Programming IMS Open Database for Optimal Performance

Kevin Hite
IBM STSM – IMS architect

Virtual IMS user group meeting

February 5, 2019

Please note

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.

The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

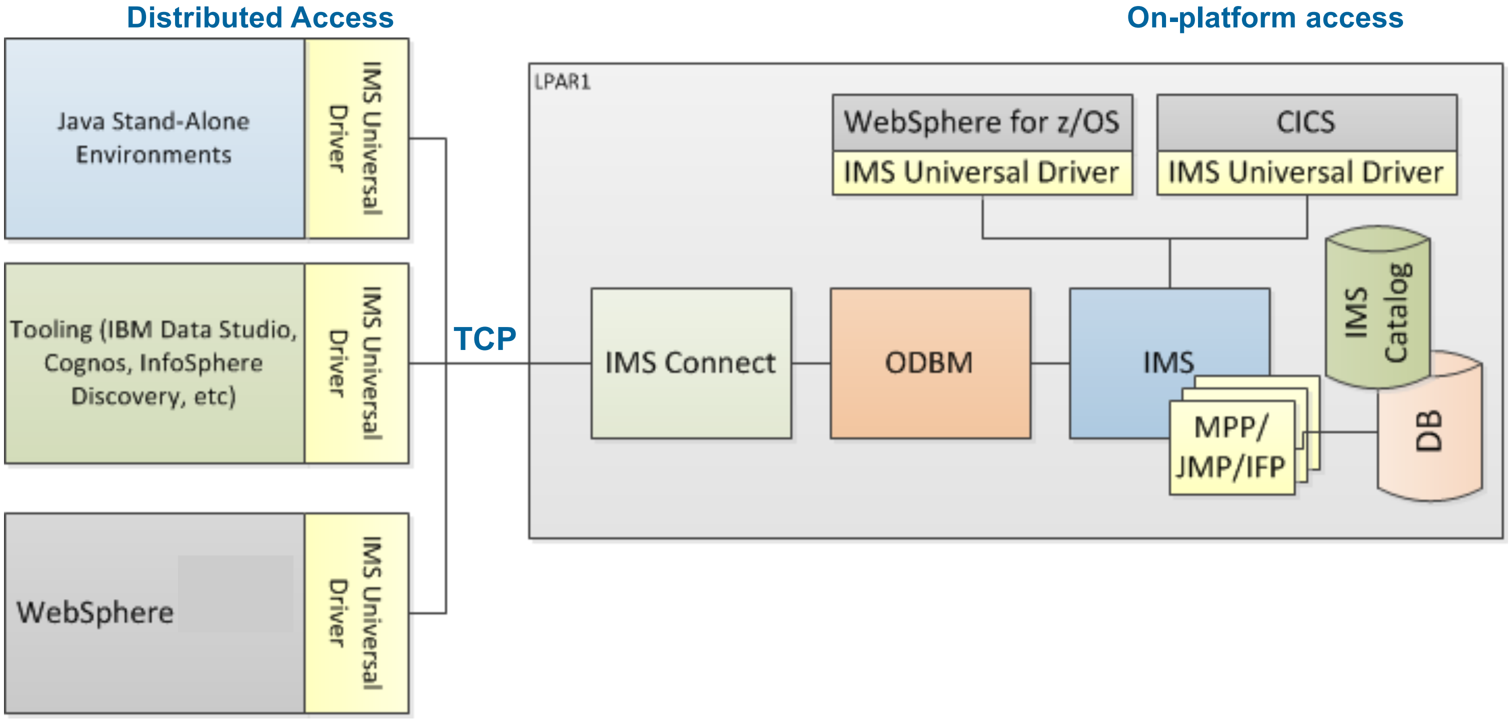
Agenda

- **IMS Java solution overview**
- Performance topics
  - Access to IMS DB off platform
  - Java enabled IMS dependent region
  - Performance at the lab
- Resources

IMS Java solutions overview

IMS Java solutions overview

– Java application developers can access IMS assets from both
  - Java running on distributed environments over TCP/IP
  - Java running on z/OS in IMS or other middleware such as CICS or WebSphere Application Server



IMS Universal Database Drivers & Resource Adapters

IMS the IMS Universal Database Drivers (imsudb.jar) include:

- JDBC interface
- DL/I for Java API - IMS DB support
- DL/I for Java API - IMS TM support

IMS ships 4 IMS Universal Database Resource Adapters

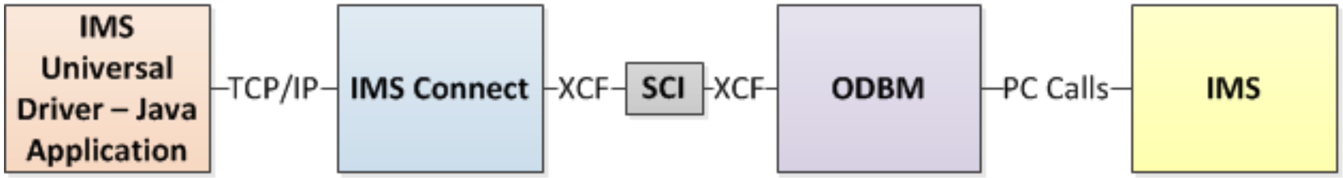
1. **imsudbJLocal.rar - JDBC with single phase commit (recommended)**
2. **imsudbJXA.rar - JDBC with two phase commit (recommended)**
3. imsudbLocal.rar - CCI with single phase commit
4. imsudbXA.rar - CCI with two phase commit

All parts shipped as part of the IMS Java

- FMID JMKxx06 where xx is the IMS release level
- no additional charge on top of the IMS base

Access to IMS DB off platform, performance topics

Distributed access to IMS database – components



- Distributed access over TCP/IP to IMS data (via IMS Connect and ODBM)
  - IMS Universal Database Driver/Resource Adapter
  - IMS Connect
  - ODBM
  - SCI
  - IMS

Performance considerations: Distributed access to IMS database

- Applications
  - SQL considerations
  - JDBC considerations
- Distributed Server
  - Thread pool settings
  - Connection pool settings
    - Number of connections
    - Connection time limits
- IMS Connect
  - Timeout settings frontend (client)
  - Timeout settings backend (IMS)
  - ODBM routing

- Open Database Manager (ODBM)
  - Max Threads
  - Fast Path buffer tuning
  - RRS=Y or RRS=N
- IMS
  - Pool tuning (PSB, PSBW, DMB)
  - MAXPST
  - PCB processing options (PROCOPT)

Performance considerations: DRDA

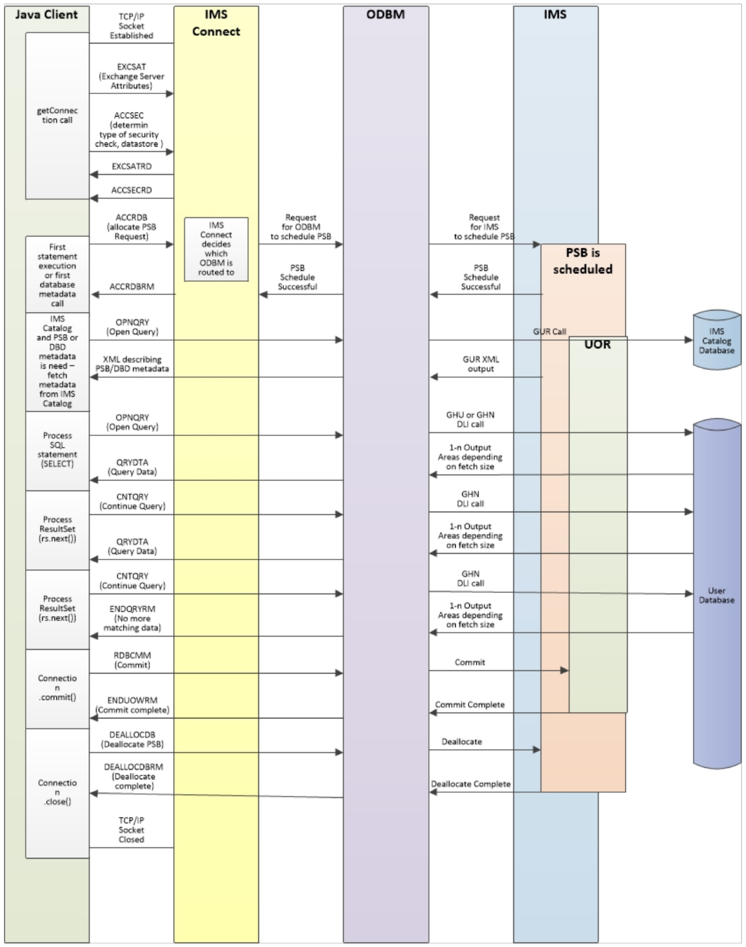
- Distributed Relation Database Architecture (DRDA)
  - Used to define the TCP/IP protocol for accessing IMS data as well as
    - Authentication/authorization
    - IMS scheduling of PSB's for data access
    - Database insert, update, delete, and retrieve requests
    - Commit/rollback

- DRDA monitoring tools
  - IMS Connect Extensions for logging DRDA activity through IMS Connect
    - Captures activity between TCP/IP client ←→IMS Connect and IMS Connect ←→ ODBM
  - IMS Universal Driver traces for analyzing DRDA activity during application development

Performance considerations: DRDA

- IMS Universal Database Driver
  - Review DRDA flows
    - While developing new transactions/services enable the IMS Universal Driver's DRDA tracing to see if the flows look to be optimized.
    - Enabling IMS Universal Database Driver trace: http://www-01.ibm.com/support/knowledgecenter/#!/SSEPH2_13.1.0/com.ibm.ims13.doc.apg/ims_odbdli4jtracing.htm
  - DRDA trace can be used to review
    - TCP/IP socket open/closed
    - PSB allocation/de-allocation
    - Type of database interactions
    - SSA Lists being used
    - How much data is fetched
    - How often data is fetched
    - Commit processing

Performance considerations: DRDA deeper dive


Java Client
IMS Connect
ODBM
IMS
getConnection call
TCP/IP Socket Established
EXCSAT (Exchange Server Attributes)
ACCSEC (determin type of security check, datastore)
EXCSATRD
ACCSECRD
ACCRDB (allocate PSB Request)
IMS Connect decides which ODBM is routed to
Request for ODBM to schedule PSB
Request for IMS to schedule PSB
PSB is scheduled
First statement execution or first database metadata call
ACCRDBRM
PSB Schedule Successful
PSB Schedule Successful
IMS Catalog and PSB or DBD metadata is need – fetch metadata from IMS Catalog
OPNQRY (Open Query)
GUR Call
IMS Catalog Database
UOR
XML describing PSB/DBD metadata
GUR XML output
Process SQL statement (SELECT)
OPNQRY (Open Query)
GHU or GHN DLI call
QRYDTA (Query Data)
1-n Output Areas depending on fetch size
Process ResultSet (rs.next())
CNTQRY (Continue Query)
GHN DLI call
QRYDTA (Query Data)
1-n Output Areas depending on fetch size
Process ResultSet (rs.next())
CNTQRY (Continue Query)
GHN DLI call
User Database
ENDQRYRM (No more matching data)
1-n Output Areas depending on fetch size
Connection .commit()
RDBCMM (Commit)
Commit
ENDUOWRM (Commit complete)
Commit Complete
Connection .close()
DEALLOCDB (Deallocate PSB)
Deallocate
DEALLOCDBRM (Deallocate complete)
Deallocate Complete
TCP/IP Socket Closed

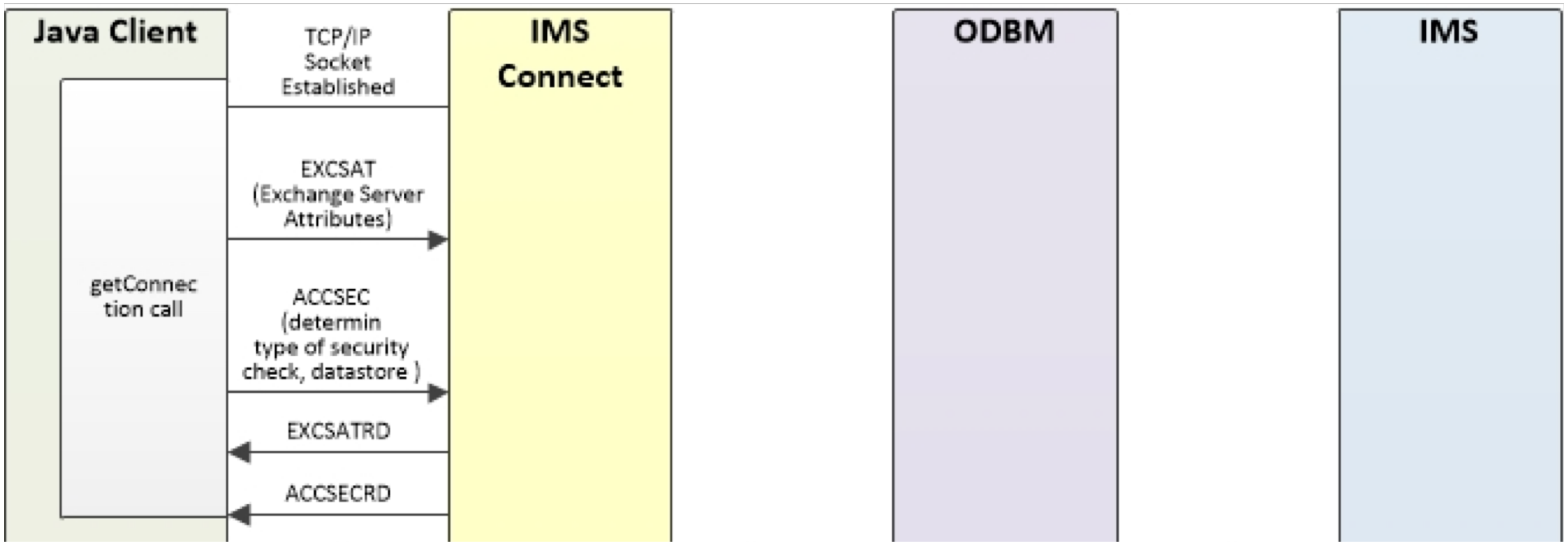
Performance considerations: DRDA deeper dive

- *Network time* - Time spent writing/reading the bytes to/from IMS Connect
- *Server time* - Time spent waiting on a response from IMS Connect

```
[ibm][ims][drda][t4][thread:35] [SystemMonitor:stop]
          network: 3.571619ms | server: 0.39184599999999997ms
```

- **Tools that help record DRDA flows (and more) in IMS Connect**
  - IMS Connect Extensions – capture/archive events (DRDA), ODBM routing exits, analyze active sessions
  - IMS Problem Investigator – merged view of IMS Connect events (DRDA), IMS logs, etc
  - I have also seen BMC Energizer being used for this as well

Performance considerations: DRDA deeper dive


Java Client
IMS Connect
ODBM
IMS
getConnection call
TCP/IP Socket Established
EXCSAT (Exchange Server Attributes)
ACCSEC (determin type of security check, datastore)
EXCSATRD
ACCSECRD

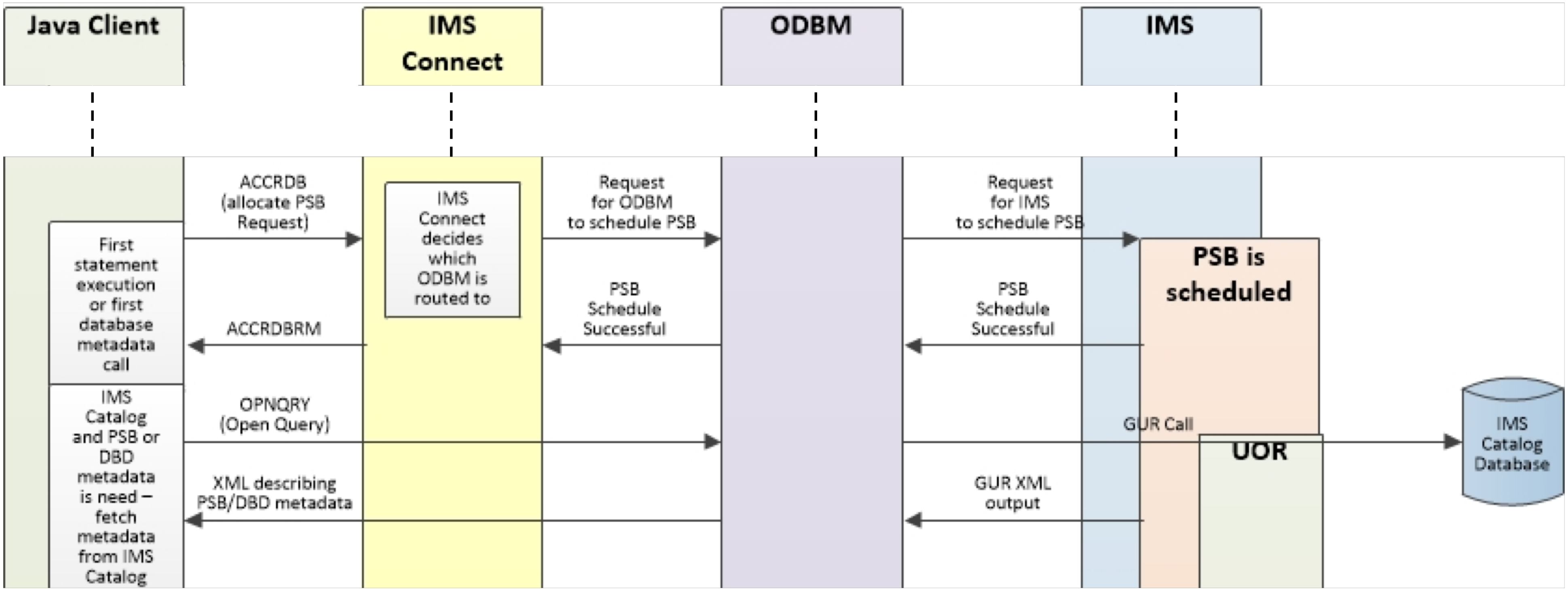
DRDA flows: Exchange Server Attributes & Access Security Check

```
        SEND BUFFER: EXCSAT                 (ASCII)           (EBCDIC)
        0 1 2 3 4 5 6 7   8 9 A B C D E F   0123456789ABCDEF  0123456789ABCDEF
0000    004DD04100010047  10410006115EF4F5  .M.A...G.A...^..  .(}..........;45
0010    0013116DC9C2D4F2  F5F660D9F9F0C7E6  ...m......`.....  ..._IBM256-R90GW
0020    C3F4C10023115AD6  C460C9C3D6D540F1  ....#.Z..`....@.  C4A...!OD-ICON 1
0030    40D6C460D6C4C2D4  40F16BF26BF36BF4  @..`....@.k.k.k.   OD-ODBM 1,2,3,4
0040    6BF56BF66BF70007  1147C4C6E2        k.k.k....G...     ,5,6,7....DFS

        SEND BUFFER: ACCSEC                 (ASCII)           (EBCDIC)
0000    0019D00100020013  106D000611A20001  .........m......  ..}......_...s..
0010    000921104BC9D4E2  F1                ..!.K....         .....IMS1

        RECEIVE BUFFER: EXCSATRD            (ASCII)           (EBCDIC)
        0 1 2 3 4 5 6 7   8 9 A B C D E F   0123456789ABCDEF  0123456789ABCDEF
0000    0052D0430001004C  14430013116DC9C2  .R.C...L.C...m..  ..}....<....._IB
0010    D4F2F5F660D9F9F0  C7E6C3F4C1000611  ....`...........  M256-R90GWC4A...
0020    5EF4F50028115AD6  C460C9C3D6D540F1  ^...(.Z..`....@.  ;45...!OD-ICON 1
0030    40D6C460D6C4C2D4  40F16BF26BF36BF4  @..`....@.k.k.k.   OD-ODBM 1,2,3,4
0040    6BF56BF66BF740C9  D4E2F100071147C4  k.k.k.@.......G.  ,5,6,7 IMS1....D
0050    C6E2                                ..                FS

        RECEIVE BUFFER: ACCSECRD            (ASCII)           (EBCDIC)
0000    0010D0030002000A  14AC000611A20003  ................  ..}..........s..
```

DRDA flows: Security Check

```
        SEND BUFFER: SECCHK                     (ASCII)           (EBCDIC)
        0 1 2 3 4 5 6 7   8 9 A B C D E F   0123456789ABCDEF  0123456789ABCDEF
0000    0030D0010001002A  106E000611A20005  .0......*.n......  ..}......>...s..
0010    000921104BC9D4E2  F1000B11A09694A5  ..!.K...........  .....IMS1....omv
0020    A2818494000C11A1  5C5C5C5C5C5C5C5C  ........\\\\\\\\ sadm...~********

        RECEIVE BUFFER: SECCHKRM                (ASCII)           (EBCDIC)
        0 1 2 3 4 5 6 7   8 9 A B C D E F   0123456789ABCDEF  0123456789ABCDEF
0000    0015D0020001000F  1219000611490000  .............I..  ..}.............
0010    000511A400                          .....             ...u.
```

Performance considerations: DRDA deeper dive


Java Client
IMS Connect
ODBM
IMS
ACCRDB (allocate PSB Request)
IMS Connect decides which ODBM is routed to
Request for ODBM to schedule PSB
Request for IMS to schedule PSB
PSB is scheduled
First statement execution or first database metadata call
ACCRDBRM
PSB Schedule Successful
PSB Schedule Successful
IMS Catalog and PSB or DBD metadata is need – fetch metadata from IMS Catalog
OPNQRY (Open Query)
GUR Call
UOR
IMS Catalog Database
XML describing PSB/DBD metadata
GUR XML output

DRDA flows: Access Relation Database

- Allocate the PSB in IMS, in this example we are requesting allocation of the PSB named **BMP255**

```
        SEND BUFFER: ACCRDB                       (ASCII)           (EBCDIC)
        0 1 2 3 4 5 6 7   8 9 A B C D E F   0123456789ABCDEF  0123456789ABCDEF
0000    0040D0010001003A  2001000F2110C2D4  .@.....: ...!...  . }...........BM
0010    D7F2F5F54BC9D4E2  F10006210F240700  ....K......!.$..  P255.IMS1.......
0020    14112EC9D4E240D6  D7C5D540C4C240E5  ......@....@..@.  ...IMS OPEN DB V
0030    F14BF0000D002FD8  E3C4E2D8D3F3F7F0  .K..../.........  1.0....QTDSQL370

        RECEIVE BUFFER: ACCRDBRM                  (ASCII)           (EBCDIC)
        0 1 2 3 4 5 6 7   8 9 A B C D E F   0123456789ABCDEF  0123456789ABCDEF
0000    0039D00200010033  22010008112E0001  .9.....3".......  ..}.............
0010    0500000611490000  001421350CFDC9A8  .....I....!5....  ..............Iy
0020    0D003050233A4948  BE26A040000D002F  ..0P#:IH.&.@.../  ...&....... ....
0030    D8E3C4E2D8D3F3F7  F0                .........         QTDSQL370
```

DRDA flows: Open Query

- Issue a **GUR** DL/I call against PCB **DFSCAT00** for SSA List **`HEADER   (RHDRSEQ EQPSB      BMP255   )`**
- Only issued if the IMS catalog is enabled
- Only issued the first time the JVM accesses a given PSB or DBD resource

```
       SEND BUFFER: OPNQRY                (ASCII)           (EBCDIC)
       0 1 2 3 4 5 6 7   8 9 A B C D E F  0123456789ABCDEF  0123456789ABCDEF
0000   002CD05100010026  200C000621410013 .,.Q...& ...!A..  ..}.............
0010   000CC907C4C6E2C3  C1E3F0F000082114 ..............!.  ..I.DFSCAT00....
0020   0000800000082156  00000014         ......!V....      ............

       SEND BUFFER: DLIFUNC               (ASCII)           (EBCDIC)
0000   000DD05300010007  CC05C7E4D9       ...S........      ..}......GUR

       SEND BUFFER: AIB                   (ASCII)           (EBCDIC)
0000   001ED05300010018  CC01000CC901C4C6 ...S............  ..}.........I.DF
0010   E2C3C1E3F0F00008  C9040000C350     .............P    SCAT00..I...C&

       SEND BUFFER: SSALIST               (ASCII)           (EBCDIC)
0000   0044D0030001003E  CC060006C9050002 .D.....>........  ..}.........I...
0010   000CC906C4C6E2D9  E34040400028C906 .........@@@.(..  ..I.DFSRT   ..I.
0020   C8C5C1C4C5D94040  4DD9C8C4D9E2C5D8 ......@@M.......  HEADER  (RHDRSEQ
0030   40C5D8D7E2C24040  404040C2D4D7F2F5 @.....@@@@@.....   EQPSB     BMP25
0040   F540405D                           .@@]              5  )
```

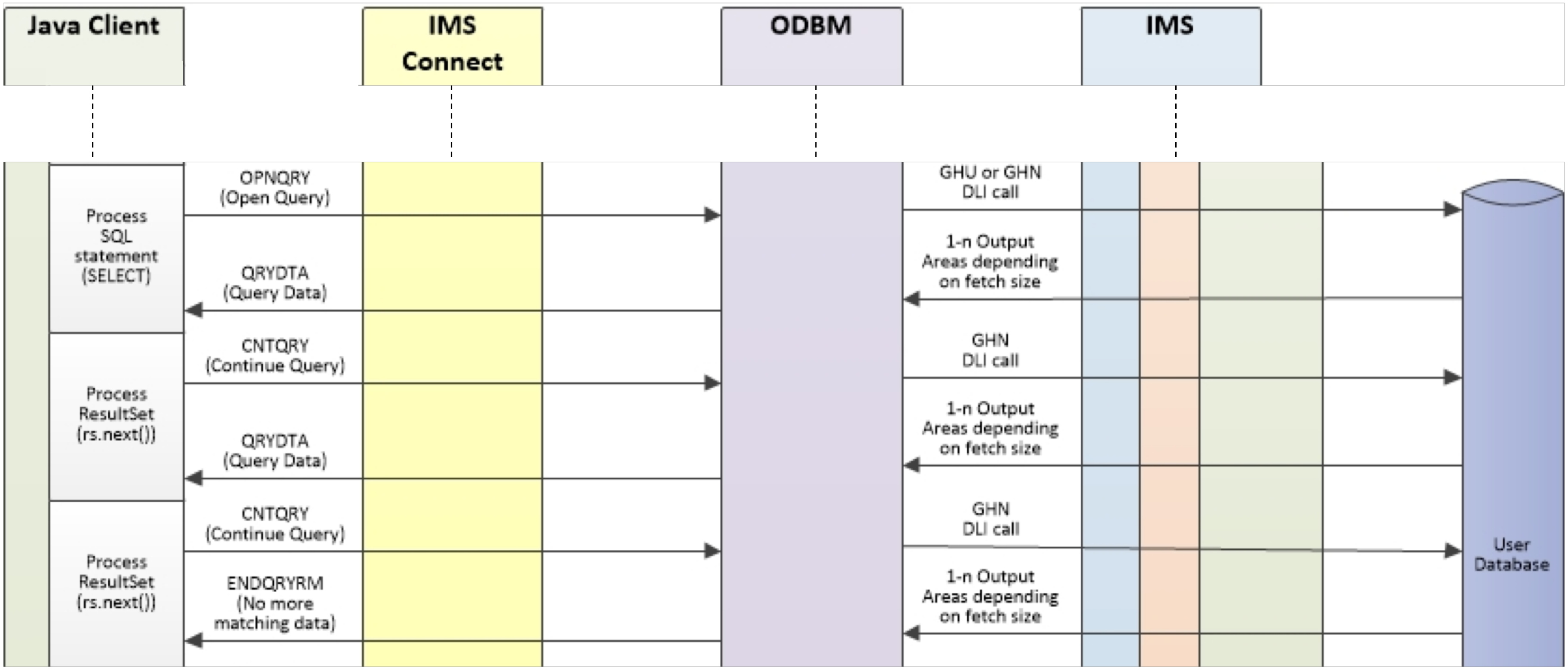
DRDA flows: Open Query

```
         RECEIVE BUFFER: OPNQRYRM              (ASCII)           (EBCDIC)
         0 1 2 3 4 5 6 7   8 9 A B C D E F     0123456789ABCDEF  0123456789ABCDEF
0000     0010D0520001000A  2205000611490000    ...R...."....I..  ..}.............

         RECEIVE BUFFER: QRYDSC                (ASCII)           (EBCDIC)
0000     001CD05300010016  241A0676D0260000    ...S....$..v.&..  ..}.........}...
0010     0671E0D000010671  F0E00000            .q.....q....      ..\}....0\..

         RECEIVE BUFFER: QRYDTA                (ASCII)           (EBCDIC)
0000     2C5FD00300012C59  241B0000002C2D00    ,_....,Y$....,-.  ..}.............
0010     0000000000000000  000000000C4C6E2     ................  .............DFS
0020     C3C4F0F0F0000040  4040404040404040    .......@@@@@@@@@  CD000..
0030     40FF3C3F786D6C20  76657273696F6E3D    @.<?xml version=   ...._%......?>.
0040     22312E302220656E  636F64696E673D22    "1.0" encoding="  .......>.?..>...
0050     4370313034372220  7374616E64616C6F    Cp1047" standalo  ........../>./%?
0060     6E653D2279657322  3F3E4C95A2F27A97    ne="yes"?>L...z.  >...`.....<ns2:p
0070     A28240A7949395A2  7A95A2F27E7F88A3    ..@.....z...~...  sb xmlns:ns2="ht
0080     A3977A6161A6A6A6  4B8982944B839694    ..zaa...K...K...  tp://www.ibm.com
0090     618994A261D7E2C2  7F4097A282D58194    a...a....@......  /ims/PSB" psbNam
00A0     857E7FC2D4D7F2F5  F57F40A3899485A2    .~........@.....  e="BMP255" times
00B0     A38194977E7FF1F7  F2F7F5F1F1F0F9F1    ....~...........  tamp="1727511091
00C0     F9F5F57F40A79493  E28388859481E585    ....@...........  955" xmlSchemaVe
……<more data>……
0490     82D7C3C26E4C6195  A2F27A97A2826E      ....nLa...z...n   bPCB></ns2:psb>
```

An XML document describing the meta data for IMS PSB BMP255 is returned.

The XML is built from the IMS catalog to describe IMS DBD and PSB resources.

Performance considerations: DRDA deeper dive

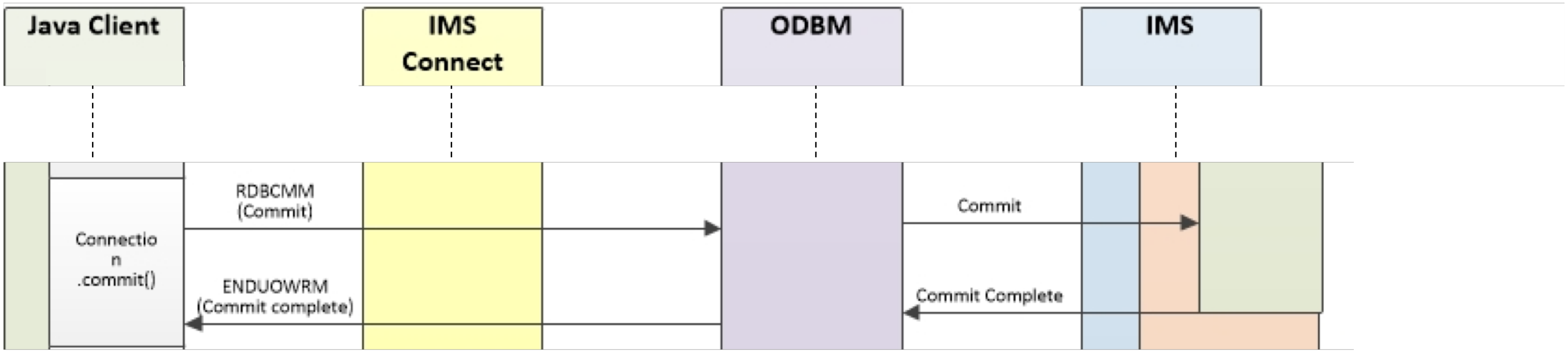

Java Client
IMS Connect
ODBM
IMS
Process SQL statement (SELECT)
OPNQRY (Open Query)
QRYDTA (Query Data)
Process ResultSet (rs.next())
CNTQRY (Continue Query)
QRYDTA (Query Data)
Process ResultSet (rs.next())
CNTQRY (Continue Query)
ENDQRYRM (No more matching data)
GHU or GHN DLI call
1-n Output Areas depending on fetch size
GHN DLI call
1-n Output Areas depending on fetch size
GHN DLI call
1-n Output Areas depending on fetch size
User Database

DRDA flows: Open Query

- Batch **retrieve** – issue a series of GU/GN calls against the IMS DB
- The DLI calls are issued against the DB PCB named **PCB01** using SSA List:
  HOSPITAL
  WARD   (WARDNO LT0004+WARDNO GE0010)

```
         SEND BUFFER: OPNQRY                   (ASCII)            (EBCDIC)
         0 1 2 3 4 5 6 7  8 9 A B C D E F     0123456789ABCDEF   0123456789ABCDEF
0000     0029D05100010023  200C00062141001F   .).Q...# ...!A..   ..}.............
0010     0009C907D7C3C2F0  F100082114000080   ...........!....   ..I.PCB01.......
0020     0000082156000003  69                 ...!V...i          .........
         SEND BUFFER: DLIFUNC                  (ASCII)            (EBCDIC)
0000     0014D0530001000E  CC05D9C5E3D9C9C5   ...S............   ..}.......RETRIE
0010     E5C5D5C8                             ....               VENH
         SEND BUFFER: AIB                      (ASCII)            (EBCDIC)
0000     001BD05300010015  CC010009C901D7C3   ...S............   ..}.........I.PC
0010     C2F0F10008C90400  000384             ...........        B01..I....d
         SEND BUFFER: RTRVFLD                  (ASCII)            (EBCDIC)
0000     0012D0530001000C  CC04000000020000   ...S............   ..}.............
0010     0013                                 ..                 ..
         SEND BUFFER: SSALIST                  (ASCII)            (EBCDIC)
0000     0048D00300010042  CC060006C9050002   .H.....B........   ..}.........I...
0010     000DC906C8D6E2D7  C9E3C1D340002BC9   ............@.+.   ..I.HOSPITAL ..I
0020     06E6C1D9C4404040  404DE6C1D9C4D5D6   .....@@@@M......   .WARD     (WARDNO
0030     4040D3E3F0F0F0F4  4EE6C1D9C4D5D640   @@......N......@    LT0004+WARDNO
0040     40C7C5F0F0F1F05D                     @......]            GE0010)
```

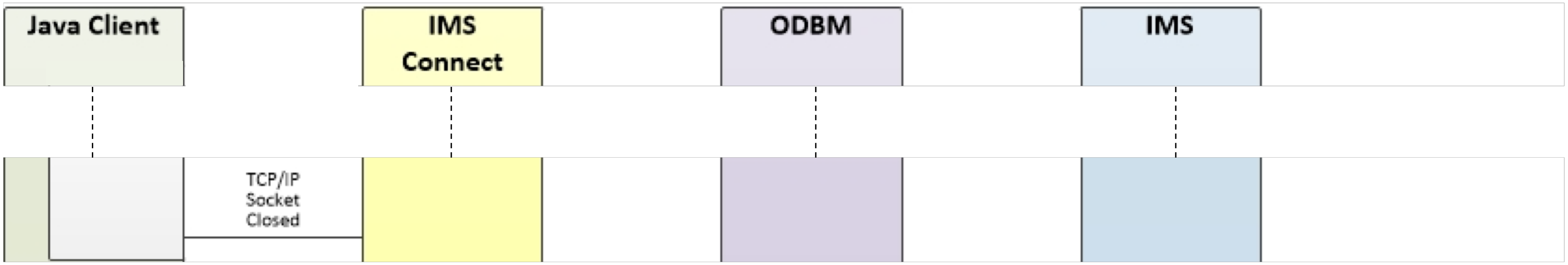
Performance considerations: DRDA deeper dive



DRDA flows: Commit

```
        SEND BUFFER: RDBCMM                          (ASCII)           (EBCDIC)
        0 1 2 3 4 5 6 7   8 9 A B C D E F    0123456789ABCDEF  0123456789ABCDEF
0000    000AD00100010004  200E               ........ .        ..}.......

        RECEIVE BUFFER: ENDUOWRM                     (ASCII)           (EBCDIC)
        0 1 2 3 4 5 6 7   8 9 A B C D E F    0123456789ABCDEF  0123456789ABCDEF
0000    002BD00200010025  220C000521150100   .+.....%"...!...  ..}.............
0010    061149000400016CC 0200000000000000   ..I.............  ................
0020    0000000000000000  0000FF             ...........       ...........
```

Performance considerations: DRDA deeper dive



```
        SEND BUFFER: DEALLOCDB                   (ASCII)           (EBCDIC)
        0 1 2 3 4 5 6 7   8 9 A B C D E F   0123456789ABCDEF  0123456789ABCDEF
0000    0014D0010001000E  C801000A2110C2D4  ............!...  ..}.....H.....BM
0010    D7F2F5F5                            ....              P255

        RECEIVE BUFFER: DEALLOCDBRM              (ASCII)           (EBCDIC)
        0 1 2 3 4 5 6 7   8 9 A B C D E F   0123456789ABCDEF  0123456789ABCDEF
0000    0032D0020001002C  CA01000C2110C2D4  .2.....,....!...  ..}...........BM
0010    D7F2F5F540400006  114900000016CC02  ....@@...I......  P255  ..........
0020    0000000000000000  0000000000000000  ................  ................
0030    00FF                                ..                ..
```

Performance considerations: IMS JDBC & SQL

- IMS Universal Database Driver
  - Query Tuning
    - For SQL calls - use IMS Explorer for Development to view what DL/I calls and SSA Lists are generated to understand what calls will be executed on the backend

```
SQL -
SELECT * FROM PCB01.HOSPITAL, PCB01.WARD WHERE PCB01.HOSPCODE = ‘xxxx’
DL/I -
     GHU HOSPITAL*D(HOSPCODE= xxxx)
         WARD
LOOP GHN HOSPITAL*D(HOSPCODE= xxxx)
         WARD
```

    - Use unique keys or indexes in the WHERE clause
    - Key all the way up to the root segment when possible
  - Fetch Size
    - If your SELECT query is designed to return a fixed number of results then set fetch size to that fixed number. This will prevent additional DL/I calls from being processed and additional data from being transferred over the network.
    - If you need your SQL to return more than one record but are unsure of the total number of results set fetch size to 0. This will help to limit the number of network trips needed to fetch all of the data.
  - Set statement concurrency appropriately
    - CONCUR_READ_ONLY will avoid IMS HOLD calls which can help reduce locks

Performance considerations: IMS Connect

- IMS Connect
  - ODBM routing
    - If you are in a data sharing environment with multiple ODBMs and IMSs across multiple LPARs consider how work is routed from IMS Connect to ODBM to IMS
    - If you would like to route to a specific IMS then use a specific/unique ODBM alias/datastoreName
  - Timeouts
    - PORTTMOT – port timeout to prevent PSBs from staying scheduled if the client remains idle for a period of time.
    - ODBMTMOT – ODBM timeout used to timeout long running request that do not respond within the timeout value.
    - These will close the socket connection and de-allocate the PSB associated with this connection. Any uncommitted work for the PSB is rolled back. If uncommitted work was in-flight, a U0210 ABEND will be issued.

Performance considerations: ODBM

- ODBM
  - Max Threads
    - Use MAXTHRDS= to throttle the maximum amount of concurrent PSBs schedules for a given IMS – see additional IMS
  - If using Fast Path, tune CNBA, FPBUF, FPBOF
    CNBA >= (MAXTHRDS * FPBUF) + FPBOF
  - If XA/2 phase commit/global transactions are not required then use RRS=N, less overhead

Performance considerations: IMS

- IMS
  - –Pool sizes (PSB, PSBW, DMB) – tune appropriately for added PSB scheduling activity
  - –MAXPST – consider the current maximum number of PSTs you allow for an IMS.
  - –PCB processing options (PROCOPT) - read only/dirty read/update, joins

Performance considerations: Java EE server

- WebSphere Application Server
  - Tune WAS thread pools to throttle the number of concurrent PSB allocation requests sent to the backend at a given time
  - Tune timeouts for the WAS transactions
  - Tune IMS JDBC connection pool settings
    - Timeouts
    - Min/max connections

Java enabled IMS dependent region, performance topics

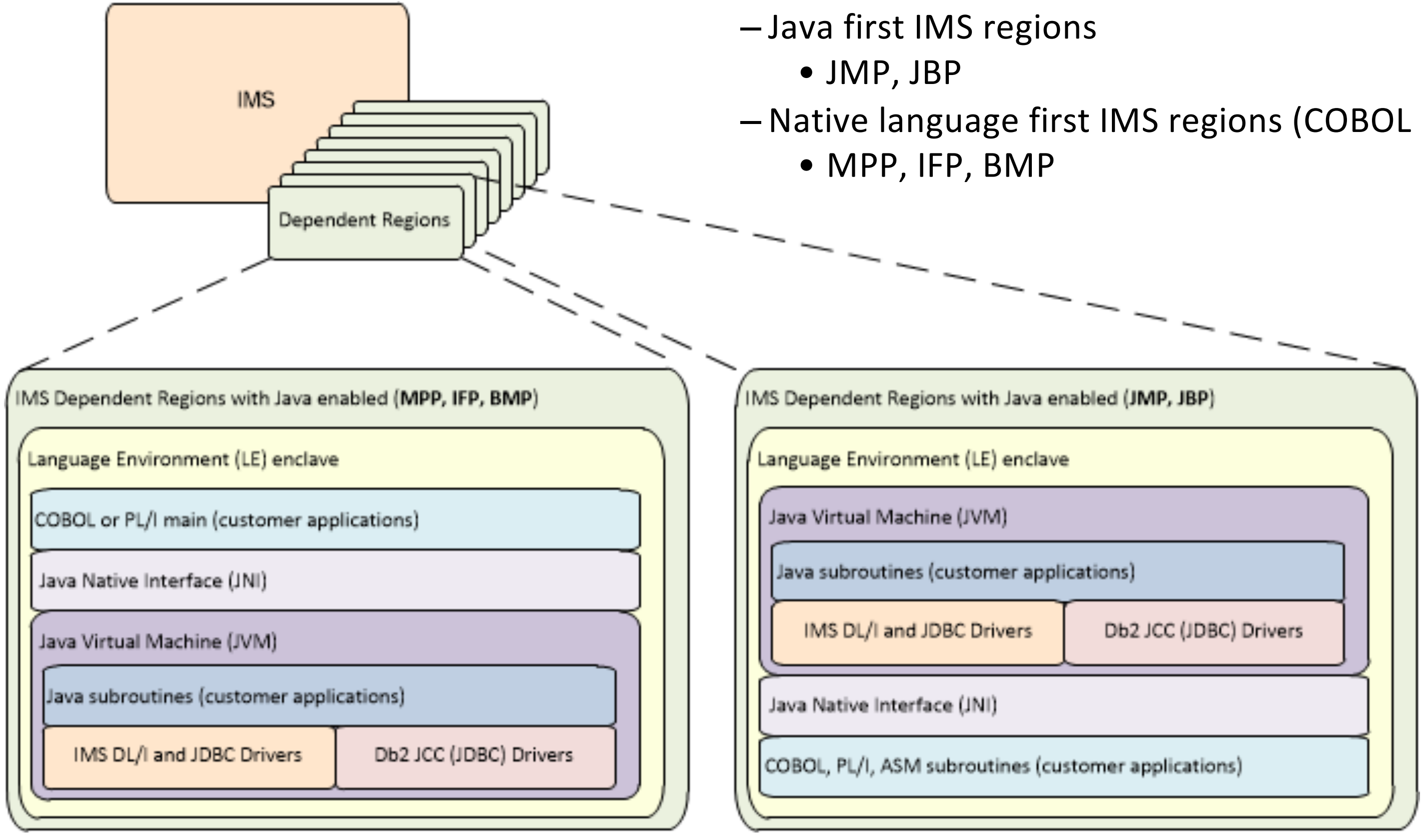
Java enabled IMS dependent regions

- Java application first regions
  - Java Batch Processing regions or **JBP regions** are used for online batch processing
  - Java Message Processing regions or **JMP regions** are used for online message/transaction processing

- Native application first regions (COBOL, PL/I)
  - Message Processing Regions or **MPP regions**
  - Fast Path Regions or **IFP regions**
  - Batch Message Processing Regions or **BMP regions**

Java enabled IMS dependent regions

- Running Java on z/OS from **Java enabled IMS dependent regions**
  - Java first IMS regions
    - JMP, JBP
  - Native language first IMS regions (COBOL or PL/I main)
    - MPP, IFP, BMP


IMS
Dependent Regions
IMS Dependent Regions with Java enabled (**MPP, IFP, BMP**)
Language Environment (LE) enclave
COBOL or PL/I main (customer applications)
Java Native Interface (JNI)
Java Virtual Machine (JVM)
Java subroutines (customer applications)
IMS DL/I and JDBC Drivers
Db2 JCC (JDBC) Drivers
IMS Dependent Regions with Java enabled (**JMP, JBP**)
Language Environment (LE) enclave
Java Virtual Machine (JVM)
Java subroutines (customer applications)
IMS DL/I and JDBC Drivers
Db2 JCC (JDBC) Drivers
Java Native Interface (JNI)
COBOL, PL/I, ASM subroutines (customer applications)

Performance considerations: Java in IMS

- Applications
  - IMS JDBC & SQL considerations
  - Java Native Interface (JNI) considerations
  - DB2 access from IMS Java environments
- JVM tuning
- Language Environment (LE) tuning

Performance considerations: IMS JDBC & SQL

- IMS Universal Database Driver
  - Query Tuning
    - For SQL calls - use IMS Explorer for Development to view what DL/I calls and SSA Lists are generated to understand what calls will be executed on the backend

```
SQL -
SELECT * FROM PCB01.HOSPITAL, PCB01.WARD WHERE PCB01.HOSPCODE = ‘xxxx’
DL/I -
     GHU HOSPITAL*D(HOSPCODE= xxxx)
         WARD
LOOP GHN HOSPITAL*D(HOSPCODE= xxxx)
         WARD
```

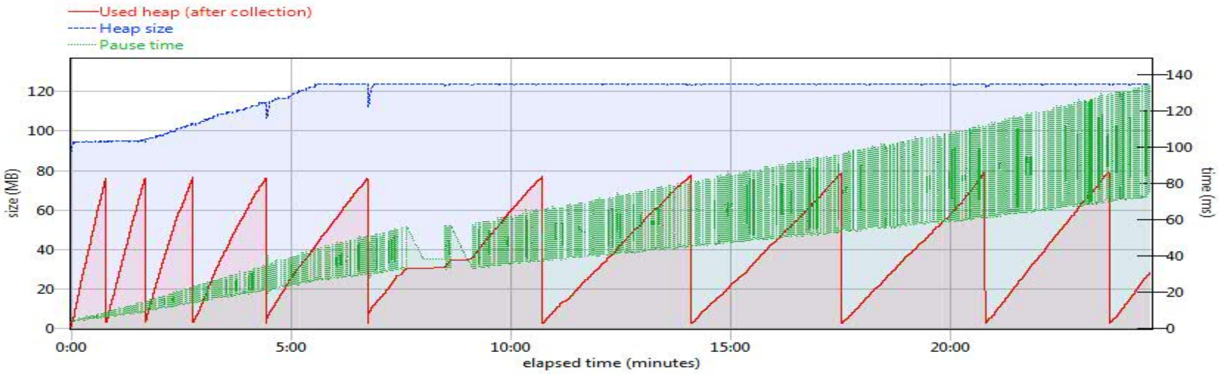
    - Use unique keys or indexes in the WHERE clause
    - Key all the way up to the root segment when possible
  - Fetch Size
    - If your SELECT query is designed to return a fixed number of results then set fetch size to that fixed number. This will prevent additional DL/I calls from being processed and additional data from being transferred over the network.
    - If you need your SQL to return more than one record but are unsure of the total number of results set fetch size to 0. This will help to limit the number of network trips needed to fetch all of the data.
  - Set statement concurrency appropriately
    - CONCUR_READ_ONLY will avoid IMS HOLD calls which can help reduce locks

Performance considerations: JNI considerations

- Java Native Interface (JNI) considerations
  - Minimize the transitions between COBOL and Java. Transitions are costly so make sure to do enough work to amortize the cost of the transition.
  - Caching is key! Cache method ID's, field ID's, and classes to limit calls to FindClass, GetXXXMethodID, CallXXXMetohd
  - Pass parameter values directly rather than using callback methods
  - When passing large blocks of data weigh the benefits of ByteBuffer.allocateDirect(int capacity) vs new Byte[capacity]
    - new Byte[capacity] provides quick allocation but more expensive to pass between Java and COBOL – shorter lived application may benefit
    - Direct ByteBuffers are more expensive to allocate but simply pass a pointer between Java and COBOL – longer lived applications with cached buffers may benefit

Performance considerations: JNI considerations

- JNI considerations
  - –Use Java options to help develop and troubleshoot
    - –Xcheck:jni – find JNI errors in development
    - –Xverbose:jni – find JNI reference leaks



  - –Best practices: https://www.ibm.com/developerworks/library/j-jni/index.html

Performance considerations: DB2 access considerations

- DB2 considerations
  - ESAF connection pooling - cache up to 50 IMS-DB2 connections (USERID+IMS PGM/DB2 PLAN)
    - Avoid ESAF create thread/terminate thread processing overhead
    - IMS 14 and higher, DB2 11 and higher

- DB2 JCC/JDBC driver considerations
  - If using ESAF connection pooling with the DB2 JCC drivers, set DB2JCC_ESAF_THREAD_NOTIFICATION=YES
    - Performs additional coordination between IMS and DB2 JCC to cache SQL statements on the cached connections
  - Limited Block Fetch – can help significantly when issuing SELECTs that fetch large amounts of data

Performance considerations: JVM tuning and monitoring

- JVM monitoring with tools such as IBM OMEGAMON for JVM on z/OS or IBM Java Health Center
  - Heap or native memory leaks
  - Method profiling to identify Java methods to optimize
  - Visualization of Garbage Collection
  - Lock contention
  - Thread activity

Performance considerations: LE tuning

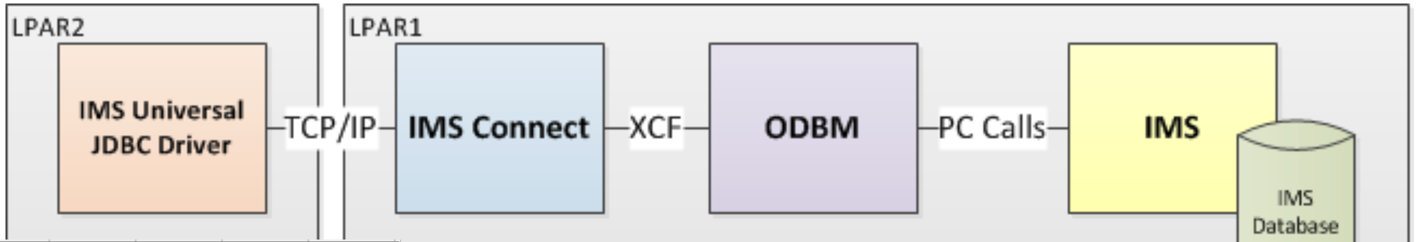
- Language Environment tuning
  - Examine the output of RPTOPTS(ON) and RPTSTG(ON) to help tune LE parameters such as HEAP, ANYHEAP, BELOWHEAP, HEAPPOOLS
  - Services available to help with this level of tuning/analysis
    - A good read on how to approach tuning LE: https://www.ibm.com/developerworks/data/library/techarticle/dm-1410languagetuning-infospheredb2/index.html

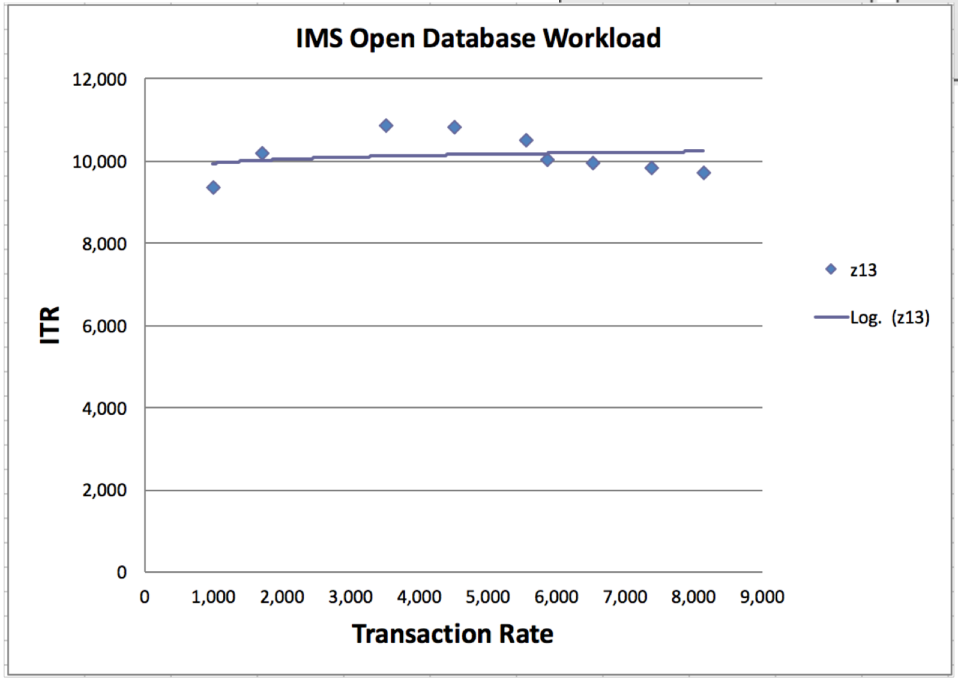
Performance at the lab

Tools used to monitor/measure performance

- Java monitoring
  - IBM OMEGAMON for JVM on z/OS
  - IBM Health Center – Java application monitoring/analysis including method profiling, garbage collection/memory leaks, lock contention, file I/O, etc
  - IBM Application Performance Analyzer – application monitoring/analysis – Java, COBOL, PL/I, etc
  - Various Java libraries available for timing sections of code
- IMS Connect
  - IMS Connect Extensions – capture/archive events (DRDA), ODBM routing exits, analyze active sessions
  - IMS Problem Investigator – merged view of IMS Connect events (DRDA), IMS logs, etc
- IMS
  - IMS Performance Analyzer – process IMS logs – monitor PSB schedules, pools space
- Test drivers
  - Rational Performance Tester, Selenium, JMeter, SOAP UI, stand alone drivers

Performance observations: Distributed access to IMS data





Distributed access to IMS database is a vary scalable solution as well.

(Controlled measurement environment, results may vary)

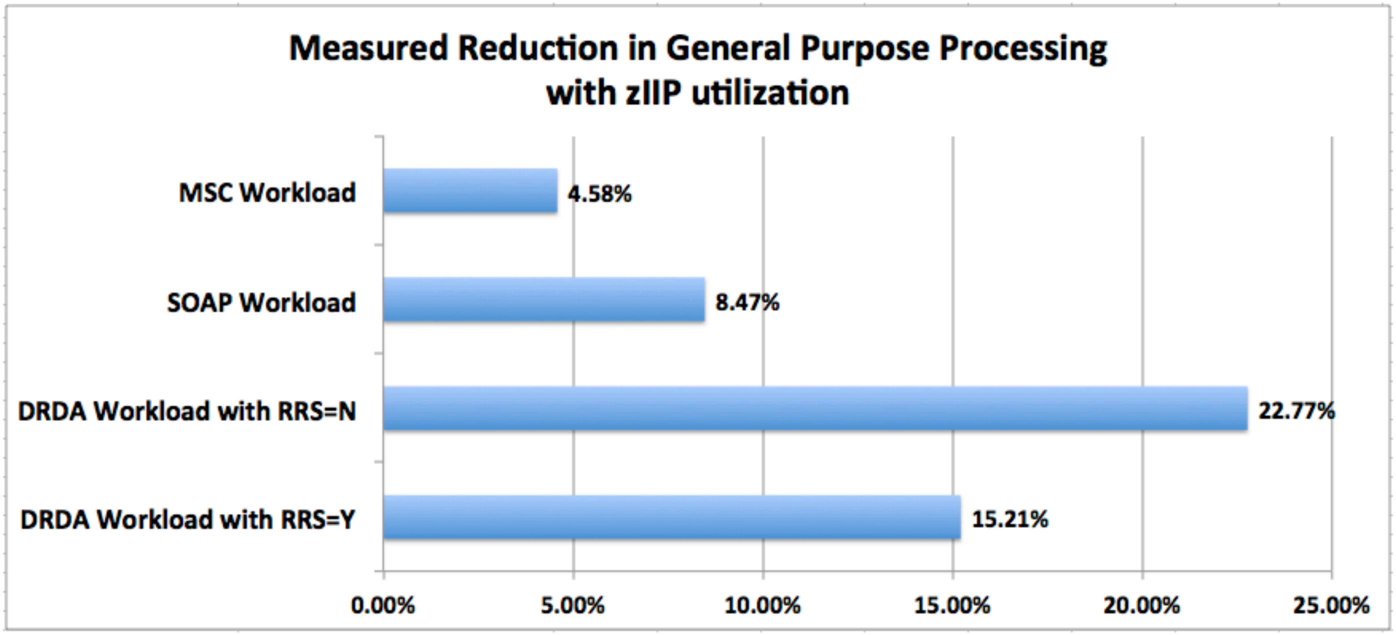
IMS V13 zIIP Eligible Workloads

When enclave SRB execution is enabled, IMS V13 will direct z/OS to authorize certain work to be processed on an available zIIP. Refer to the Notices on Slide 2. Portions of the following processing can execute under an enclave SRB in V13:

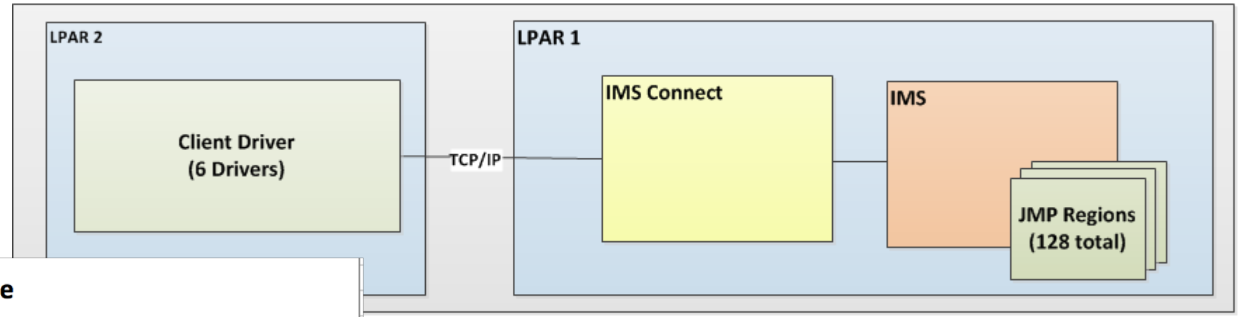
- **DRDA Workload -** The processing of IMS Connect and ODBM address space Distributed Relational Database Architecture ("DRDA") threads for DRDA requests arriving via TCP/IP (“DRDA Workload”)
- **SOAP Workload -** The processing of IMS Connect address space SOAP message threads for SOAP messages arriving via TCP/IP (“SOAP Workload”)
- **MSC Workload -** The processing of IMS Connect address space Multiple System Communication ("MSC") threads for MSC messages arriving via TCP/IP (“MSC Workload”)
- **ISC Workload -** The processing of IMS Connect address space Intersystem Coupling ("ISC") threads for ISC messages arriving via TCP/IP (“ISC Workload”)
- **CSLDMI Workload -** The processing of ODBM address space threads for requests arriving through the CSLDMI API (“CSLDMI Workload”)

Note that any user exits called by the above processing will not execute under an enclave SRB. User exits are always given control in TCB mode, and such exit instructions are not authorized to be processed on a zIIP. Also note that certain processing can not, due to technical restrictions, execute under enclave SRBs. Such processing includes calling z/OS Resource Recovery Services ("RRS"), IMS DL/I call processing, and z/OS supervisor calls ("SVCs"). IMS switches from SRB mode into TCB mode to perform such processing, and thus such processing will not execute on a zIIP.

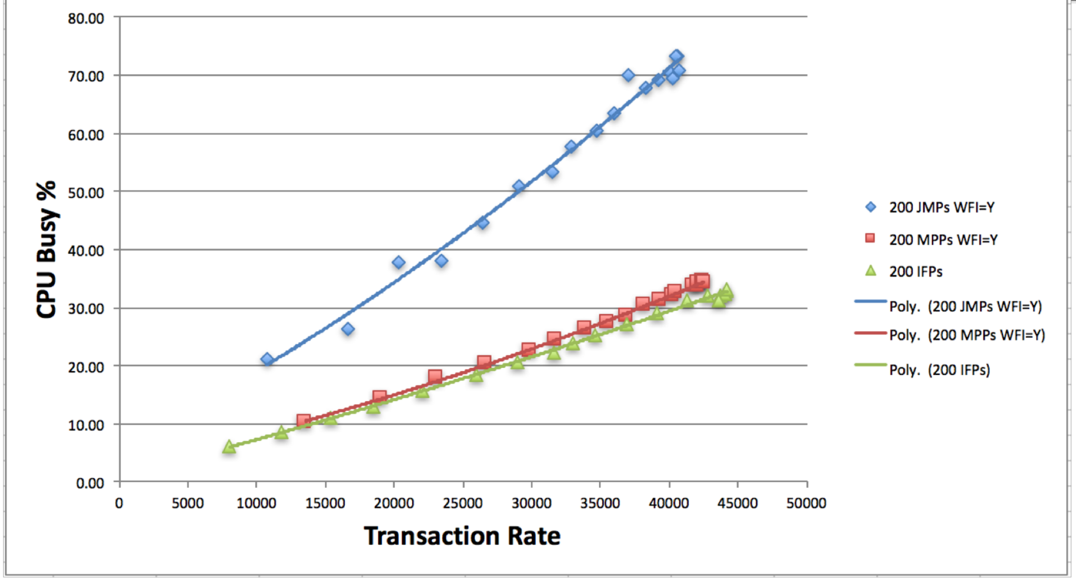
IMS 13 zIIP Eligible Workloads



Performance observations: Java in IMS



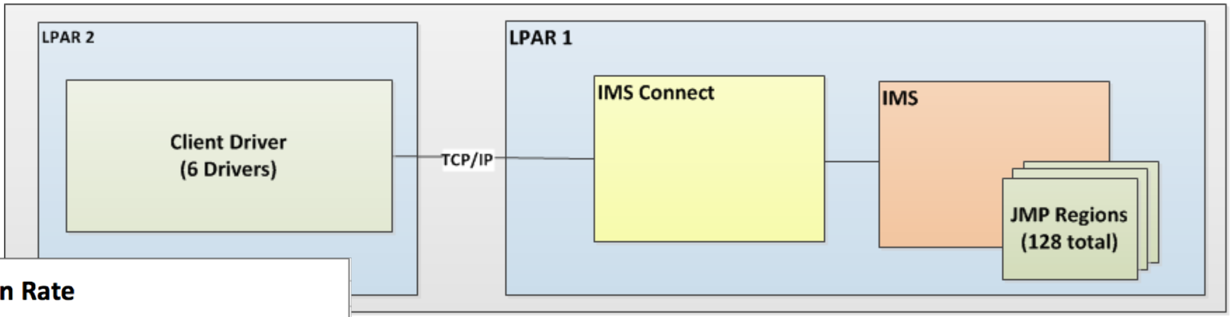
CPU Busy % vs Transaction Rate

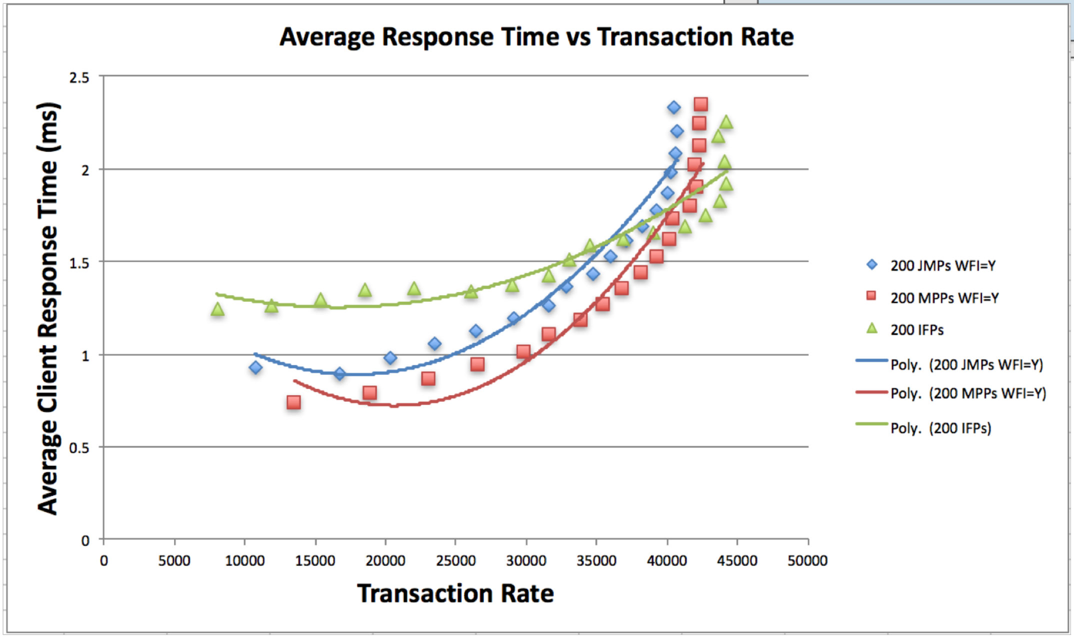


- *Java transaction running in a* ***JMP*** *region vs*
  COBOL applications running in ***IFP*** *or* ***MPP*** *regions*
- *CPU Busy is ~2-2.5x more higher for Java but Java processing is offloadable zIIP*
- *How much can be offloaded highly depends on how much Java processing/business logic*

(Controlled measurement environment, results may vary)

Performance observations: Java in IMS



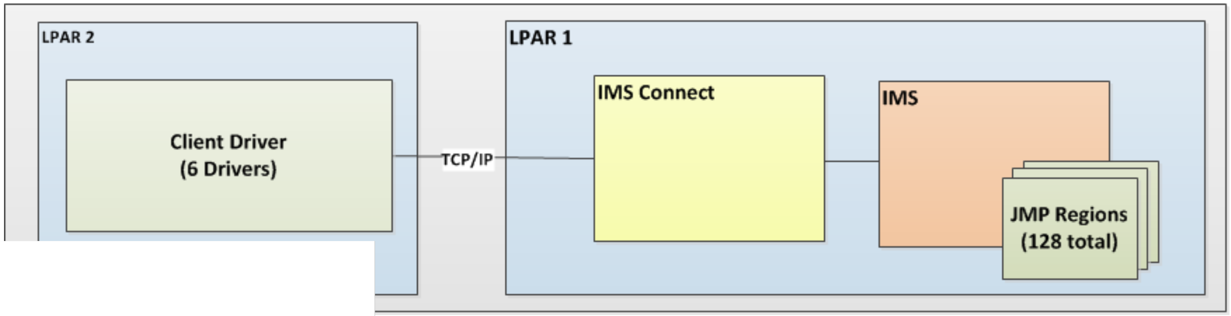


- *Java transaction running in a **JMP** region*
  vs
  *COBOL applications running in **IFP** or **MPP** regions*

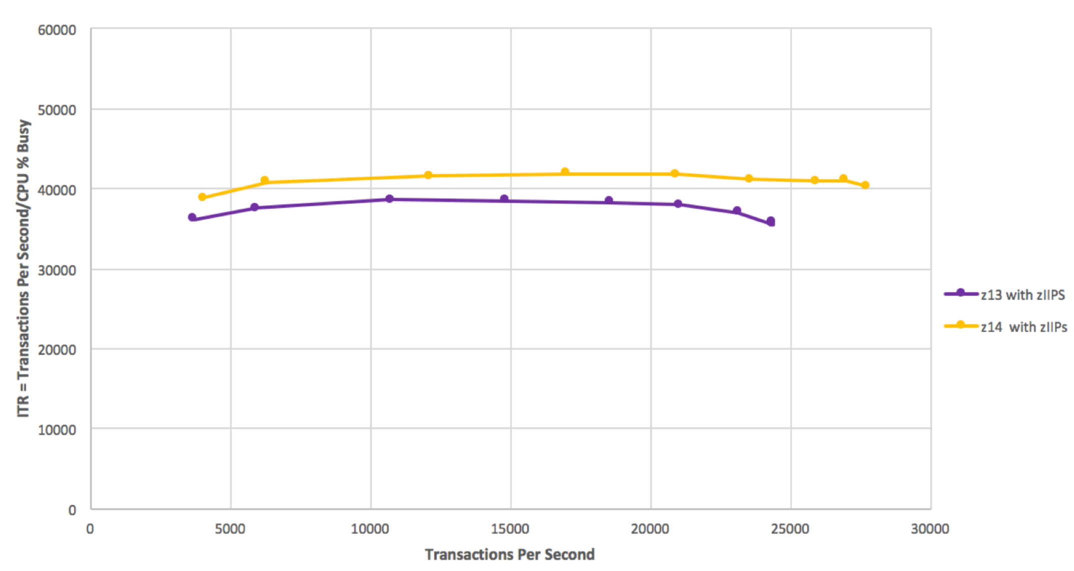
- *Average transaction response times are extremely comparable*

(Controlled measurement environment, results may vary)

Performance observations: Java in IMS



JMP Workload ITR vs Transaction Rate Comparison



- *Z13 vs z14*

(Controlled measurement environment, results may vary)

IMS Makerspace

IMS **Modernization** through **Education** & **Co-Creation**

FREE

- Meet the experts and learn how simple it is to modernize IMS assets
- Define your digital transformation strategy with IMS
- Jump-start with hands-on workshop and POC
- Guided deployment for production

Transform IMS for the Digital World

API

Open IMS transaction and database access as API

Java

Extend existing or develop new IMS applications with Java

DevOps

Integrate IMS assets into enterprise DevOps pipeline

Open Database

Open access to IMS DB with JDBC and SQL

Administrate IMS database with catalog and DDL

Makerspace

A multi-day framework that emphasizes IMS modernization and incorporates education of assets, design thinking exercises and a potential POC & Production delivery.

Education

Understand the assets available for modernizing the existing IMS System

Design Thinking

Reflect on current pain points and business scenarios best related to specific Makerspace agenda

Prototype

Use customer assets to create and deliver on a Prototype that solves a business need

Production (DPO)

Solve a complex use case(s) and deploy into production

Sample Makerspace Schedule for modernizing IMS application with Java

Day 1

Education

For example:

- *Java in IMS – Overview, Use cases, Development, Setup and Deployment*

Design Thinking

- Persona Feedback
- Collect Pain Points
- Prioritize Needs
- As-is/To-be

Day 2 – 3 (Optional)

Prototype

For example:

- *Develop a sample Java application Deploy and run as a JMP in IMS*
- *Rewrite your existing (simple) IMS transaction to use Java and SQL and run in IMS*

Day 4 – 20 (Optional)

Deployment (DPO)

Complex Use Cases

DevOps

Security

Production

Session summary

Key thoughts

- Leverage the vast amount of Java skills in the market
- Leverage the thousands of Java libraries available
- It is possible to write well performing Java applications for IMS
- Start small with
  - converting simple transactions
  - converting simple batch jobs
  - using language interoperability to start converting subroutines

Notice and disclaimers

U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed "as is" without any warranty, either express or implied. In no event shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity. IBM products and services are warranted according to the terms and conditions of the agreements under which they are provided.**

IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer is in compliance with any law.

Notice and disclaimers cont.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products**. IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a particular, purpose.**

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, ibm.com, AIX, BigInsights, Bluemix, CICS, Easy Tier, FlashCopy, FlashSystem, GDPS, GPFS, Guardium, HyperSwap, IBM Cloud Managed Services, IBM Elastic Storage, IBM FlashCore, IBM FlashSystem, IBM MobileFirst, IBM Power Systems, IBM PureSystems, IBM Spectrum, IBM Spectrum Accelerate, IBM Spectrum Archive, IBM Spectrum Control, IBM Spectrum Protect, IBM Spectrum Scale, IBM Spectrum Storage, IBM Spectrum Virtualize, IBM Watson, IBM Z, IBM z Systems, IBM z13, IMS, InfoSphere, Linear Tape File System, OMEGAMON, OpenPower, Parallel Sysplex, Power, POWER, POWER4, POWER7, POWER8, Power Series, Power Systems, Power Systems Software, PowerHA, PowerLinux, PowerVM, PureApplica- tion, RACF, Real-time Compression, Redbooks, RMF, SPSS, Storwize, Symphony, SystemMirror, System Storage, Tivoli, WebSphere, XIV, z Systems, z/OS, z/VM, z/VSE, zEnterprise and zSecure are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both. Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Thank you

ibm.com/training/systems