DB2 for z/OS Technical Conference

Application Development Best Practices

Maria Sueli de Almeida

sueli@us.ibm.com

1-408-463-5192

Important Disclaimer

- THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.
- WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.
- IN ADDITION, THIS INFORMATION IS BASED ON IBM’S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE.
- IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION.
- NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, OR SHALL HAVE THE EFFECT OF:
  - CREATING ANY WARRANTY OR REPRESENTATION FROM IBM (OR ITS AFFILIATES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS); OR
  - ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF IBM SOFTWARE.

Introduction

- DBAs, system programmers and application developers
- Variety of ways DB2 data can be accessed
- Variety of programming languages
  - Java, C, Perl, Python, OHP, Ruby, etc..
- DB2 features that support application development
- Impact of those features in the DB2 system
- Application functions that should be synchronous and asynchronous
- SOA concepts
- Tooling
  - Eg. IBM Data Studio for stored procedure and web services

Introduction continue

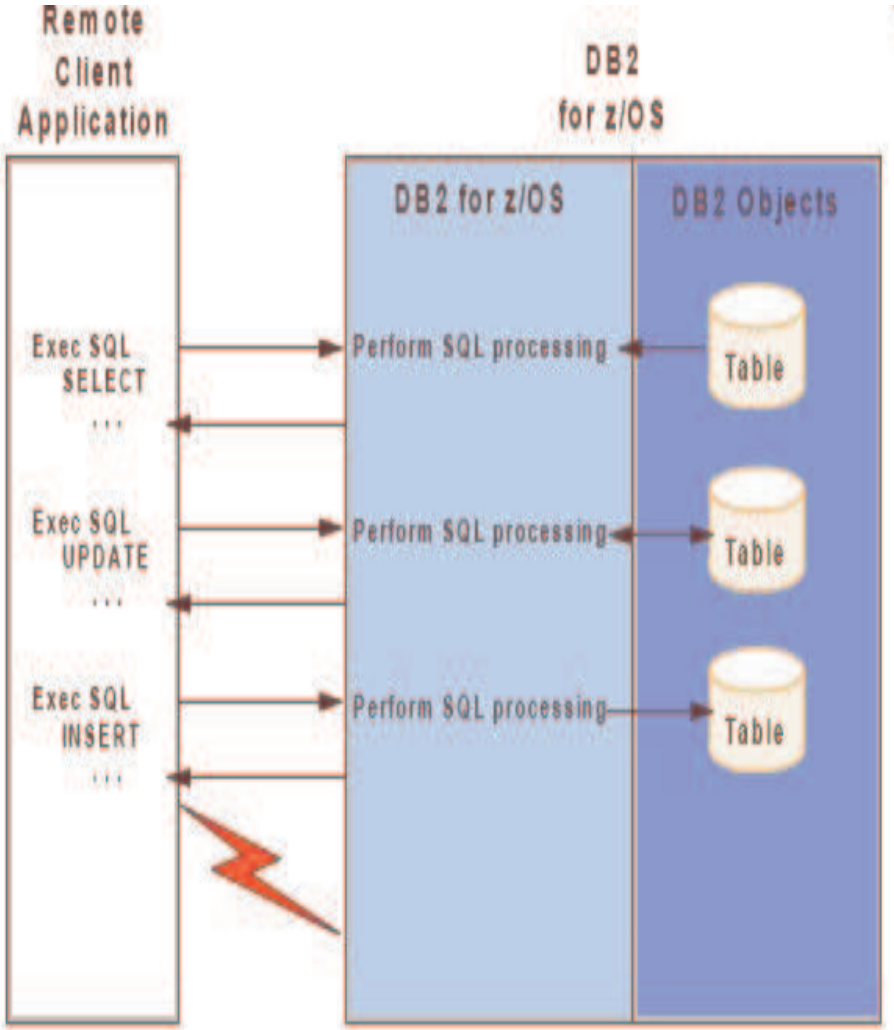
- Selected topics
  - Usage of
    - Stored procedures, UDFs and triggers
    - Dynamic SQL
    - Multi Row Fetch/Multi Row Insert
    - LOBs

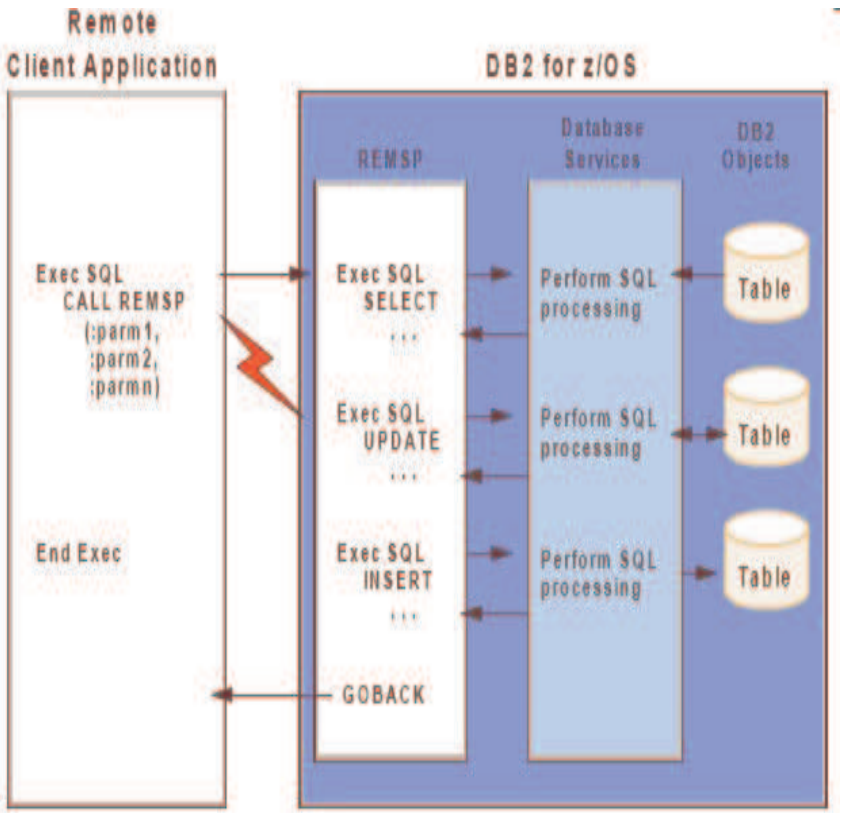
Stored Procedures, UDFs and Triggers

What are Stored Procedures

- It is a user-written program that can be called by an application with an SQL CALL statement.
- It is a compiled program that is stored at a DB2 server, and can execute SQL statements.
- Stored procedures can be called:
  - locally (on the same system where the application runs) and
  - remotely (from a different system).
    - Reducing the traffic of information across the communication network
    - Splitting the application logic and encouraging an even distribution of the computational workload
    - Providing an easy way to call a remote program

Processing without & with Stored Procedures





The same SQL previously executed by the client has been stored on the server and is called by the client whenever necessary. The invocation is treated as a regular external call:

- The application waits for te stored procedure to terminate
- Parameters can be passed back and forth

Benefits of stored procedures

- Modularity in application development
- Data will be processed always in a consistent way according to the rules defined in the stored procedure
- Reduced network traffic for distributed applications
  - Typical application requires two trips across the network for each SQL statement
  - Grouping SQL statements into a stored procedure results in two trips across the network for each group of statement, resulting in better performance for applications
- Improved application security
  - Sensitive business logic runs on the DB2 server
  - End users do not need table privilege

Benefits of stored procedures cont…

- Access to features that exist only on the server:
  - Stored procedures can have access to commands that run only on the server.
  - They might have the advantages of increased memory and disk space on server machines.
  - They can access any additional software installed on the server.

- Enforcement of business rules:
  - You can use stored procedures to define business rules that are common to several applications.
  - This is another way to define business rules, in addition to using *constraints* and *triggers*.

Benefits of stored procedures cont…

- Application integration solutions:
  - You can use stored procedures to easily access non-DB2 resources.
  - With the use of WebSphere® MQ, you can coordinate accesses to multiple data and platforms.
- Cost of ownership reduction
  - DRDA® activity is a candidate for zIIP re-routing. A smaller percentage of work is redirected to zIIP for remote non SQL-native-procedures, just the CALL, COMMIT and result set processing.
  - Stored procedures written in Java can take advantage of zAAP engines
  - Native SQL procedures have richer SQL functions and remote native SQL procedures, running as enclaves in DBM1 address space, are candidate for zIIP reroute with DB2 V9.

Use of stored procedures

- Distributed applications to:
  - Distribute the logic between a client and a server
  - Perform a sequence of operations at a remote site
  - Combine results of query functions at a remote site
  - Control access to database objects
  - Remove SQL applications from the workstation and prevent workstation users from manipulating the contents of sensitive SQL statements and host variables
  - Dynamically invoke static SQL rather than use Java Data Base Connectivity (JDBC) dynamic SQL approach

- To access non-DB2 resources:
  - VSAM files
  - Flat files
  - IMS or CICS transactions
  - DL/I databases
  - MVS/APPC conversations
  - Utilize Recoverable Resource Services (RRS) to coordinate two-phase commit processing of recoverable resources.

Use of stored procedures cont…

- When the details of trigger and User Defined Function (UDF) processing go beyond the scope of SQL statements, stored procedures can be called for the application logic.
- To transport messages using MQSeries® functions that:
  - Notify other business processes that an event has taken place
  - Forward information from one process to many other processes
  - Aggregate information from multiple sources to create warehouses and Operational Data Stores (ODSs).

Stored procedures types

- External high level language (EHL-L) procedures
  - COBOL, PL/I, C, C++, Assembler, REXX, and Java
- External SQL language (ESQL-L) procedures
- Native SQL language stored procedures
  - Introduced by DB2 9 for z/OS

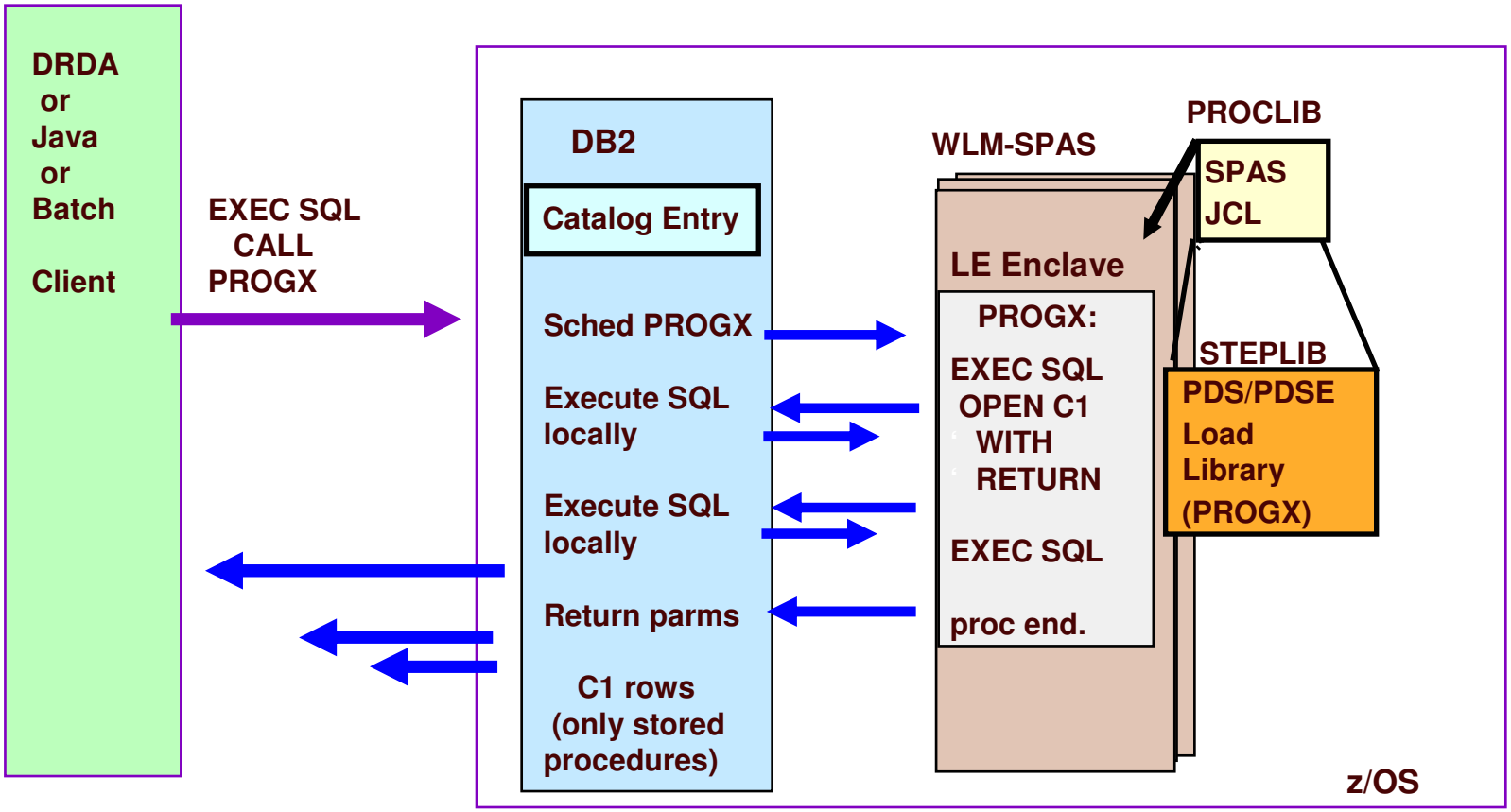
Stored Procedure types: Considerations

- EHL-L
  - Static or dynamic SQL statements
  - IFI calls and DB2 commands issued through IFI
  - Bound to a package
    - Uses the invoking plan's thread
  - Whenever possible, should be prepared as *reentrant programs*
    - Single copy can be shared by multiple tasks in the WLM SPAS
      - This decreases the amount of virtual storage used for code in the SPAS
    - The stored procedure does not have to be loaded into storage every time it is called
  - However, if your stored procedure cannot be reentrant
    - linkedit it as *non-reentrant* and *non-reusable*
      - The non-reusable attribute prevents multiple tasks from using a single copy of the stored procedure at the same time

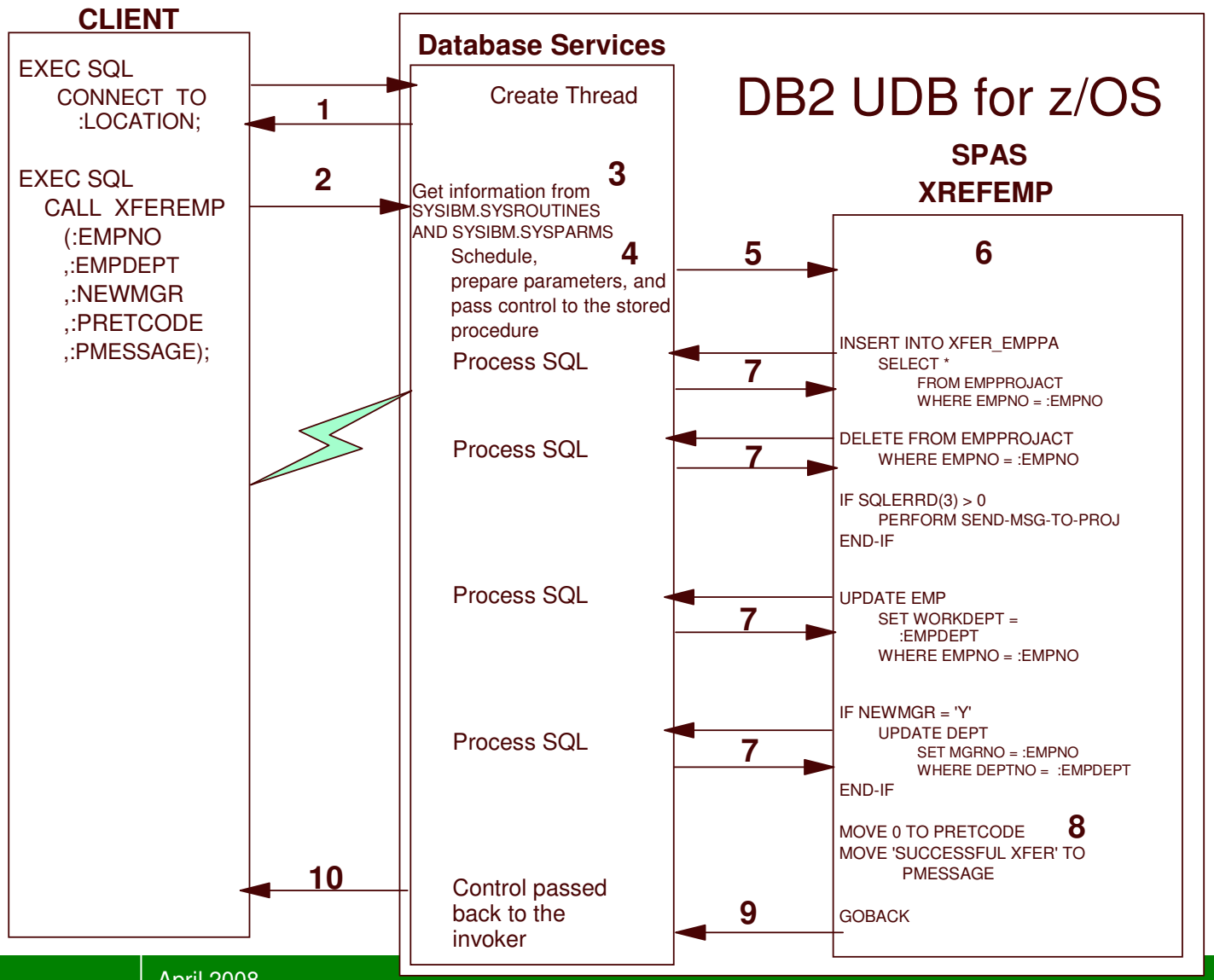
Stored Procedure types: Considerations

- ESQL-L
  - Very similar to EHL-L, except that source code and DDL are tight together
  - Implementation supports Constructs that are common to most programming languages
    - ✓declaration of local variables,
    - ✓assignment of expression results to variables,
    - ✓statements to control the flow of the procedure,
    - ✓receiving and returning of parameters from and to the invoker,
    - ✓returning result sets
    - ✓error handling
- Native SQL
  - DB2 9 for z/OS
  - Very similar to ESQL-L in that the code source is part of the DDL
  - Differences are:
    - richer SQL language
    - no external load module
    - in the executables → entire executable is contained in DB2
    - Easier deploy process → No code level management in load libraries and in WLM application environments

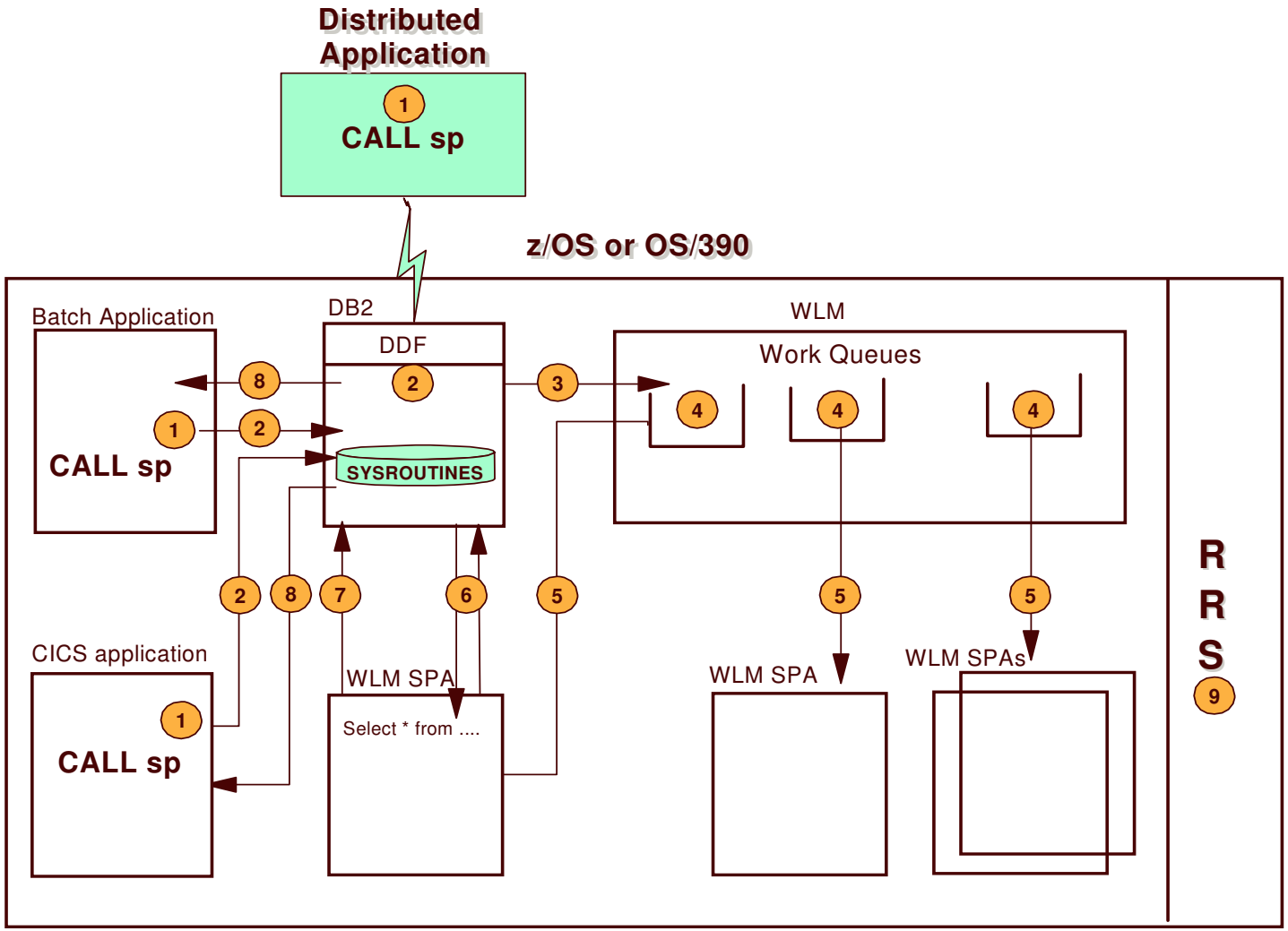
DB2 z/OS Stored Procedures and User-Defined Functions execution



Stored procedure flow



System management of stored procedures



Changes in V8

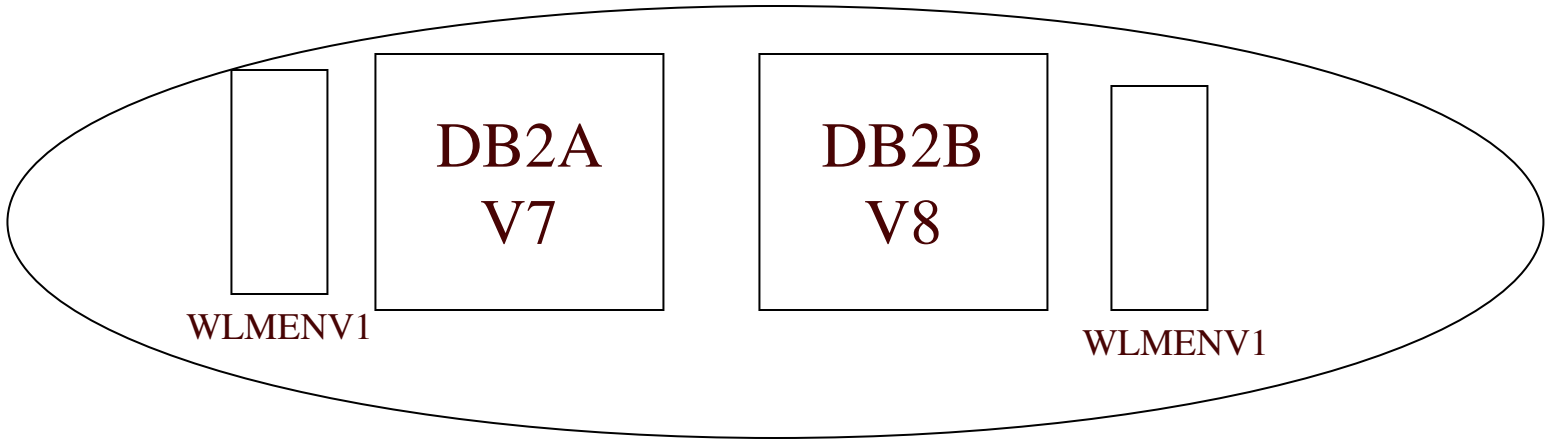
- MAX_ST_PROC zparm (2000 per thd)
  - Causes -904 if exceeded without commit
  - PK08328 provided relief
- Specify number of allowed abends per procedure
- WLM management of tasks, not address spaces
- Can't create DB2-managed stored procedures
  - After V8, no execution either

Datasharing Coexistence

- Example of JCL used after one member of a datasharing group has migrated, so the same DSN needs to refer to different datasets

```
//STEPLIB  DD DISP=SHR,DSN=DSNT2.&DB2SSN..SDSNEXIT
//         DD  DISP=SHR,DSN=DSNT2.&DB2SSN..SDSNLOAD
//         DD  DISP=SHR,DSN=DSNT2.&DB2SSN..SDSNLOD2
//         DD  DISP=SHR,DSN=DSNT2.RUNLIB.LOAD
```

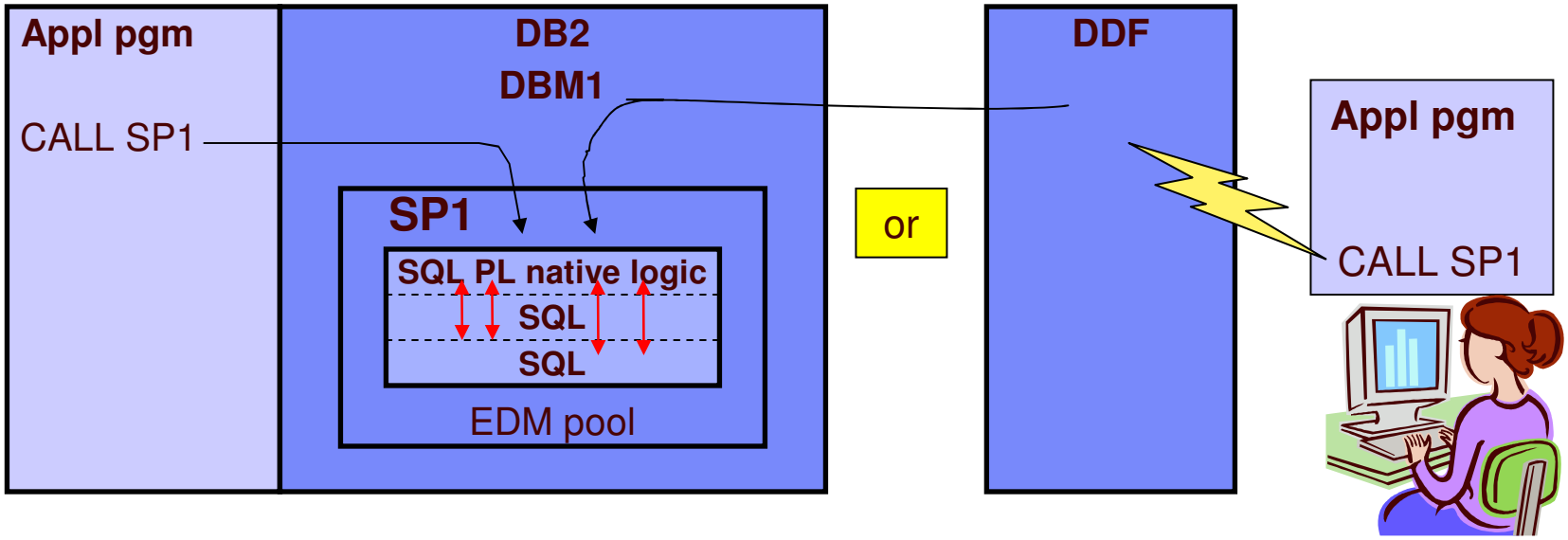
```
DSNT2.DB2A.SDSNLOAD                    *ALIAS
DSNT2.DB2B.SDSNLOAD                    *ALIAS
```

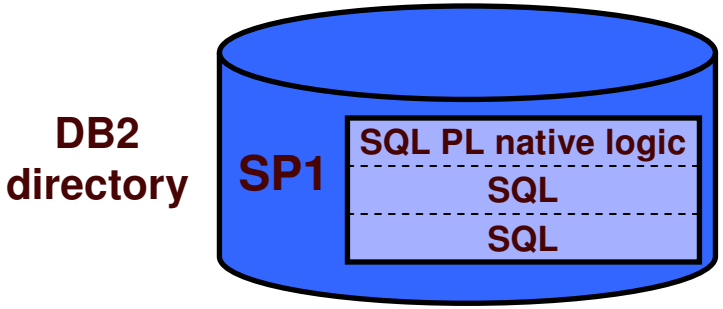


Using V8 current package path

- SP program made up of multiple parts
- Each could be in development/test/production
- Prior to V8:
  - Each program must try SET CURRENT PACKAGESET and handle -805
- In V8:
  - Invoker 1 : development,test,production
  - Invoker 2: test,production
  - Invoker 3: production

Native SQL procedure execution → DB2 9 for z/OS





Stored procedure hot topics

- C compiler for SQL procedures – gone in DB2 9
- zIIP processor
  - TCB execution is not eligible for offload
  - Stored procedure benefits still apply!
    - Less network trips
    - Better concurrency
    - Static SQL packages invoked from dynamic
  - Native SQL procedures in V9 are zIIP eligible *when invoked over DDF*
  - Processing on DDF SRB is eligible
    - Commit, result sets
    - We Measured 10-13%

Java Stored procedure hot topics

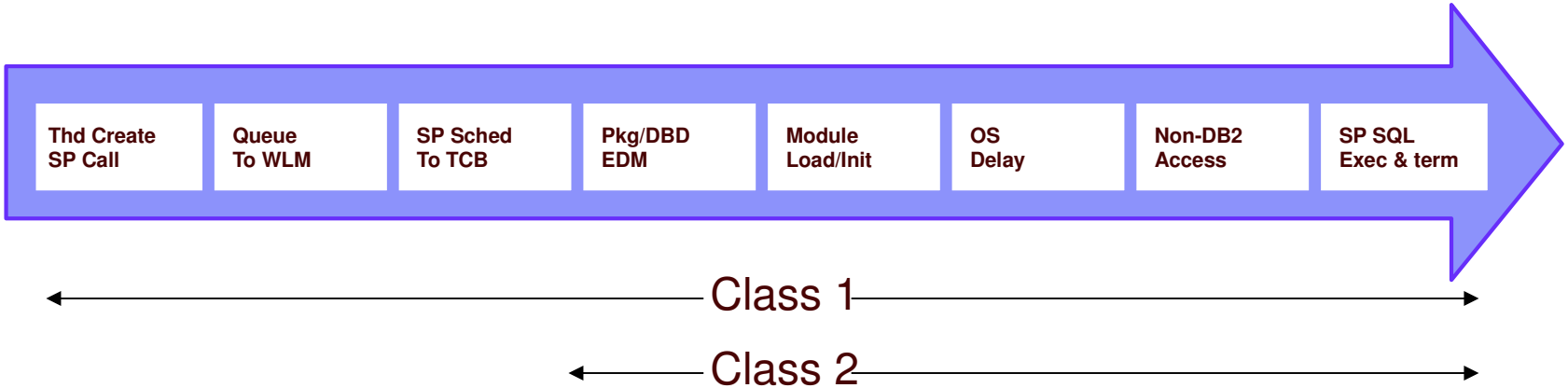
- IBM's zAAP processor

  http://www-1.ibm.com/servers/eserver/zseries/zaap/
- Persistent Reusable JVM topics
  - Gone from JVM 5!!
- Java Shared Classes – shared memory
- Build for Universal Driver and DB2 z/OS V7
- Deployment development/production
- DB2 9: Multiple/common Jars for an application
- DB2 9: Debugging

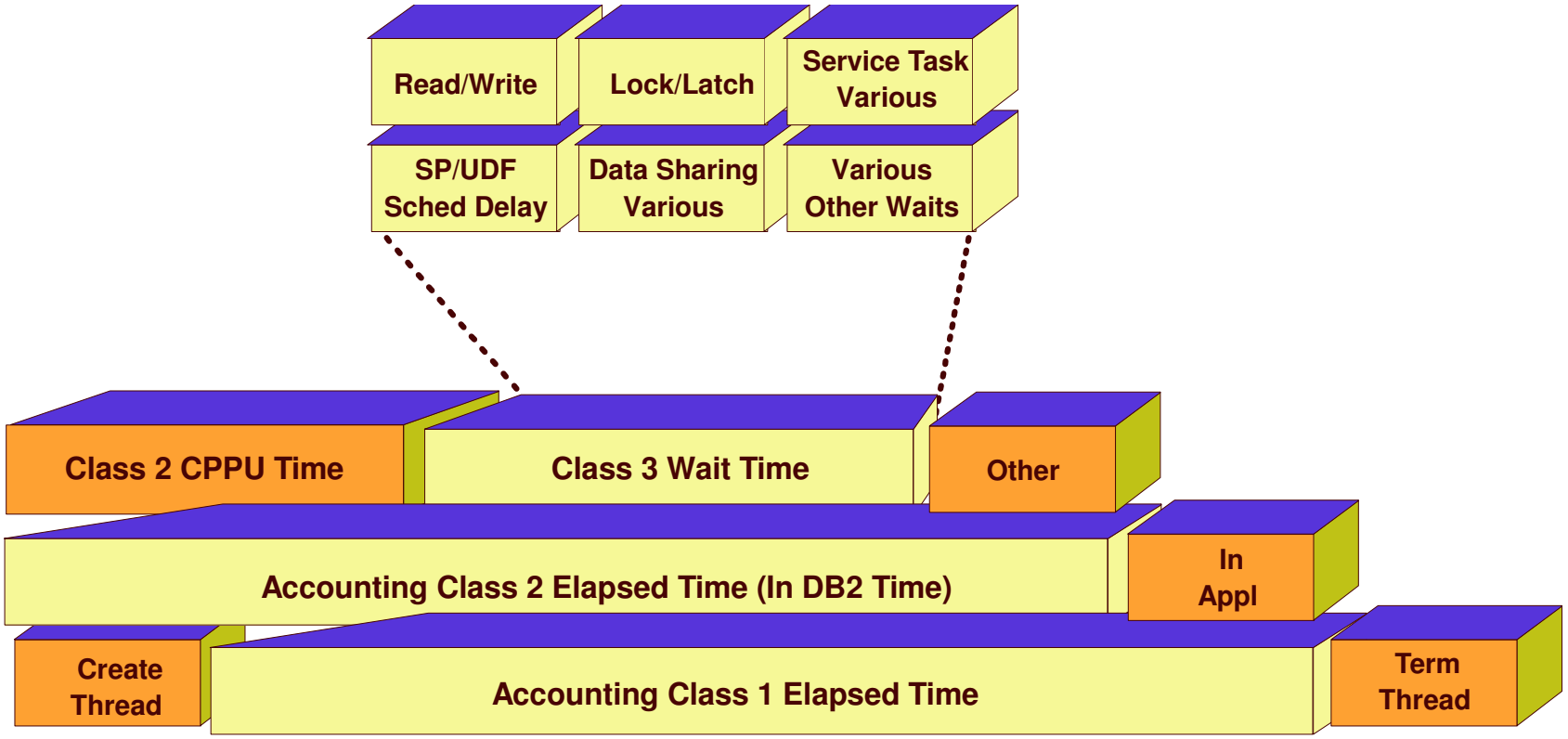
Cost of Various Alternatives

Language	Base Billable Cost	Billable Cost after zIIP and/or zAPP acceleration
Cobol stored proc	1x	.9x
SQLJ stored proc	1.7x	1.3x (zIIP + zAAP)
External SQL stored proc	1.75x	1.6x
Native SQL stored proc	1.2x	.6x

Where does the time go?

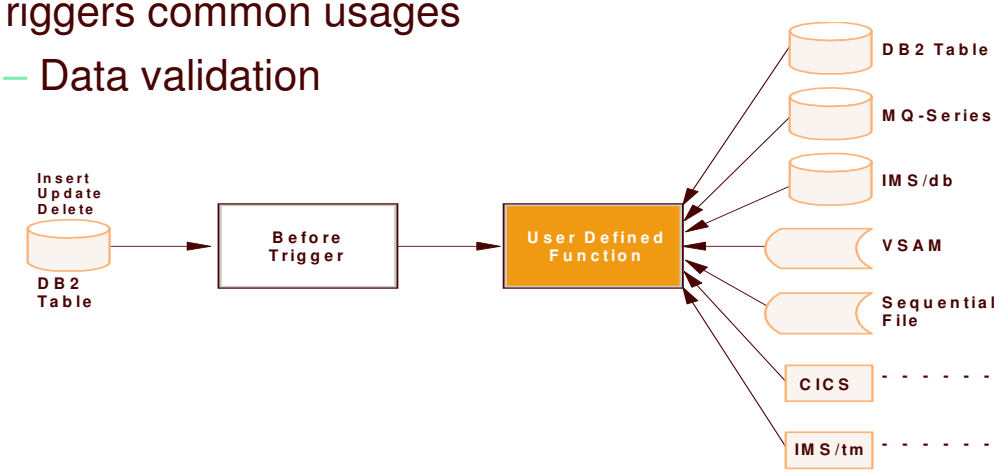


Important DB2 Accounting Traces

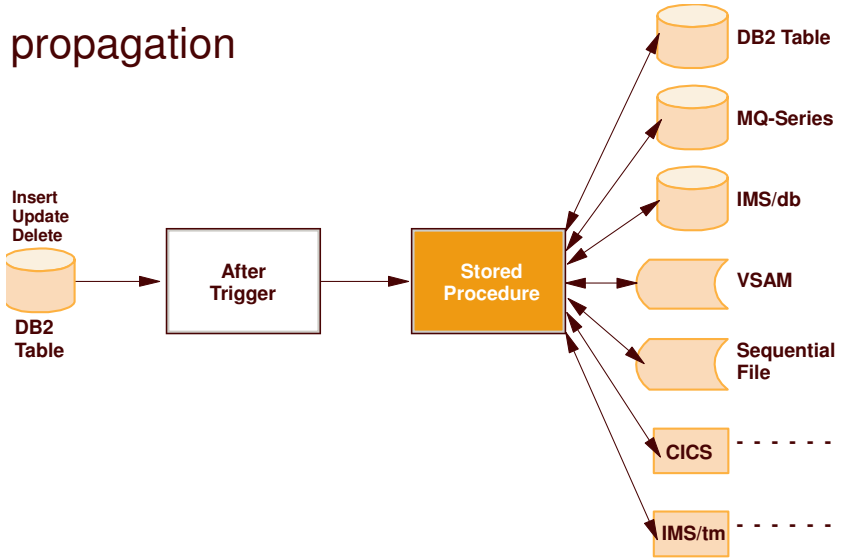


Using triggers and UDFs

- Triggers common usages
  - Data validation



  - Data propagation



Best Practices→Performance Checklist

- Consider pathlength of each invocation
- Tune the SQL
- Don't call the metadata SPs
- Use the SP authorization cache
- Don't println() / DISPLAY in production
- Use PROGRAM TYPE SUB
- Split up SUB to WLMENV by run options
- Use STAY RESIDENT YES
- Use SECURITY DB2
- No more than 512 SPs in one WLMENV

Java Considerations:

- Don't use JSPDEBUG in production
- Make sure the JVM is not destroyed between invocations
  - Test it out with JSPDEBUG
- Use a non-resettable JVM

Controlling the runaways

- Use ASUTIME to control looping
  - Specified on CREATE PROCEDURE
  - Only monitored every 20 seconds

- If no ASU limit specified and looping:
  - Cancel Thread
    (doesn't do anything if program is processing outside DB2)
  - Refresh WLM environment
  - Cancel of WLM-SPAS address space is effective as a **last resort**

In This Order!

Performance Checklist

- Consider pathlength of each invocation
- Tune the SQL
- Don’t call the metadata SPs
- Don’t println() / DISPLAY in production
- Use PROGRAM TYPE SUB
- Split up SUB to WLMENV by run options
- Use STAY RESIDENT YES
- Use SECURITY DB2
- Use the SP authorization cache
- No more than 512 SPs in one WLMENV

Should we use stored procedures locally ?

- When code reuse benefits outweigh additional overhead
  - (additional 30K+ instructions)
- If executing from a high-overhead environment, such as Java on z/OS
- If executing from dynamic SQL (JDBC, ODBC) and prefer static SQL
- Other options for common or I/O modules:
  - Multiple linkedits for use as both SP and non-SP
  - Use DYNAM and entry point DSNHLI

Nested and concurrent routines

- **Multiple stored procedures and UDFs require multiple times the resources**
  - One task per routine
  - Scheduling time for each routine
  - 2004 WLM and DB2 change for “dependent” routines: OA04555, PQ80631
- **Cancel scenarios become more complex**
  - Follow recommended order of events
  - Stay current on cancel maintenance
    - PK22811

Performance Checklist

- Consider pathlength of each invocation
- Tune the SQL
- Don’t call the metadata SPs
- Don’t println() / DISPLAY in production
- Use PROGRAM TYPE SUB
- Split up SUB to WLMENV by run options
- Use STAY RESIDENT YES
- Use SECURITY DB2
- Use the SP authorization cache
- No more than 512 SPs in one WLMENV

SQLPROCEDURECOLS

- Metadata stored procedure supplied by server
- Invoked by V8 CLI client – initially always
- Not invoked by universal java driver
  - Legacy client java driver uses CLI
- Invoked if CLP used to invoke SP
- Fixed in DB2 LUW client V8 Fixpack 4
  - Only invoked if first attempt to call fails, gets parm types
    Setting `DescribeParam=0` in `db2cli.ini` eliminates call

Performance Checklist

- Consider pathlength of each invocation
- Tune the SQL
- Don’t call the metadata SPs
- Use PROGRAM TYPE SUB
- Split up SUB to WLMENV by run options
- Use STAY RESIDENT YES
- Use SECURITY DB2
- Don’t println() / DISPLAY in production
- Use the SP authorization cache
- No more than 512 SPs in one WLMENV

Why SQL stored procedures ?

- Improve the development and usability of stored procedures
  - IBM Data Studio
- Ensure portability of SQL Stored Procedures across DB2 platforms
  - Collaboration across the products
- Simplify migration to DB2
  - Migration tool (http://www-306.ibm.com/software/data/db2/migration/mtk/)
- DB2 9 eliminates C program requirement (native SQL procedures)

Performance Checklist

- Consider pathlength of each invocation
- Tune the SQL
- Don't call the metadata SPs
- Don't println() / DISPLAY in production
- Use PROGRAM TYPE SUB
- Split up SUB to WLMENV by run options
- Use STAY RESIDENT YES
- Use SECURITY DB2
- Use the SP authorization cache
- No more than 512 SPs in one WLMENV

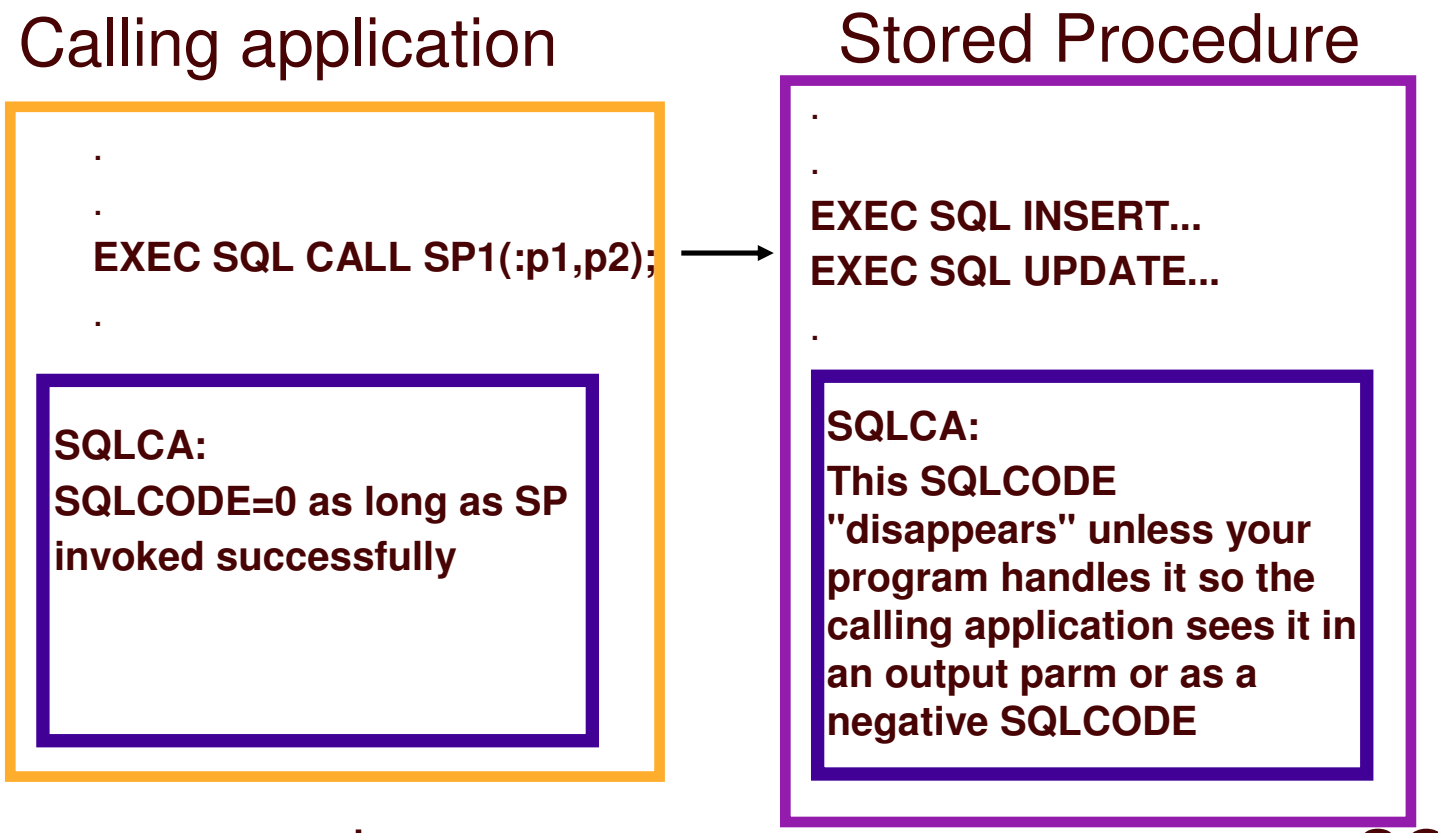
Interactive Debugging

►SQL procedures debugging in Version 8!

- IBM Debug Tool
  - Debug interactively from the workstation
  - How-to in the DB2 "Application Programming and SQL Guide"
  - See www.software.ibm.com/awdtools/debugtool
  - Works with COBOL, C/C++, PL/I

►Unified debugger support in DB2 9: SQL, Java

Stored Procedures need to signal failures



Return error code as parameter or return error on SQL CALL

"What happens in a stored procedure…"

Performance Checklist

- Consider pathlength of each invocation
- Tune the SQL
- Don't call the metadata SPs
- Use the SP authorization cache
- Don't println() / DISPLAY in production
- Use PROGRAM TYPE SUB
- Split up SUB to WLMENV by run options
- Use STAY RESIDENT YES
- Use SECURITY DB2
- No more than 512 SPs in one WLMENV

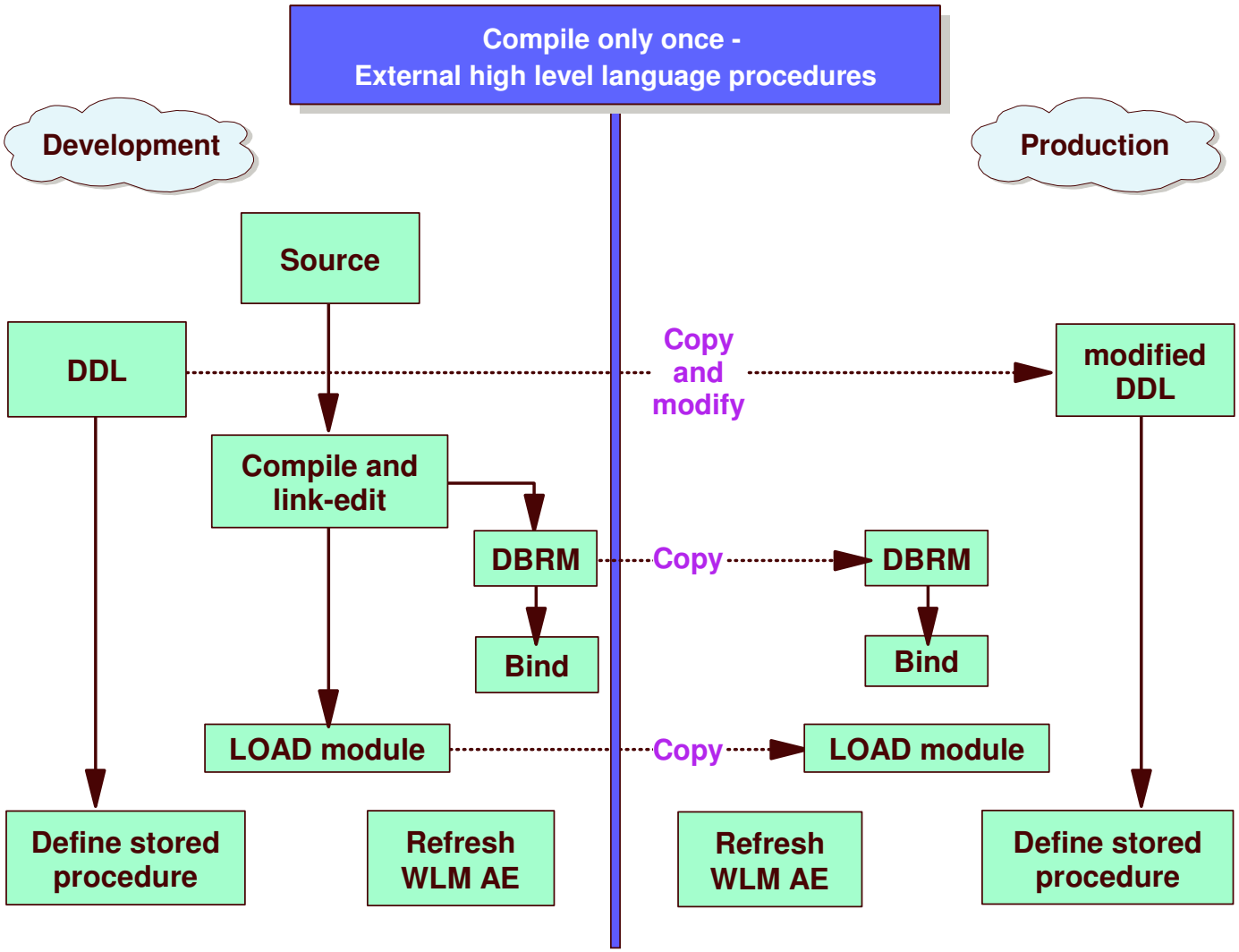
Stored Procedure Authorization Cache

- ZPARM
- ROUTINE AUTH CACHE option on the INSTALL DB2 PROTECTION panel (DSNTIPP).
- If set to zero no caching is done.
- Default is 100KB, maximum is 5MB
- Field name is CACHERAC
- Maximum of 5 authorization IDs cached for each routine

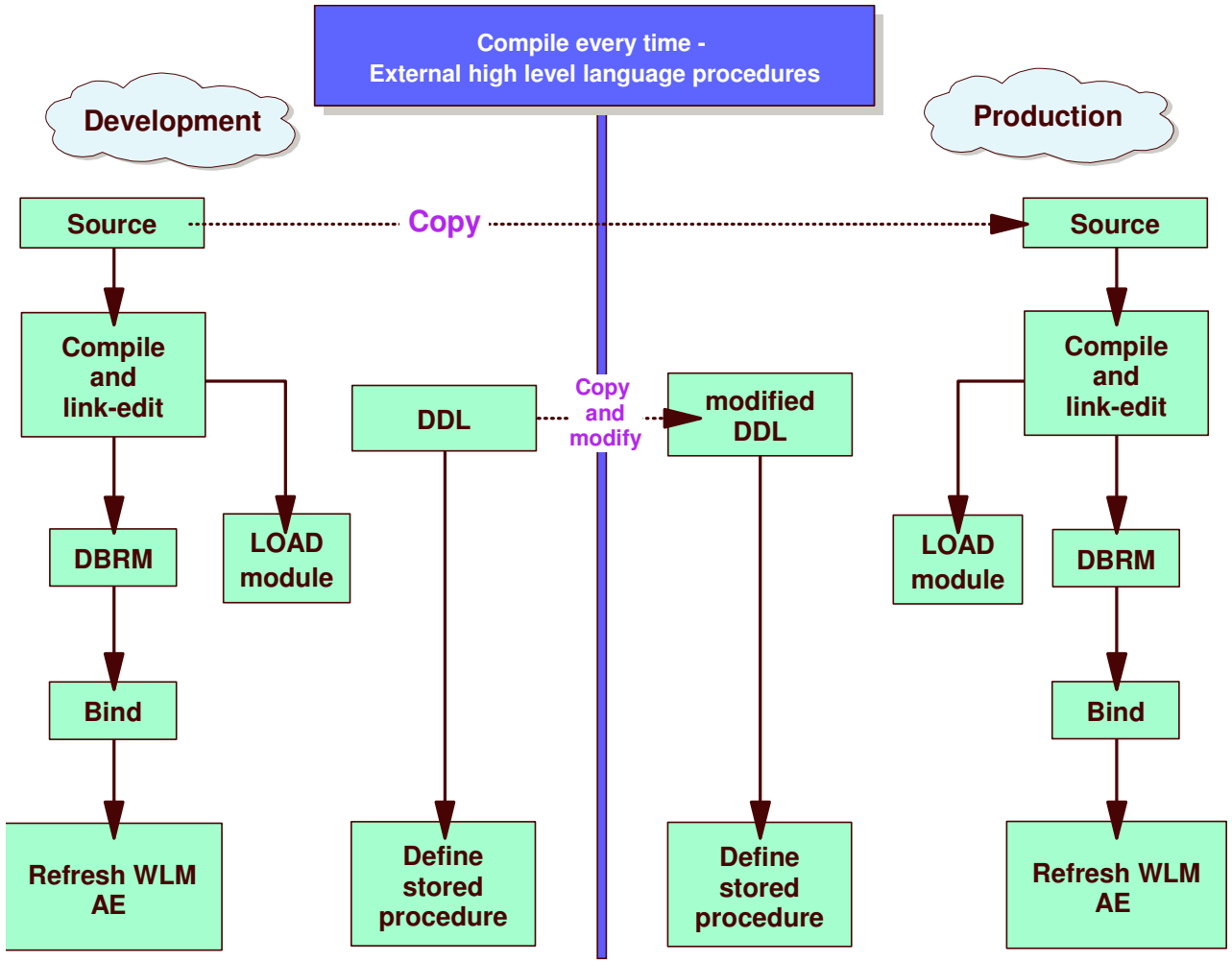
Promotion of stored procedures

- Components to consider
  - External high level language
    - Source code + DDL
  - External SQL language
    - DDL only
      - Source code is part of the CREATE PROCEDURE statement
  - Native SQL language
- Depends on the combination of two categories:
  - Compile only once in development and copy all components
  - Compile in each environment

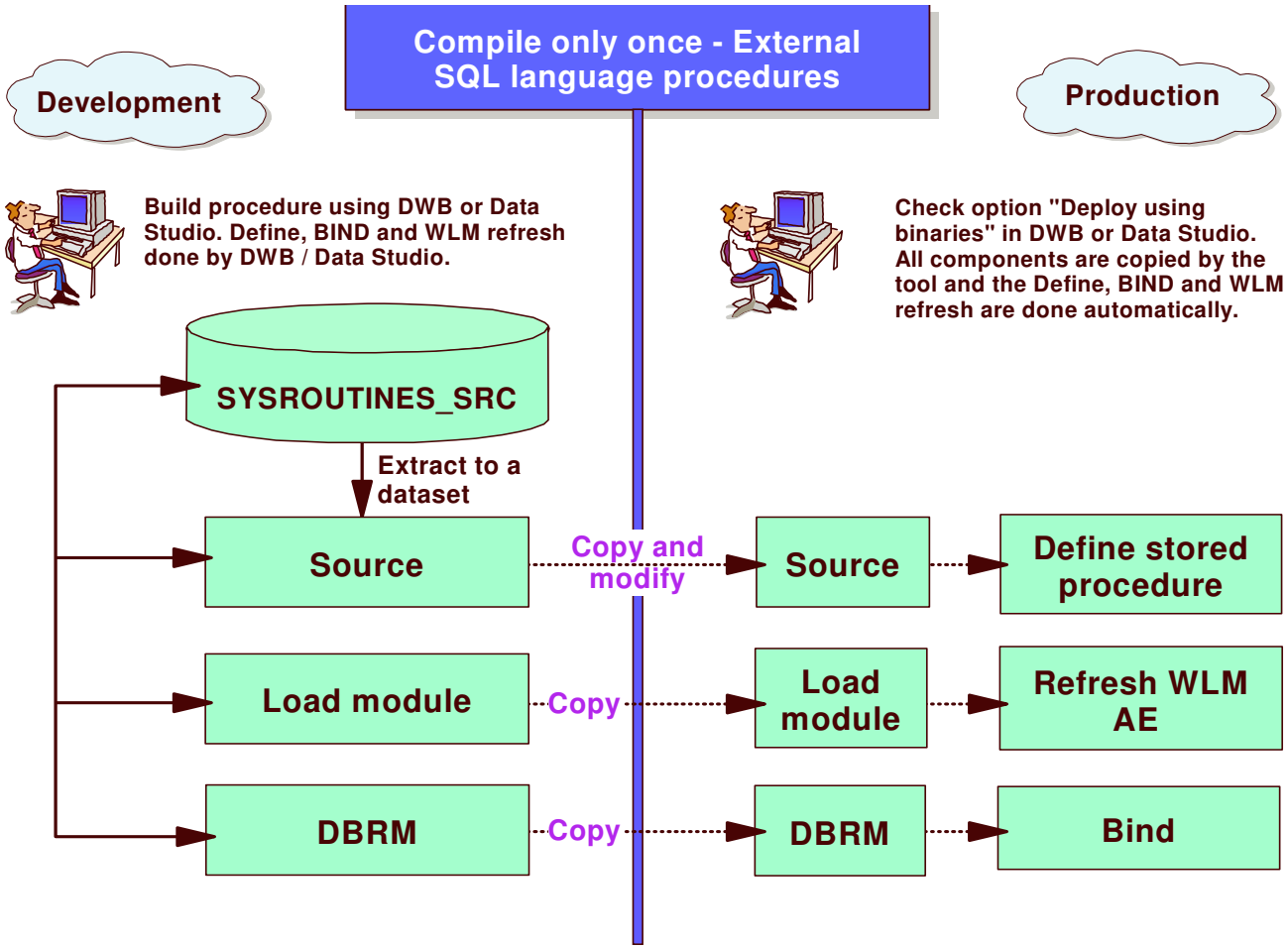
CLM: Compile *only once* - EHLL



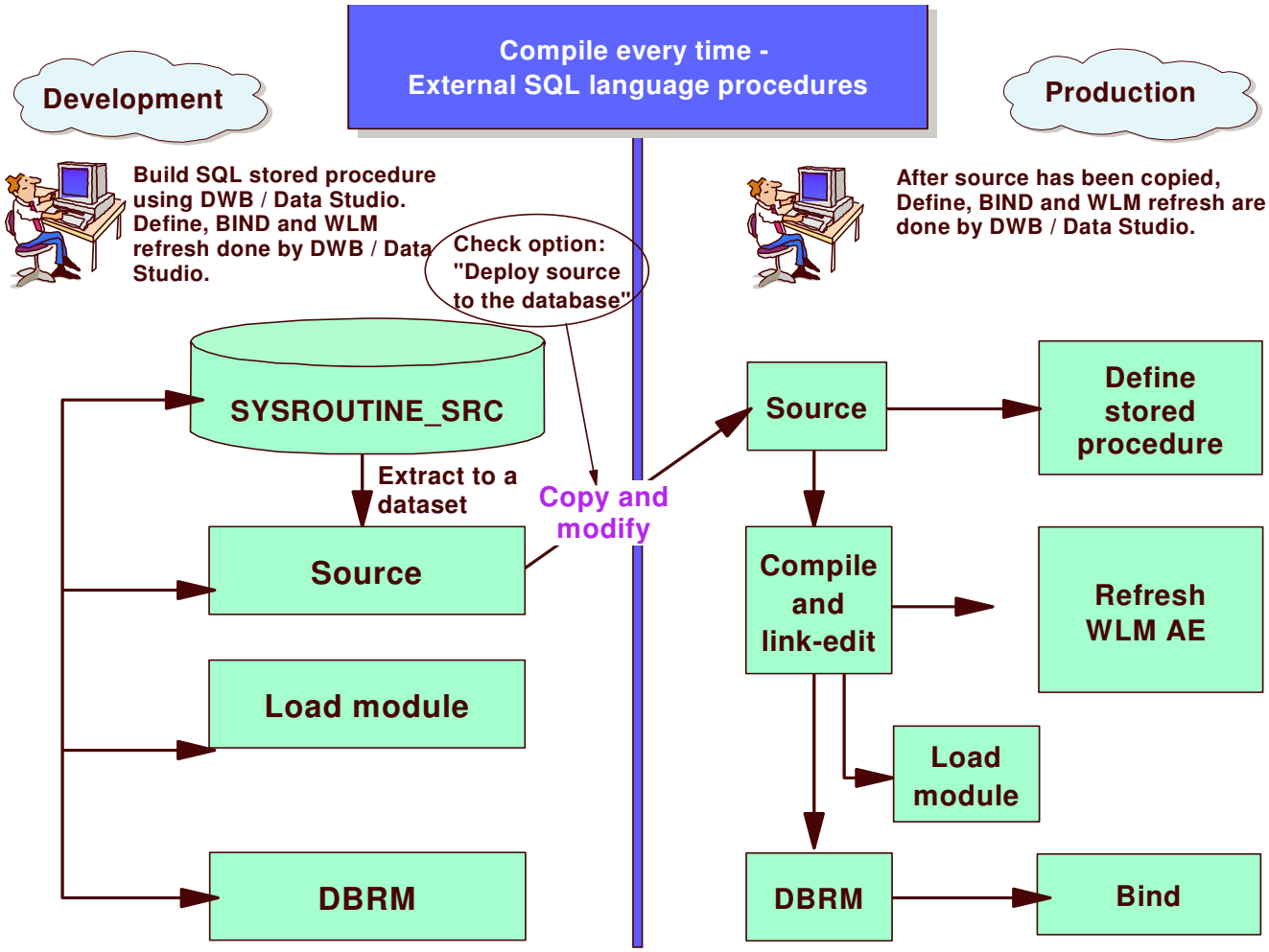
CLM: Compile *every time* once - EHLL



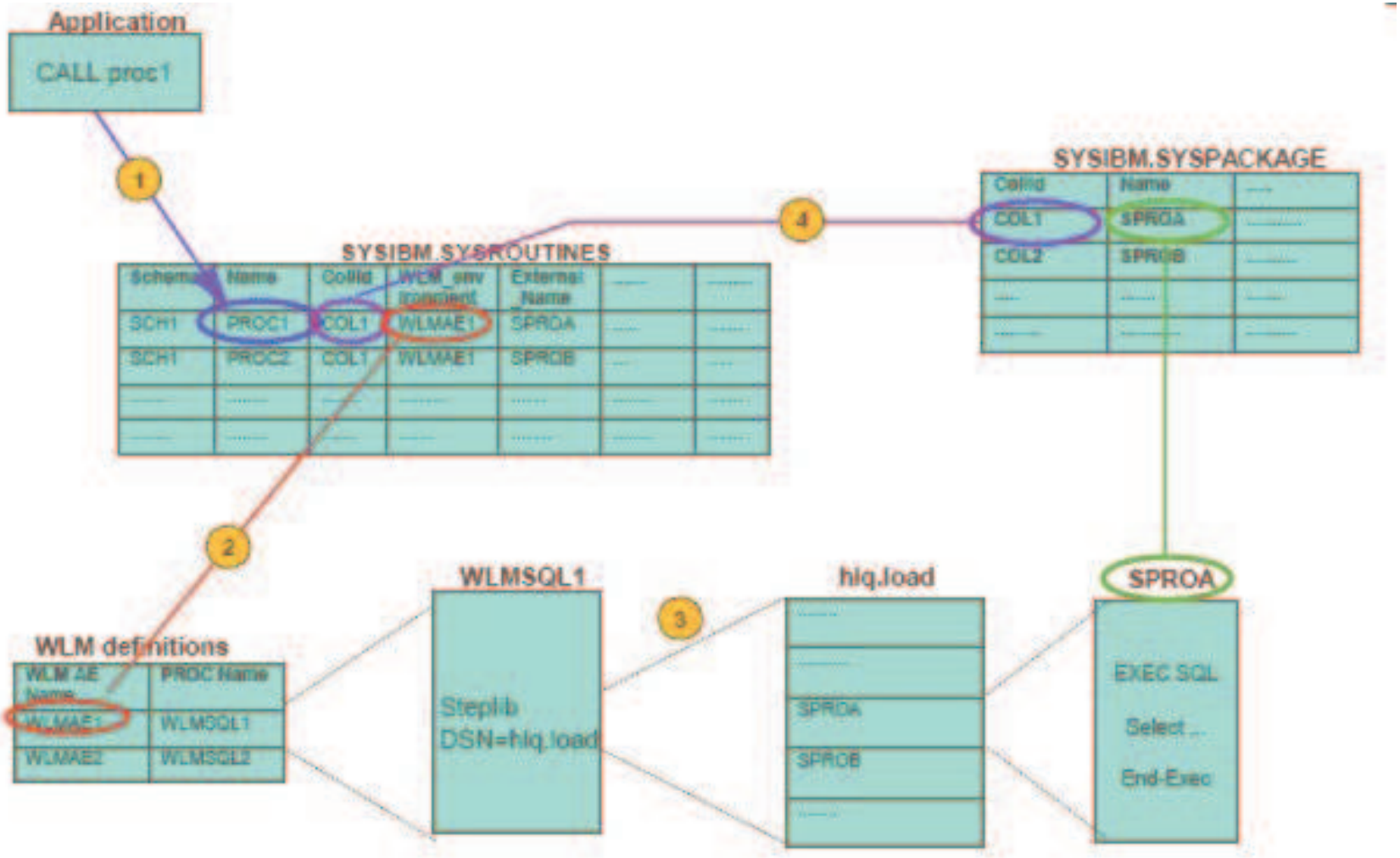
CLM: Compile *only once* - ESQLL



CLM: Compile *every time* - ESQLL



CLM: Relationship between schema, collid, and WLMAE @ runtime



References

- Redbooks
  - www.redbooks.ibm.com
  - SG24-7083 Through the call and beyond
  - Scheduled for update this fall
- New! Collaboration site
  - www.developerworks.com/spaces/db2zos
- DB2 for z/OS: www.ibm.com/software/db2zos
  - Follow ‘support’ link for FAQ’s
- DB2 Developer Domain: www.ibm.com/software/data/developer

DB2 for z/OS information resources

Take advantage of the following information resources available for DB2 for z/OS:

- **Information center**
  http://publib.boulder.ibm.com/infocenter/dzichelp/index.jsp
- **Information roadmap**
  ibm.com/software/data/db2/zos/roadmap.html
- **DB2 UDB for z/OS library page**
  ibm.com/software/data/db2/zos/library.html
- **Examples trading post**

ibm.com/developerworks/exchange/dw_categoryView.jspa?categoryID=25

- **DB2 for z/OS support**
  ibm.com/software/data/db2/zos/support.html
- **Official Introduction to DB2 for z/OS**
  ibm.com/software/data/education/bookstore

DB2 for z/OS Technical Conference

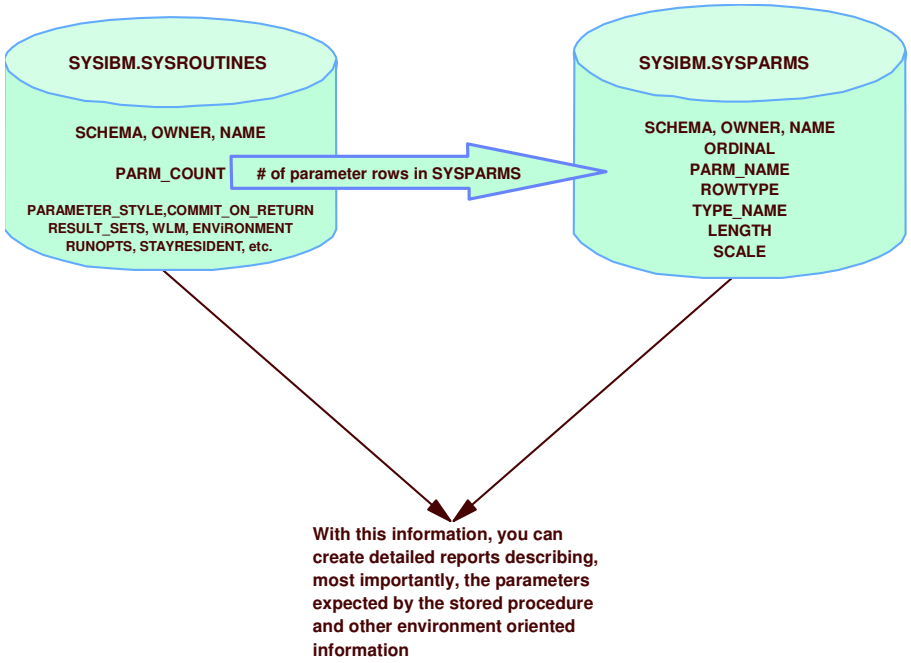
Backup slides

Debugging Stored Procedures

- Add (DISPLAY, println, etc.) debugging statements
- Take these out in production – it's expensive
- To allow multiple Stored Procedures to write output messages to the same sysout file without abend 02A
  - Change runopts to include 'MSGFILE(SYSOUT,,,, ENQ)'
  - Suggestion: put timestamps on messages

Stored procedures & DB2 Catalog tables

- Stored Procedures are DB2 Objects, therefore they must be defined with DDL
- The DDL execution affect two DB2 Catalog tables:
  - SYSIBM.SYSROUTINES
  - SYSIBM.SYSPARMS



Additional DB2 Catalog Tables

- For Java stored procedures
  - SYSIBM.SYSJARCLASS_SOURCE
  - SYSIBM.SYSJARCONTENTS
  - SYSIBM.SYSJARDATA
  - SYSIBM.SYSJAROBJECTS
  - SYSIBM.SYSJAVAOPTS
  - SYSIBM.JAVAPATHS
- For external SQL language and native SQL language stored procedures
  - SYSIBM.SYSENVIRONMENT
  - SYSIBM.SYSROUTINES_OPTS
  - SYSIBM.SYSROUTINES_SRC
  - SYSIBM.SYSROUTINESTEXT
  - SYSIBM.SYSROUTINESAUTH

Using DSNHLI and DYNAM

- COBOL DYNAM is OK (the DB2 book is wrong)
- COBOL dynamically loads modules that are external references, including DSNELI,/DSNHLI/DSNRLI/etc
- By default, DSNHLI
  - Not correct module for either CICS or stored procedures, so books should recommend NODYNAM.
- Options:
  - Using the ATTACH(RRSAF) precompiler option
  - COPY DSNRLI module into a load library concatenated in front of the DB2 libraries and call it DSNHLI.

Summary

- Key Rules:
  - Consider pathlength of each invocation
  - Use WLM environments
  - Specify ASUTIME LIMIT
  - Cancel following prescribed order
- Closing thoughts :
  - Many large companies in very high production environments
  - Pitfalls are mostly when getting started