Understanding The Impact Of The Network On DB2 Performance

Ed Woods
Consulting IT Specialist
IBM Corporation
Agenda

- Introduction
- The Network And The DB2 Application Time Line
- Understanding The Network
- Sources Of Analysis Information And Analysis Examples
- Useful Commands
- Summary And Recommendations
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
  - How best to do alerting, problem isolation, and root cause analysis
DB2 Distributed Data Base Access

- DB2 supports two different types of remote access between the requesting application and the serving DBMS
  - DRDA - Distributed Relational Database Architecture
    - A database interoperability standard from The Open Group
    - With DRDA access, an application can access data at any server that supports DRDA
    - DRDA supports all SQL features, including user-defined functions, LOBs, stored procedures, and XML data
    - DRDA minimizes the number of binds required, and supports multi-row FETCH
    - DRDA is the preferred method for remote access with DB2
  - DB2 private protocol
    - Private protocol is an older method for remote access. It can be used only between DB2 subsystems and only over a SNA network.
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.

A query sent to a remote system can sometimes take longer to execute than the same query on the local DB2 subsystem:
- Time is required to send messages across the network.
- Overhead processing, including communication subsystem session management.

Key questions to ask:
- Is the problem in the network, the host DBMS, the application, or somewhere else?
Understanding TCP/IP
A Layered Network Protocol

Web Browser

HTTP Server

Application
Port
App Interface
Port

TCP
Transmission Control Protocol

IP
Internet Protocol

Network Interface

WWW, FTP, Telnet
End to End Acknowlegement
Datagram Delivery
Physical Connection

Application
Port
App Interface
Port

TCP
Transmission Control Protocol

IP
Internet Protocol

Network Interface
Understanding TCP/IP
Three Way Handshake

- TCP assigns a SEQ number to each segment sent
- Each segment received is ACK’ed by the receiver
- Time Algorithm
The Performance Characteristics Of TCP/IP

- Visualize as a mesh-connected network
  - No defined mechanism to limit the number of users on the network
  - Path to destination may have multiple hops
  - Data routes are not necessarily symmetric
- Network routers have a finite amount of memory to store information
  - IP allows routers to drop packets when buffers overflow
  - Congestion avoidance cannot help when demand outstrips network capacity
- Only two ways to correct the problem:
  - Increase capacity and add additional bandwidth
  - Reduce network demand through DB2 and application optimization and tuning
DB2 Has Several Potential Performance Bottlenecks

DB2 Performance Issues
- DB2 thread connection issues
- Long running threads
- DB2 SQL delays and long running SQL
- DB2 database I/O delays and BP perf.
- Network and/or OS delays
- DB2 lock conflicts

z/OS
- IMS
- CICS
- Batch
- TSO
- WebSphere

MSTR
- Connections
- Threads
- Logging

DBM1
- SQL
- EDM
- PTs, SKPTs, CTs, SKCTs
- DSC – Dynamic SQL
- Sort Pool
- RID Pool

DDF
- Distributed threads
- SP Addr Space(s)
- Stored Procedures
- UDFs
- SP Sched Delays

DB2 Connect
- Network delays
- Connect GW bottlenecks

DB2 App
- Network delays
- Connection bottlenecks

Network delays
- DB, BP I/O delays
- DB2 thread connection issues
- Long running threads
- DB2 SQL delays and long running SQL
- DB2 database I/O delays and BP perf.
- Network and/or OS delays
- DB2 lock conflicts

Connection bottlenecks
- Lock Conflicts

z/OS
- IMS
- CICS
- Batch
- TSO
- WebSphere

MSTR
- Connections
- Threads
- Logging

DBM1
- SQL
- EDM
- PTs, SKPTs, CTs, SKCTs
- DSC – Dynamic SQL
- Sort Pool
- RID Pool

DDF
- Distributed threads
- SP Addr Space(s)
- Stored Procedures
- UDFs
- SP Sched Delays

DB2 Connect
- Network delays
- Connect GW bottlenecks

DB2 App
- Network delays
- Connection bottlenecks

Network delays
- DB, BP I/O delays
- DB2 thread connection issues
- Long running threads
- DB2 SQL delays and long running SQL
- DB2 database I/O delays and BP perf.
- Network and/or OS delays
- DB2 lock conflicts

Connection bottlenecks
- Lock Conflicts

z/OS
- IMS
- CICS
- Batch
- TSO
- WebSphere

MSTR
- Connections
- Threads
- Logging

DBM1
- SQL
- EDM
- PTs, SKPTs, CTs, SKCTs
- DSC – Dynamic SQL
- Sort Pool
- RID Pool

DDF
- Distributed threads
- SP Addr Space(s)
- Stored Procedures
- UDFs
- SP Sched Delays

DB2 Connect
- Network delays
- Connect GW bottlenecks

DB2 App
- Network delays
- Connection bottlenecks

Network delays
- DB, BP I/O delays
- DB2 thread connection issues
- Long running threads
- DB2 SQL delays and long running SQL
- DB2 database I/O delays and BP perf.
- Network and/or OS delays
- DB2 lock conflicts

Connection bottlenecks
- Lock Conflicts
Optimizing Network Processing

- Minimizing the number of messages sent between the application requester and the DBMS is a primary way to optimize performance.

- **BIND options**
  - Consider the `DEFER(PREPARE)` bind option to reduce the number of messages that must be sent back and forth across the network.
  - Consider `ISOLATION(CS)` to reduce contention and message overhead.

- **SQL statement options**
  - Commit frequently to avoid holding resources at the server.
  - Minimize SQL statements:
    - Avoid using several SQL statements when one well-tuned statement can do the job.
    - Consider using stored procedures.

- **Block fetching of result sets**
  - Block fetch can significantly decrease the number of messages sent across the network.
  - Block fetch is used only with cursors that do not update or delete data.
  - With block fetch, DB2 groups the rows that are retrieved by an SQL query into a “block” of rows that fit in a message buffer and will transmit the block over the network.
Exploit The Power Of SQL To Optimize 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

Diagram:
- Multiple SQLs vs. Single SQL
- OR
- Network flow between HTTP Server, App Server, and DB Server
- Cross-layer communication implications
Sources Of Analysis Information

- DB2 Traces
  - Statistics traces – subsystem level information
  - Accounting traces – application level information

- Useful network commands
  - PING and NETSTAT

- Other trace options
  - DRDA trace
  - CL/I trace
  - JDBC trace

- Sample transactions and robotic techniques
  - Create a simple application to display times for testing purposes

- ARM – application instrumentation
  - Instrumentation of applications for ongoing measurement and analysis
  - Breakout application performance by component layer
An effective trace gathering, retention, and analysis strategy is important
Traces have costs, so used the appropriate tool in the right manner
Different traces have different levels of granularity
DB2 Statistics Traces
DB2 Subsystem Performance Analysis

- System-wide performance data gathered at specified intervals (set in ZPARMs)
- Written to SMF with type 100 header
  - Two IFCIDs written per period
- Overhead small (approximately 1%)

SSAS Statistics
DB2 CPU, Storage, Log, & IRLM statistics

DBAS Statistics
SQL, Buffer, EDM & RDS statistics

-START TRACE (STAT)

IFCID 1

IFCID 2

SMF Type 100
DB2 Statistics Data
Distributed Stats For The Entire Subsystem

<table>
<thead>
<tr>
<th>DISTRIBUTED DATA FACILITY STATISTICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start: 06/04 13:40:46</td>
</tr>
<tr>
<td>End: 06/04 13:40:49</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>+ Collection Interval: REALTIME</td>
</tr>
<tr>
<td>+ Report Interval: 4 sec</td>
</tr>
<tr>
<td>+</td>
</tr>
<tr>
<td>+ DCDB203</td>
</tr>
<tr>
<td>+ DDF Status</td>
</tr>
<tr>
<td>+ Inactive DBATs</td>
</tr>
<tr>
<td>+ DDF Send Rate</td>
</tr>
<tr>
<td>+ DDF Receive Rate</td>
</tr>
<tr>
<td>+ Resync Attempts</td>
</tr>
<tr>
<td>+ Resync Successes</td>
</tr>
<tr>
<td>+ Cold Start Connections</td>
</tr>
<tr>
<td>+ Warm Start Connections</td>
</tr>
<tr>
<td>+ DBAT Queued</td>
</tr>
<tr>
<td>+ Conversations Dealloc</td>
</tr>
<tr>
<td>+ HWM All DBATs</td>
</tr>
<tr>
<td>+ HWM Active DBATs</td>
</tr>
<tr>
<td>+ Max DB Access (MAXDBAT)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RDA REMOTE LOCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binds for Remote Access</td>
</tr>
<tr>
<td>Block Mode Switches</td>
</tr>
<tr>
<td>Rollbacks/Remote</td>
</tr>
</tbody>
</table>

<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>1746</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>
Accounting Traces
The Starting Point For DB2 Application Analysis

<table>
<thead>
<tr>
<th>Accounting Class 1</th>
<th>Thread elapsed times, SQL stats, Buffer &amp; lock stats</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accounting Class 2</td>
<td>“In-DB2” Time</td>
</tr>
<tr>
<td>Accounting Class 3</td>
<td>“Wait” times and counts</td>
</tr>
<tr>
<td>Accounting Class 7</td>
<td>“In-DB2” Package level</td>
</tr>
<tr>
<td>Accounting Class 8</td>
<td>“Wait” times Package level</td>
</tr>
</tbody>
</table>

-START TRACE (ACCTG) CLASS (1,2,3,7,8)

IFCID 3

SMF Type 101

- Data written @ thread termination
- Five main classes of accounting traces
- Written to SMF as a type 101 record
First Example - Accounting Information

```
> DISTRIBUTED THREAD DETAIL
PLAN
+ Thread: Plan=DISTSERV Connid=SERVER Corrid=db2bp.exe Authid=DNET581
+ Dist : Type=DATABASE ACCESS, Luuid=G941491B.PA10.090604181240=168
+ Location : 9.65.73.27
rsum
+ Distributed TCP/IP Data
+Location IP Addr Port Ctbuser Srvclsname Prod ID Workstation Name
+----------------- ------ ------ ---------- ---- ------------------------
+9.65.73.27 0941491B 448 dnet581 NT SQL09013 IBM-1E47754C52F
+ Transaction name: db2bp.exe
+ TCP/IP Userid: dnet581
+ Distributed SQL Statistics
+ Remote Location Name = 9.65.73.27 Remote Location Luname =
+ Protocol Used
+ Block Mode Switches = 0 Message Buffer Rows = 0
+ Bind Remote Access = Max Allocated Conv = 0
+ Conv Allocated = Conv Deallocated = 0
+ Indoubt/Remote = Commit/Remote = 0
+ Rollback/Remote = 0
+ Tran SQL Row Message Byte Commit Abort Conv Blocks
+ ------ ------ ------ ---------- ------- ------- ------ -------- -------- --------
+ Sent 0 0 4 54 47953 0 0 0 20
+ Recv 1 40 0 54 4625 10 0 1 0
+ 2-PHASE COMMIT:
Prepares 2-Phase Commit/Backout
+ Sent 0 0 0 0 0 0 0 0 0
+ Recv 0 0 0 0 0 0 0 0 0
```

Multiple executions of a single-row SELECT statement
SELECT * FROM TABLEA WHERE NAME = ‘AAA’
Sample invoked 10 times

Note no buffering
Recv – received into DB2
Sent – out to client/app

# SQL calls
# of messages sent
DRDA blocks for queries
First Example – Accounting SQL Counts

```
> CURRENT SQL COUNTS
PLN
+ Thread: Plan=DISTSERV Connid=SERVER Corrid=db2bp.exe Authid=DNF7561
+ Dist : Type=DATABASE ACCESS, Luwid=G941491B.PA10.090604181240=168
+ Location : 9.65.73.27
sqls
+ Commit = 10 Abort = 0 Select = 0
+ Open Cursor = 10 Close Cursor = 0 Fetch = 10
+ Insert = 0 Delete = 0 Update = 0
+ Describe = 10 Lock Table = 0 Prepare = 10
+ Grant = 0 Revoke = 0 Set Rules = 0
+ Increm Bind = 0 Label/Comm On = 0 Set SQLID = 0
+ Set Host Var = 0 Set Connection = 0 Set Degree = 0
+ Connect Type 1 = 0 Connect Type 2 = 0 Set Path = 0
+ Rename Table = 0 Hold Locator = 0 Free Locator = 0
+ Release = 0 Assoc Locator = 0 Allocate Cursor = 0
+
+ SQL Call Statements = 0 SQL Calls Timed Out = 0
+ Stored Proc SQL Reqs = 0 SQL Calls Rejected = 0
+ Stored Procedures Abended = 0
+
+ UDFs Executed = 0 UDFs ABENDed = 0
+ UDFs Timed Out = 0 UDFs Rejected = 0
+
+ Stmt Triggers Activated = 0 Row Triggers Activated = 0
+ SQL Error in Trigger = 0 Maximum Nested SQL = 0
+
+ Prepare Statistics:
+ Copied from Cache = 10 Implicit - KEEP_DYNAMIC(YES) = 0
+ No Match = 0 Avoided - KEEP_DYNAMIC(YES) = 0
+ Discarded - MAXKEEPD = 0 Purged - DROP/ALTER/REVOKE = 0
+
```

Sample call was invoked 10 times. This can be easily seen from the pattern of call counts.

SPs were not called in this example.

Same call invoked multiple times results in DSC cache hits.
<table>
<thead>
<tr>
<th>User Defined Functions</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TCB Time (SQL)</td>
<td>00:00:00.000</td>
</tr>
<tr>
<td>Wait for TCB Time</td>
<td>00:00:00.000</td>
</tr>
<tr>
<td>Elapsed Time</td>
<td>00:00:00.000</td>
</tr>
<tr>
<td>TCB not in Enclave</td>
<td>00:00:00.000</td>
</tr>
<tr>
<td>Elapsed not in Enclave</td>
<td>00:00:00.000</td>
</tr>
<tr>
<td>TCB prior to Enclave</td>
<td>00:00:00.000</td>
</tr>
</tbody>
</table>

Elapsed time includes ‘think’ time between calls to DB2.

IN-DB2 time represents time executing the calls.

CPU times shows general CP time and time on zIIP.

No delays in this example.
Thread Enclave Information

What service class is this thread executing in?

WLM qualifiers used to select service class

See enclave token, WLM service class, and service class performance index (PI)
Don’t Overlook The Importance Of WLM And Enclave Priority To Performance

- Each Enclave is an individual transaction to be managed by priorities as defined in Workload Manager
  - As a DDF request arrives in DB2 an enclave is created to tie together the SRB tasks and to allow WLM to set its priority
Second Example – Accounting Information

DISTRIBUTED THREAD DETAIL

<table>
<thead>
<tr>
<th>Transaction name:</th>
<th>db2bp.exe</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP/IP Userid:</td>
<td>dnet581</td>
</tr>
<tr>
<td>Protocol Used:</td>
<td>Conversations Queued = 0</td>
</tr>
<tr>
<td>Block Mode Switches</td>
<td>0</td>
</tr>
<tr>
<td>Bind Remote Access</td>
<td>0</td>
</tr>
<tr>
<td>Conv Allocated</td>
<td>0</td>
</tr>
<tr>
<td>Indoubt/Remote</td>
<td>Commit/Remote = 0</td>
</tr>
<tr>
<td>Rollback/Remote</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sent</th>
<th>Recv</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<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>0</td>
<td>0</td>
<td>33280</td>
<td>270</td>
<td>8457981</td>
<td>0</td>
<td>0</td>
<td>260</td>
</tr>
<tr>
<td>1</td>
<td>264</td>
<td>0</td>
<td>270</td>
<td>26329</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2-PHASE COMMIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prepare</td>
</tr>
<tr>
<td>Agent</td>
</tr>
<tr>
<td>Commit</td>
</tr>
<tr>
<td>Backout</td>
</tr>
<tr>
<td>Forget</td>
</tr>
<tr>
<td>Commit</td>
</tr>
<tr>
<td>Backout</td>
</tr>
<tr>
<td>Resp</td>
</tr>
<tr>
<td>Resp</td>
</tr>
<tr>
<td>Sent</td>
</tr>
<tr>
<td>Recv</td>
</tr>
</tbody>
</table>

Note now data is buffering

Note the quantity of data being sent

Two executions of a multi-row SELECT statement

SELECT * FROM TABLEA

# SQL calls

# of messages sent

DRDA blocks for queries
Second Example – Accounting SQL Counts

```
> PLAN
+ Thread: Plan=DISTSERV Connid=SERVER Corrid=db2bp.exe Authid=DNET581
+ Dist : Type=DATABASE ACCESS, LuwId=G941491B.PC10_090604180216-169
+ Location : 9.65.73.27
+ sqls
+ Commit = 2 Abort = 0 Select = 0
+ Open Cursor = 2 Close Cursor = 0 Fetch = 258
+ Insert = 0 Delete = 0 Update = 0
+ Describe = 2 Lock Table = 0 Prepare = 2
+ Grant = 0 Revoke = 0 Set Rules = 0
+ Incrm Bind = 0 Label/Comm On = 0 Set SQLID = 0
+ Set Host Var = 0 Set Connection = 0 Set Degree = 0
+ Connect Type 1 = 0 Connect Type 2 = 0 Set Path = 0
+ Rename Table = 0 Hold Locator = 0 Free Locator = 0
+ Release = 0 Assoc Locator = 0 Allocate Cursor = 0
+ SQL Call Statements = 0 SQL Calls Timed Out = 0
+ Stored Proc SQL Reqs = 0 SQL Calls Rejected = 0
+ Stored Procedures Abended = 0
+ UDFs Executed = 0 UDFs ABENDed = 0
+ UDFs Timed Out = 0 UDFs Rejected = 0
+ Stmt Triggers Activated = 0 Row Triggers Activated = 0
+ SQL Error in Trigger = 0 Maximum Nested SQL = 0
+ Prepare Statistics:
+ Copied from Cache = 2 Implicit - KEEPDYNAMIC(YES) = 0
+ No Match = 0 Avoided - KEEPDYNAMIC(YES) = 0
+ Discarded - MAXKEEPD = 0 Purged - DROP/ALTER/REVOKE = 0
```
**Second Example Accounting Information**

Elapsed time includes ‘think’ time between calls to DB2.

IN-DB2 time represents time executing the calls. Most of this time is I/O delays (see below).

CPU times shows general CP time and time on zIIP.

I/O and Lock/latch delays in this example due to larger result set being retrieved.
Network Analysis Tools
PING Command

- PING is a simple, but highly useful command
- Verifies connection between hosts by sending ICMP packets to the specified address (IP address or hostname)
- PING shows the time to echo a packet
- Beware - in some shops PING may have limitations

```
64 bytes from 198.210.45.27: icmp_seq=0  ttl=253  time=0.345 ms
64 bytes from 198.210.45.27: icmp_seq=1  ttl=253  time=0.345 ms
64 bytes from 198.210.45.27: icmp_seq=2  ttl=253  time=0.345 ms
```

```
PING hostname  -l # of bytes in echo packet
PING hostname  -n # of packets to echo
PING hostname  -r record the route of the packet
PING hostname  -s report timestamps of hops
```
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.

```
C:\Documents and Settings\woodse>netstat
Active Connections
Protocol  Local Address          Foreign Address        State
TCP        IBM-1E47754C52F:4138 demomvs.demopkg.ibm.com:telnet ESTABLISHED
TCP        IBM-1E47754C52F:4251 d01m1253.pok.ibm.com:1352 ESTABLISHED
TCP        IBM-1E47754C52F:4255 demomvs.demopkg.ibm.com:448 ESTABLISHED
TCP        IBM-1E47754C52F:1035 localhost:1036 ESTABLISHED
TCP        IBM-1E47754C52F:1036 localhost:1035 ESTABLISHED
TCP        IBM-1E47754C52F:1920 localhost:3416 ESTABLISHED
TCP        IBM-1E47754C52F:1920 localhost:3768 ESTABLISHED
TCP        IBM-1E47754C52F:3416 localhost:1920 ESTABLISHED
TCP        IBM-1E47754C52F:3417 localhost:3661 ESTABLISHED
TCP        IBM-1E47754C52F:3661 localhost:3417 ESTABLISHED
TCP        IBM-1E47754C52F:3661 localhost:3769 ESTABLISHED
TCP        IBM-1E47754C52F:3768 localhost:1920 ESTABLISHED
TCP        IBM-1E47754C52F:3769 localhost:3661 ESTABLISHED
TCP        IBM-1E47754C52F:1097 204.146.166.107:443 CLOSE_WAIT
TCP        IBM-1E47754C52F:1098 129.42.208.236:https ESTABLISHED
TCP        IBM-1E47754C52F:1100 racol01.attglobal.net:https CLOSE_WAIT
TCP        IBM-1E47754C52F:1188 www.live365.com:80 CLOSE_WAIT
TCP        IBM-1E47754C52F:1188 ss32.live365.com:80 ESTABLISHED
TCP        IBM-1E47754C52F:4204 58.mtl-mg05.streamtheworld.net:80 ESTABLISHED
```
NETSTAT Command
Display Connections To A Specific Port

`netstat conn (port 448)`

<table>
<thead>
<tr>
<th>EZZ2350I</th>
<th>MVS TCP/IP NETSTAT CS V1R10</th>
<th>TCPIP Name: TCPIP</th>
<th>19:10:22</th>
</tr>
</thead>
<tbody>
<tr>
<td>EZZ2585I</td>
<td>User Id Conn Local Socket Foreign Socket State</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EZZ2586I</td>
<td>------- ---- ----------- -----------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EZZ2587I</td>
<td>DSNCDIST 0000C90E 9.39.68.147.448 9.65.73.27.4255 Establ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EZZ2587I</td>
<td>DSNCDIST 0000005B 0.0.0.0..448 0.0.0.0..0 Listen</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Connection to DB2 on z/OS

Command can be filtered a variety of ways including IP address and port number
NETSTAT Connection Detail

**netstat all (port 448)**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last Touched:</td>
<td>19:14:58</td>
</tr>
<tr>
<td>BytesIn:</td>
<td>00000006973</td>
</tr>
<tr>
<td>SegmentsIn:</td>
<td>00000003423</td>
</tr>
<tr>
<td>RcvNxt:</td>
<td>3808791478</td>
</tr>
<tr>
<td>ClientRcvNxt:</td>
<td>3808791478</td>
</tr>
<tr>
<td>InitRcvSeqNum:</td>
<td>3808784504</td>
</tr>
<tr>
<td>CongestionWindow:</td>
<td>0000017349</td>
</tr>
<tr>
<td>IncomingWindowNum:</td>
<td>3808824236</td>
</tr>
<tr>
<td>SndW1:</td>
<td>3808791478</td>
</tr>
<tr>
<td>SndWnd:</td>
<td>00000065482</td>
</tr>
<tr>
<td>StdUna:</td>
<td>2538223807</td>
</tr>
<tr>
<td>MaximumSegmentSize:</td>
<td>0000001310</td>
</tr>
<tr>
<td>Smooth trip time:</td>
<td>184.000</td>
</tr>
<tr>
<td>ReXmt:</td>
<td>0000000002</td>
</tr>
<tr>
<td>DupACKs:</td>
<td>0000000284</td>
</tr>
<tr>
<td>SockOpt:</td>
<td>8D</td>
</tr>
<tr>
<td>TcpSig:</td>
<td>04</td>
</tr>
<tr>
<td>TcpDet:</td>
<td>EC</td>
</tr>
<tr>
<td>QOSPolicy:</td>
<td>No</td>
</tr>
<tr>
<td>RoutingPolicy:</td>
<td>No</td>
</tr>
<tr>
<td>ReceiveBufferSize:</td>
<td>0000016384</td>
</tr>
<tr>
<td>ReceiveDataQueued:</td>
<td>0000000000</td>
</tr>
<tr>
<td>SendDataQueued:</td>
<td>0000000000</td>
</tr>
</tbody>
</table>

**Byte counts**

- **BytesOut:** 0008457981
- **SegmentsOut:** 0000006614
- **SndNxt:** 2538223807
- **ClientSndNxt:** 2538223807
- **InitSndSeqNum:** 252976825

**Network segment counts**

- **SlowStartThreshold:** 00000002620
- **OutgoingWindowNum:** 2538289289
- **SndW12:** 2538223807
- **MaxSndWnd:** 0000131070
- **rtt_seq:** 2538223753
- **DSField:** 00

**Network response time info**

- **SmoothTripVariance:** 84.000
- **ReXmtCount:** 0000000000
- **RcvWnd:** 0000032758
- **TcpTimer:** 00
- **TcpSel:** 40
- **TcpPol:** 00

**Retransmission count**

- **SendBufferSize:** 0000065536
<table>
<thead>
<tr>
<th>Message Code</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>EZZ2350I</td>
<td>MVS TCP/IP NETSTAT CS V1R10</td>
<td>TCP/IP Name: TCPIP</td>
</tr>
<tr>
<td>EZZ2720I</td>
<td>TCP Configuration Table:</td>
<td></td>
</tr>
<tr>
<td>EZZ2721I</td>
<td>DefaultRcvBufSize:</td>
<td>00016384</td>
</tr>
<tr>
<td>EZZ2721I</td>
<td>DefaultSndBufSize:</td>
<td>00016384</td>
</tr>
<tr>
<td>EZZ2727I</td>
<td>DefltMaxRcvBufSize:</td>
<td>00262144</td>
</tr>
<tr>
<td>EZZ2722I</td>
<td>MaxReTransmitTime:</td>
<td>120.000</td>
</tr>
<tr>
<td>EZZ2723I</td>
<td>RoundTripGain:</td>
<td>0.125</td>
</tr>
<tr>
<td>EZZ2724I</td>
<td>VarianceMultiplier:</td>
<td>2.000</td>
</tr>
<tr>
<td>EZZ2725I</td>
<td>DefaultKeepAlive:</td>
<td>00000120</td>
</tr>
<tr>
<td>EZZ2726I</td>
<td>RestrictLowPort:</td>
<td>Yes</td>
</tr>
<tr>
<td>EZZ2728I</td>
<td>TcpTimeStamp:</td>
<td>Yes</td>
</tr>
<tr>
<td>EZZ2729I</td>
<td>TTLS:</td>
<td>No</td>
</tr>
<tr>
<td>EZZ2730I</td>
<td>UDP Configuration Table:</td>
<td></td>
</tr>
<tr>
<td>EZZ2731I</td>
<td>DefaultRcvBufSize:</td>
<td>00065535</td>
</tr>
<tr>
<td>EZZ2732I</td>
<td>CheckSum:</td>
<td>Yes</td>
</tr>
<tr>
<td>EZZ2733I</td>
<td>RestrictLowPort:</td>
<td>Yes</td>
</tr>
<tr>
<td>EZZ2735I</td>
<td>IP Configuration Table:</td>
<td></td>
</tr>
<tr>
<td>EZZ2736I</td>
<td>Forwarding:</td>
<td>Yes</td>
</tr>
<tr>
<td>EZZ3073I</td>
<td>IpSecurity:</td>
<td>No</td>
</tr>
<tr>
<td>EZZ2753I</td>
<td>ArpTimeout:</td>
<td>01200</td>
</tr>
<tr>
<td>EZZ2738I</td>
<td>IgRedirect:</td>
<td>No</td>
</tr>
<tr>
<td>EZZ2739I</td>
<td>StopClawEr:</td>
<td>No</td>
</tr>
<tr>
<td>EZZ2740I</td>
<td>MultiPath:</td>
<td>No</td>
</tr>
<tr>
<td>EZZ2750I</td>
<td>DynamicXCF:</td>
<td>No</td>
</tr>
<tr>
<td>EZZ2746I</td>
<td>IQDIORoute:</td>
<td>No</td>
</tr>
<tr>
<td>EZZ3070I</td>
<td>TcpStackSrcVipa:</td>
<td>No</td>
</tr>
</tbody>
</table>

**TCP CONFIG**

Keepalive timer

**IP CONFIG**

**UDP CONFIG**

No garbage send
Providing Workload Classification Attributes To DB2

- Send accounting, user, and application information to DB2
  - Provides more information for analysis purposes
  - Allows the system more information for WLM prioritization
- ODBC/CLI/VB/ADO ... applications
  - Use SQLSetConnectionAttr on:
    - SQL_ATTR_INFO_ACCTSTR - accounting string (AI)
    - SQL_ATTR_INFO_APPLNAME - application name (PC)
    - SQL_ATTR_INFO_USERID - client userid
    - SQL_ATTR_INFO_WRKSTNNAME - client workstation name
- Universal Driver for Java applications (JCC T2 or T4)
  - Use methods against connection class instance
    - setClientUser
    - setClientApplicationInformation,
    - setClientWorkStation
    - setClientAccountingInformation
An Example Providing Attributes To DB2

!DIS THD(*)
DSNV401I ! DISPLAY THREAD REPORT FOLLOWS -
DSNV402I ! ACTIVE THREADS  241
NAME ST A REQ ID AUTHID PLAN ASID TOKEN
SERVER  RA * 33 javaw.exe DNET581 DISTSERV 008D 282
        V437-WORKSTATION=Ed@IBM, USERID=dnet581,
        APPLICATION NAME=javaw.exe
        V445-G941491B.E205.090605130859=282 ACCESSING DATA FOR 9.65.73.27
CICSACB7 N 3 SYSSTC 009E 0
CICSACB6 N 3 SYSSTC 009D 0

> DISTRIBUTED THREAD DETAIL
PLAN
+ Thread : Plan=DISTSERV Connid=SERVER Corrid=javaw.exe Auth
+ Dist : Type=DATABASE ACCESS, Luwid=G941491B.E205.090605130859=
+ Location : 9.65.73.27
rsum
+ Location IP Addr Port Cthuser Srvclsnm Prod ID Workstation Name
+----------------- -------------- -------- ------------- ---------------
+ 9.65.73.27 0941491B 448 dnet581 NT SQL09013 Ed@IBM
+ Transaction name: javaw.exe
+ TCP/IP Userid: dnet581
+ Distributed SQL Statistics
+ Remote Location Name = 9.65.73.27 Remote Location Luname =

Use CL/I settings to set the workstation name.
What Is ARM?

- ARM – An open industry standard
  - The Open Group ARM Standard V4.0
  - Application Response time Monitoring
  - A standard for how to enable the instrumentation of applications

- ARM enablement provides important performance analysis information
  - The ability to measure and break down performance by application component layer
  - Enable the ability to determine the application timeline

- Two key components to ARM
  - Application instrumentation to provide timing information
  - Collection agent to gather response time information
Summary - Optimizing Performance Tuning The Requesting System

- **Minimize message traffic between the requestor and the server**
  - Optimize SQL
  - Reduce the number of SQL statements
  - Use Stored Procedures to reduce SQL traffic
  - Use RELEASE SQL statement to release remote connections no longer needed
  - Use DEFER(PREPARE) to reduce back and forth message traffic
  - Do not request more columns of data than needed
  - Optimize for ‘n’ rows
Summary - Optimizing Performance Tuning The Server

- Consider DB2 thread connection issues
  - Queuing for an available thread
- Consider DB2 SQL delays and long running SQL
  - SQL cache performance
  - Database I/O and database wait analysis
  - CPU, memory, and resource availability
  - Database lock contention
  - SQL call optimization and Explain analysis
- Operating system delays
  - Proper WLM classification and goals
    - Goals for threads and DB2 address spaces
  - SP address space management
Thank You!