



IBM Software Group

Understanding The Importance Of Workload Manager And DB2

Ed Woods / IBM Corporation

Tivoli software

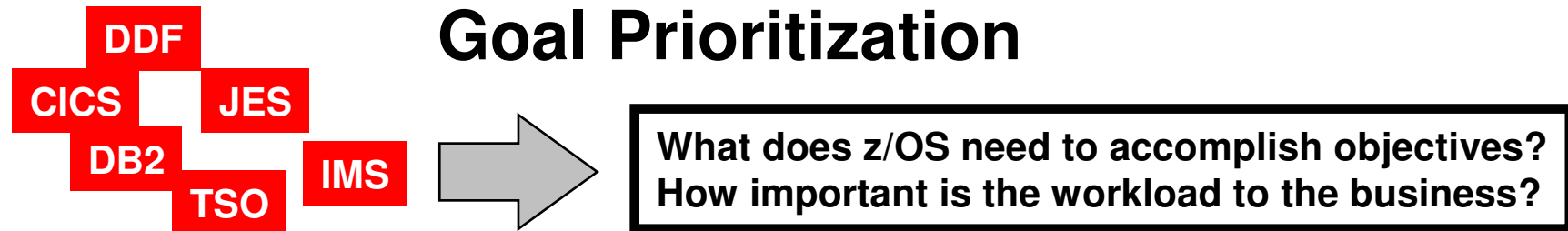


Agenda

- Workload Manager Overview
- Important WLM Concepts And Terminology
- How DB2 Exploits Workload Manager
- WLM/DB2 Usage Recommendations
- What is new in WLM and how it impacts DB2
- Summary



Why Workload Manager (WLM)?



- Complexity of systems has increased dramatically
 - ▶ In the past priority managed by PARMLIB options
 - ▶ Too much work to 'micro-manage' each z/OS system
 - ▶ Multiple LPARs, Data sharing, large sophisticated workloads
- Workload Manager improves the performance management process
 - ▶ Prioritize workload based upon goals and business objectives
 - ▶ Let the system optimize and prioritize resource management
 - ▶ Make sure that the most important workload gets the appropriate resource

DB2 And Workload Manager

- Workload Manager (WLM) is the priority and resource manager for z/OS and implicitly for DB2 as well
- Workload Manager manages CPU, I/O, and memory resources as needed by the workload
- WLM manages critical DB2 address spaces
 - ▶ DB2 subsystem address spaces
 - ▶ DB2 Stored Procedure address spaces
- These resources have an impact on how DB2 applications perform
- Workload Manager is used to manage DB2 workflow
 - ▶ DB2 Distributed and DB2 Stored Procedure workload
 - ▶ Priority and performance of allied address spaces (CICS, IMS, batch, TSO) that call DB2
- Important to have a basic understanding of WLM



Workload Manager

Terminology

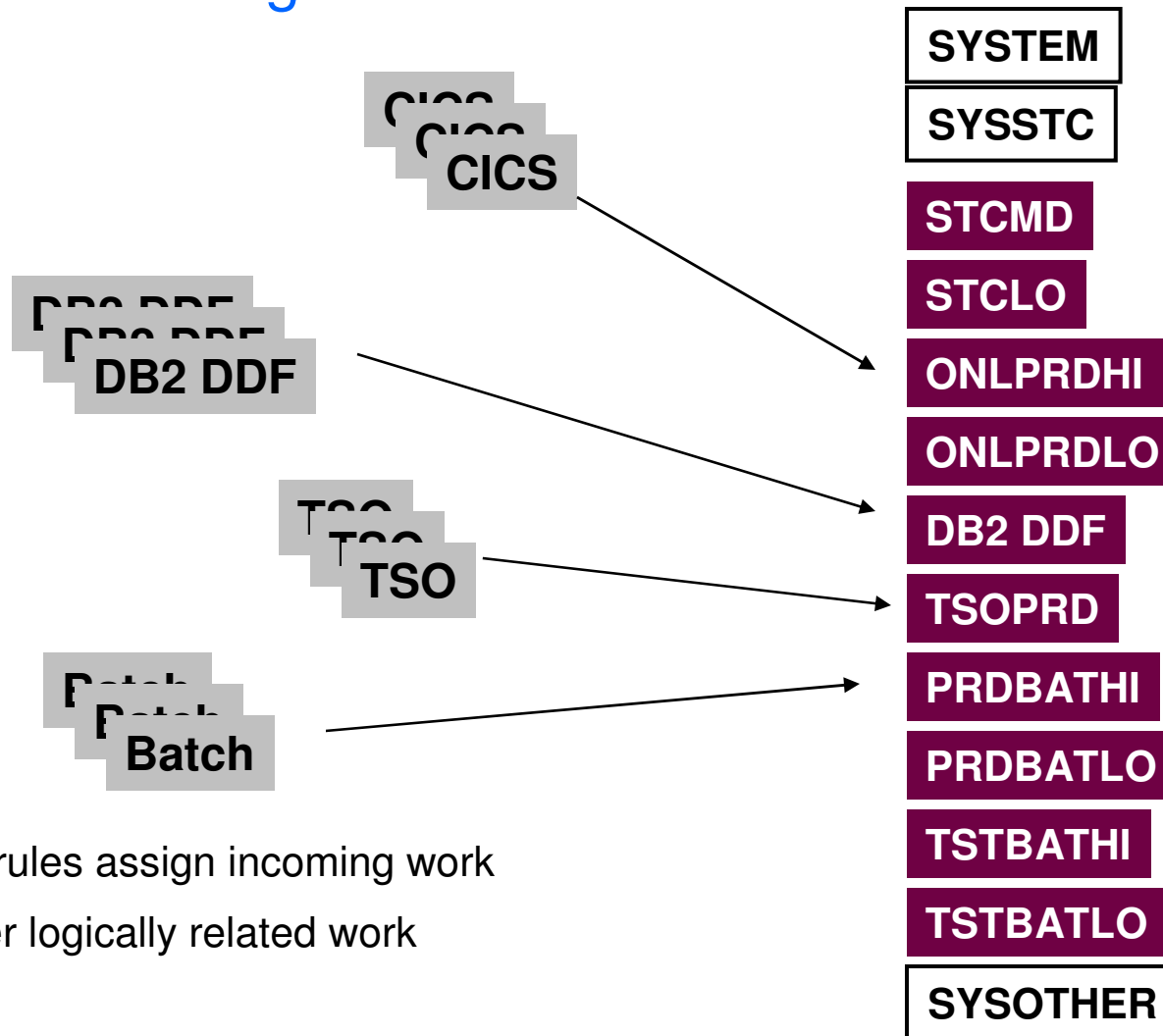
- A SERVICE DEFINITION consists of one or more SERVICE POLICIES
- A service policy contains several WORKLOADS
- Each workload consists of one or more SERVICE CLASSES
- Each service class has at least one PERIOD and each period has one GOAL
- There are five types of goals
 - ▶ System, Average Response Time, % Response Time, Execution Velocity, Discretionary
- Address spaces and transactions are assigned to service classes by CLASSIFICATION RULES

SERVICE DEFINITION
SERVICE POLICIES
WORKLOADS
SERVICE CLASSES
PERIOD
GOAL
CLASSIFICATION RULES



WLM

Service Classes Categorize Workload

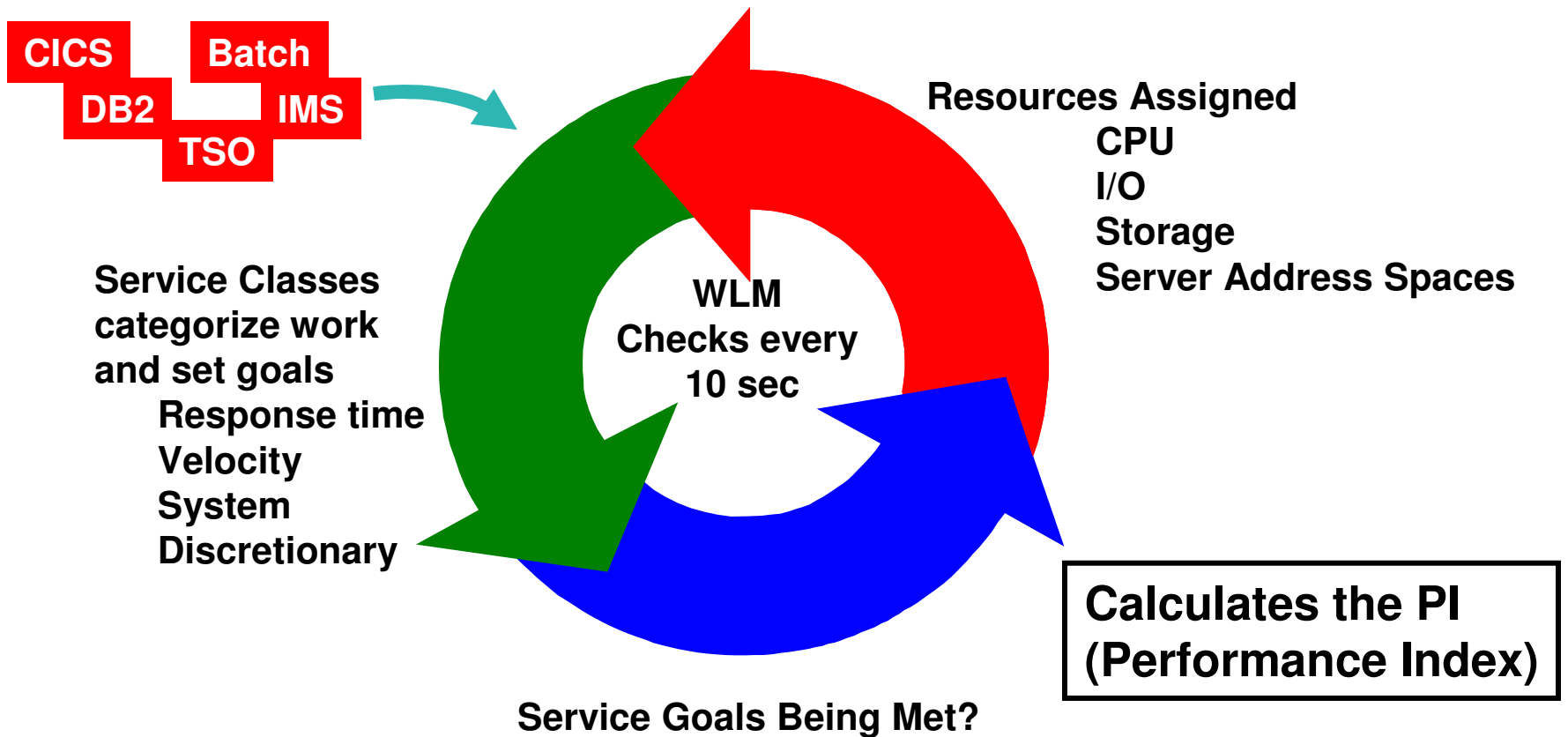


- Classification rules assign incoming work
- Group together logically related work



Workload Manager Service Classes And Goals

- Z/OS resources assigned based upon goals defined in WLM



Key WLM Constructs

- WLM is built upon two key constructs
 - ▶ DEFINITION - WLM provides mechanisms to categorize, prioritize, and manage workload
 - These are the service definitions managed by the WLM dialogs
 - ▶ FEEDBACK - The components managed by WLM provide information (samples) to help WLM determine how well it is doing
 - These are feedback mechanisms provided by each component, subsystem, and operating system



Understanding WLM Goals

The Performance Index

- Service Class periods are compared by calculating a Performance Index (PI) for each
- PI gives WLM a common way to track how well the work is doing regardless of goal type
- Importance parameter
 - ▶ Defined as part of the Service Class - 1 (high) to 5 (low)
 - ▶ Assigned to a Service Class Period
 - ▶ A way to prioritize critical goals
 - ▶ For work at the same importance level, WLM attempts to equalize the PIs

- | | |
|--------------------------------|---|
| • The PI equals 1 | -> The work in the period is meeting its goal exactly |
| • The PI is less than 1 | -> The work is doing better than its goal |
| • The PI is more than 1 | -> The work is missing its goal |



The Importance Of Importance

- Importance parameter
 - ▶ A way to prioritize critical goals
- Not analogous to Dispatching Priority
 - ▶ What is the importance of achieving the goal?
- WLM attempts to meet importance 1 goals first, and so on.....
- Helps WLM determine donors and receivers of resources
 - ▶ Donors – workload that can give up resource
 - ▶ Receivers – workload that needs resource



Types Of WLM Goals

Velocity Goals

- The percentage of time workload is ready and able to run, and is not delayed for lack of resources
 - ▶ Example - Velocity of 50 means that 50% of the time resources should be available for work to run
- Measure of acceptable delay
- Based on samples
- High velocity goals (example 90) are unreasonable
- Use high velocity goals for DBM1 and SSAS
 - ▶ Example - set around 60



Velocity Goal Challenges

- At first glance easy to set
- Actually requires more analysis
- Systems, environments, and workloads will change over time
 - ▶ Operating system
 - ▶ Memory, CPU, DASD
 - ▶ Size and nature of workloads
- Velocity goals require regular review
 - ▶ As systems change, velocity goals may need 'fine-tuning'



Types Of WLM Goals

Response Time Goals

- Average response time
 - ▶ Average response time for a given set of transactions
 - ▶ Include queue time and execution time
- Percentile response time
 - ▶ Percentile of transactions that need to complete within a desired response time
 - ▶ Reduces the impact of 'outliers'
- Rule of thumb
 - ▶ Work should have at least 10 completions in a 20 minute time frame to have adequate samples
- Consider Response time goals where possible for DB2 workloads
 - ▶ DDF requests, even batch jobs



Special Service Classes SYSTEM, SYSSTC, & SYSOTHER

- SYSTEM
 - ▶ For selected high priority system address spaces
 - ▶ Get highest CPU and I/O dispatching priority in system
- SYSSTC
 - ▶ For selected high priority started tasks and workload
 - ▶ Second highest priority behind SYSTEM
 - ▶ Place very high importance workload items here
 - DB2 Example - place IRLM here
- SYSOTHER
 - ▶ Unclassified work falls here
 - ▶ Bottom of the resource food chain



Specifying The WLM Objectives

```

File Utilities Notes Options Help
-----
Functionality LEVEL011          Definition Menu          WLM Appl LEVEL013
Command ==> _____

Definition data set . . . : none

Definition name . . . . . DEMO      (Required)
Description . . . . . No overrides

Select one of the
following options. . . . . ___  1.  Policies
                                2.  Workloads
                                3.  Resource Groups
                                4.  Service Classes
                                5.  Classification Groups
                                6.  Classification Rules
                                7.  Report Classes
                                8.  Service Coefficients/Options
                                9.  Application Environments
                               10.  Scheduling Environments
    
```

All the various WLM constructs on z/OS are defined using the WLM ISPF dialogs

- WLM provides an ISPF interface to define and manage the WLM Service Definition

WLM Service Class Definitions

```

Service-Class  View  Notes  Options  Help
-----
-----
1 to 14 of 27
Command ==>>

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

Action  Class      Description      Workload
-----  -
___    BATHI      ExVel (70) Imp (2)      BATCH
___    BATLOW     ExVel (40) Imp (2,Disc)  BATCH
___    BATMED     ExVel (40) Imp (2,3,4)  BATCH
___    CICSHI     ExVel (70) Imp (2)      CICS
___    CICSLow    ExVel (60) Imp (3)      CICS
___    CICSMED    ExVel (70) Imp (2)      CICS
___    DB2HI      ExVel (80) Imp (2)      DB2
___    DB2LOW     ExVel (60) Imp (3)      DB2
___    DB2MED     ExVel (60) Imp (2)      DB2
___    DDFHI      ExVel (80) Imp (2)      DB2
___    DDFLOW     ExVel (30) Imp (3)      DB2
___    DDFMED     ExVel (60) Imp (2)      DB2

```

- WLM Service Class definitions
 - ▶ Define the work to z/OS
 - ▶ Specify the goals of the workload
 - ▶ Specify the relative importance of the workload to the system



Workload Manager As A DB2 Priority Mechanism

```

Subsystem-Type  Xref  Notes  Options  Help
-----
                Create Rules for the Subsystem Type          Row 1 to 5 of 5

Subsystem Type . . . . . DDF      (Required)
Description . . . . . Example DB2
Fold qualifier names? . . . . Y  (Y or N)

Enter one or more action codes: A=After  B=Before  C=Copy  D=Delete
M=Move  I=Insert rule  IS=Insert Sub-rule  R=Repeat

Action      -----Qualifier-----          -----Class-----
Type      Name      Start          Service      Report
-----
          DB2A          _____  PRDBATCH
          ONLINE _____  PRDBATCH
          SYSADM _____  PRDONLIN
          QMFOS2 _____  PRDONLIN
          DB2B          _____  TSTQUERY
          _____  TESTUSER
    
```

Thread Attributes

- AI (Accounting Information)
- CI (Correlation Information)
- CN (Collection Name)
- CT (Collection Type)
- LU (LU Name)
- NET (Net ID)
- PK (Package Name)
- PN (Plan Name)
- SI (Subsystem Instance)
- UI (Userid)
- Procedure Name – Be aware of limitations

Thread attributes in WLM allow for considerable granularity in the classification of DB2 workloads



DB2 Workload Priority

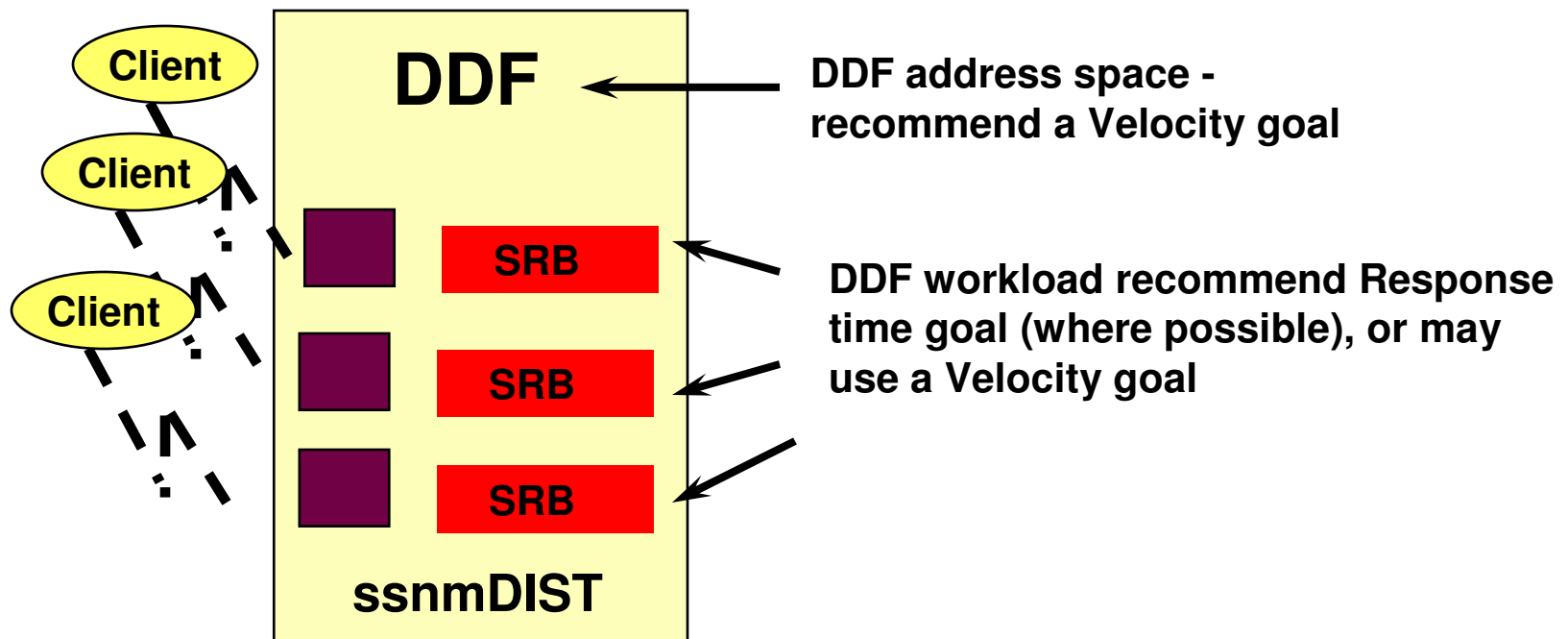
How Does WLM Assign Priority To DB2 Workload?

- The priority of the DB2 workload will vary depending upon the origin of the workload
- DB2 workload originating from a local application (IMS, CICS, TSO, Batch)
 - ▶ Priority is inherited from the invoking application
 - ▶ This applies to Stored Procedures invoked locally
- DB2 Distributed requests (DDF)
 - ▶ Priority controlled by DDF Service Class definitions
- DB2 Stored Procedure request via DDF
 - ▶ Priority controlled by Service Class definitions



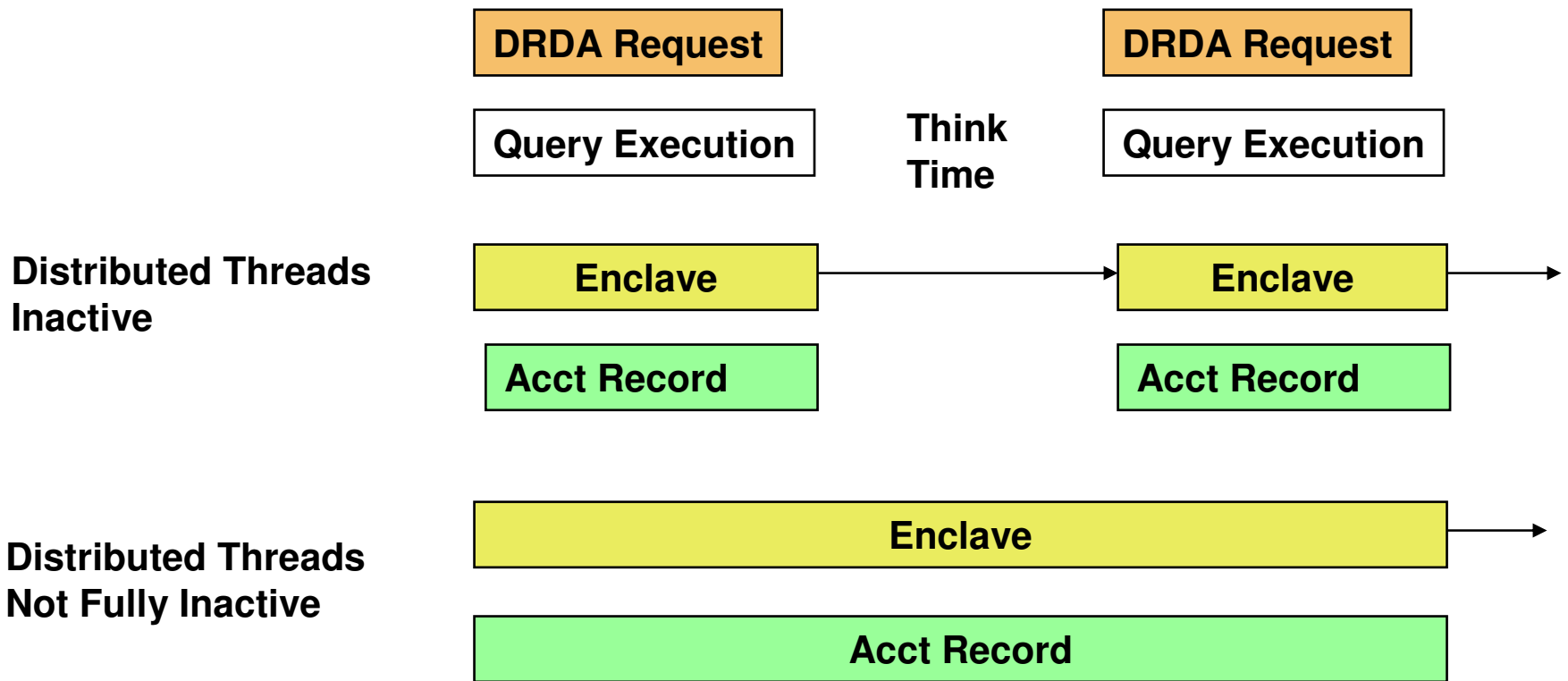
DDF Goal Considerations

- DDF workload may call for a combination of Velocity and Response time goals
 - ▶ DDF address space versus DDF workload



DDF WLM Goal Considerations

- DB2 Distributed thread options control enclave creation and how DB2 accounting records get created
 - ▶ Impacted by KEEP DYNAMIC options, cursor with hold, and zparm settings
 - ▶ This impacts whether to use velocity or response time goals



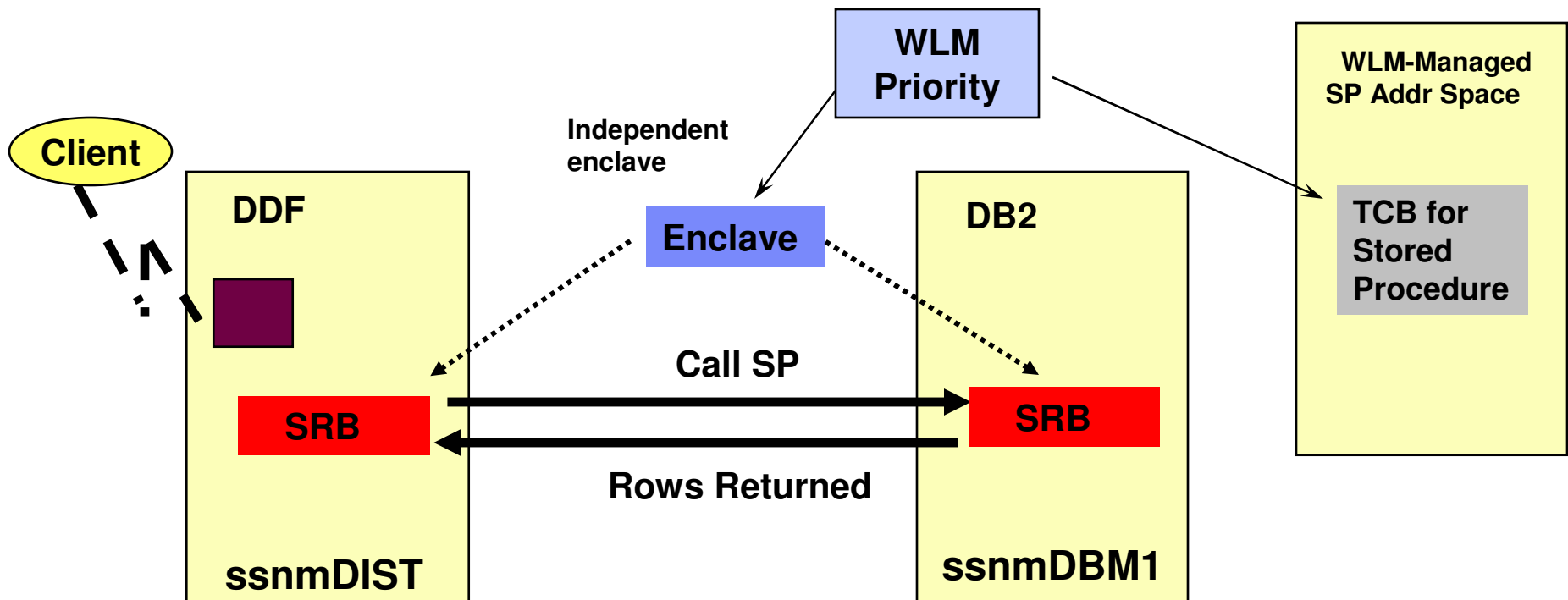
DDF WLM Goal Considerations

- THREADS=INACTIVE and DBAT is pooled (Connection inactive)
 - ▶ DDF creates one enclave per active interval
 - ▶ Response times do not include user think time
 - ▶ Response time goals and multiple periods can be used
- THREADS=ACTIVE
 - ▶ DDF creates one enclave for the life of the thread
 - ▶ Enclave response time includes user think time
 - ▶ Response time goals should not be used
 - ▶ Multiple periods should not be used



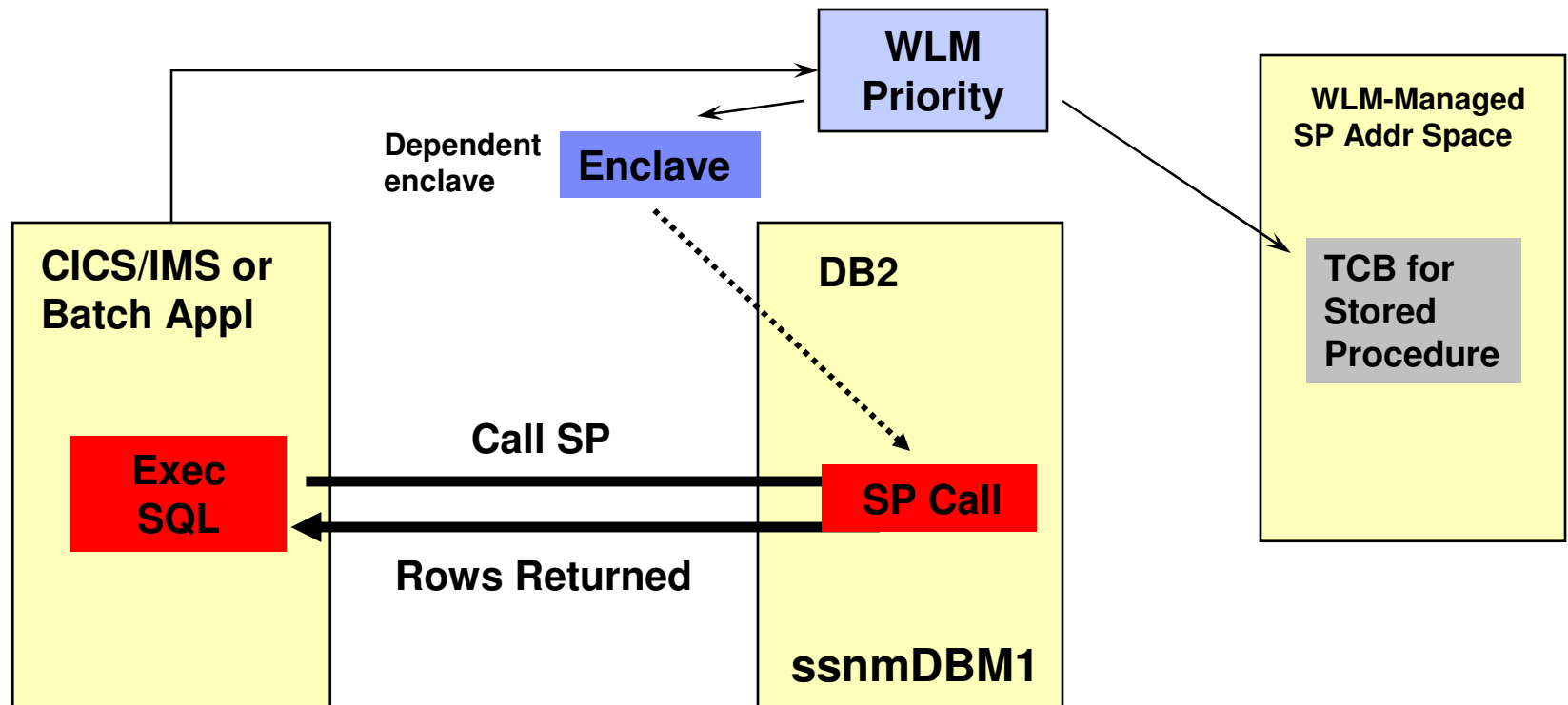
DDF Stored Procedure Priority

- When a Stored Procedure is called from DDF thread
 - DB2 references the enclave created for the DDF request for Stored Procedure
 - Stored Procedure priority is the priority of the DDF request



Stored Procedure Priority Called From A Local Application

- When a Stored Procedure is called from an application on z/OS or OS/390
 - ▶ DB2 creates an enclave for use by the Stored procedure
 - ▶ Stored Procedure priority is the priority of the calling application address space



Service Classes And Thread Priorities

```

_____ ZENCLD  VTM    O2      V310./C D81G 02/22/06  7:13:42  2
> Help PF1                                           Back PF3

>          THREAD INFORMATION:  Enter a selection letter on the top line.

> A-THREAD DETAIL B-LOCK COUNTS C-LOCK WAITS      D-LOCKS OWNED  E-GLOBAL LOCKS
> F-CURRENT SQL   G-SQL COUNTS  H-DISTRIBUTED     I-BUFFER POOL  J-GROUP BP
> K-PACKAGES      L-RES LIMIT   M-PARALLEL TASKS N-UTILITY    O-OBJECTS
> P-CANCEL THREAD Q-DB2 CONSOLE R-DSN ACTIVITY   S-APPL TRACE  *-ENCLAVE
=====
                                ENCLAVE DETAIL INFORMATION
PLAN
+ Thread:  Plan=DEMO1           Connid=DB2CALL  Corrid=DEMOENCL  Authid=CXE12AUR
+ Attach:  BATCH                JOB Name=DEMOENCL  JOB Asid= 49
+ Package: DEMO1                Collection=
enc
+ ENCLAVE TOKEN:  2000000001    Enclave Type:    Dependent
+ Owing System:   SP12           Owing Job:       DEMOENCL
+ WLM Mode:       Goal           Enclave CPU Time 00:00:00.000
+
+ SERVICE PERIOD INFORMATION
+ Period(s) for Service Class BATCH:  2
+ Current Period for This Thread:     1
+ Performance Index This Period:      .20
    
```

Service Class

Period

PI

- Depending upon how an enclave is created (local allied address space or via DDF) controls what service class, etc that is assigned to a given thread

DB2 DDF Considerations

Things To Note

- Look for overly simplistic Service Class definitions
 - ▶ Example – type DDF and nothing more than DB2 subsystem name
 - ▶ This does little to exploit the ability of WLM to prioritize DB2 workloads
 - ▶ Some workloads will inherently be more important than others
- Look for Distributed workloads that run longer than expected but use little resource
 - ▶ Indicative of workload that drops into less than optimal Service Classes – example SYSOTHER



DB2 Velocity Goals Setting Optimal Goals

- Use Velocity goals for always running and long running work
- Velocity goals at first glance seem easy to set
 - ▶ Require more ongoing review
 - ▶ Should be validated as the operating environment changes – changes to operating system, hardware, and workload
- If using Velocity goals make them realistic
- Review Velocity goals regularly



DB2 Subsystem Address Spaces Using Velocity Goals

- Use a high Velocity goal for
 - ▶ DB2 address spaces (SSAS and DBM1)
 - ▶ CICS and IMS regions (if not using response time goals)
- Use the SYSSTC service class
 - ▶ Typical for VTAM & DB2 IRLM address space
 - ▶ Suggested for high performance/always running work
- Use a Velocity goal for the DB2 DDF address space
 - ▶ DDF address space has internal tasks that govern thread creation that should have high performance goal



DB2 Response Time Goals Setting Optimal Goals

- Use Response Time goals when possible
 - ▶ Less need for ongoing maintenance and review
 - ▶ WLM will manage resources dynamically to achieve goals
- Response Time goals work well for certain types of DB2 workloads
 - ▶ DB2 Distributed workloads in e-business and WebSphere transactional type workloads
 - ▶ Transactional type workloads in general including distributed workloads that invoke Stored Procedures
 - ▶ Repetitive workloads that have multiple events for WLM to measure and manage



General DB2 & WLM Recommendations

- Define realistic goals
- Understand the difference between Velocity and Response Time goals
- Use Velocity goals for always running work
 - ▶ Example - DB2 subsystem address spaces
- Review Velocity goals regularly
- Use Response time goals when possible
- Watch for workload going to SYSOTHER
 - ▶ Indication of unclassified work - low priority



WLM

General Recommendations

- KISS method (Keep It Simple Stupid)
 - ▶ Avoid overly complex implementations
 - ▶ Avoid overly simplistic standards
 - Example – a service class and/or application environment for each application
 - ▶ Too many Application Environments and Service Classes can result in an over abundance of server address spaces
 - ▶ Increases number of queues that WLM must manage
- Avoid too many service classes
 - ▶ WLM analyzes service classes in a round-robin manner
 - ▶ Too many and WLM is unable to manage them all effectively



About Intelligent Resource Director

- LPAR Weight management
 - ▶ The “weight” of an LPAR may be moved across LPARs
 - ▶ Manages the number of CPs for an LPAR
 - ▶ LPAR weight is part of WLM CPU delay analysis
- Dynamic Channel Path management
 - ▶ Lets WLM move channel paths from one I/O control unit to another
- Channel Subsystem I/O Priority Queuing
 - ▶ Allows WLM to assign a priority to an I/O request
 - ▶ Channel subsystem may use a priority managed queue as opposed to FIFO queue
 - Complementary to other I/O queuing mechanisms in the I/O subsystem

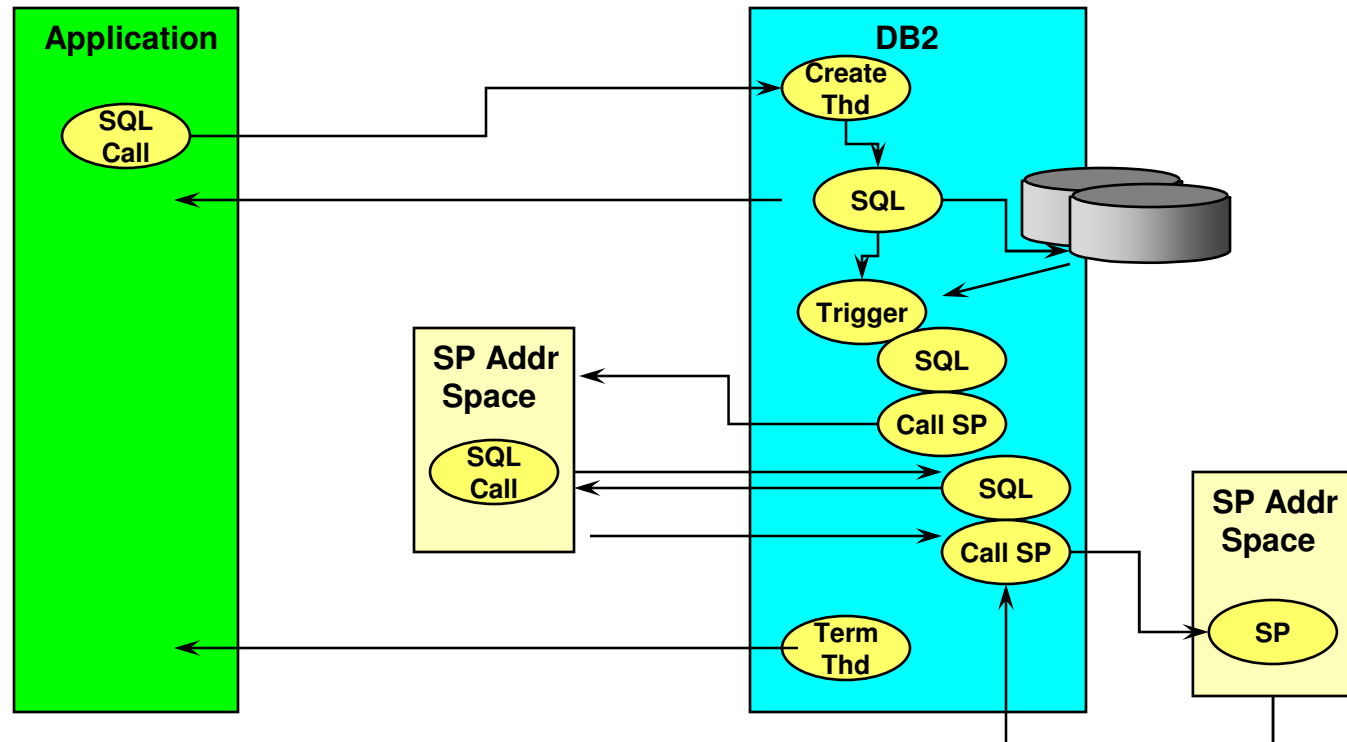


WLM Contention Management Enhancements

- WLM Contention Management helps addressing chronic or long lasting contention situations
 - ▶ WLM provides interfaces to allow resource managers (for example – DB2) to signal contention situations
 - ▶ WLM has had the ability to promote (increase the DP) for a short duration to resolve the issue
- DB2 example scenario
 - ▶ Lock/latch contention in DB2 may impact performance
 - ▶ Often contention may be resolved with a short boost of resource
 - ▶ DB2 may notify WLM if a contention occurs
 - ▶ WLM may optionally raise the priority for the holder to complete the work
- In z/OS 1.10 WLM can promote units of work for longer periods of time, and promote them to the priority of the highest-priority units of work waiting for a resource they are holding.



WLM Enhancement For Nested Stored Procedure Requests



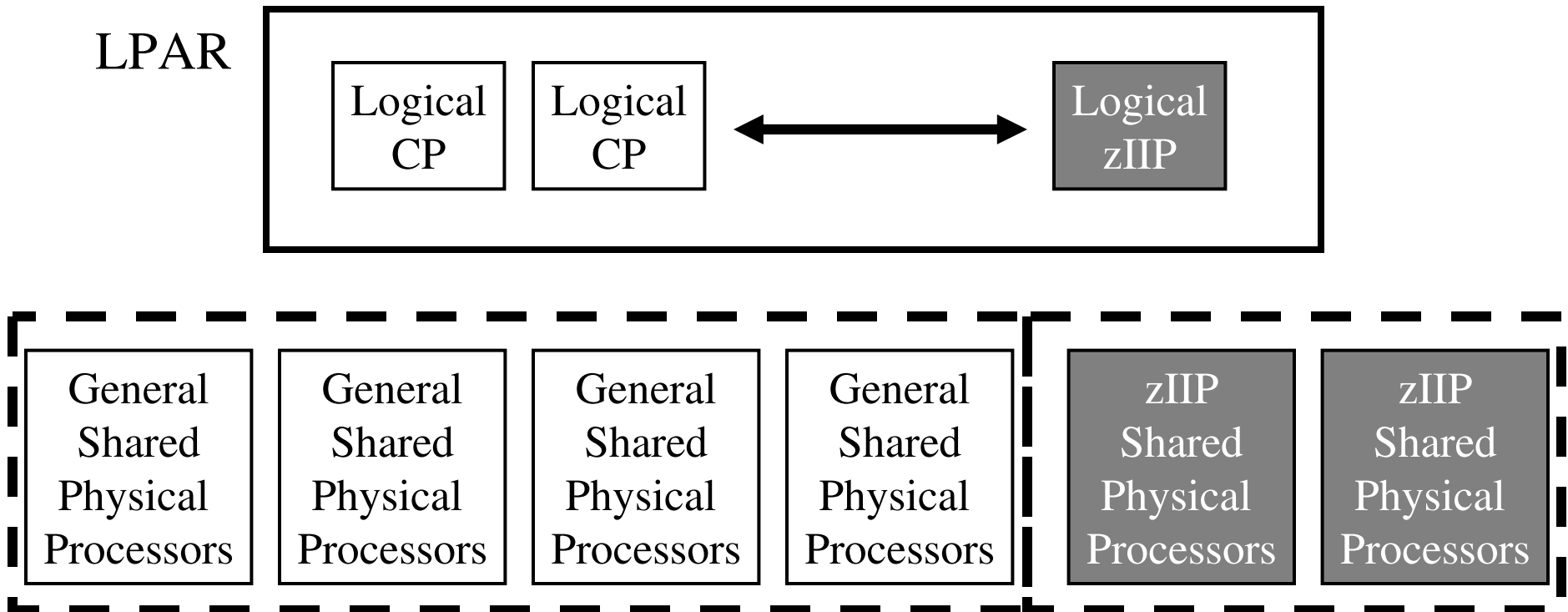
- Triggers, Stored Procedures, and UDFs actions may be nested, sometimes multiple layers of nesting
- WLM has had the ability to promote workload to resolve an issue

WLM Stored Procedure Enhancements

- The original assumption
 - ▶ All work requests inserted by DB2 (example – Stored Procedures) were independent requests
- The reality
 - ▶ Procedures may recursively call other procedures
 - ▶ The processing may be inter-dependent
- The new logic
 - ▶ DB2 tells WLM about dependent stored procedure requests
 - ▶ WLM gives dependent requests priority
 - ▶ WLM may start server regions more aggressively, if needed



About zIIP Processors



- A specialty processor for offloading certain types of work
- DB2 for z/OS V8 was one of the first IBM exploiters of the zIIP

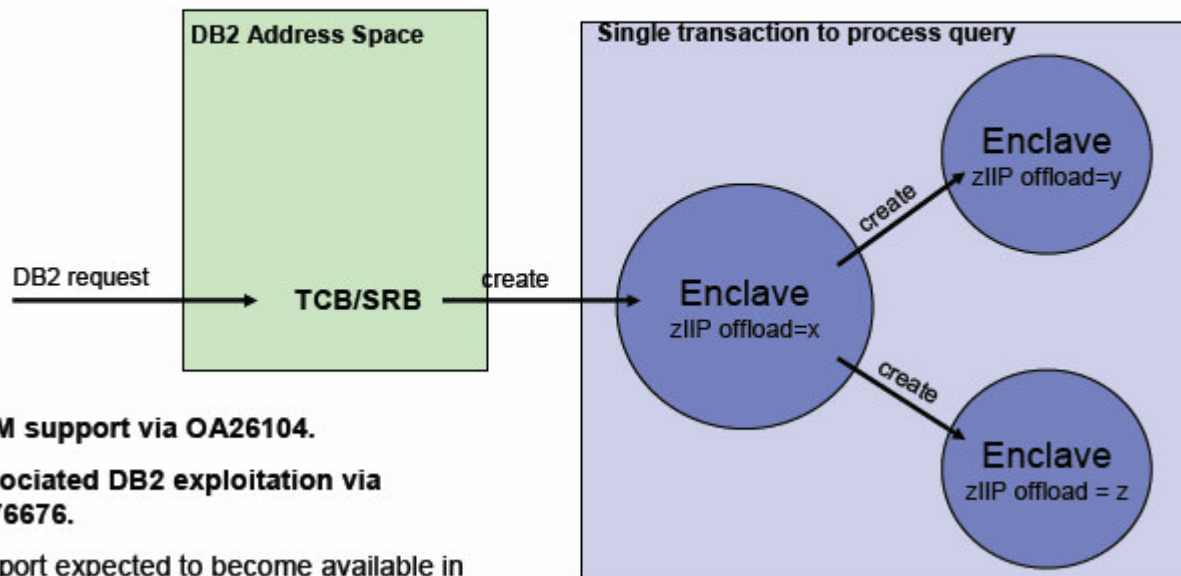
More About zIIP Processors

- Work on z/OS may have all or a portion of its resource usage on an enclave Service Request Block
 - ▶ Enclave SRB work may be directed to the zIIP
- Certain types of DB2 work may take advantage of zIIP
 - ▶ DRDA - Queries that access DB2 for z/OS V8/V9 via DRDA over a TCP/IP connection are dispatched within z/OS in enclave SRBs. z/OS directs a portion of this work to the zIIP
 - ▶ Complex parallel queries
 - ▶ DB2 utilities for index maintenance
 - LOAD, REORG, and REBUILD
- WLM management of zIIP is similar to zAAP processors
 - ▶ z/OS 1.10 provides more WLM zIIP management



About Work-dependent Enclaves

- Extension to an independent, dependent, or other work-dependent enclave
 - ▶ Extends the transaction the creating enclave.
 - ▶ Runs like an independent enclave when created by non-enclave work
- Allows control of zIIP offload by entitled products.



WLM support via OA26104.
Associated DB2 exploitation via PK76676.
 Support expected to become available in 1H09



WLM Managed BP

- DB2 V9 will provide a WLM-managed buffer pool management capability
 - ▶ AUTOSIZE(YES) option
- The AUTOSIZE(YES) option of the ALTER BUFFERPOOL command allows dynamic buffer pool size adjustments based on real time workload monitoring
- Potential benefits
 - ▶ Frees the storage for use by mission-critical subsystems on the same LPAR
 - ▶ DB2 and WLM will "fine tune" the buffer pool size, based on long term trends and steady state growth
- Currently disabled
 - ▶ PK75626 will eventually re-enable function when available



Summary

Workload Manager As The Priority Manager Of DB2

- Workload Manager (WLM) is the priority and resource manager for z/OS and DB2
- WLM manages critical DB2 address spaces
- These resources have an impact on how DB2 applications perform
- Workload Manager is used to manage DB2 workflow
- Important to have a basic understanding of WLM
- WLM is constantly being enhanced to provide new features and functions

