DB2 Connections to DB2 for z/OS
(Distributed Functions of DB2 z/OS)

Paul Wirth
wirthp@us.ibm.com
Disclaimer and Trademarks

Information contained in this material has not been submitted to any formal IBM review and is distributed on "as is" basis without any warranty either expressed or implied. Measurements data have been obtained in laboratory environment. Information in this presentation about IBM's future plans reflect current thinking and is subject to change at IBM's business discretion. You should not rely on such information to make business plans. The use of this information is a customer responsibility.

IBM MAY HAVE PATENTS OR PENDING PATENT APPLICATIONS COVERING SUBJECT MATTER IN THIS DOCUMENT. THE FURNISHING OF THIS DOCUMENT DOES NOT IMPLY GIVING LICENSE TO THESE PATENTS.

TRADEMARKS: THE FOLLOWING TERMS ARE TRADEMARKS OR ® REGISTERED TRADEMARKS OF THE IBM CORPORATION IN THE UNITED STATES AND/OR OTHER COUNTRIES: AIX, AS/400, DATABASE 2, DB2, e-business logo, Enterprise Storage Server, ESCON, FICON, OS/390, OS/400, ES/9000, MVS/ESA, Netfinity, RISC, RISC SYSTEM/6000, System i, System p, System x, System z, IBM, Lotus, NOTES, WebSphere, z/Architecture, z/OS, zSeries

The FOLLOWING TERMS ARE TRADEMARKS OR REGISTERED TRADEMARKS OF THE MICROSOFT CORPORATION IN THE UNITED STATES AND/OR OTHER COUNTRIES: MICROSOFT, WINDOWS, WINDOWS NT, ODBC, WINDOWS 95

For additional information see ibm.com/legal/copytrade.phtml
Agenda

- Overview
- DB2 Clients
- DB2 Connect
- DB2 z/OS and Distributed Connections
- DB2 z/OS and Connection Resources
- z/OS Workload Manager
- DB2 Private Protocol
- DB2 Connect Considerations
- DB2 Connect Tips
Overview:

The purpose of this presentation is to provide an overview of distributed connections to DB2 z/OS. Given distributed transactions cross many different business organizations, gaining an understanding the “big picture” of a distributed transaction is difficult. Frequently I get the comment; “We have a DB2 Connect problem!”. However, the comment is really stating we are have a problem with distributed transactions; and the issue may actually not be with DB2 Connect but some other item. The goal of this presentation is to provide a high level end-to-end view of a distributed transactions and to try and de-mystifying the behavior of DB2 z/OS distributed workloads.
5 “Simple” parts to DB2 Connections

1. **Network**
   - SNA or TCP/IP

2. **DB2 Clients**
   - **Connect** Tools: Control Center, Configuration Assistant, DB2 Command Line Editor & etc…
   - Drivers/Providers:
     - DB2 Call Level Interface (CLI)
     - Microsoft: ODBC, OLE DB and DB2 .NET Data Providers
     - JDBC type 2 and JDBC Type 4 Drivers

3. **DB2 Connect**
   - DB2 Connect PE (This product is specifically designed and licensed for enabling two-tier client/server applications running on individual workstations and, as such, is not appropriate for use on servers.)
     - DB2 Connect Server (Same code – different licenses)
       - DB2 Connect Enterprise Edition (No. Users license)
       - DB2 Connect Unlimited Edition for zSeries (MSU – Includes DB2 Connect PE)
     - JDBC Type 4 Driver (A client that can connect directly to DB2 for z/OS (with DB2 Connect license))

4. **DB2 z/OS**
   - DDF – Network Connections - CONDBAT
   - DBM1 – Database Access Threads – DBATs

5. **z/OS Workload Manager (WLM)**
   - Service Classes and Classification Rules
Typical Topologies for Application Connections to DB2 z/OS

- This example has DB2 Connect Enterprise Server Edition, being use as a gateway Access
- Optionally, DB2 Connect Personal Edition allows client to connect directly to DB2 z/OS
- Optionally, DB2 Connect using Java Universal driver TYPE-4 allows Java network connects to connect directly to DB2 z/OS
- DB2 SYSPLEX used for scalability and high availability

Client Applications:
- Brio
- Excel
- Access

WEB Browser
Client
Application Server
DB2 Connect Servers
DB2 SYSPLEX
Shared Storage

Client 3270 Terminal
CICS
IMS/TM

Database Server
(SQL/Oracle/etc… Federation)
Summary of DB2 Connect Topologies - Roadmap:

1. **Web Application Server with DB2 Connect or Type-4 Java Driver**
   - Client (Internet Explorer)
   - Web Application Server
   - DB2 Connect Gateway Servers

2. **Web Application Server with DB2 Client**
   - Client
   - Web Application Server
   - DB2 Connect Gateway Servers

3. **DB2 Connect PE**
   - Client
   - DB2 Connect Gateway Server on zLinux
   - Hypersockets

4. **DB2**
   - zIIP
   - DB2
   - DDF
   - DBM1
   - WLM

**Tip:** When agents are on the same computer as DB2 Connect set the following DB2 registry variable: `db2connect_in_app_process=n`

**Tip:** Hypersockets are used for DB2 sub systems local to the CEC. Remote DB2s use normal TCP/IP
Things to consider for: availability, redundancy and resiliency in DB2 Connect Gateway Topologies:

1. Excel DB2 Client
   Web Application Sever With DB2 Connect or Type-4 Java Driver
   Edge Sever

2. DB2 Connect Gateway Sever
   IP Sprayer (Load Balancing)
   Web Application Sever With DB2 Connect or Type-4 Java Driver

3. Web Application Sever With DB2 Connect or Type-4 Java Driver
   DB2 Client

*** Tip: When setting up a high-availability system you want to eliminate single points of failure. A potential single point of failure is your network controller. You can Use Virtual IP Addressing (VIPA) instead of using a particular network controller. In case of a network controller failure, another controller can be used.
Things to consider for: availability, redundancy and resiliency in DB2 Connect Gateway Topologies…:

1. Installing DB2 Connect/Type-4 on the application server avoids the additional network hop through a DB2 Connect gateway and avoids the DB2 Connect gateway as a single point of failure. (SAP does this for their DBs on DB2 z/OS)

2. Using DB2 Connect gateways centralizes thread pool management and gateway software maintenance, Use IP Sprayers to avoid DB2 Connect gateway as single point of failure. Note: make sure and rigorously test your IP Sprayer solution because there are lots of combinations available and some work better than others.

3. DB2 Connect on zLinux exploits hypersockets for local DB2 systems in the same CEC. Note: Think about redundancy of DB2 Connect even on zLinux…

- Use Dynamic VIPA for network component transparency for DB2 data sharing
- Use Sysplex Distributor for application transparency (DB2 data sharing)
- Life is not “either or other” do both – Try using DB2 Connect/Type-4 driver on application servers and DB2 Connect gateways for end-user connections
Agenda

- Overview
- DB2 Clients
- DB2 Connect
- DB2 z/OS and Distributed Connections
- DB2 z/OS and Connection Resources
- z/OS Workload Manager
- DB2 Private Protocol
- DB2 Connect Considerations
- DB2 Connect Tips
What are we talking about on the “Roadmap”?

1. Edge Server
   - Web Application Server with DB2 Connect or Type-4 Java Driver
   - Virtual IP Addressing (VIPA) & Sysplex Distributor

2. Web Application Server
   - IP Sprayer (Load Balancing)
   - DB2 Connect Gateway Server on zLinux and Hypersockets

3. Client
   - Internet Explorer
   - Excel
   - DB2 Client
   - Web Application Server
   - DB2 Connect Gateway Server
   - Web Application Server & DB2 Client
DB2 Clients

DB2 V8:

- The Administration Client provides extensive DB2 administrative ability. It supports the following:
  - full function administration of DB2 UDB for UNIX, Windows, and OS/2
  - DB2 for OS/390
  - the entire DB2 family for replication services

- The Run-Time Client contains a subset of functions provided by the DB2 product that enables pre-coded applications to run on a remote node and access DB2 database servers. The Run-Time Client supplies the Command Line Processor (CLP) with the capability to catalog and uncatalog node and database directories, and bind packages.

- The Software Developer Kits (SDK) for a specific platform are provided on each of the DB2 server CDs. The SDK provides a development environment for the supported operating system. If you have a need to develop applications on a machine other than the server, you should acquire the Developer's Edition product appropriate to your needs.
DB2 Clients (Continued)

DB2 9:

- **DB2 9 Runtime Client**
  The best option if your only requirements are to enable applications to access DB2 9 data servers. They provide the APIs necessary to perform this task, but his client comes with no management tools.

- **DB2 9 Client**
  Includes all the functions found in the DB2 Runtime Client plus functions for client-server configuration, database administration, and application development through a set of rich graphical tools. The DB2 9 Client replaces the functions found in both the DB2 8 Application Development and DB2 8 Administration clients.

- **Java Common Client (JCC)**
  This 2 MB fully redistributable client provides JDBC and SQLJ applications access to DB2 data servers without installing and maintaining DB2 client code. If you are connecting to a DB2 for System i or DB2 for System z data server, you are still required to purchase the DB2 Connect product.
DB2 Clients Provide

- Support for common database access interfaces: JDBC, ADO.NET, OLE DB, ODBC, and DB2 Command Line Interface (CLI). This includes drivers and capabilities to define data sources. For example, for ODBC, installing a DB2 client installs the DB2 ODBC driver and registers the driver. Application developers and other users can use the Windows ODBC Data Source Administrator tool to define data sources.
- Base client support to handle database connections, SQL statements, XQuery statements and DB2 commands.
- LDAP exploitation.
- Support for common network communication protocols: TCP/IP, Named Pipe.
- Versions that run on 32-bit and 64-bit operating systems.
- License terms that allow free redistribution of DB2 Runtime Client with your application.
The DB2 V9 Providers/Drivers

1. **DB2 .NET Data Provider**: A high performance, managed ADO.NET data provider. This is the recommended* .NET data provider for use with DB2 family databases. ADO.NET database access using the DB2 .NET Data Provider has fewer restrictions, and provides significantly better performance than the OLE DB and ODBC .NET bridge providers.

3. **OLE DB .NET Data Provider**: A bridge provider that feeds ADO.NET requests to the IBM OLE DB Provider (by way of the COM interop module). This .NET data provider is not recommended* for access to DB2 family databases. The DB2 .NET Data Provider is faster and more feature-rich.

5. **ODBC .NET Data Provider**: A bridge provider that feeds ADO.NET requests to the IBM ODBC Driver. This .NET data provider is not recommended* for access to DB2 family databases. The DB2 .NET Data Provider is faster and more feature-rich.

7. **JDBC type 2** driver are translated to Java native methods. The Java applications that use this driver must run on a DB2 client, through which JDBC requests flow to the DB2 server. DB2 Connect™ must be installed before the DB2 JDBC application driver can be used to access DB2 UDB for iSeries data sources or data sources in the DB2 for OS/390® or z/OS® environments.

9. **JDBC Type 4** JDBC driver is a two-tier pure Java(TM) JDBC driver that allows a Java client to communicate directly with DB2 servers via the DRDA(R) protocol.

11. **DB2 Call Level Interface (CLI)** is IBM's callable SQL interface to the DB2 family of database servers. It is a 'C' and 'C++' application programming interface for relational database access that uses function calls to pass dynamic SQL statements as function arguments. It is an alternative to embedded dynamic SQL, but unlike embedded SQL, DB2 CLI does not require host variables or a precompiler. DB2 CLI is based on the Microsoft(R)** Open Database Connectivity** (ODBC) specification, and the International Standard for SQL/CLI.

13. **New: IBM DB2 Driver for ODBC and CLI** The IBM DB2 Driver for ODBC and CLI provides runtime support for the DB2 CLI application programming interface (API) and the ODBC API. Though the DB2 Client and DB2 Runtime Client both support the DB2 CLI and ODBC APIs, this driver is not a part of either DB2 client. It is available separately, installed separately, and supports a subset of the functionality of the DB2 clients.

*Source: DB2 Version 9 for Linux, UNIX, and Windows - Developing ADO.NET and OLE DB Applications, Developing ADO.NET and OLE DB Applications
DB2 Client … The New “IBM DB2 Driver for ODBC and CLI” in Version 9

Advantages of the IBM DB2 Driver for ODBC and CLI:
- The driver has a much smaller footprint than the DB2 Client and the DB2 Runtime Client.
- You can have multiple installations of the driver on a single machine.
- You can install the driver on a machine that already has a DB2 client installed.
- You can include the driver in your database application installation package, and redistribute the driver with your applications. Under certain conditions, you can redistribute the driver with your database applications royalty-free.
- The driver can reside on an NFS mounted filesystem.

Additional Details:
- Small, 8 meg zipped, 32 meg unzipped
- Supports ODBC and CLI applications today
- Will support .NET applications in 9.X
- Connects natively to DB2 LUW or DB2 for z/OS
- Unzips into a subdirectory … can co-exist with other DB2 Clients on same desktop
- More Info CALL LEVEL INTERFACE AND REFERENCE VOLUME 1, See Chapter 27 for details.
DB2 Client … The New “IBM DB2 Driver for ODBC and CLI” in Version 9…

- Installation/Configuration
  - Unzip file
  - Add valid DB2 Connect license key
  - Run `db2oreg1.exe -i` to add to Windows Registry
  - To `db2cli.ini`:
    - `[ZOS_DB]`
    - `database=ndcdb202`
    - `protocol=tcpip`
    - `hostname=9.39.64.151`
    - `servicename=446`
Updating db2cli.ini

1. Open DB2 Configuration Assistant
2. Right click on desired DB2 z/OS
3. Select “CLI Settings…”
4. Select “Settings” tab
5. Select “Add”
6. Select desired parameter and adjust setting

NOTE: Avoid editing db2cli.ini manually; parameters do have requirements of position in the file, use DB2 Configuration Assistant to avoid mistakes
A brief overview of optional information on DB2 Client tools…
DB2 Client Tools – Used for DB2 LUW and z/OS

- **Control Center** – is used for administration of DB2/IMS Databases. Control Center can also be used to launch all of the addition DB2 tools.

- **Configuration Assistant** - is used for setting up client/server communications and maintaining registry variables, though it can do more. This is used for DB2 LUW, DB2 400 and DB2 z/OS systems.

- **DB2 Command Line Editor** - is an application you can use to run DB2 commands, operating system commands, or SQL statements.

- **Development Center (V8) / DB2 Developer Workbench (V9)** - is used to create business logic for databases (stored procedures and user defined functions).

- **Visual Explain (DB2 LUW version included with client)** – lets you view the access plan for an explained SQL statement as a graph. You can use the information available from the graph to tune your SQL query for better performance.

- **DB2 Command Editor** - Use the DB2 Command Editor to build and execute DB2 commands and SQL statements, and to view a graphical representation of the access plan for an SQL statement.
DB2 V8 - Intelligent Visual Explain – V9 z/OS will have OSC

- Significant improvements to Visual Explain tool:
  - Much more detailed information about access path
  - More statistical details for each node in the graph
  - XML document describing query access path selected
  - Easier collection of information to send for help
EXPLAIN for global prepare cache

- Enhancements to the EXPLAIN statement allow you to obtain EXPLAIN information for entries in the DB2 global prepare cache.
- Visual Explain is enhanced to exploit this new function.

```
EXPLAIN STMTCACHE
   STMTID=integer
   STMTTOKEN=string
   ALL
```

**ALL** requires a DSN_STATEMENT_CACHE_TABLE to be created in advance. See the SQL Reference Guide, Ch 5, EXPLAIN for details.
Some Examples Of The Many DB2 Client Tools
Configuration Assistant

In addition to configuring database and testing connections, connection profiles can be created too. Connection profiles are useful for replication of client setup.
DB2 Command Line Editor (DB2 LUW 9)
DB2 Command Line Processor (DB2 LUW V8)
Command Window & CLP are handy tools for checking connections to DB2 databases and running scripts. Note that the connection string and commands begin with “db2” in Command Window.
Rapid Application Development Tools…
Developer Workbench &
Microsoft Visual Studio

Developer Workbench is a handy tool to test Java Type 2 and 4 connections to DB2 databases.
Clients Can Be Downloaded from:
http://www.ibm.com/software/data/db2/support/db2_9/
Response Files Are A Tool To Help Standardize Client Configurations

- Using an exported profile with a response file will automate client installation

```
Z:\CONPE\setup /U Z:\CONPE\db2conpe.rsp /L C:\db2conpeFP.log /F /T C:\db2conpe_traceFP.log /W
```

- The DB2 V8 manual documenting remote software installation is “DB2 Installation and Configuration Supplement”
db2iprune

db2iprune is a command line utility that you can use to reduce the size of your DB2 Windows product installation image. This tool is useful for large-scale deployments of DB2, as well as for embedding DB2 within an application.

It consists of an input file and a db2iprune executable. The input file, or ".prn" file, contains a full list of removable components and is used to indicate which features and languages you would like removed from the installation image.

The db2iprune executable (db2iprune.exe) removes the cabinet (.cab) files associated with those features and languages. The result is a new, smaller DB2 installation image that can be installed using the regular DB2 installation methods.

Note: db2iprune is a Windows only program.
db2iprune

- Saved over 200MB on the CD image selecting only needed items...
- Instructions are with the utility...
- Available for selected DB2 LUW Windows V8 FP7+ & V9 products

The file directory were db2iprune is stored:

```plaintext
?:\product name\DB2\WINDOWS\UTILITIES\DB2IPRUNE
```
Please wait... The product image is being copied to the destination specified:
C:\_pwirth\Software\DB2 V8 Conn PE\CONPE\db2\Windows\utilities\db2iprune> db2iprune
  -r db2conpe.prn -p "C:\_pwirth\Software\DB2 V8 Conn PE\CONPE" -o "C:\_pwirth\Software\DB2 V8 Conn PE\CONPES_SMALL"

Deleting .cab files for the feature whose token is SYSTEM_BIND_FILES...
Deleting .cab files for the feature whose token is APPLICATION_DEVELOPMENT_TOOLS...
Deleting .cab files for the feature whose token is LDAP_EXPLOITATION...
Deleting .cab files for the feature whose token is CLIENT_TOOLS...
Deleting .cab files for the feature whose token is DB2_WEB_TOOLS...
Deleting .cab files for the feature whose token is DATA_WAREHOUSE_CENTER...
Deleting .cab files for the feature whose token is INFORMATION_CATALOG_CENTER...
Deleting .cab files for the feature whose token is INFORMATION_CATALOG_CENTER_WEB...
Deleting .cab files for the feature whose token is SPATIAL_EXTENDER_CLIENT_SUPPORT...
Deleting .cab files for the feature whose token is XML_EXTENDER...
Deleting .cab files for the feature whose token is FIRST_STEPS...
Deleting .cab files for the feature whose token is CONFIGURATION_ASSISTANT...
Deleting .cab files for the feature whose token is COMMAND_CENTER...
Deleting .cab files for the feature whose token is CONTROL_CENTER...
Deleting .cab files for the feature whose token is ACTIVITY_MONITOR...
Deleting .cab files for the feature whose token is EVENT_ANALYZER...
Deleting .cab files for the feature whose token is DEVELOPMENT_CENTER...
Deleting .cab files for the feature whose token is DB2_SAMPLE_APPLICATIONS...
Deleting .cab files for the feature whose token is SQLJ_SAMPLES...
Deleting .cab files for the feature whose token is WAREHOUSE_SAMPLES...
Deleting .cab files for the feature whose token is INFORMATION_CATALOG_SAMPLES...
Deleting .cab files for the feature whose token is SPATIAL_EXTENDER_SAMPLES...
Deleting .cab files for the feature whose token is XML_EXTENDER_SAMPLES...
Deleting .cab files for all features of language: BR...
Deleting .cab files for all features of language: CN...
Deleting .cab files for all features of language: CZ...
Deleting .cab files for all features of language: DE...
Deleting .cab files for all features of language: DK...
Deleting .cab files for all features of language: FI...
Deleting .cab files for all features of language: FR...
Deleting .cab files for all features of language: ES...
Deleting .cab files for all features of language: IT...
Deleting .cab files for all features of language: JP...
Deleting .cab files for all features of language: KR...
Deleting .cab files for all features of language: NO...
Deleting .cab files for all features of language: PL...
Deleting .cab files for all features of language: SE...
Deleting .cab files for all features of language: TW...

DOS Command driven...
Agenda

- Overview
- DB2 Clients
- **DB2 Connect**
- DB2 z/OS and Distributed Connections
- DB2 z/OS and Connection Resources
- z/OS Workload Manager
- DB2 Private Protocol
- DB2 Connect Considerations
- DB2 Connect Tips
1. What are we talking about on the “Roadmap”?

2. Web Application Server With DB2 Connect or Type-4 Java Driver

3. DB2 Connect Gateway Server on zLinux and Hypersockets (Load Balancing)

Virtual IP Addressing (VIPA) & SysPlex Distributor

zIIP
DB2-DDF
-DBM1
WLM
Virtual IP Addressing (VIPA) & SysPlex Distributor
DB2 Connect

- DB2 Connect provides Distributed Relational Database Architecture (DRDA) Application Requester (AR) function. With Communications Support, DB2 Connect enables applications running on DB2 UDB for UNIX, OS/2, and Windows client platforms to access and update data on the following:
  - DB2 for MVS
  - DB2 for OS/390
  - DB2 for VM and VSE
  - DB2 for OS/400
  - DRDA compliant database management systems
  - Two version: DB2 Connect Personal Edition and DB2 Connect Server Edition available
DB2 Connect
In a nutshell

- Desktop PCs
- Application servers
- Web application servers
- APIs
- Communication
- Data

- DB2 Connect functions
DB2 Connect
3 tiers of a solution

DB2 Run-Time Client

DB2 Connect Server

DB2 for zOS

Data

DB2 address space

VD.NET, C#
ADO.NET

VD Application

C Application

Java Application

Java Application

ADO

CLIDrvr.

JDBC Drvr.

SQLJ Drvr.

DRDA

TCP/IP

DB2 Connect engine

DRDA

TCP/IP

DDF (DRDA)

TCP/IP
DB2 z/OS V8 Server LOCATION Aliases

- BSDS for each DB2 member can include multiple LOCATION names for the DB2 server
- Each LOCATION name is registered with WLM separately for WLM sysplex workload balancing information
- Useful for:
  - consolidating DB2 systems
  - workload routing within sysplex

The extra location alias names are defined in the BSDS with the Change Log Inventory utility (DSNJU003).
DB2 Connect Server
Basic Architecture Overview

- `db2start` starts instance
- Multiple instances/machine
- 1-1-1 relationship between applications, coordinating agents and DB2 threads
- Capacity of the server is determined by the MAXAGENTS (1 to 64K) (default 200)
- `LIST DCS APPLICATIONS` display info for all agents

DB2 Connect Agents:
- On Unix, each agent is a process consuming approximately 700 K of memory.
- On Windows, each agent is a thread consuming approximately 300 K of memory.
DB2 Connect Server
Connection Pooling Overview

- 1-1-1 relationship between applications, coordinating agents and DB2 threads
- Initial pool size is controlled by NUM_INITAGENTS
- Max pool size is controlled by NUM_POOLAGENTS
- Coordinating Agents (CA) and corresponding connection returned to the pool on disconnect
DB2 Connect Server
Connection Concentrator

- N-1 relationship between applications and threads
- CA and corresponding connection returned to the pool on commit/rollback
- Connection concentrator is activated when MAX_CONNECTIONS > MAX_COORDAGENTS
- Initial pool size is controlled by NUM_INITAGENTS
- CA=Coordinating Agents
- CO=Connections

Note: The MAX_LOGICAGENTS parameter was renamed to MAX_CONNECTIONS in V8
**DB2 Connect Connection Concentrator**

**Reduce mainframe resource usage (Saves on DBATs)**

- Connection Concentrator reduces number of DB2 threads
- Saving a thread saves:
  - Active: 250K
  - Inactive: 7K
- You decide how many threads to keep around by setting DB2 Connect parameter `MAX_COORDAGENTS`
- There is a cost (special registers and user id switching)
Sysplex + Concentrator
24*7 for the DB2 for OS/390

- DB2 datasharing: DB2 cluster for high availability
- DB2 Connect Connection
  Concentrator is datasharing aware:
  - Balances workload
  - Routes transactions around failed members

To enable DB2 SYSPLEX (data sharing) support; set the sixth parameter to ‘SYSPLEX’ in the DCS catalog parameter string for the cataloged database on the DB2 Connect Server.
How do I monitor agent activity?
Two tools DB2 snapshot monitor or db2pd

Example of db2pd agent information:

SYSPLEX host information is available using: db2pd --sysplex
How do I monitor agent activity?
One included tool DB2 snapshot monitor...

Example of snapshot monitor:
Can I monitor connection activity?

Step 1 – Setup monitoring
Can I monitor connection activity? (continued…)

Step 2 – Get snapshot…
Can I monitor connection activity? (continued…)

```sql
<table>
<thead>
<tr>
<th>Statement</th>
<th>Static Commit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Section number</td>
<td>0</td>
</tr>
<tr>
<td>Application creator</td>
<td></td>
</tr>
<tr>
<td>Package name</td>
<td></td>
</tr>
<tr>
<td>SQL compiler cost estimate in timerons</td>
<td>0</td>
</tr>
<tr>
<td>SQL compiler cardinality estimate</td>
<td>0</td>
</tr>
<tr>
<td>Statement start timestamp</td>
<td>05/29/2007 23:34:23.381757</td>
</tr>
<tr>
<td>Statement stop timestamp</td>
<td>05/29/2007 23:34:23.471175</td>
</tr>
<tr>
<td>Host response time (sec.ms)</td>
<td>0.089118</td>
</tr>
<tr>
<td>Elapsed time of last completed stmt (sec.ms)</td>
<td>0.089418</td>
</tr>
<tr>
<td>Physical fetches for statement</td>
<td>0</td>
</tr>
<tr>
<td>Time spent on gateway processing</td>
<td>0.000307</td>
</tr>
<tr>
<td>Inbound bytes received for statement</td>
<td>10</td>
</tr>
<tr>
<td>Outbound bytes sent for statement</td>
<td>10</td>
</tr>
<tr>
<td>Outbound bytes received for statement</td>
<td>54</td>
</tr>
<tr>
<td>Inbound bytes sent for statement</td>
<td>54</td>
</tr>
<tr>
<td>Blocking cursor</td>
<td>NO</td>
</tr>
<tr>
<td>Outbound blocking cursor</td>
<td>NO</td>
</tr>
<tr>
<td>Host execution elapsed time</td>
<td>0.000000</td>
</tr>
</tbody>
</table>
```

```
Commit statements = 3
Rollback statements = 0
Inbound bytes received = 2198
Outbound bytes sent = 2246
Outbound bytes received = 9312
Inbound bytes sent = 9205
Number of open cursors = 2
Application idle time = 3 minutes 52 seconds
UOW completion status = Committed - Commit Statement
Previous UOW completion timestamp = 05/29/2007 23:34:23.268331
UOW start timestamp = 05/29/2007 23:34:23.275026
UOW stop timestamp = 05/29/2007 23:34:23.471175
Elapsed time of last completed uow (sec.ms) = 0.196149
Host execution elapsed time = 0.006028
```
Can I monitor connection activity? (continued…)

```
Info for statements taking 2 transmissions:
Number of SQL statements = 9
Total Outbound bytes sent = 2246
Total Outbound bytes received = 9312
Outbound bytes sent high water mark = 1441
Outbound bytes received high water mark = 4126
Outbound bytes sent low water mark = 10
Outbound bytes received low water mark = 54
Maximum network time for statement = 0.116325
Minimum network time for statement = 0.089118
Total Host response time (sec.ms) = 0.687695
Total elapsed times for stmts (sec.ms) = 0.004301

Number of statements with outbound bytes sent:
<table>
<thead>
<tr>
<th>Range</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Between 1 and 128 bytes</td>
<td>3</td>
</tr>
<tr>
<td>Between 129 and 256 bytes</td>
<td>2</td>
</tr>
<tr>
<td>Between 257 and 512 bytes</td>
<td>1</td>
</tr>
<tr>
<td>Between 513 and 1024 bytes</td>
<td>0</td>
</tr>
<tr>
<td>Between 1025 and 2048 bytes</td>
<td>1</td>
</tr>
<tr>
<td>Between 2049 and 4096 bytes</td>
<td>0</td>
</tr>
<tr>
<td>Between 4097 and 8192 bytes</td>
<td>0</td>
</tr>
<tr>
<td>Between 8193 and 16384 bytes</td>
<td>0</td>
</tr>
<tr>
<td>Between 16385 and 31999 bytes</td>
<td>0</td>
</tr>
<tr>
<td>Between 32000 and 64000 bytes</td>
<td>0</td>
</tr>
<tr>
<td>Greater than 64000 bytes</td>
<td>0</td>
</tr>
</tbody>
</table>

Number of statements with outbound bytes received:
<table>
<thead>
<tr>
<th>Range</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Between 1 and 128 bytes</td>
<td>3</td>
</tr>
<tr>
<td>Between 129 and 256 bytes</td>
<td>0</td>
</tr>
<tr>
<td>Between 257 and 512 bytes</td>
<td>0</td>
</tr>
<tr>
<td>Between 513 and 1024 bytes</td>
<td>1</td>
</tr>
<tr>
<td>Between 1025 and 2048 bytes</td>
<td>1</td>
</tr>
<tr>
<td>Between 2049 and 4096 bytes</td>
<td>1</td>
</tr>
<tr>
<td>Between 4097 and 8192 bytes</td>
<td>1</td>
</tr>
<tr>
<td>Between 8193 and 16384 bytes</td>
<td>0</td>
</tr>
<tr>
<td>Between 16385 and 31999 bytes</td>
<td>0</td>
</tr>
<tr>
<td>Between 32000 and 64000 bytes</td>
<td>0</td>
</tr>
<tr>
<td>Greater than 64000 bytes</td>
<td>0</td>
</tr>
</tbody>
</table>

Number of statements with network time:
<table>
<thead>
<tr>
<th>Range</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up to 1 ms</td>
<td>0</td>
</tr>
<tr>
<td>Between 1 and 4 ms</td>
<td>0</td>
</tr>
<tr>
<td>Between 4 and 16 ms</td>
<td>0</td>
</tr>
<tr>
<td>Between 16 and 100 ms</td>
<td>6</td>
</tr>
<tr>
<td>Between 100 and 500 ms</td>
<td>1</td>
</tr>
<tr>
<td>Greater than 500 ms</td>
<td>0</td>
</tr>
</tbody>
</table>
```

DB2 Universal Driver For Java - Type 4 Driver

DB2 Universal Driver Connection Concentrator

- IBM DB2 Driver for JDBC and SQLJ provides connection pool and connection concentrator functionality similar to DB2 Connect (DB2 V8 FP10 and later)
- Ability to reuse server agents at application commit or rollback
Which Connection Pool Should I use?

- To exploit DB2 Data Sharing workload balancing and transparent failover, both, application server connection pool AND connection concentrator/ connection pool should be used.

Diagram:

- Application Server
  - JVM
  - DB2 Universal Driver JDBC/SQLJ
  - Logical Connection 1, 2, 3
  - Transport 1, 2
  - Pooled connections to DB2 Data Sharing Group
  - JCA Connection Manager
  - DB Connection Pool
  - Resource Adapter
  - Application
DB2 Universal Driver Connection Concentrator

- New Global properties defined in Global Properties File:
  - `db2.jcc.maxTransportObjects` - equivalent to MAXAGENTS - max # of connections to DB2 server across all datasources (default value is -1, meaning no limit)
  - `db2.jcc.minTransportObjects` - equivalent to NUM_POOLAGENTS - # of connections kept in the pool across all datasources - # of transport objects will grow as requested but always stay (default value is 0)
  - `db2.jcc.maxTransportObjectIdleTime` - time in sec., a connection stays idle in the pool before it is closed, until minTransportObject is reached (default value is 60 sec)
  - `db2.jcc.maxTransportObjectWaitTime` - if maxTransportObjects is reached - time in sec., an application waits to get a connection before throwing a SQLException (default value is 5 sec)

- New DataSource Properties:
  - `maxTransportObjects` - max # of connections to DB2 server from this DataSource. Can not be bigger than db2.jcc.maxTransportObjects (default value is -1, meaning no limit)
  - `enableConnectionConcentrator` - enables connection concentrator functionality. Not allowed for DB2 LUW (default value is "false" - disabled)
  - `enableSysplexWLB` - enables Sysplex Workload Balancing functionality. Not allowed for DB2 LUW (default value is "false" - disabled)

** Helps saves DBATs on DB2 z/OS
WebSphere Connection Pooling...
The Java Type 4 Driver

- Connection Concentrator is independent of Sysplex Workload Balancer, either feature can be set on or both
  - **Connection Concentrator**
    - Heavyweight Reuse vs Lightweight Reuse
    - Heavyweight Reuse avoided when there are multiple commit scopes within a connect scope. If this isn’t the case, do not use connection concentration, just WebSphere connection pooling
    - Use JCC global transport pool statistics
  - **Sysplex Workload Balancer**
    - Unlike the DB2 Connect gateway, SWB does not depend on connection concentration being activated
    - Use VIPA to always assure a 1st connection
    - WLM sends back a server list
    - Workload balancing as well as failover
    - DB2 data sharing group name must be specified as the WebSphere datasource database
  - Subsetting Alias can be referenced by application instead of the BSDS location name
  - Both features are turned off by default
Type 4 vs DB2 Connect Gateways

- If a Type 4 driver connects through a DB2 Connect gateway, then JCC connection concentration and WSB are ‘turned off’ at the JCC level
- Type 4
  - Better granularity of transport agents that each JVM can use
  - Each JCC driver can specify the maximum number of allowed transport objects (which = DBATs) per datasource
  - Slightly better performance (1-2% faster)
- Gateway
  - Bigger Scope – connection concentration supports multiple JVMs from multiple WAS servers … bigger DBAT savings
  - Easier to manage upgrades and changes in one place (one gateway versus multiple JCC drivers)
Type 4 Driver files

Type 4 drivers are pure Java and implement the network protocol for a specific data source. The client connects directly to the data source.

- **db2jcc.jar** – Type 4 driver
- **sqlj.zip** – SQLJ Support
- **db2jcc_license_cu.jar** – License for: Cloudscape All DB2 Database for Linux, UNIX, and Windows servers
- **db2jcc_license_cisuz.jar** – License for: Cloudscape All DB2 Database for Linux, UNIX, and Windows servers **DB2 for z/OS DB2 UDB for iSeries** (Included with DB2 Connect)

How to find IBM DB2 Driver for JDBC and SQLJ version and environment information:

```
 java -classpath com.ibm.db2.jdbc.DB2Jcc:version=configuration=help=
```
How can I monitor Type 4 driver activity

- To assist you in monitoring the performance of your applications with the DB2 Universal JDBC Driver, the DB2SystemMonitor interface is provided. This interface contains methods that collect the data about a connection.

- To monitor the DB2 Universal JDBC Driver connection concentrator and Sysplex workload balancing, you need to monitor the global transport objects pool. You can monitor the global transport objects pool in either of the following ways:
  - Using traces that you start by setting DB2 Universal JDBC Driver configuration properties
  - Using an application programming interface
  - Configuration properties for monitoring the global transport objects pool:

See DB2 Manual: Application Programming Guide and Reference For Java
Additional DB2 Client and Connect Settings

Client Settings:

- **CLISchema=XXX** A shadow catalog can be created by using the DB2OCAT tool from IBM at the tools site (ftp://ftp.software.ibm.com/ps/products/db2/tools/). This shadow catalog can be maintained automatically by DPROP. This speeds up catalog processing for ODBC and JDBC applications.

- **AutoCommit Usage.** By default, autocommit=1 meaning a commit will flow automatically in the same block in which the SQL is sent to the host. This is good for OLTP environments. But this can be very bad for applications that do a huge amount of INSERTs.

- **Schemalist.** SQL submitted through a dbname where a schema list is specified will be further qualified to only access the tables with the schema’s specified. This can speed up system catalog access considerably.

- **Dbname=PRODUCTION.** Same idea as schemalist, except with regard to a database.

- **DB2_ENABLE_LDAP=Y** within the DB2 Registry can elongate connect times for unpooled connections, so set it to N if not needed. Many circumstances can cause the need for an unpooled connection to be created. These circumstances are detailed in the **DB2 Connect User’s Guide**.

- **Explicitly bind the DDCSMVS.LST with BLOCKING ALL with a userid that has bindadd authority in the NULLID collection.** If you don’t do this and someone connects to the target DB2 for zOS subsystem who just happens to have this needed authority and this person runs a query then the DDCSMVS.LST packages will be bound implicitly without BLOCKING. All subsequent users of the gateway will experience poor query performance since rows will be returned on a one row per block basis.

- **Always use DB2BATCH, not the CLP, when estimating performance from the client desktop to the host DB2 subsystem.**

- **MS ACCESS does not use the most efficient ODBC instruction for pulling data from a database.** Other tools vendors (even MS Query) have corrected this problem a long time ago. Consequently, for large result sets that query will appear to run much faster in any query tool than in MS ACCESS.

**NOTE:** Avoid editing db2cli.ini manually, parameters do have requirements of position in the file, use DB2 Configuration Assistant to avoid mistakes.
Additional DB2 Client and Connect Settings…

Connect Settings:

- TCPKEEPALIVE setting in operating system on which the DB2 Connect EE software runs is important in eliminating DB2 hanging threads under certain circumstances. If tcpipkeepalive requests by that operating system are not responded to, then DB2 Connect running on that operating system will tell DB2 on zOS to kill the DB2 threads that were being used by the applications on the non-responding platforms. A five minute setting is suggested.
- Configure enough memory on the DB2 Connect gateway to prevent paging.
- When collocating DB2 Connect EE on the server as MTS or WAS, set the following DB2 registry variable `db2connect_in_app_process=n`. This will allow local applications to connect through the DB2 instance. You must do a db2start in order to accomplish connectivity. This will also allow local connections to take advantage of two other performance enhancements within DB2 Connect, namely connection pooling and connection concentration. Since local connections will appear to DB2 Connect as though they were remote connections, you can use the LIST DCS APPLICATIONS and GET SNAPSHOT FOR ALL DCA APPLICATIONS.
- `TCPIPRECEIVEWINDOW` (NT – 8k, UNIX – 16k defaults) – raise to 64k or better invoke tcpip window scaling by issuing a: `db2set –g db2sosndbuf=65537 and db2set –g db2sorcvbuf=65537` if your operating system supports it. Check the db2diag.log as this may be done automatically as of DB2 Connect V8.1. This raises to 64k, the amount of data that can be transmitted between the mainframe and DB2 Connect gateway as well as gateway and client, before an acknowledgment must be sent.
- Remove the queuing that DELAYACK (on DB2 for zOS IP port as well on the DB2 Connect IP port), that affects incoming traffic.
Additional DB2 Client and Connect Settings…

**IP Network Settings:**

- CISCO Router defaults, eg. + PROTOCOL BLOCKING=yes. Change to no!!! If no, then traffic is queued for up to 5 ms.
- Firewall port timeout values for firewalls between the mainframe and the DB2 Connect gateway … set to never timeout. or you’ll see DSNL511I error messages on the mainframe DB2 log
- Change the default incoming Queue Length on the device card in DB2 Connect gateways
  UNIX On UNIX systems (especially AIX), this is accomplished by changing the device characteristics such as Transmit and/or Receive queue depth. The defaults are 32 and a recommended value is 150.
- As mentioned before, change the TCPIPKEEPALIVE in DDF and on DB2 Connect gateway.
Additional DB2 Client and Connect Settings…

Application Coding Practices:

- Use parameter markers instead of literals with SQL (set Patch2=18 for automatic insert of SQL literal to parameter marker substitution. For SELECT SQL, programmers must code parameter markers. This makes Access Plan reuse made possible on DB2 for zOS
- Use the SET CLIENT INFO api or java method within applications. This is useful for identifying pooled threads, doing performance or problem diagnosis. It is also possible to write a WLM policy to give better performance for certain users or applications.
- Only bring back the rows needed within the application by limiting results with predicates. Sort and aggregate using SQL, not your program. Use SQL and api constructs to limit rows by using: Fetch First n Rows, Optimize for n Rows, ODBC: set SQL_MAX_ROWS statement option, or JDBC: setMaxRows on a statement object.
- Use embedded SQL instead of dynamic SQL to eliminate access plan overhead.
- Avoid the use of CURSOR WITH HOLD unless needed.
- Commit your work often, commit even if you are reading, commit on return from stored procedures, exception: mass inserts then use appropriate commit scope, control your commit scope within your program instead of relying on autocommit.
- For bulk insert, update, delete activity use COMPOUND SQL to reduce network time, but remember DB2 for zOS supports non-atomic COMPOUND SQL so your application is responsible for transaction scope
- Use CLI trace for problem determination and performance tuning. This is supported for CLI, ODBC, OLEDB and .NET Data Provider environments. Use JDBC tracing for java environments.
Agenda

- Overview
- DB2 Clients
- DB2 Connect

Coffee!

- DB2 z/OS and Distributed Connections
- DB2 z/OS and Connection Resources
- z/OS Workload Manager
- DB2 Private Protocol
- DB2 Connect Considerations
- DB2 Connect Tips
What are we talking about on the “Roadmap”?

1. Edge Sever
2. Web Application Sever With **DB2 Connect or Type-4 Java Driver**
3. DB2 Connect Gateway Sever on zLinux and Hypersockets
   - Virtual IP Addressing (VIPA) & SysPlex Distributor

Client
Internet Explorer
Excel
Web Application Sever
**DB2 Client**
IP Sprayer (Load Balancing)
**DB2 Connect Gateway Sever**

Client
Web Application Sever
**DB2 Client**

Client
Web Application Sever
**DB2 Client**
DB2 z/OS Thread Pooling –
In addition to DB2 Connect Pooling

DB2 thread pooling is a mechanism to share DBATs between connected applications. It is implemented wholly within DB2 for z/OS and OS/390, and can be exploited by connections from any DRDA Application Requestor. This support is sometimes called type 2 inactive thread support, but DB2 thread pooling is a far better and more accurate description of this type of pooling.

The benefits for DB2 z/OS Thread Pooling are:
- CPU savings in DB2, by avoiding repeated creation and destruction of DBATs
- Real memory savings in z/OS, by reducing the number of DBATs
- Virtual storage savings in DBM1, by reducing the number of DBATs
- Greater capacity to support DRDA connections
DB2 z/OS V8 Address Spaces: Zooming in on DBM1 & DDF

Threads:
- DBATs
- CONDBATs

System engines
- User threads
- Pipe Manager
- Local DSC

System storage
- FIXED
- STACK
- Global DSC

DBD cache
- EDM DBD cache
- Compression Dictionaries
- Castout Buffers

Virtual Buffer Pools
- Buffer Pool Control Blocks
- Virtual Buffer Pools

EDM Pool
- EDM Pool
- RID lists

RID pool
- RDS OP pool

RDS OP pool
- Buffer and DM Trace tabe

Pipe Manager
- Local DSC

Connections Network
- Connections Network
# Key DB2 DDF Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Possible values</th>
<th>Default - V8*</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDF</td>
<td>NO / AUTO / COMMAND</td>
<td>No</td>
<td>DDF Startup</td>
</tr>
<tr>
<td>CMTSTAT</td>
<td>ACTIVE / INACTIVE</td>
<td>INACTIVE*</td>
<td>Thread Pooling - Pool inactive threads (CICS, IMS, TSO, Batch attach, SPUFI, Classic QMF, etc)</td>
</tr>
<tr>
<td>CTHREAD</td>
<td>1-2000</td>
<td>200*</td>
<td>Max remote active DDF Threads - DBM1 Address Space</td>
</tr>
<tr>
<td>MAXDBAT</td>
<td>0-1999</td>
<td>200*</td>
<td>Max remote connections - DDF Address Space</td>
</tr>
<tr>
<td>CONDBAT</td>
<td>0-150000</td>
<td>10000*</td>
<td>Max inactive DBATs, these are used for private protocol. DRDA uses inactive connections.</td>
</tr>
<tr>
<td>MAXTYPE1</td>
<td>0-CONDBAT</td>
<td>0</td>
<td>Approximate time, in seconds that a DBAT can remain idle in the pool before it is terminated.</td>
</tr>
<tr>
<td>POOLINAC</td>
<td>0-9999</td>
<td>120</td>
<td>Idle thread timeout. – Could set a “little” higher than TCPKPALV</td>
</tr>
<tr>
<td>IDTHTOIN</td>
<td>0-9999</td>
<td>120*</td>
<td>TCP/IP already verified user identification</td>
</tr>
<tr>
<td>TCPALVER</td>
<td>YES / NO</td>
<td>No</td>
<td>TCP/IP keep alive (Goes hand-in-hand with IDTHTOIN)</td>
</tr>
<tr>
<td>TCPKPALV</td>
<td>ENABLE / DISABLE / 1-65534</td>
<td>120*</td>
<td>Periodically contract each thread’s working storage area.</td>
</tr>
<tr>
<td>CONTSTOR</td>
<td>YES / NO</td>
<td>No</td>
<td>Use storage management algorithms that minimize the amount of working storage consumed by individual threads.</td>
</tr>
<tr>
<td>MINSTOR</td>
<td>YES / NO</td>
<td>No</td>
<td>Use storage management algorithms that minimize the amount of working storage consumed by individual threads.</td>
</tr>
</tbody>
</table>
Thread Types:

- There are two kinds of inactive threads used in DB2 for OS/390 and z/OS, called type 1 (Private Protocol) and type 2 inactive threads. Type 2 inactive threads are only for DRDA connections, and type 2 inactive threads are preferred over type 1 inactive threads.

- A type 1 inactive thread is really an inactive thread. If a thread becomes type 1 inactive, the DBAT remains in DBM1, and remains linked to the connection information in the DDF address space. The only thing that happens when the thread becomes type 1 inactive, is that DB2 tries to clean up as much storage that is no longer in used or required by the thread as possible.

- Type 2 inactive threads, on the other hand, are not really inactive threads. They are inactive connections in the DDF address space. At commit time the connection information inside DDF is separated from the DBAT. The DBAT itself is returned to a pool of (inactive) DBATs and can be reused by another connection.

DB2 z/OS V8 renamed Type 1 / 2 Inactive Threads to: Inactive DBATs & Inactive Connections (respectively)
DB2 9 for z/OS and Private Protocol

- Prepare for elimination of PRIVATE protocol requester
  - Includes a tool for identifying which packages need to be bound at remote servers
    - Tool DSNTP2DP can be executed with DSNTIJPD
Thread Monitoring:

- **SQL1226N** - Issued by DB2 Connect if the maximum number of agents has already been started (“the maximum number of client connections is already started”).

- **-DIS DDF DETAIL** provides real time information on DBAT and CONDBAT metrics.

- **Hung connection** MAXDBAT is reached, a new connection request to DB2 for z/OS and OS/390 will simply wait for a DBAT to become available...forever. Using the DB2 Statistics Trace and tools like DB2 PE will report the number of times that DBATs queued. No error message issued. Customers complain system slow...

- **Maximum concurrent connections** - represents the number of times the maximum number of connections (CONDBAT) was reached. DB2 PE will report this condition also. Customers complain system slow...
APAR-PK01230: DISPLAY THREAD SERVICE(WAIT) BOOST ENHANCEMENT

- The “-DISPLAY THREAD SERVICE(WAIT)” command can issue a priority boost for agents holding a latch for 2x IRLM timeout limit or a minimum of 1 minute.
- An MVS SYSEVENT ENQHOLD enhancement was needed for DB2 to boost agents effectively.
- Problem conclusion:
  - The -DISPLAY THREAD SERVICE(WAIT) command exploits the SYSEVENT ENQHOLD enhancements introduced in APAR OA08949.
  - By using this boost DB2 will:
    1. Not deboost an agent after issuing a priority boost. The boost is now self-expiring.
    2. Boost at enclave and address space granularity.
    3. Not be susceptible to timing windows.
TCPIP Keepalive Considerations

- Set TCPKPALV in DDF parms
  - DB2 can detect when DB2 Connect gateways die
  - DB2 can clean up affiliated DBATs
- Set TCPKEEPALIVE on DB2 Connect server’s operating system
  - DB2 Connect server can detect when desktops and Web Application servers die
  - DB2 can clean up affiliated DBATs
How do I figure out thread activity?

Some type of monitor tool is needed like Omegamon or DB2 PE (like below) is needed.
Agenda

- Overview
- DB2 Clients
- DB2 Connect
- DB2 z/OS and Distributed Connections
- DB2 z/OS and Connection Resources
- z/OS Workload Manager
- DB2 Private Protocol
- DB2 Connect Considerations
- DB2 Connect Tips
How Many threads can DB2 z/OS handle?
(Is 800 DBATs high?)

\[ CTHREAD + MAXDBAT \leq 2000 \]  
“well probably not 2000”

How many threads can be supported will vary for each DB2 system:

- Available Virtual Storage
  - DBM1 Address Space (V7 – Bufferpools, Dynamic Statement Cache)
- Thread footprint (size of the thread)
  - Stored Procedures, Large SQL Statements
- Duration of a thread
  - Long running program, Commit Frequency, CURSORHOLD
- Number of allied threads (threads started locally)
  - CAF/RRSAF, TSO QMF Users, Batch Jobs, IBM Regions, CICS transactions, DB2 Utilities and …
- Number of Remote threads
What happens if you don’t have enough storage for threads? (bad things…) 

- Examples of problems:
  - Abends due to either E20003 or E20016 abends of user transactions
  - 878 abends of DBM1 if no storage available for must complete work (backout)

- Key Thread Parameters:
  - CTHREAD - Number of allied threads (threads started at the local)
  - MAXDBAT – Database Access Thread (DBAT)
  - \( CTHREAD + MAXDBAT \leq 2000 \) is the maximum
    - Many customers support MAXDBATs 100s to 400
    - *Note: this is very dependent on site thread profile*
Where Do Treads Live in DB2 z/OS – V8

DBM1 64 bit Virtual Memory Map

2^{64} < 16EB "beam" (?)

- Buffer pools, BM control blocks
- Castout buffers
- RID pool
- EDM DBD cache (OBDs)
- Global dynamic stmt cache
- Sort pool
- Trace tables (Global, Lock, BB)
- Accounting blocks
- Compression dictionaries
- IRLM locks

2^{32} < Dead Space

2^{31} < Below 2GB

< 2GB "bar"

- Most thread storage. Local DSC
- Other EDM and RID component
How Do I figure out thread usage and virtual storage available?

See Judy Ruby-Brown’s Teleconference: Data Sharing Health Checks….What We Have Learned


- Memory Reporting Tool: http://www.ibm.com/support/docview.wss?rs=64&uid=swg27007819

- Section “Virtual Storage in DBM1” reviews thread storage utilization and provides a guide to compute utilization and use of memory tool.

- If there is a need to have many more DBATs than a single DB2 subsystem can hold, data sharing can help this by scaling a DB2 system wide (with more members).
How Many Treads Can DB2 z/OS Handle

<table>
<thead>
<tr>
<th>Dataspace Lookaside Pool Size</th>
<th>Current Number of Active DBATs</th>
<th>Maximum Number of Active DBATs</th>
<th>Full System Contraction</th>
<th>Storage Critical</th>
<th>Basic Storage Cushion</th>
<th>Upper Limit Total</th>
<th>Total Fixed</th>
<th>Upper Limit Variable</th>
<th>Thread Footprint</th>
<th>Maximum Number Threads</th>
<th>Total Active Threads</th>
</tr>
</thead>
<tbody>
<tr>
<td>36</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>197</td>
<td>1117</td>
<td>218</td>
<td>699</td>
<td>1.2</td>
<td>749</td>
<td>11</td>
</tr>
<tr>
<td>36</td>
<td>4</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>197</td>
<td>1117</td>
<td>218</td>
<td>699</td>
<td>1.37</td>
<td>656</td>
<td>10</td>
</tr>
<tr>
<td>36</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>197</td>
<td>1117</td>
<td>219</td>
<td>898</td>
<td>1.36</td>
<td>660</td>
<td>9</td>
</tr>
<tr>
<td>36</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>197</td>
<td>1117</td>
<td>219</td>
<td>698</td>
<td>1.05</td>
<td>855</td>
<td>13</td>
</tr>
<tr>
<td>36</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>197</td>
<td>1117</td>
<td>220</td>
<td>897</td>
<td>1.33</td>
<td>674</td>
<td>9</td>
</tr>
<tr>
<td>36</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>197</td>
<td>1117</td>
<td>220</td>
<td>897</td>
<td>1.51</td>
<td>594</td>
<td>15</td>
</tr>
<tr>
<td>36</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>197</td>
<td>1117</td>
<td>223</td>
<td>894</td>
<td>1.48</td>
<td>604</td>
<td>14</td>
</tr>
<tr>
<td>36</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>197</td>
<td>1117</td>
<td>224</td>
<td>893</td>
<td>1.28</td>
<td>608</td>
<td>16</td>
</tr>
<tr>
<td>36</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>197</td>
<td>1117</td>
<td>224</td>
<td>893</td>
<td>1.4</td>
<td>638</td>
<td>15</td>
</tr>
<tr>
<td>36</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>197</td>
<td>1117</td>
<td>224</td>
<td>893</td>
<td>1.26</td>
<td>700</td>
<td>12</td>
</tr>
<tr>
<td>36</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>197</td>
<td>1117</td>
<td>224</td>
<td>893</td>
<td>1.43</td>
<td>624</td>
<td>12</td>
</tr>
<tr>
<td>36</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>197</td>
<td>1117</td>
<td>224</td>
<td>893</td>
<td>1.22</td>
<td>732</td>
<td>11</td>
</tr>
<tr>
<td>36</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>197</td>
<td>1117</td>
<td>224</td>
<td>893</td>
<td>1.3</td>
<td>687</td>
<td>11</td>
</tr>
<tr>
<td>36</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>197</td>
<td>1117</td>
<td>224</td>
<td>893</td>
<td>1.4</td>
<td>638</td>
<td>9</td>
</tr>
<tr>
<td>36</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>197</td>
<td>1117</td>
<td>224</td>
<td>893</td>
<td>1.46</td>
<td>612</td>
<td>9</td>
</tr>
<tr>
<td>36</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>197</td>
<td>1117</td>
<td>224</td>
<td>893</td>
<td>1.37</td>
<td>652</td>
<td>10</td>
</tr>
<tr>
<td>36</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>197</td>
<td>1117</td>
<td>224</td>
<td>893</td>
<td>1.25</td>
<td>714</td>
<td>13</td>
</tr>
<tr>
<td>36</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>197</td>
<td>1117</td>
<td>225</td>
<td>892</td>
<td>1.21</td>
<td>738</td>
<td>12</td>
</tr>
<tr>
<td>36</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>197</td>
<td>1117</td>
<td>225</td>
<td>892</td>
<td>1.39</td>
<td>642</td>
<td>10</td>
</tr>
<tr>
<td>36</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>197</td>
<td>1117</td>
<td>225</td>
<td>892</td>
<td>1.39</td>
<td>642</td>
<td>10</td>
</tr>
<tr>
<td>36</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>197</td>
<td>1117</td>
<td>225</td>
<td>892</td>
<td>1.25</td>
<td>714</td>
<td>13</td>
</tr>
<tr>
<td>36</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>197</td>
<td>1117</td>
<td>225</td>
<td>892</td>
<td>1.49</td>
<td>599</td>
<td>9</td>
</tr>
<tr>
<td>36</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>197</td>
<td>1117</td>
<td>225</td>
<td>892</td>
<td>1.53</td>
<td>583</td>
<td>10</td>
</tr>
<tr>
<td>36</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>197</td>
<td>1117</td>
<td>225</td>
<td>892</td>
<td>1.32</td>
<td>670</td>
<td>11</td>
</tr>
<tr>
<td>36</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>197</td>
<td>1117</td>
<td>225</td>
<td>892</td>
<td>1.51</td>
<td>501</td>
<td>10</td>
</tr>
<tr>
<td>36</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>197</td>
<td>1117</td>
<td>225</td>
<td>892</td>
<td>1.5</td>
<td>595</td>
<td>9</td>
</tr>
<tr>
<td>36</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>197</td>
<td>1117</td>
<td>225</td>
<td>892</td>
<td>1.29</td>
<td>681</td>
<td>13</td>
</tr>
<tr>
<td>36</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>197</td>
<td>1117</td>
<td>225</td>
<td>892</td>
<td>1.51</td>
<td>591</td>
<td>8</td>
</tr>
<tr>
<td>36</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>197</td>
<td>1117</td>
<td>226</td>
<td>891</td>
<td>1.45</td>
<td>614</td>
<td>10</td>
</tr>
</tbody>
</table>
How Many Threads Can DB2 z/OS Handle
Agenda

- Overview
- DB2 Clients
- DB2 Connect
- DB2 z/OS and Distributed Connections
- DB2 z/OS and Connection Resources
- z/OS Workload Manager
- DB2 Private Protocol
- DB2 Connect Considerations
- DB2 Connect Tips
What are we talking about on the “Roadmap”?

1. Web Application Server with DB2 Connect or Type-4 Java Driver

2. DB2 Connect Gateway Server on zLinux and Hypersockets

3. Virtual IP Addressing (VIPA) & Sysplex Distributor

- Edge Server
- Web Application Server
- IP Sprayer (Load Balancing)
- DB2 Connect Gateway Server
- zIIP
- DB2 DDF
- DBM
- WLM
- Virtual IP Addressing (VIPA)
- Sysplex Distributor

Client
Internet Explorer
Excel
DB2 Client
Web Application Server
DB2 Client
Web Application Server
DB2 Client
Web Application Server
DB2 Client
Web Application Server
DB2 Client
Workload Manager Overview

IWEB
DB2
DDF
JES2
OMVS
STC

Service Class
- Critical
- Ad hoc
- Accounting & Payroll
- Marketing & Sales

Report Class
- Marketing
- Sales
- Headquarters
- Test

Business objectives
Monitoring
Workload Manager...

50 Concurrent Users

Killer Queries

Runaway queries cannot monopolize system resources aged to low priority class

<table>
<thead>
<tr>
<th>period</th>
<th>duration</th>
<th>velocity</th>
<th>importance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5000</td>
<td>80</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>50000</td>
<td>60</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>1000000</td>
<td>40</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>10000000</td>
<td>30</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>discretionary</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Avg response time in seconds

- trivial: 12, 10
- small: 48, 46
- medium: 101, 95
- large: 162, 155
Workload Manager (WLM)

Attention: If you do not define classification rules for DDF work and you are running in WLM goal mode, then all your DDF work will run in the SYSOTHER service class, which has a discretionary goal. This means that your DDF work will only run if the system has no other work to do, which normally means that your DDF work is executed at a very low priority.

Before DB2 V4 and MVS/ESA V5.2, all work in the DB2 distributed data facility address space was running with the same dispatching priority. Obviously this was not desirable, since you could not distinguish between your high-priority DDF work and lower-priority DDF work. With the introduction of enclaves, DDF transactions can be managed separately from the DDF address space itself. Work coming into DDF is classified as an assigned service class by workload manager (WLM). Depending on the classification rules you specify, WLM can assign service classes to DDF work, and associate different goals with these service classes according to the type of your DDF work.

When the enclave is deleted depends on whether the DBAT can become pooled, type 2 connections only (DRDA). If the DBAT becomes pooled, the enclave is deleted. If the DBAT cannot become pooled, the enclave is only deleted at thread termination time. Since WLM assigns the performance goals to the enclaves, it is the lifetime of the enclave that WLM takes as the duration of the work. Therefore, when you run with CMTSTAT=INACTIVE, DDF creates one enclave per transaction, and response time goals and multiple time periods can be used.
How do I manage different workloads?  
(AKA – Identifying the loved ones.)

1) Create Service Classes  
   Rule of thumb 3 groups – High, Medium & Low  

2) Identify work:  
   User Id., Program Name, etc..  

3) Assign service class to identified work
Service Classes

Service classes identify the priority of different types of work

Service Class Selection List

Row 1 to 11 of 11

Command ===> 

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete, /=Menu Bar

<table>
<thead>
<tr>
<th>Action</th>
<th>Class</th>
<th>Description</th>
<th>Workload</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ASCH</td>
<td>APPC Transaction Programs</td>
<td>STCTASKS</td>
</tr>
<tr>
<td>1</td>
<td>BATCH</td>
<td>Batch Workload</td>
<td>BATCH</td>
</tr>
<tr>
<td>1</td>
<td>CICS</td>
<td>CICS Transactions</td>
<td>ONLINE</td>
</tr>
<tr>
<td>1</td>
<td>DB2QUERY</td>
<td>DB2 Sysplex Queries</td>
<td>DATABASE</td>
</tr>
<tr>
<td>1</td>
<td>DDFDEF</td>
<td>DDF Default Requests</td>
<td>DATABASE</td>
</tr>
<tr>
<td>1</td>
<td>DDFBAT</td>
<td>DDF low priority</td>
<td>DATABASE</td>
</tr>
<tr>
<td>3</td>
<td>DDFONL</td>
<td>DDF High priority</td>
<td>DATABASE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Service class</th>
<th>Period</th>
<th>Duration</th>
<th>Importance</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDFONL</td>
<td>1</td>
<td>500</td>
<td>2</td>
<td>90% within 1 second</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
<td>3</td>
<td>Velocity 40</td>
</tr>
<tr>
<td>DDFDEF</td>
<td>1</td>
<td>500</td>
<td>3</td>
<td>80% within 2.5 seconds</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
<td>4</td>
<td>Velocity 20</td>
</tr>
<tr>
<td>DDFBAT</td>
<td>1</td>
<td>500</td>
<td>4</td>
<td>80% within 4 seconds</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
<td>5</td>
<td>Velocity 10</td>
</tr>
</tbody>
</table>
### Work Qualifiers

Work qualifiers are used to help identify a thread or work.

- **AI** - Accounting Information
- **CI** - Correlation Information
- **CN** - Collection Name
- **CT** - Connection Type
- **CTG** - Connection Type Group
- **LU** - LU Name
- **LUG** - LU Name Group
- **NET** - Net ID
- **NETG** - Net ID Group
- **PC** - Process Name
- **PF** - Perform
- **PFG** - Perform Group
- **PK** - Package Name
- **PKG** - Package Name Group
- **PN** - Plan Name
- **PNG** - Plan Name Group
- **PR** - Procedure Name
- **PX** - Sysplex Name
- **SI** - Subsystem Instance
- **SIG** - Subsystem Instance Group
- **SSC** - Subsystem Collection
- **UI** - Userid
Providing Workload Classification Attributes from Client

- 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
- Non-ODBC... use sqleseti Administrative API function
- Universal Driver for Java applications (JCC T2 or T4)
  - Use methods against connection class instance
    - setClientUser
    - setClientApplicationInformation,
    - setClientWorkStation
    - setClientAccountingInformation

DISPLAY Thread(*) DETAIL
Details for thread token 41731
V437-WORKSTATION=IBM-WIRTHP, USERID=dba031,
   APPLICATION NAME=db2bp.exe
V441-ACCOUNTING=ME@PAUL
V445-G94382E0.AC0C.070530023901=41731 ACCESSING DATA FOR 9.67.130.224
V447-LOCATION SESSID A ST TIME
V448--9.67.130.224 448:3244 W R2 0714921383699
Providing Workload Classification Attributes from Client (Continued…)

To send accounting strings from your client applications to the DB2 Connect server, use the API-specific means for setting accounting information. The API-specific means perform faster than setting the DB2ACCOUNT environment variable.

- **IBM DB2 Driver for JDBC and SQLJ**
  - `com.ibm.db2.jcc.DB2BaseDataSource.clientAccountingInformation` property
- **DB2 .NET Data Provider**
  - `DB2Connection.ClientAccountingInformation` property
- **CLI/ODBC**
  - `ClientAcctStr` CLI/ODBC configuration keyword
- **Embedded SQL (C, C++, and COBOL)**
  - `sqlesact` function
Classification Rules

Classification rules connect Service Classes with work qualifiers, or DB2 threads/transactions. When setting up classification rules, plan for “defaults” to accommodate new unplanned for production applications.

<table>
<thead>
<tr>
<th>Subsystem-Type</th>
<th>Xref</th>
<th>Notes</th>
<th>Options</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>====&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subsystem Type</td>
<td>.. : DDF</td>
<td>Fold qualifier names?</td>
<td>Y (Y or N)</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>... DDF Work Requests</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Action codes: A=Affect \ C=Copy \ M=Move \ I=Insert rule \ B=Before \ D=Delete row \ R=Repeat \ IS=Insert Sub-rule

<table>
<thead>
<tr>
<th>Action</th>
<th>Type</th>
<th>Name</th>
<th>Start</th>
<th>Service</th>
<th>Report</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>SSC</td>
<td>D7F1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>PN</td>
<td>CASH*</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>PN</td>
<td>LOAN*</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>PC</td>
<td>BI</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>PC</td>
<td>OLTP</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>SI</td>
<td>D82G</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>UE</td>
<td>DEV1*</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>UE</td>
<td>DEV2*</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>UE</td>
<td>BART*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The Service Class performance data can be reported in RMF which can provide information on z/OS system level thread activities. Reporting Classes can be created in addition to Service Classes for details within Service Classes.
Sample DDF Definition – Available on the web

How can I figure out thread Enclave activity?

RMF Monitor III

- Overview
- Enclave Report
The RMF Service class reports can provide information on thread activities too. Reporting Classes can be created in addition to Service Classes for details within Service Classes.
IBM System z9 Integrated Information Processor (IBM zIIP)

- New specialty engine for the System z9 mainframe:
  - Customers integrate data across the enterprise
  - Improve resource optimization and lower the cost of ownership for eligible data serving workloads

- z/OS manages and directs work between the general purpose processor and the zIIP
  - Number of zIIPs per z9 not to exceed number of standard processors.

- DB2 for z/OS V8 will be first user of the zIIP with
  - System z9 109
  - z/OS 1.6 or later
  - DB2 for z/OS V8
  - DB2 zIIP information APAR: ii14219

- Webcast replay:
DB2 Eligible DB2 z/OS V8 Workloads

- ERP, CRM and multi tier application serving*
  - For applications, running on z/OS, UNIX, Linux, Intel, or Linux on System z, that access DB2 for z/OS V8 on a System z9, via DRDA over a TCP/IP connection DB2 gives z/OS the necessary information to have portions of these SQL requests directed to the zIIP.

- Data warehousing applications*
  - Requests that utilize DB2 for z/OS V8 star schema parallel queries may have portions of these SQL requests directed to the zIIP when DB2 gives z/OS the necessary information.
    - This function has been enhanced to include all parallel queries.

- Some DB2 for z/OS V8 utilities*
  - A portion of DB2 utility functions used to maintain index maintenance structures (example LOAD, REORG, and REBUILD INDEX) typically run during batch, can be redirected to zIIP.

* The zIIP is designed so a program can work with z/OS to have a portion of it’s Service Request Block (SRB) enclave work directed to the zIIP. The above types of DB2 V8 work are those executing in SRB enclaves, portions of which can be sent to the zIIP.
Specialty engines work together

The IBM System z9 specialty engines can run independently or complement each other
(shown at left, instances where specialty engines can be employed)
DB2 z/OS V9 Continues to add zIIP features –
(DRDA invoked SQL-PL stored procedures zIIP eligible)

Native SQL Procedural Language

- Eliminates generated C code and compilation
- Fully integrated into the DB2 engine
- Extensive support for versioning:
  - VERSION keyword on CREATE PROCEDURE
  - CURRENT ROUTINE VERSION special register
  - ALTER ADD VERSION
  - ALTER REPLACE VERSION
  - ALTER ACTIVATE VERSION
- BIND PACKAGE with new DEPLOY keyword

See Roger Miller’s May 26, 2006 web cast for more DB2 z/OS V9 details
Agenda

- Overview
- DB2 Clients
- DB2 Connect
- DB2 z/OS and Distributed Connections
- DB2 z/OS and Connection Resources
- z/OS Workload Manager
- DB2 Private Protocol
- DB2 Connect Considerations
- DB2 Connect Tips
Private Protocol

- **Inbound/outbound private protocol still works**
  - V5, V6, V8 and V9 officially deprecated private protocol
- **DBPROTCL ZPARAM has been removed from V9**
  - DBPROTOCOL(DRDA) will be assumed for any BIND/REBIND PLAN/PACKAGE where DBPROTOCOL option is not specified
  - If DBPROTOCOL(PRIVATE) explicitly specified, DSNT226I message will be issued and BIND will complete with a warning return code (4)
  - Therefore, in V9 you cannot have a default DB Protocol of Private anymore
Private protocol to DRDA protocol Catalog Analysis Tool (DSNTP2DP)

- Tool provided in DB2 V9
- DB2 REXX support must be enabled
- APAR PK40433 will permit program to be run against V7 and V8 subsystems
- APAR PK44544 (not yet closed) will change DSNTIJPD (sample job to run tool)
  - Include optional step to create additional indexes on catalog to improve tool’s performance
  - Include optional step to REBUILD INDEX with inline statistics on indexes and RUNSTATS on index that cannot have inline statistics during rebuild
  - Include optional step to delete indexes

- DB2 for z/OS Examples Trading Post allows V7/V8 subsystem user to download tool (with no service provided) and JCL

- Program currently will only generate commands to migrate plans or packages which can be determined to have a remote location dependency
  - Plans or packages which utilize embedded dynamic SQL programming will usually not have a remote location dependency that can be determined by querying the catalog
Private Protocol Considerations

- Since DB2 V6, you can use aliases and three-part names in combination with the DRDA protocol. This provides some form of location transparency and makes migration from DB2 PP to DRDA easier.
- Don’t forget remote binds of plans or packages could increase the size of remote catalogs – not a bad thing just think about it
Functions not available in the DB2 Private Protocol

- Connections using TCP/IP
- Connections to RDBMS other than DB2 for z/OS and OS/390
- Connection using the SQL CONNECT statement
- DDL (for example, CREATE, DROP) and DCL (for example, GRANT, REVOKE) SQL statements
- Stored procedures
- Remote bind
- Static SQL
- User-defined and LOB data type
- Thread pooling
- Any new features new in V7, V8 or V9
Agenda

- Overview
- DB2 Clients
- DB2 Connect
- DB2 z/OS and Distributed Connections
- DB2 z/OS and Connection Resources
- z/OS Workload Manager
- DB2 Private Protocol
- DB2 Connect Considerations
- DB2 Connect Tips
Planning for DB2 Connect Server
Hardware Platform Dilemmas

- **Choices:**
  - Windows
  - UNIX: AIX, HP-UX, SUN 32 and 64-bit
  - Linux Intel and AMD
  - Linux on zSeries

- **Questions:**
  - Is UNIX better than Windows?
  - Is 64-bit good?
  - Is Linux on zSeries a good idea?
  - How much memory should I plan for?
  - What should we spend money on and what is not worth it?
DB2 Connect Server Planning
Choosing 32 vs. 64-bit

- DB2 Connect server is available in both 32 and 64-bit
- Can create 32 and 64-bit instances on the same machine
- Don’t run 32-bit on Itanium 2 hardware
- Have not observed significant performance advantage on 64-bit
Planning for DB2 Connect Server
Is Linux on zSeries a good idea?

- Runs on the same hardware as the DB2 itself.
- Very flexible:
  - Allocate unused capacity to DB2 Connect.
  - Easy to do capacity on demand.
- IFL hardware is cheaper than z/OS CPUs
- HiperSockets are great: <1ms latency and very high throughput
- Good choice for server consolidation (especially older UNIX boxes) and when looking for flexible capacity allocations
Planning for DB2 Connect ServerLinux on zSeries and HiperSockets

- In-memory TCP/IP
- Very low latency
- High throughput
- Reduces locking contention
- Good performance especially for latency sensitive workload

![Graph comparing Gigabit Ethernet and HiperSockets]
DB2 Connect Server Planning
Criteria for Choosing Hardware

- Choose CPU by looking at SpecInt numbers
- Make sure that you have enough memory:
  - Windows: 250K per connection
  - UNIX: 750K per connection
- Strive for a balanced system (Intel servers)
- Take care of network connectivity:
  - Gigabit Ethernet is great and relatively inexpensive (watch out for latency issues on Windows)
  - 100BT Ethernet is inexpensive and extra capacity and fault tolerance can be gained by combining up to 4 adapters
  - Some adapters will off-load IPSec encryption to hardware
- Don’t worry about anything else, it does not matter
DB2 Connect Planning
Check your network connection!

- Simple tools:
  - PING: simple response time measurement. Vary packet sizes (-l parameter)
  - TRACERT: tells you about network hops
  - ROUTE PRINT: see how your TCP/IP routing table

- NetIQ Qcheck (free from http://www.netiq.com/qcheck/default.asp)

- Check that DDF has sufficient priority

- DB2PING command
  - Like ping but helps you see both network + DDF time
Running Applications on a DB2 Connect Server

When you run an application on a DB2 Connect Server server as a local client, we recommend setting DB2CONNECT_IN_APP_PROCESS to NO. (The default value is YES.) Specifying NO means that local DB2 Connect clients on a DB2 Connect Enterprise Edition are forced to run within an agent. This allows these local clients to be monitored, and they can also use the SYSPLEX support. The default value is YES.

When using YES, an application programs sends requests using threads in its process. In that case, you cannot monitor the performance of an application or use the SYSPLEX support. YES is only recommended in cases where there are only a small number of clients.

Note: SAP uses this configuration where DB2 Connect is installed on the application server with their application servers to simplify network setup and improve performance.
DB2 Connect V8
CLI Performance

Query performance

Client CPU load

transactions

CPU utilization

clients

V8  V7.2 FP7

V8  V7.2 FP7

clients
DB2 Connect
Considerations for migrating to V9

- DB2 Connect version 8 supports connectivity to DB2 for z/OS 7.1 (RSU0606), 8.1 (RSU0606) and 9
- Migrate DB2 Connect servers to V9 first, then migrate clients
- Take advantage of capability to run multiple versions on the same server (UNIX, LINUX)
- When DB2 UDB and DB2 Connect are on the same server run them in separate instances
Agenda

- Overview
- DB2 Clients
- DB2 Connect
- DB2 z/OS and Distributed Connections
- DB2 z/OS and Connection Resources
- z/OS Workload Manager
- DB2 Private Protocol
- DB2 Connect Considerations
- DB2 Connect Tips
Application Development Tips

A few tips that have proven to be useful.
Tip # 1: Do not fear COMMITment

- **Do**
  - Commit your work often
  - Commit even if you are reading
  - Commit on return from stored procedures

- **Don’t**
  - Use Autocommit unless you have to
Tip # 2: Do not hold on to things you don’t need

- **Connections:**
  - applications must release connections or connection pooling will not work
    - (eg. VB programmers must destroy their connection objects and must not put them in to things like global.asa)
  - Use connection concentrator to reduce connection resources

- **Cursors:**
  - Avoid CURSOR WITH HOLD at all costs
  - CURSOR WITH HOLD is the default
  - Turn it off with CURSORHOLD=NO
  - “BECAREFUL SOME ISVs REQUIRE CURSORHOLD=YES”
Tip # 3: Only ask for what you need and get the database to do the job

- Limit results with predicates
- Sort and aggregate using SQL not in your application
- Fetch only what you need:
  - Embedded SQL:
    - FETCH FIRST n ROWS,
    - OPTIMIZE FOR n ROWS
  - ODBC: set SQL_MAX_ROWS statement option
  - ADO: set MaxRecords property of the Recordset
  - JDBC: setMaxRows on a Statement object
Tip # 4: Optimize network transfers

- Blocking is key: 32K blocks instead of a single row at a time
- DB2 Connect pre-fetch capability (automatic)
- Use Stored Procedures for business logic that requires more than one SQL statement
- Extra Query Blocks:
  - Enable on DB2 for OS/390 by setting EXTRA BLOCKS SRV parameter on the DB2 DDF installation panel
  - Enable in application via:
    - Embedded SQL: FETCH FIRST n ROWS, OPTIMIZE FOR n ROWS
    - ODBC: SQL_MAX_ROWS
    - ADO: set MaxRecords property of the Recordset
    - JDBC: setMaxRows
Tip # 4: Optimize network transfers (part 2)

- TCP/IP: RFC-1323 Window Scaling:
  - On DB2 Connect server: set DB2SORCVBUF to greater than 64K (eg. `db2set DB2SORCVBUF =65537`)
  - On z/OS: set TCPRCVBUFSIZE to any value above 64K
  - Measure improvement before putting in production

- Defer prepares: set `DEFEREDPREPARE=1` in the `DB2CLI.INI`
Tip # 4: Use Stored Procedures (part 3)

- Greatly minimize network traffic
- Can write using static SQL
- Excellent way to component based development:
  - Use the same stored procedures from batch, interactive and client-server applications
  - Present it to programmers as SQL, Java Session Bean or a Web Service
- Leverage WLM to manage priority and resource allocation
Tip # 5: Help DB2 do things better

- Help DB2 reuse cached statements by using parameter markers instead of literals:
  - prepare from something already cached is 10K instructions
  - prepare of a simple statement not in cache is 150K instructions
  - prepare of a complex query not in cache can be millions of instructions
- Help static profiler better match statements by using parameter markers in your SQL
Tip # 6: Use CLI trace for problem determination and tuning

- Enable through DB2CLI.INI or by using DB2 CA.
  - [Common]
  - ;TRACEFLUSH=1
  - TRACE=1
  - TRACEPATHNAME=D:\Temp\Traces
  - TRACECOMM=1
- Restart the application after turning on trace
- Don’t set TRACEFLUSH if you don’t have to
- Set TRACECOMM to see network transmissions (very handy for spotting PREPAREs that are not deferred)
- Use TRACEPATHNAME to have every process and thread trace to their own file
Tip # 6 (continued): Making sense of the CLI trace

```
SQLConnectW( hDbc=0:1, szDSN="DWCTRLDB" - X"440057004300540052004C0044004200", cbDSN=8, szUID="db2admin" -
X"640062003200610640060D069006E00", cbUID=-3, szAuthStr="*", cbAuthStr=-3 )
    ---> Time elapsed - +1.390000E-004 seconds
    sqlccsend( ulBytes - 1618 )
    sqlccsend( Handle - 21474176 )
    sqlccsend( ) - rc - 0, time elapsed - +7.740000E-004
    sqlccrecv( )
    sqlccrecv( ulBytes - 1262 ) - rc - 0, time elapsed - +1.250000E-004
    sqlccsend( ulBytes - 609 )
    sqlccsend( Handle - 21474176 )
    sqlccsend( ) - rc - 0, time elapsed - +7.260000E-004
    sqlccrecv( )
    sqlccrecv( ulBytes - 237 ) - rc - 0, time elapsed - +1.518200E-001
    ( DBMS NAME="DB2/NT", Version="07.02.0003", Fixpack="0x23040105" )
    ( Application Codepage=1252, Database Codepage=1252, Char Send/Recv Codepage=1252, Graphic Send/Recv
      Codepage=1252, Application Char Codepage=1252, Application Graphic Codepage=1252 )

SQLConnectW( )
    <--- SQL_SUCCESS Time elapsed - +1.833160E-001 seconds
    ( DSN="DWCTRLDB" )
    ( UID="db2admin" )
    ( PWD="********" )
    ( DBALIAS="DWCTRLDB" )
```
Tip #7: Know where the work is coming from (SQLSETI API)

- Provide information about origin of work for each:
  - Connection
  - Transaction
- Can use WebSphere Admin console to set parameters
- Types of information that can be provided:
  - SQLE_CLIENT_INFO_USERID (16 char max)
  - SQLE_CLIENT_INFO_WRKSTNNNAME (18 char max)
  - SQLE_CLIENT_INFO_APPLNAME (32 char max)
  - SQLE_CLIENT_INFO_ACCTSTR (200 char max)
Tip #8: querytimeoutinterval

- querytimeoutinterval CLI/ODBC configuration keyword
- An application can use the SQLSetStmtAttr() function to set the SQL_ATTR_QUERY_TIMEOUT statement attribute.
- The QueryTimeoutInterval configuration keyword is used to indicate how long the CLI driver should wait between checks to see if the query has completed.
- For instance, suppose SQL_ATTR_QUERY_TIMEOUT is set to 25 seconds (timeout after waiting for 25 seconds), and QueryTimeoutInterval is set to 10 seconds (check the query every 10 seconds). The query will not time out until 30 seconds (the first check AFTER the 25 second limit).
- For instance, suppose SQL_ATTR_QUERY_TIMEOUT is set to 25 seconds (timeout after waiting for 25 seconds), and QueryTimeoutInterval is set to 10 seconds (check the query every 10 seconds). The query will not time out until 30 seconds (the first check AFTER the 25 second limit).
- Microsoft applications often have SQL_ATTR_QUERY_TIMEOUT set to 30, and therefore terminate before any DB2 z/OS timeout parameters. SQLSTATE 57014 - Processing was canceled as requested.
Tip #9: Cutting down latency in sending data from a Mainframe

- **Use fastest connection to mainframe**
  - DB2 Connect on Linux for zSeries (Hipersockets)
  - OSA Express 1 GB or 10 GB adapters
  - Configure jumbo frame support on DB2 Connect server (set MTU to 8192)
  - For large result sets, set DMKRIIO to 64k and set OPTIMIZEFORNROWS=10000
  - Change default queue length for network card to 150
Tip #9: Cutting down latency in sending data from a Mainframe …

- Set delayack=n for the port DB2 listens to on the mainframe (default is 446)
- Use connection pooling and connection concentration to speed up the connection processing
- Use multi-row insert and multi-row update
Tip #9: Cutting down latency in sending data from a Mainframe …

- Setup jumbo frames (MTU=8192) for Incoming DB2 Connect traffic to the mainframe
  - 1. Set MTU for each interface separately in TCP profile in the Gateway section
    - This allows regular ethernet connections to have MTU=1500
    - This allows hipersockets and 10Gbe connections to have MTU=8192
  - 2. Turn on PATHMTUDISCOVERY in IPCONFIG so MTU negotiation can take place and fragmentation is avoided
Tip #10: Removing overhead of access plan creation

- Code applications with parameter markers
  - If not possible, use PATCH2=18 for parameter marker substitution for inserts
  - Consider KEEPDYNAMIC=1 in db2cli.ini as well as CACHEDYN=Y … consider second set of ODBC packages to limit usage to certain apps … there is a write-up I can send to interested parties
Tip #11: Making your application DB2 Connect “Friendly”

- Do not use autocommit=1, make application responsible for commit processing
- Use multi-row array insert syntax as well as multi-row update syntax
- Fetch only the rows needed … Fetch N Rows or Optimize for N Rows
Tip #12: Reducing DB2 catalog access overhead

- Use Schemalist="’Payroll’,’Accounting’"
- Dbname=Production
- CLISCHEMA=xxx … shadow catalog created by DB2OCAT tool
Tip #13: Mainframe Settings suggestions

- If using keepdynamic, set cachdyn=y in zParms
- Set Idle Thread Timeout (IDTHTOIN) reasonable low, eg. 5 minutes to clean-up hanging threads
- Set up RACF with a large cache, cachpac=32768
- Setup WLM policies for enclave SRBs
Tip #14: General recommendations

- DB2degree=any to exploit CPU and IO parallelism
Tip #15: So your experiencing slowdowns

- **Recommended:**
  - Repeatable set of queries with historical metrics running behind different api’s, like JDBC, ODBC, and OLEDB. DB2BATCH can be used for ODBC/CLI metrics. There are freeware tools for JDBC and OLEDB. Don’t use the CLP for measuring query response time.
  - Good change management
Tip #15: So your experiencing slowdowns …

- Check CLASS 3 times for high IP activity or disk activity on the mainframe
- Is the CPU pegged and WLM policies not explicitly specified?
- Is the DB2 Connect server paging?
- Is the network card in the DB2 Connect server pegged?
- DB2TRC ALLOC needed for non-AIX, non-Windows clients
Tip #15: So your experiencing slowdowns...

- Is it the application, mainframe or network?
  - Take a representative query, run it in DB2BATCH with the –CLI option and a CLI trace turned on
    - The CLITRACEPARSER tool will tell you where the bulk of the time is being spent (app vs MF/Network)
    - Running the same query through SPUFI will tell you how much time was spent on the MF
Tip #16: Something to be aware of

Application running on a DB2 client performs poorly due to low level semget calls issued by DB2

**Technote (FAQ)**

**Problem**
This document provides troubleshooting information for the situation where an application running on a remote DB2 client performs poorly. In particular, it describes a scenario where performance is poor because of failed semget() calls to the operating system.

**Cause**
DB2® Universal Database™ (DB2 UDB) Version 8 and DB2 Version 9 have an internal trace facility called "db2trc". This trace facility acquires Inter Process Communication (IPC) resources (both semaphore and shared memory). Even though a DB2 trace is not turned on, it may issue semget() calls to the operating system. These calls will fail, though, since no IPC resources were acquired.

**Solution**
The trace facility has an "alloc" option that can allocate necessary IPC resources. When you issue `db2start`, the command `db2trc alloc` is automatically issued, and when you issue `db2stop`, the trace segment will be removed if no one is using it.

Since a DB2 client does not necessarily have the ability to perform the `db2start` command, no IPC's will be automatically allocated for the trace facility, and hence the problem.

The solution is to issue the following command at the DB2 client:

`db2trc alloc`
Tip #17: Other virtues of DB2 Connect

- Federation Support + OLEDB Consumer
  - Make any SQLServer or Oracle table appear to be a DB2 for zOS table (R/O support)
- OLEDB Provider Support
  - Through a SQLServer LINKED SERVER definition, any DB2 for zOS table can appear to be a SQLServer table (R/W support)
- Only DB2 LUW and DB2 Connect provide MS Visual Studio 2005 plug-ins
Additional Information

This is the BOOK to have!
Available at: ibm.com/redbooks
SG24-6952

Too bad they don’t have “DB2 Connect” in the title…
Additional Information

- **Redbook** – Distributed Functions of DB2 for z/OS and OS/390 SG24-6952-00
- John Campbell’s IOD Conference Presentation 1282 - DB2 for z/OS V8 Migration and Experiences
- John Campbell’s - Virtual Storage Relief in DB2 V8...What to Expect
- Judy Ruby-Brown’s - Data Sharing Health Checks….What We Have Learned
- Mark Rader - DB2 and WLM: An Introduction to Enclaves at New England DB2 User Group
  June 21, 2005
- Maryela Weihrach’s - WebSphere and DB2 Pooling and Caching…Help with Your Decision
  [ibm.com/software/zseries/telecon/fecb6](http://ibm.com/software/zseries/telecon/fecb6)
- Hugh Smith Share presentation: An Introduction to Distributed Processing with DB2 for z/OS
- Ed Woods - DB2 and Workload Manager: setting optimal goals, part two