>
<tr>
<td>EZZ2559I</td>
<td>SndWnd: 0000065482</td>
<td>MaxSndWnd: 0000131070</td>
</tr>
<tr>
<td>EZZ2560I</td>
<td>SndUna: 2538223807</td>
<td>rtt_seq: 2538223753</td>
</tr>
<tr>
<td>EZZ2561I</td>
<td>MaximumSegmentSize: 00000001310</td>
<td>DSField: 00</td>
</tr>
<tr>
<td>EZZ2563I</td>
<td>Round-trip information:</td>
<td>SmoothTripTime: 184.000</td>
</tr>
<tr>
<td>EZZ2564I</td>
<td>Smooth trip time: 184.000</td>
<td>SmoothTripVariance: 84.000</td>
</tr>
<tr>
<td>EZZ2565I</td>
<td>ReXmt: 0000000002</td>
<td>ReXmtCount: 0000000000</td>
</tr>
<tr>
<td>EZZ2572I</td>
<td>DupACKs: 0000000284</td>
<td>RcvWnd: 0000032758</td>
</tr>
<tr>
<td>EZZ2566I</td>
<td>SockOpt: 8D</td>
<td>TcpTimer: 00</td>
</tr>
<tr>
<td>EZZ2567I</td>
<td>TcpSig: 04</td>
<td>TcpSel: 40</td>
</tr>
<tr>
<td>EZZ2568I</td>
<td>TcpDet: EC</td>
<td>TcpPol: 00</td>
</tr>
<tr>
<td>EZZ2537I</td>
<td>QOSPolicy: No</td>
<td></td>
</tr>
<tr>
<td>EZZ2542I</td>
<td>RoutingPolicy: No</td>
<td></td>
</tr>
<tr>
<td>EZZ2570I</td>
<td>ReceiveBufferSize: 00000016384</td>
<td>SendBufferSize: 0000065536</td>
</tr>
<tr>
<td>EZZ2538I</td>
<td>ReceiveDataQueued: 0000000000</td>
<td></td>
</tr>
<tr>
<td>EZZ2539I</td>
<td>SendDataQueued: 0000000000</td>
<td></td>
</tr>
</tbody>
</table>
IMS Has Many Potential Bottlenecks (Including Network)
IMS Connect And The Interaction With TCP/IP

IMS Control Center
IMS DB Adapters
WebSphere IMS TMRA
IMS SOAP Gateway
IMS Connect API Clients

TCP/IP
IMS Connect
IMS Connect Extensions

IMS
OTMA
TM
IMS Application
IMS DB

ODBM
Operations Manager
IMS Connect Extensions OMEGAMON IMS
IMS Connect
Types Of Connections & Message Considerations

- IMS Connect – types of connections
  - Non-Persistent socket
    - Closes after each send to the ICON client
  - Transaction socket
    - Close after each transaction or conversation - The default
    - Has connect/disconnect overhead for each message
  - Persistent socket
    - Read/writes for multiple transactions
    - Typically more efficient
    - Will keep the socket open – make sure you have enough sockets

- Message considerations
  - General ROT – use one send for the entire message
  - If doing multiple writes then specify NODELAYACK on PORT statement in z/OS
    - If not specified then may wait up to 300ms for each transmission
IMS Network Examples Of Relevant Options And Parameters

- **PROFILE.TCPIP parameters**
  - **PORT**
    - Reserve ports for IMS Connect
      - Include the NODELAYACK parameter for multi-message applications
      - Example benchmark
  - **SOMAXCONN**
    - Controls the queue depth for listening sockets at the LPAR level
      - Works in conjunction with TCPIPQ parm

- **IMS Connect parameter MAXSOC**
  - Total number of sockets IMS Connect supports across all ports at the same time
  - IMS Connect issues warning message HWSS0772W when the number of sockets reaches the default warning threshold of 80 percent of MAXSOC
  - When the number of sockets reaches the MAXSOC limit, IMS Connect refuses any new connections and issues message HWSS0771W

- **TCPIPQ** – configure the depth of the queue for connection requests
IMS Connect Monitoring
An Example

Note – This display requires IMS Connect Extensions And OMEGAMON IMS
The Network Impacts CICS Processing

- Network potentially impacts CICS in a variety of ways
  - Connections to CICS – connections via a variety of means
  - Communication within CICS - ISC and MRO
    - InterSystems Communication - system to system, Multi-Region Operation - region to region, and IPIC – IP InterCommunications

Network (SNA or TCP/IP)

TOR

AOR

Network delays

AOR

Network delays

TOR

Network delays

AOR

Network delays

XCF/MRO – Cross System CF
CICS Sockets Versus CICS Sockets Domain

- **CICS Sockets** – a component of Communications Server for z/OS
  - General purpose socket API for use by CICS programmers

- **CICS Sockets Domain** – a component of CICS TS
  - Does not have direct access to the socket
  - Communicates with CICS Socket Domain Services
CICS Network
Examples Of Relevant Parameters

- TCPIPSERVICE (TCPDEF) parameter defines services
  - ECI over TCP/IP (for CICS Clients), IIOP, CICS Web support (HTTP), IPIC (ISC)
  - For use only with the CICS-provided TCP/IP services, and have nothing to do with the z/OS Communications Server IP CICS Sockets interface

- SOCKETCLOSE parameter
  - Set to NO (the default) or a time interval to avoid socket creation overhead

- BACKLOG parameter
  - Specifies the maximum number of inbound TCP/IP connection requests that can be queued in TCP/IP for CICS processing
  - When the maximum number is reached, TCP/IP rejects additional connection requests

- MAXSOCKETS – CICS SIT parameter
  - Maximum number of IP sockets that can be managed by the CICS sockets domain
  - If the CICS region userid does not have superuser authority, the maximum possible value is the value of the MAXFILEPROC parameter in SYS1.PARMLIB member BPXPRMxx.
  - MAXSOCKETS and maximum tasks (MXT)
    - Recommendation - MAXSOCKETS should not be a subset of MXT
CICS Transaction Server 5.2 And Liberty

- **Liberty**
  - Initially introduced in CICS 51
  - Light weight web container
  - Provides Servlets and JSPs support for CICS
  - Runs in a JVMServer
  - Access to remote DB2 resources with T4 driver (JDBC 1.0)

**CICS may access DB2 remote databases - Network implications?**
MQSeries Performance Bottlenecks

MQ Channels
MQ queue processing
MQ queue depth
Application processing

MQ Channels
MQ queue processing
MQ queue depth
Application processing

MQ Channels
MQ queue processing
MQ queue depth
Application processing

Application bottlenecks

Application bottlenecks

Application bottlenecks

Network delays

Network delays

Network delays
MQSeries Performance Bottlenecks

- WebSphere MQ is very dependent on the network
  - Network speed, network traffic and message volume are all key components

**Optimization options**
- Increase network speed
- Compress messages - decreases network transmission by reducing the size of the message.
- Channel parameters
  - Batch size defines the maximum number of messages sent within a batch.
  - Reduces the amount of channel processing required.
  - Note – batching for small applications may result in delays and spikes
MQ Series
Configuration/Application Options And Network Impact

- Consider MQCONN and MQPUT patterns
  - MQCONN connects the application program to the MQ queue manager
    - Note - Cost of MQCONN high
  - MQPUT puts a message on a queue that was opened using MQOPEN
    - Similar to DB2 SQL call scenario
      - Consider cost of back and forth activity versus application logic
- Channel parameters
  - Batchsz – defines the maximum number of messages sent within a batch
    - Reduces the amount of channel processing required
  - Channel message compression
    - Some compression can be CPU heavy - how compressible is the data?
- Persistent versus non-persistent messages
  - Persistent messages are written to logs and queue data files
    - May be recovered by the queue manager after restart from failure
    - Persistent messages may have I/O and logging bottlenecks
  - Non-persistent messages are discarded after a failure
- Fast non-persistent messages
  - NPMSPEED – specifies speed at which non-persistent messages are sent
Defining A Monitoring Strategy
Many Factors May Impact z/OS Application Processing

- Host processing bottlenecks
  - Transaction bottlenecks, application failures/stopped resources, high I/O and poor BP ratios, transaction/message queues, concurrency/lock conflicts

- Network performance
  - Network congestion, data fragmentation, data retransmission

- Network hardware issues
  - Adapter hardware errors, hardware configuration errors, hardware congestion issues

- Application subsystem connection issues
  - Application errors, subsystem configuration errors

- Application issues
  - Application design and logic problems
Defining A Monitoring Strategy
Monitoring At Multiple Levels

- Monitor at the host application subsystem level
  - IMS, CICS, DB2, WebSphere, WebSphere MQ
  - Response time, transaction rates, message rates, queues

- Monitor host application network connection activity
  - Connection activity, connection counts, connection backlogs

- Monitor at the interface level
  - OSA adapters, error counts, fragmentation counts, retransmission counts

- Monitor at the network connection level
  - Response time, traffic counts, error counts, fragmentation counts, retransmission counts

- Integrate host and network monitoring

- Monitor from an end-to-end perspective

Subsystem Monitoring
Both
Network Monitoring
Dashboard level monitoring
Composite level monitoring
Example – Monitoring At The Application Level

- **Mainframe network monitoring**
  - Network time for IMS transactions

- **IMS subsystem monitoring**
  - IMS host response time including queue and processing time for the transaction

Including network monitoring detail provides a more complete analysis of IMS response time.
Another Example Monitoring At The Application Level – DB2

DB2 thread level monitoring

DB2 network level monitoring
Monitor At The Subsystem Level – Configuration And Backlogs

- Connection activity, connection counts, connection backlogs
  - Look for applications with connection failures and backlogs
Monitor For Network Issues
Monitor At The Interface Level

- Monitor for interface status, bandwidth utilization, and errors
- Look for potential problems at the interface level

<table>
<thead>
<tr>
<th>Interface Name</th>
<th>Interface Type</th>
<th>Current State</th>
<th>Transmit Packet Rate</th>
<th>Receive Packet Rate</th>
<th>Transmit Bandwidth Utilization</th>
<th>Receive Bandwidth Utilization</th>
<th>Inbound Packets Discarded</th>
<th>Inbound Packet Discard Rate</th>
<th>Outbound Packets Discarded</th>
<th>Outbound Packet Discard Rate</th>
<th>Percent Packets Discarded in Error</th>
<th>Outbound Packets Discarded in Error</th>
<th>Transmit Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>L0PEBACK</td>
<td>Loopback</td>
<td>Up</td>
<td>7677</td>
<td>7677</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>L0PEBACK2</td>
<td>Loopback</td>
<td>Up</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>EZ6OSX01</td>
<td>OSA_QDIO_ethemet_OSM</td>
<td>Up</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>EZ6OSX02</td>
<td>OSA_QDIO_ethemet_OSM</td>
<td>Up</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>EELEK</td>
<td>Ethemet_virtual</td>
<td>Up</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>OSAFXCDIL</td>
<td>OSA_QDIO_ethemet_OSM</td>
<td>Up</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>OSG3200P</td>
<td>OSA_QDIO_ethemet_OSM</td>
<td>Up</td>
<td>611</td>
<td>524</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>OSG3340P</td>
<td>OSA_QDIO_ethemet_OSM</td>
<td>Up</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>HIPERL5</td>
<td>Hipersocket</td>
<td>Down</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>EZASAMEMVS</td>
<td>MPC_plb_samehost</td>
<td>Up</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>IGDIOLN_KD0A301B3</td>
<td>Hipersocket</td>
<td>Up</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>EAXCF2</td>
<td>MPC_plb_xcf</td>
<td>Up</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>EAXCF3</td>
<td>MPC_plb_xcf</td>
<td>Up</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Dashboard Level Monitoring
Creating An Integrated Performance Interface

- Creating an integrated performance management display allows for the easy inclusion of network detail into various mainframe monitoring displays.

- Integrated monitoring takes several forms:
  - Integrated displays pulling together performance detail from multiple sources (host and network monitoring).
  - Integrated cross monitoring tool navigation.
  - History integrated with real time performance information.
  - Integrated alerts, alert correlation, and corrective actions.
Real time monitoring provides a view of current utilization, status, and alerts.

Provides a view of current status, but is not necessarily ‘predictive’ in nature.
Summary

- Monitor the network from the z/OS perspective
- The network is an essential part of the overall mainframe application time line
  - Each network application/subsystem has interactions with the network
- It’s important to understand how the mainframe interacts with the network
  - Application/subsystem configuration and options
- It is useful to have an integrated monitoring strategy that pulls together core mainframe and network monitoring information
  - Integrated dashboard views, integrated analysis, integrated alerts and automated corrections
  - Defining an end to end analysis strategy
Thank You!
IBM Software Group

Check Out My Blog
http://tivoliwithaz.blogspot.com

Tivoli With A z
This is a blog to discuss what is happening in the area of IBM zSeries, Tivoli, OMEGAMON monitoring, System Automation, and other relevant IBM Tivoli technology for z/OS performance and availability management.

Friday, February 5, 2010
OMEGAMON DB2 Near Term History

OMEGAMON DB2 has a very useful Near Term History (NTH) function. NTH provides an easy way to be able 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. How far back the history goes depends upon the size of the files and the amount of data being written to these files. Now some of the data volume 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 records 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.

Posted by Ed Woods at 3:13 PM 0 comments