Overview

- DB2 on z/OS Distributed Processing Intro
- DDF (Distributed Data Facility) Introduction
- Distributed Workflows
- WLM and enclave SRBs
- Performance and Tuning Considerations
Distributed DB2 on z/OS Concepts
What is the DDF (Distributed Database Facility)?

- DB2 on z/OS Component that supports connectivity with other databases across the network
- Implements the full DRDA Application Requester / Application Server function types
- Is DB2’s transaction manager for distributed connections and workloads
- Supported network protocols
  - SNA (VTAM) for DB2 on z/OS to DB2 on z/OS workloads, if desired
  - TCP/IP for everything, recommended approach
DB2 on z/OS Architecture

- 3 DB2 address spaces and many allied agents in the beginning
- DB2 distributed database support and a new address space
  - DB2 2.2 supported DB2 on MVS to DB2 on MVS only using private protocol
  - DB2 2.3 introduced Remote Unit of Work Support via DRDA protocol (level 1)
  - DB2 V3 introduced Distributed Unit of Work via DRDA protocol
- DB2 V4 implemented DRDA support of stored procedures, DBAT priorities, increased connections (25,000)
- DB2 V5 (DRDA level 3) supported TCP/IP, ODBC, CLI and JDBC
DB2 on z/OS
Connection Options

Data Server Drivers include:
• IBM Data Server Client
• IBM Data Server Runtime Client
• IBM Data Server Driver for JDBC and SQLJ
• IBM Data Server Driver for ODBC and CLI

**** Other vendors provide distributed data access drivers

DRDA

DB2 on other platform PRD1
Distributed Fundamentals
A Tale of Two Protocols

**Private Protocol (PP)**
- First delivered in DB2 2.2
- Uses 3 part names
  - Select C1,C2,C3 from DB2G.APPL1.TABLE1
- Dynamic SQL only
  - No remote bind
- DB2 on z/OS to DB2 on z/OS only
- No stored procedure support
- DB2 V10 eliminates private protocol

**DRDA (Distributed Relational Database Architecture)**
- Introduced in DB2 2.3
- Supports 3-part names and explicit CONNECT statements
- Dynamic and Static SQL
  - Remote bind capabilities
- Supports stored procedures
- Supports all RDBMS implemented using DRDA protocol
- Supports SNA and TCP/IP
- Is the strategic architecture for distributed

**Specifying the protocol**
- Default can be set at subsystem level using DBPROTCL DSNZPARM value (DRDA or PRIVATE)
  - With DB2 9.1 PRIVATE can no longer be specified
- Can also be specified in the DBPROTOCOL parm of the BIND statement (D or P)
  - Only option in DB2 9
Database Access Threads (DBATs)

- Service distributed workloads
- Implemented as an MVS WLM enclave running in preemptive SRBs originating in the DIST address space
- DBAT Types
  - DBAT (Server)
  - DBAT (Dist)

Allied Threads

- Service local attachment facilities
- Run at the dispatching priority of the requesting application
- Allied Agent Types
  - Allied
  - Allied Dist (requester)
DDF Implementation
DSNZPARM Considerations

- **Specific DDF ZPARMs**
  - MAXDBAT – Max remote database access threads
  - CONDBAT – Max remote connections
  - DDF – DDF startup option
  - CMTSTAT – DDF Threads
  - IDTHTOIN – Idle Thread Timeout
  - TCPKPALV – TCP/IP Keepalive
  - POOLINAC – Pool Thread Timeout
  - ACCUMACC and ACCUMUID
  - EXTRAREQ/EXTRASRV – Extra Blocks REQ / SRV

- **Application-related**
  - KEEP_DYNAMIC(YES) / MAX_KEEPD

- **Specified on DSNTIPR and DSNTIP5**
## DDF ZPARM Definitions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Location</td>
<td></td>
</tr>
<tr>
<td>DDF Startup Facility Name</td>
<td></td>
</tr>
<tr>
<td>DDF Start Option</td>
<td></td>
</tr>
<tr>
<td>Database Protocol for 3-Part Names</td>
<td></td>
</tr>
<tr>
<td>DDF Max Number of Facility Entries</td>
<td></td>
</tr>
<tr>
<td>DBAT Status after Commit</td>
<td></td>
</tr>
<tr>
<td>Idle Thread Timeout (Seconds)</td>
<td></td>
</tr>
<tr>
<td>Minutes between Resync Periods</td>
<td></td>
</tr>
<tr>
<td>TCP/IP KEEPALIVE</td>
<td></td>
</tr>
<tr>
<td>DDF Interval Cycle Frequency</td>
<td></td>
</tr>
<tr>
<td>DDF Queued Conversation Time</td>
<td></td>
</tr>
<tr>
<td>DDF Receive Buffer Size</td>
<td></td>
</tr>
<tr>
<td>Max Extra DRDA Query Blocks for DB2 Req</td>
<td></td>
</tr>
<tr>
<td>Max Extra DRDA Query Blocks for DB2 Svr</td>
<td></td>
</tr>
<tr>
<td>Check Connection State</td>
<td></td>
</tr>
<tr>
<td>Accounting for DDF and RRSAF threads</td>
<td></td>
</tr>
<tr>
<td>Aggregate field used for DDF/RRSAF thread</td>
<td></td>
</tr>
</tbody>
</table>

© Copyright 4/20/2010 BMC Software, Inc
DDF ZPARAM Definitions – page 2

<table>
<thead>
<tr>
<th>DBAT Thread Controls</th>
<th>STATS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Concurrent Database Access Threads (CONDBAT)</td>
<td>10000</td>
</tr>
<tr>
<td>Maximum Remote Database Access Threads (MAXDBAT)</td>
<td>200</td>
</tr>
<tr>
<td>Maximum Type 1 Inactive Threads (MAXTYPE1)</td>
<td>0</td>
</tr>
<tr>
<td>DDF Pool Thread Timeout Value (POOLINAC)</td>
<td>120</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DDF-Related Authorization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extended Security (EXTSEC)</td>
</tr>
<tr>
<td>ID Sent to Second Server (HOPAUTH)</td>
</tr>
<tr>
<td>Accept Already Verified TCP/IP Connects (TCPALVER)</td>
</tr>
<tr>
<td>DDF RLF Access Error Parameter (RLFERRD)</td>
</tr>
<tr>
<td>DDF RLF Service Unit Limit (RLFERRD)</td>
</tr>
</tbody>
</table>
Enabling Distributed Processing
DDF Local Site Information

- Recorded in the BSDS (Boot Strap Data Set)
- DDF settings for each local subsystem are built during installation (DSNTIPR)

DDF Parameters
- Site location name: DIA
- Luname for site to use: DIALL
- Password for luname: -NONE- (specify "-NONE-" for no password)
- Generic LU Name: -NONE- (specify "-NONE-" for no generic luname support)
- Idle thread status: INACTIVE (ACTIVE or INACTIVE)
- TCP/IP PORT Number: 7240 (0-65534)
- TCP/IP RESPORT Number: 7241 (0-65534)
- TCP/IP SECPORT Number: __________ (0-65534)

- Change Log Inventory utility can be used to modify these values post installation
DDF Security Considerations

- Security Mechanisms differ for DRDA and SNA environments
  - DRDA supports authentication using SNA security mechanisms or DRDA mechanisms

- For DB2 on z/OS as a requester
  - Both SNA and TCP/IP protocols support authentication
    - User ID only (already verified)
    - User ID/password
    - User ID and PassTicket
  - For TCP/IP with z/OS Integrated Cryptographic Service Facility
    - Encrypted user ID and encrypted security-sensitive data

- For DB2 on z/OS as a server
  - Accepts either SNA or DRDA authentication mechanisms
  - Supports the following authentication
    - User ID only (already verified)
    - User ID/password
    - User ID and PassTicket

- Communications database entries tell DB2 what to expect
DDF Security Considerations

For DB2 on z/OS as a server

- Accepts either SNA or DRDA authentication mechanisms
- Supports the following authentication
  - User ID only (already verified)
  - User ID/password
  - User ID and PassTicket
  - Kerberos tickets
  - Unencrypted user ID and encrypted password
  - Encrypted user ID and encrypted password
  - User ID, password, and new password
- z/OS Integrated Cryptographic Service Facility supported authentication mechanisms
  - Combinations of encrypted user ID, password, and security-sensitive data

Communications Database entries tell DB2 what to expect
Enabling Distributed Processing
The CDB – Communications Database

CDB stored in DSNDB06.SYSDDF tablespace
- LOCATIONS
- IPNAMES
- IPLIST
- LUNAMES

TCP/IP side of the CDB
- SYSIBM.IPLIST
  • Contains list of multiple IP addresses specified for a given location
- SYSIBM.IPNAMES
  • Defines host names and outbound security to connect other systems using TCP/IP
- SYSIBM.LULIST
  • Supports member specific access to remote data sharing group
- SYSIBM.LULIST
  • Supports member specific access to remote data sharing group
- SYSIBM.LUNAMES
  • Identifies remote DB2s for access via VTAM
  • Default row can be used to serve requests from any system not listed
- SYSIBM.USERNAMES
  • Support AUTHID Translations

VTAM side of the CDB
- SYSIBM.LOCATIONS
  • One entry for every location
  • LINKNAME establishes relationship with other CDB table entries
DBAT Processing Modes

- Mode is defined with the ZPARM CMTSTAT
  - Very critical option for DDF that defines the flow of DBAT processing
  - “DDF Threads” on panel DSNTIPR

- Two choices:
  - INACTIVE – highly recommended
    - Provides DBAT pooling for DRDA access
    - More effective WLM classification per UOW
    - Reduced Resource usage
  - ACTIVE
    - DBAT created for each new client application
    - DBAT held through commits
    - Use this only if the applications require it

- ZPARM Considerations
  
<table>
<thead>
<tr>
<th>ZPARM</th>
<th>INACTIVE / ACTIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMTSTAT</td>
<td>Used / not applicable</td>
</tr>
<tr>
<td>IDTHTOIN</td>
<td>Used / not recommended</td>
</tr>
<tr>
<td>CONDBAT:MAXDBAT</td>
<td>&gt; / = *****CONDBAT should be much higher</td>
</tr>
</tbody>
</table>
DB2 Thread Pooling

- Driven by CMTSTAT INACTIVE in DSNZPARM
  As active DBATs are freed they remain in the pool and can be immediately reused

- Often called Type 2 Inactive Thread support
  - Use the Thread Pooling terminology as it’s more accurate

- DB2 Connect ESE and JDBC Type 4 Driver also provide thread pooling functions that can (and often) should be used with DDF Thread Pooling

Benefits
- Reduces CPU required to create and destroy DBATs
- Reduced number of DBATs decreases real and virtual storage requirements
- Better scalability for distributed connections
DBAT Workflow
CMTSTAT INACTIVE

CHARACTERISTICS
- No end user “think time” included
- Enclave is created when the first SQL is received
- Enclave is deleted at commit/rollback (thread complete)
- New enclave for each UOW, recategorized by WLM
- Use multi-period response time or velocity goals

**New Connection**
- > CONDBAT?
  - Yes: Reuse DBAT
  - No: Pooled DBAT Avail?
    - Yes: Reuse DBAT
    - No: MAXDBAT Reached?
      - Yes: Queue
      - No: Create DBAT
        - First SQL/UOW?
          - Yes: Reply “ready” to client
          - No: Process SQL
            - Yes: Commit/Rollback?
              - Yes: End enclave / Write Acctg
              - No: Pool DBAT / Inactv. Conn.
            - No: End enclave / Write Acctg
DBAT Workload
CMTSTAT ACTIVE

CHARACTERISTICS
- End User “think time” is included
- Enclave is created when DBAT is created
- Enclave is deleted at thread termination
- Only one enclave, no reclassification
- Can only use a single-period velocity goal

New Connection

MAXDBAT Reached?
Yes
No
Queue

Create DBAT

Reply “ready” to client

Process SQL

Commit / Rollback?
Yes
No

Term Thread?
Yes
No

Term. DBAT & Connection

End enclave / Write Acctg
WLM Enclaves

- WLM Enclaves are independent, dispatchable units-of-work that span multiple address spaces and can include combinations of SRBs and TCBs
  - Concept of “Business Unit-of-Work”

- DB2 on z/OS uses enclaves for work coming through DDF
  - Controlled by WLM
  - Eligible for zIIP processors

- Thread priority set by WLM workload classification
  - Providing good DDF classifications is vital
  - No DDF workload classification defaults to SYSOTHER which means minimal service
### Setting Service Classes for WLM-Managed Distributed Workloads

#### Modify Rules for the Subsystem Type

**Subsystem Type:** DDF  
**Fold qualifier names:** Y (Y or N)  
**Description:** DB2 DDF

**Action codes:**  
A=After  
B=Before  
C=Copy  
D=Delete row  
M=Move  
R=Repeat  
I=Insert rule  
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>SI</td>
<td>DIA1</td>
<td>___</td>
<td>DB2PRDDF</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PN</td>
<td>DIST*</td>
<td>___</td>
<td>DB2PRMED</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UI</td>
<td>RND*</td>
<td>___</td>
<td>DB2PRHI</td>
<td>HIDIST</td>
</tr>
<tr>
<td></td>
<td>SI</td>
<td>DHN1</td>
<td>___</td>
<td>DB2PTLOW</td>
<td>RNDDIA1</td>
</tr>
<tr>
<td></td>
<td>PR</td>
<td>SPRARPT1</td>
<td>___</td>
<td>SCPRTMAX</td>
<td>ARHIPRI</td>
</tr>
<tr>
<td></td>
<td>PR</td>
<td>ARRP1</td>
<td></td>
<td>ARRP1</td>
<td></td>
</tr>
</tbody>
</table>

**Defaults:**
- DB2PRDDF
- DB2PRMED
- DB2PRHI
- DB2PTLOW
- SCPRTMAX
- ARHIPRI
- ARRP1
<table>
<thead>
<tr>
<th>Sel</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI</td>
<td>Accounting Information</td>
<td></td>
</tr>
<tr>
<td>CI</td>
<td>Correlation Information</td>
<td></td>
</tr>
<tr>
<td>CN</td>
<td>Collection Name</td>
<td></td>
</tr>
<tr>
<td>CT</td>
<td>Connection Type</td>
<td></td>
</tr>
<tr>
<td>CTG</td>
<td>Connection Type Group</td>
<td></td>
</tr>
<tr>
<td>LU</td>
<td>LU Name</td>
<td></td>
</tr>
<tr>
<td>LUG</td>
<td>LU Name Group</td>
<td></td>
</tr>
<tr>
<td>NET</td>
<td>Net ID</td>
<td></td>
</tr>
<tr>
<td>NETG</td>
<td>Net ID Group</td>
<td></td>
</tr>
<tr>
<td>PC</td>
<td>Process Name</td>
<td></td>
</tr>
<tr>
<td>PK</td>
<td>Package Name</td>
<td></td>
</tr>
<tr>
<td>PKG</td>
<td>Package Name Group</td>
<td></td>
</tr>
<tr>
<td>PN</td>
<td>Plan Name</td>
<td></td>
</tr>
<tr>
<td>PNG</td>
<td>Plan Name Group</td>
<td></td>
</tr>
<tr>
<td>PR</td>
<td>Procedure Name</td>
<td></td>
</tr>
<tr>
<td>PX</td>
<td>Sysplex Name</td>
<td></td>
</tr>
<tr>
<td>SI</td>
<td>Subsystem Instance</td>
<td></td>
</tr>
<tr>
<td>SIG</td>
<td>Subsystem Instance Group</td>
<td></td>
</tr>
<tr>
<td>SPM</td>
<td>Subsystem Parameter</td>
<td></td>
</tr>
<tr>
<td>SSC</td>
<td>Subsystem Collection</td>
<td></td>
</tr>
<tr>
<td>UI</td>
<td>Userid</td>
<td></td>
</tr>
<tr>
<td>UIG</td>
<td>Userid Group</td>
<td></td>
</tr>
</tbody>
</table>
More Identifiers from non-z/OS Clients

Clients can flow other identifiers to DB2 for z/OS
- Values can be set by DDF threads via “Server Connect” and “Set Client” calls
  - ODBC/CLI/VB (SQLSetConnectionAttr)
  - Non-OBDC (sqlseti)
  - JDBC (DB2Connection)
  - DRDA (ACCRDB prddta / sqlstt in EXCSQLSET)

Most important IDs supported in V8 with special registers
- Workstation Client Accounting
- Workstation Userid **WLM SPM 1-16**
- Workstation Name **WLM SPM 17-34**
- Workstation Application **WLM PC 1-32**
Setting and Passing Client Identifiers

- **java.sql.Connection.setClientInfo**
  - Sets values for client info properties
  - Previous set methods are deprecated

<table>
<thead>
<tr>
<th>Method</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>setDB2ClientUser</td>
<td>ClientUser CURRENT CLIENT_USERID</td>
</tr>
<tr>
<td>setDB2ClientWorkstation</td>
<td>ClientHostname CURRENT CLIENT_WRKSTNNAME</td>
</tr>
<tr>
<td>setDB2ClientApplicationInformation</td>
<td>ApplicationName CURRENT CLIENT_APPLNAME</td>
</tr>
<tr>
<td>setDB2ClientAccountingInformation</td>
<td>ClientAccountingInformation CURRENT CLIENT_ACCTNG</td>
</tr>
</tbody>
</table>
WAS 6.1 - `setClientInformation` API can be used to pass client registers
- API is defined on the `WSConnection` class
- Part of the `com.ibm.websphere.rsadapter` package
- Values that can be passed include
  - `WSConnection.CLIENT_ID`
  - `WSConnection.CLIENT_LOCATION`
  - `WSConnection.CLIENT_APPLICATION_NAME`
  - `WSConnection.CLIENT_ACCOUNTING_INFO`
- Client Information can be reset by calling method with a NULL parameter
- Trace Specification
  - `WAS.clientinfo=all` to pass client information to the backend without tracing of client activities.
DBATs and Accounting

- **ACTIVE mode**
  - Only cut at thread termination, not at commit

- **INACTIVE mode**
  - DRDA – at “clean” COMMIT or ROLLBACK
    - “Type 2 inactive”
  - DRDA with KEEPDYNAMIC(YES)
    - At “clean” commit (DB2 V8 and above)
  - PP DBAT – at commit or termination
    - At commit, if “Type 1 Inactive” (MAXTYPE1) allowed
    - Else only at termination

- Active thread is idle too long and is canceled
  - At “Idle Thread Timeout” (IDTHTOIN), if allowed
    - Checked every 2 minutes
Accounting and DDF Rollup

Option in DB2 V8 to reduce accounting volume
- Turned on if ZPARM ACCUMACC > 1
- Just going to V8 the ACCUMACC parm defaults to 10 which means it’s on
  - You could see a big drop in SMF 101 records

Data accumulated for specified # of threads
- For matching IDs, based on ACCUMUID
- Combination of the 3 workstation IDs
  - Total of 10 combinations in V8
  - 7 new options in DB2 9
  - Accounting for DDF and RRSAF threads......(ACCUMACC).. 10
  - Aggregate field used for DDF/RRSAF thread..(ACCUMUID).. 0

Accounting written when
- “Too old” (staleness threshold)
- “Too much” (internal storage threshold reached)
- “Just enough” (limit threshold reached)

One accounting record reflects one or more threads
- Currently no DDF statistics (QLAC) or QMDA accounting
- Only one “ROLLUP” package

Active thread data only shows the current thread counts
DBAT Thread Status

- **Assigned** to a remote client *(RA or RX)*
  - Actively processing SQL or waiting for more
  - Waiting for more work after “clean” commit, if:
    - INACTIVE mode – only:
      - **KEEPDYNAMIC(YES)** – all resources & DBAT kept
    - ACTIVE mode – even after commit
      - All resources & DBAT kept until thread termination

- **Pooled** *(DA)*
  - DRDA clients only, with INACTIVE mode
    - Freed or newly created DBATs are pooled (also called ‘DBAT slots’)
  - Available for reuse by any new / resumed request
  - Still uses resources (esp. DBM1 storage)
  - Still shown and counted as “active threads”
    - But connection name is “DISCONN”
    - Can be terminated if not used (POOLINAC)
DISPLAY THREAD
Command Output

DSNV401I  *DIA1 DISPLAY THREAD REPORT FOLLOWS -
DSNV402I  *DIA1 ACTIVE THREADS -
NAME  ST  A  REQ ID  AUTHID PLAN  ASID TOKEN
SERVER  RA * 12 db2jcc_appli RNDWDA DISTSERV 00AC 119862
V437-WORKSTATION=barlodge-AUS-73, USERID=RNDWDA,
   APPLICATION NAME=db2jcc_application
V445-AC1C4162.H120.C5BA7D75FF0E=119862 ACCESSING DATA FOR
   ::172.28.65.98

POOLED
DBAT

DSNV401I  *DIA1 DISPLAY THREAD REPORT FOLLOWS -
DSNV402I  *DIA1 ACTIVE THREADS -
NAME  ST  A  REQ ID  AUTHID PLAN  ASID TOKEN
DISCONN DA * 287 NONE  NONE DISTSERV 00AC 119863
V471-USBMCN01.DIA1LU.C5BA7D766DC9=119863

EXECUTING
DBAT
## Inactive Connections

- **Inactive connections (R2)**
  - Associated with a remote requester
  - Waiting for more work
  - Speeds up response to additional SQL

- **Previously called Inactive Type 2 DBATs**
  - Not DBATs at all

### Inactive Connection

<table>
<thead>
<tr>
<th>NAME</th>
<th>ST</th>
<th>A</th>
<th>REQ ID</th>
<th>AUTHID</th>
<th>PLAN</th>
<th>ASID</th>
<th>TOKEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>SERVER</td>
<td>R2</td>
<td>0</td>
<td>db2jcc_appli</td>
<td>RNDWDA</td>
<td>DISTSERV</td>
<td>00AC</td>
<td>119862</td>
</tr>
<tr>
<td>V437-WORKSTATION=barledge-AUS-73, USERID=RNDWDA, APPLICATION NAME=db2jcc_application</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V445-AC1C4162.H120.C5BA/D75FF0E=119862 ACCESSING DATA FOR ::172.28.65.98</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Enclave Views (MAINVIEW for z/OS)

<table>
<thead>
<tr>
<th>Enclave Token</th>
<th>Service</th>
<th>Owner</th>
<th>Cumulative Exectn</th>
<th>Total</th>
<th>%Idle</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000000000012D54</td>
<td>DDF</td>
<td>DIAIDIST</td>
<td>00:00:00.4</td>
<td>33.33</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Timeframe...</th>
<th>Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jobname......</td>
<td>ENCLAVE</td>
</tr>
<tr>
<td>Type.........</td>
<td>ENCLAVE</td>
</tr>
<tr>
<td>Serv. Class.</td>
<td>DDF</td>
</tr>
<tr>
<td>Rept. Class.</td>
<td>Enclave Cnt.</td>
</tr>
<tr>
<td>ASID.........</td>
<td>0</td>
</tr>
<tr>
<td>Dmn..........</td>
<td>0</td>
</tr>
<tr>
<td>Period No....</td>
<td>2</td>
</tr>
<tr>
<td>Workload....</td>
<td>DDF</td>
</tr>
<tr>
<td>Resource....</td>
<td>MPL Delays..</td>
</tr>
<tr>
<td>Trxn RPN...</td>
<td>0</td>
</tr>
<tr>
<td>Userid RPNG...</td>
<td>0</td>
</tr>
<tr>
<td>TrxC RPN...</td>
<td>0</td>
</tr>
<tr>
<td>Acct RPN...</td>
<td>0</td>
</tr>
<tr>
<td>Status......</td>
<td>Ended</td>
</tr>
</tbody>
</table>
Analyzing DDF Thread Data

- The accounting data is the first source

- Still analyze other application considerations
  - Elapsed and CPU times, I/O, SQL counts . . .

- But in addition:
  - Elapsed time inside / outside the DB2 server
  - Number of messages and blocks sent / received

- Batch reports summarized by
  - The important DDF identifiers for your workloads
**Thread Accounting – DBAT Specifics**

**Summary Information**

**Distributed Indicators/Identifiers**

**DBAT Accounting Metrics**
- “Application” processing is occurring in the workstation client
- Class 2 – Class 1 CPU measures time not processing SQL in DB2
  - Includes parts of thread creation and termination and moving data from and to communication buffers

**zIIP Metrics**
- Actual CPU time on the zIIP
  - Not included in “standard” class 1 & 2 CPU times
- CPU eligible for the zIIP

---

**Runtime Analysis**

<table>
<thead>
<tr>
<th></th>
<th>IN DB2</th>
<th>IN APPL.</th>
<th>TOTAL</th>
<th>% IN DB2(=)</th>
<th>TOTAL(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ELAPSED TIME</strong></td>
<td>807 ms</td>
<td>2,798 ms</td>
<td>3,605 ms</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CPU TIME</strong></td>
<td>83 ms</td>
<td>1,099 us</td>
<td>84 ms</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DB2 WAIT TIME</strong></td>
<td>620 ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ZIIP CPU TIME</strong></td>
<td>0 us</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ZIIP-ELIGIBLE CP</strong></td>
<td>46 ms</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Thread Accounting – DBAT Specifics

## DDF Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CURRENT LOCATION</td>
<td>DIA (DBAT SERVER)</td>
</tr>
<tr>
<td>WORKSTATION USER ID</td>
<td>AFDQA User</td>
</tr>
<tr>
<td>WORKSTATION NAME</td>
<td>AFDQA Workstation</td>
</tr>
<tr>
<td>WORKSTATION TRANSACTION ID</td>
<td>AFDQA TEST appdemo</td>
</tr>
<tr>
<td>MVS WLM SERVICE CLASS NAME</td>
<td>DDF</td>
</tr>
<tr>
<td>Client</td>
<td>DB2 Universal JDBC Driver. / Vers. 02 Rel. 10 Mod. 0</td>
</tr>
<tr>
<td>Client Platform</td>
<td>AFDQA Workstation</td>
</tr>
<tr>
<td>Job accounting data</td>
<td>AFDQA CLIENT ACCOUNTING01234567891011121314151617181920212223242526272829303132333435363738394041424344454647484950515253545</td>
</tr>
</tbody>
</table>

### Remote

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCATION</td>
<td>:192.168.7.5</td>
</tr>
<tr>
<td>SQL STATEMENTS</td>
<td>8</td>
</tr>
<tr>
<td>WAIT ON MAX DBAT</td>
<td>0 us</td>
</tr>
<tr>
<td>REMOTE PROD-ID</td>
<td>JCC02100</td>
</tr>
<tr>
<td>DB AGENT CPU</td>
<td>n/a</td>
</tr>
<tr>
<td>ROWS</td>
<td>n/a</td>
</tr>
<tr>
<td>Bytes</td>
<td>1,771</td>
</tr>
<tr>
<td>Elapsed Local</td>
<td>0 us</td>
</tr>
<tr>
<td>Transactions</td>
<td>1</td>
</tr>
<tr>
<td>Elapsed Remote</td>
<td>n/a</td>
</tr>
<tr>
<td>Messages</td>
<td>12</td>
</tr>
<tr>
<td>Conversations Queued</td>
<td>n/a</td>
</tr>
<tr>
<td>Conversations</td>
<td>1</td>
</tr>
<tr>
<td>Block Fetch %</td>
<td>100%</td>
</tr>
<tr>
<td>Conmits</td>
<td>0</td>
</tr>
<tr>
<td>Max Conversations</td>
<td>n/a</td>
</tr>
<tr>
<td>Aborts</td>
<td>0</td>
</tr>
<tr>
<td>Unsuccessful Conv’s</td>
<td>n/a</td>
</tr>
</tbody>
</table>

### DBAT Details

- Wait on MAX DBATs
- PROD-ID = IBM Universal Driver for JDBC and SQLJ
- Traffic Information

---

Client Identifiers Available in DB2 V8

WLM Service Class
Tracing Distributed Workloads

- Additional focus on one workload
  - Summary exception trace (accounting)
  - Detail trace with important event IFCIDs

- All the usual qualifiers are available

- For DDF, important to reduce the data:
  - Filter by requesting location
  - Filter by Workstation ID(s)
    - In V9, DB2 also allows qualification by these IDs

- Exception Filters can be used to keep only threads that may need analysis
  (high In-DB2 elapsed, etc.)
Detail Traces

- Detail traces can include selected event groups
  - Basic thread flow and SQL
  - Also can choose to add scans, I/O, locks

- Another group to include specific DDF events
  - The volume can be high
  - Use it only when needed
  - To understand the conversation flow
  - Includes IFCIDs 157 – 163
  - IFCID 163, for example, traces events like
    - DBAT creation
    - Commit request from coordinator
    - Backout request from coordinator
    - DBAT creation queued
    - Deallocation initiated

- Each event has a pop-up view with the IFCID details
The next place to look are the statistics

Global statistics
- Critical DB2 subsystem tuning information

Location statistics
- Application impact on DB2 and network
  - DRDA_Remote_Locs (combined)
  - Private Protocol locations (separate)

DDF Address Space CPU usage
- TCB and SRB
### Global DDF Statistics - STDISTD

<table>
<thead>
<tr>
<th>Status - Current and High Water Mark</th>
<th>Current</th>
<th>HWM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total DBATs - Active &amp; Pooled</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>DBATs Pooled for Reuse (Type 2)</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Inactive DBATs (Type 1)</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Total Remote Connections</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>Type 2 Inactive Connections</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Type 2 Connections Queued for DBAT</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Maximums Reached</th>
<th>Interval</th>
<th>Session</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queued for DBAT (MAXDBAT Reached)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Connections Deallocated (CONDBAT Reached)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Type 1 Connections Terminated (MAXTYPE1 Reached)</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DBAT Usage Statistics</th>
<th>Interval</th>
<th>Session</th>
</tr>
</thead>
<tbody>
<tr>
<td>New DBATs Created</td>
<td>0</td>
<td>18</td>
</tr>
<tr>
<td>Pooled DBATs Reused</td>
<td>256</td>
<td>5065</td>
</tr>
<tr>
<td>New/Resumed (Type 2) Requests</td>
<td>128</td>
<td>2533</td>
</tr>
</tbody>
</table>
Extended Reporting

- Distributed workloads are often volatile
  - Less insight and control
- Can be useful to track activity over time
  - Store and query summary data in DB2 tables
- When needed, distributed traces and monitoring
- z/OS reporting on WLM can be helpful
  - Enclaves – SMF 30
  - Workloads by service class – SMF 72
Summary

**SQL is SQL, but:**

- Distributed Workloads are increasing at most DB2 shops
- DDF processing has a unique set of characteristics that differ from traditional DB2 applications
- New technologies (for DBAs) like WLM
Questions?
BUSINESS RUNS ON I.T.
I.T. RUNS ON BMC™