Understanding The Impact Of The Network On z/OS Performance

Ed Woods - IBM Corporation
Session 12780
Tuesday, February 5th: 4:30 PM - 5:30 PM

Tivoli software
Agenda

- Introduction
- Looking at the application time line
- Examples of mainframe/network interaction
  - Analysis scenarios using commonly available commands
  - Optimization considerations 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 impacts network usage and performance
- Be aware of the potential impact of SSL and IPSec
DB2 Has Several Potential Performance Bottlenecks

Many methods of connection to DB2

Network delays

Connect GW bottlenecks

DB2 Connect

Network delays

Application

Application

z/OS

IMS

Network delays

CICS

Batch

TSO

WebSphere

MSTR

Connections

Threads

Logging

IRLM

Locking

DBM1

SQL

EDM

PTs, SKPTs, CTs, SKCTs

DSC – Dynamic SQL

Sort Pool

RID Pool

Lock Conflicts

DB, BP I/O delays

Network delays

Connection bottlenecks

DDF

Distributed threads

SP Addr Space(s)

Stored Procedures

UDFs

SP Sched Delays

Network delays
Network Analysis Tools

**NETSTAT Command**

- NETSTAT reports TCP/IP connections and protocol statistics
- Get status information on connections and statistics on packets sent, packets received, fragmentation, etc.

```plaintext
C:\Documents and Settings\woodse>netstat

Active Connections

<table>
<thead>
<tr>
<th>Proto</th>
<th>Local Address</th>
<th>Foreign Address</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>IBM-1E47754C52F:4138</td>
<td>demomvs.demopkg.ibm.com:telnet</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>IBM-1E47754C52F:4251</td>
<td>d0lm1253.pok.ibm.com:1352</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>IBM-1E47754C52F:4255</td>
<td>demomvs.demopkg.ibm.com:448</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>IBM-1E47754C52F:1035</td>
<td>localhost:1036</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>IBM-1E47754C52F:1036</td>
<td>localhost:1035</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>IBM-1E47754C52F:1920</td>
<td>localhost:3416</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>IBM-1E47754C52F:1920</td>
<td>localhost:3768</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>IBM-1E47754C52F:3416</td>
<td>localhost:3661</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>IBM-1E47754C52F:3417</td>
<td>localhost:3661</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>IBM-1E47754C52F:3661</td>
<td>localhost:3417</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>IBM-1E47754C52F:3661</td>
<td>localhost:3769</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>IBM-1E47754C52F:3768</td>
<td>localhost:1920</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>IBM-1E47754C52F:3769</td>
<td>localhost:3661</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>IBM-1E47754C52F:1097</td>
<td>204.146.166.107:http</td>
<td>CLOSE_WAIT</td>
</tr>
<tr>
<td>TCP</td>
<td>IBM-1E47754C52F:1098</td>
<td>129.42.208.236:https</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>IBM-1E47754C52F:1100</td>
<td>raccol01.attglobal.net:http</td>
<td>CLOSE_WAIT</td>
</tr>
<tr>
<td>TCP</td>
<td>IBM-1E47754C52F:1188</td>
<td>as32.live365.com:http</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>IBM-1E47754C52F:4204</td>
<td>58.mtl-mg05.streamtheworld.net:http</td>
<td>ESTABLISHED</td>
</tr>
</tbody>
</table>
```
### NETSTAT Connection Detail

**netstat all (port 448)**

<table>
<thead>
<tr>
<th>EZZ2350I</th>
<th>MVS TCP/IP NETSTAT CS V1R10</th>
<th>TCPIP Name: TCPIP</th>
<th>19:16:24</th>
</tr>
</thead>
<tbody>
<tr>
<td>EZZ2550I</td>
<td>Client Name: DNSCDIST</td>
<td>Client Id: 0000C90E</td>
<td></td>
</tr>
<tr>
<td>EZZ2551I</td>
<td>Local Socket: 9.39.68.147..448</td>
<td>Foreign Socket: 9.05.73.27..4255</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EZZ2552I</th>
<th>Last Touched: 19:14:58</th>
<th>State: Establish</th>
</tr>
</thead>
<tbody>
<tr>
<td>EZZ2577I</td>
<td>BytesIn: 0000006973</td>
<td>BytesOut: 0008457981</td>
</tr>
<tr>
<td>EZZ2584I</td>
<td>SegmentsIn: 0000003423</td>
<td>SegmentsOut: 0000006614</td>
</tr>
<tr>
<td>EZZ2613I</td>
<td>RcvNxt: 3808791478</td>
<td>SndNxt: 2538223807</td>
</tr>
<tr>
<td>EZZ2583I</td>
<td>InitRcvSeqNum: 3808784504</td>
<td>ClientSndNxt: 2538223807</td>
</tr>
<tr>
<td>EZZ2566I</td>
<td>CongestionWindow: 0000017349</td>
<td>SlowStartThreshold: 00000002620</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>SndW2: 0000065482</td>
<td>MaxSndWnd: 0000131070</td>
</tr>
<tr>
<td>EZZ2560I</td>
<td>SndUna: 2538223807</td>
<td>rtt_seq: 2538223753</td>
</tr>
<tr>
<td>EZZ2565I</td>
<td>MaximumSegmentSize: 0000001310</td>
<td>DSField: 00</td>
</tr>
<tr>
<td>EZZ2563I</td>
<td>Round-trip information: Smooth trip time: 184.000</td>
<td>SmoothTripVariance: 84.000</td>
</tr>
<tr>
<td>EZZ2564I</td>
<td>ReRcv: 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>EZZ2576I</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>EZZ2577I</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: 0000016384</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>
An Example - Looking At The Numbers

DB2 Distributed Performance Statistics Trace Data For The DB2 Subsystem

<table>
<thead>
<tr>
<th>DFST</th>
<th>Start: 06/04 13:40:46</th>
<th>End: 06/04 13:40:49</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection Interval</td>
<td>REALTIME</td>
<td></td>
</tr>
<tr>
<td>Report Interval</td>
<td>4 sec</td>
<td></td>
</tr>
<tr>
<td>DCDB203</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDF Status</td>
<td>ACTIVE</td>
<td></td>
</tr>
<tr>
<td>Dist Allied Threads</td>
<td>30%</td>
<td></td>
</tr>
<tr>
<td>Inactive DBATs</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>DDF Send Rate</td>
<td>0K/sec</td>
<td></td>
</tr>
<tr>
<td>DDF Receive Rate</td>
<td>0K/sec</td>
<td></td>
</tr>
<tr>
<td>Resync Attempts</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Resync Successes</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Cold Start Connections</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Warm Start Connections</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>DBAT Queued</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Conversations Dealloc</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>HWM All DBATs</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>HWM Active DBATs</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Max DB Access (MAXDBAT)</td>
<td>500</td>
<td></td>
</tr>
<tr>
<td>HWM Inactive DBATs</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

<p>| RDA REMOTE LOCS     |                        |                     |
| Conversations Queued | 0                      |                     |
| Binds for Remote Access | 0                  |                     |
| Block Mode Switches | 0                      |                     |
| Rollbacks/Remote   | 0                      |                     |
| Indoubts/Remote    | 0                      |                     |</p>
<table>
<thead>
<tr>
<th>Tran</th>
<th>SQL</th>
<th>Row</th>
<th>Message</th>
<th>Byte</th>
<th>Commit</th>
<th>Abort</th>
<th>Conv</th>
<th>Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sent</td>
<td>0</td>
<td>0</td>
<td>174927</td>
<td>2486</td>
<td>43164569</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Recv</td>
<td>55</td>
<td>1714</td>
<td>0</td>
<td>2285</td>
<td>238429</td>
<td>363</td>
<td>8</td>
<td>55</td>
</tr>
</tbody>
</table>

- Thread creation queues?
- HWM of DBAT usage
- Is buffering occurring?
- Recvd – received into DB2
- Sent – out to client/apps
- Bytes and messages sent and received
Looking At The Application
DB2 Accounting Information Analysis

> DISTRIBUTED THREAD DETAIL

PLAN
+ Thread: Plan=DISTSERV Connid=SERVER Corrid=db2bp.exe Authid=DNET581
+ Dist : Type=DATABASE ACCESS, Lwuid=G941491B.PC10.090604182432=169
+ Location : 9.65.73.27
+ rsum
  + Distributed TCP/IP Data
    + Location           IP Addr  Port Ctbuser Srvcslnam Prod ID  Workstation Name
      +------------------------+--------+------+----------+----------+--------------+------------------------+
      +9.65.73.27              0941491B 448 dnet581 NT     SQL09013 IBM-1E47754C52F
      + Transaction name: db2bp.exe
      + TCP/IP Userid: dnet581
      +
      + Remote Loc: 9.65.73.27
      + Remote Location Luname =
      + Protocol Used = Conversations Queued = 0
      + Block Mode Switches = 0 Message Buffer Queued = 33280
      + Bind Remote Access = 0 Max Allocated Conv = 0
      + Conv Allocated = 0 Conv Deallocated = 0
      + Indoubt/Remote = 0 Commit/Remote = 0
      + Rollback/Remote = 0
      +
      + Tran SQL Row Message Byte Commit Abort Conv Blocks
      + Sent 0 0 33280 270 8457981 0 0 0 260
      + Recv 1 264 0 270 26329 2 0 1 0
      + 2-PHASE COMMIT
        + List 2-Phase
        + Commit Backout
        + Prepare Agent Commit Backout Forget Resp Resp
        + Sent 0 0 0 0 0 0 0 0
        + Recv 0 0 0 0 0 0 0 0

# SQL calls
# of messages sent
DRDA blocks for queries

Note row versus message/block counts to determine blocking

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

Note the quantity of data being sent
Example - Optimize DB2 Applications To Minimize Network Traffic

- Do more with SQL to eliminate redundant back/forth activity
- Crossing more layers will mean more overhead
- Don’t put too much business logic in the DB layer
DB2 Subsystem Configuration
The Interaction Of CONDBAT
And MAXDBAT

Note – DB2 10 changes

Increases the maximum setting for MAXDBAT to 19999. (Similar increases for IDFORE, IDBACK, and CTHREAD)

New zparm for queue wait time
IMS Has Many Potential Bottlenecks (Including Network)

- IMS Connect
- OTMA
- APPC
- Telnet
- IMS Control Region
  - Message In
  - IMS Message & BMP Regions
  - IMS Control Region
  - IMS DLI
  - IRLM

- Network delays
- Connection bottlenecks
- Thread bottlenecks
- DB, BP I/O delays
- Lock Conflicts
- App Init & execution
IMS Connect And The Interaction With TCP/IP

- IMS Control Center
- IMS DB Adapters
- WebSphere IMS TMRA
- IMS SOAP Gateway
- IMS Connect API Clients
- IMS Connect
  - IMS Connect Extensions
  - IMS Connect Extensions
  - ODBM
  - Operations Manager
  - IMS Connect Extensions
  - OMEGAMON IMS
- TCP/IP
  - IMS
    - TM
      - IMS Application
      - IMS DB
      - ODBM
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
    - Multiple 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
    - The number of connection requests that can be queued because IMS Connect has not yet issued the accept call - Default setting is 10

- IMS Connect parameter MAXSOC
  - Total number of sockets IMS Connect supports across all ports at the same time
    - Note - USS parameter MAXFILEPROC must be equal to or greater than the value of the IMS Connect parameter MAXSOC
  - IMS Connect issues warning message HWSS0772W when the number of sockets reaches the default warning threshold of 80 percent of MAXSOC
IMS Connect Monitoring
An Example

<table>
<thead>
<tr>
<th>Transaction Code</th>
<th>Target Database</th>
<th>Client ID</th>
<th>Port Number</th>
<th>User ID</th>
<th>Collection Level</th>
<th>Message Received Time</th>
<th>Response Time</th>
<th>Input Pre-OTMA Time</th>
<th>Input Read Sustain Time</th>
<th>Input Read Exit Time</th>
<th>Input Read Exit Name</th>
<th>Input SAF Time</th>
<th>Process OTMA Time</th>
<th>Output Confirm Time</th>
<th>Output Post-OTMA Time</th>
<th>XMIT Exit Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>PART 11Y</td>
<td>ICTDRVY</td>
<td>4713</td>
<td>1701</td>
<td>JMAHE</td>
<td>Maximum</td>
<td>01/14/09 12:08:08</td>
<td>0.000130</td>
<td>0.000130</td>
<td>0.00021</td>
<td>0.026154</td>
<td>HVIMSD00</td>
<td>0.000000</td>
<td>0.113476</td>
<td>0.0000629</td>
<td>0.00025</td>
<td></td>
</tr>
<tr>
<td>PART 11Y</td>
<td>ICTDRVY</td>
<td>4713</td>
<td>1701</td>
<td>JMAHE</td>
<td>Maximum</td>
<td>01/14/09 12:08:08</td>
<td>0.0000623</td>
<td>0.0000623</td>
<td>0.000019</td>
<td>0.000019</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000019</td>
<td></td>
</tr>
<tr>
<td>PART 11Y</td>
<td>ICTDRVY</td>
<td>4713</td>
<td>1701</td>
<td>JMAHE</td>
<td>Maximum</td>
<td>01/14/09 12:08:08</td>
<td>0.000000632</td>
<td>0.000000632</td>
<td>0.000000</td>
<td>0.000019</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000019</td>
<td></td>
</tr>
<tr>
<td>PART 11Y</td>
<td>ICTDRVY</td>
<td>4713</td>
<td>1701</td>
<td>JMAHE</td>
<td>Maximum</td>
<td>01/14/09 12:08:08</td>
<td>0.000000632</td>
<td>0.000000632</td>
<td>0.000000</td>
<td>0.000019</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000019</td>
<td></td>
</tr>
</tbody>
</table>

Note – This display requires IMS Connect Extensions And OMEGAMON IMS

© 2013 IBM Corporation
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
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 Sockets Versus CICS Sockets Domain

- CICS Sockets – a component of Communications Server for z/OS
- CICS Sockets Domain – a component of CICS TS

CICS Application Program

CICS Sockets

CICS Sockets Domain – component of CICS TS

Sockets API provided by USS

TCP/IP Network

CICS Socket APIs

Traditional CICS APIs

CICS Sockets Listener

CICS Web Listener

CICS IIOP Listener

ECI Over IP Listener

IPIC Listener

EZASOKET

UIforward
CICS Socket Interface Example

- Socket API available for C, COBOL, PL/I and ASM applications
- Listener is a CICS transaction
- Conversion routines – ASCII/EBCDIC
A WebSphere Example With CICS Transaction Gateway
CICS Network
Examples Of Relevant Parameters

- **TCPIPSERVICE** parameter defines services
  - ECI over TCP/IP (for CICS Clients), IIOP, CICS Web support (HTTP), IPIC (ISC), or a user-defined protocol.
  - 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.

- **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** 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
MQSeries Performance Bottlenecks

MQ Channels
MQ queue processing
MQ queue depth
Application processing

Network delays

Channel

Application bottlenecks

Transmission Queue

Reply-To-Queue

© 2013 IBM Corporation
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
Address Network Latency Concerns With zEnterprise
Defining A Monitoring Strategy
Many Factors May Impact Response Time

- 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
  - Dashboard level monitoring
  - Composite level monitoring

Subsystem Monitoring
Both
Network Monitoring
Example - Understanding IMS Response Time

Mainframe network monitoring
Network time for IMS transactions

Including network monitoring detail provides a more complete analysis of IMS response time

IMS subsystem monitoring
IMS host response time including queue and processing time for the transaction
Another Example Combining Host And Network Level Monitoring

DB2 thread level monitoring

DB2 network level monitoring

© 2013 IBM Corporation
Monitor Host Application Network Connection Activity

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

- Monitor for interface status, bandwidth utilization, and errors
- Look for potential problems at the interface level
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
Dashboard Level Monitoring
Integrate Host And Network Monitoring

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.
Collect History For Trending And Analysis

- Collect history data at multiple levels
  - System/LPAR level
  - Application level
  - Interface/adapter level
  - Connection level
- Frequency and retention will vary by type
- Useful for trending and alert threshold analysis

Select desired group of information, collection interval, and destination

Collect at the TEMA or the TEMS
To warehouse or not to warehouse: Hourly, Daily, or not at all

Specify summarization and pruning along with collection interval

Graphics make it easier to spot peaks and valleys over time
End-to-End Monitoring, Tracking and Diagnosis

1. **Response Time Measurement**
   - Start by monitoring transaction performance and end-user problems.

2. **Transaction Tracking**
   - Correlate data from app server, MQ, CICS, IMS, custom instrumentation, etc. to show topology and isolate problems.

3. **Deep Dive Diagnostics**
   - Launch in context to SME tools where appropriate.
   - In this scenario, the problem is a WebSphere JEE memory leak.

### Transaction Root Cause Analysis

<table>
<thead>
<tr>
<th>1. <strong>Sense</strong></th>
<th>2. <strong>Isolate</strong></th>
<th>3. <strong>Diagnose and repair</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>End User Experience and alert on threshold violation</td>
<td>by measuring performance data against baseline through entire infrastructure</td>
<td>through launch-in-context into deep-dive diagnostics</td>
</tr>
</tbody>
</table>
Summary

- 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!
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 TFCIDs 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

© 2013 IBM Corporation