IBM Chat with Lab for Greater China Group

- **Host:** Frank Ning, Manager, DB2 LUW Install and Up/Running Development

- **Executive introduction**

  Sal Vella, Vice President, Development, Distributed Data Servers and Data Warehousing

- **Presentation: Best Practices for Workload Management Using IBM Optim Performance Manager**

  Xiaomei Wang, Senior Program Manager, IBM Toronto Lab Data Warehouse Team

Slides are available at https://www.ibm.com/db2/labchats/gcg

November 9, 2010

Executive Introduction

Sal Vella

Vice President, Development, Distributed Data Servers and Data Warehousing

IBM Software Group

Best Practices for Workload Management Using IBM Optim Performance Manager

- Chat with the Toronto Lab for the Greater China Group

Xiaomei Wang, Senior Program Manager
IBM Toronto Lab Data Warehouse Team

November 9, 2010

Agenda

- **Overview**

- **Step by step methodology**

- **Additional scenarios, special cases**

- **Summary**

Benefits of Workload Management

- **Protect data server from overload**
  - Large warehouses are vulnerable to thrashing w/o WLM
  - Set limits on how long a query can run
- **Monitor your data server**
  - Understand what is running
  - Diagnose and correct performance problems
- **Meet business objectives according to their priority**
  - High priority work addressed in the time required
  - Other work completed without compromising response time of high priority work
  - Restrict CPU resources for a specific line of business, group of users or application

Real life results of applying WLM Best Practices
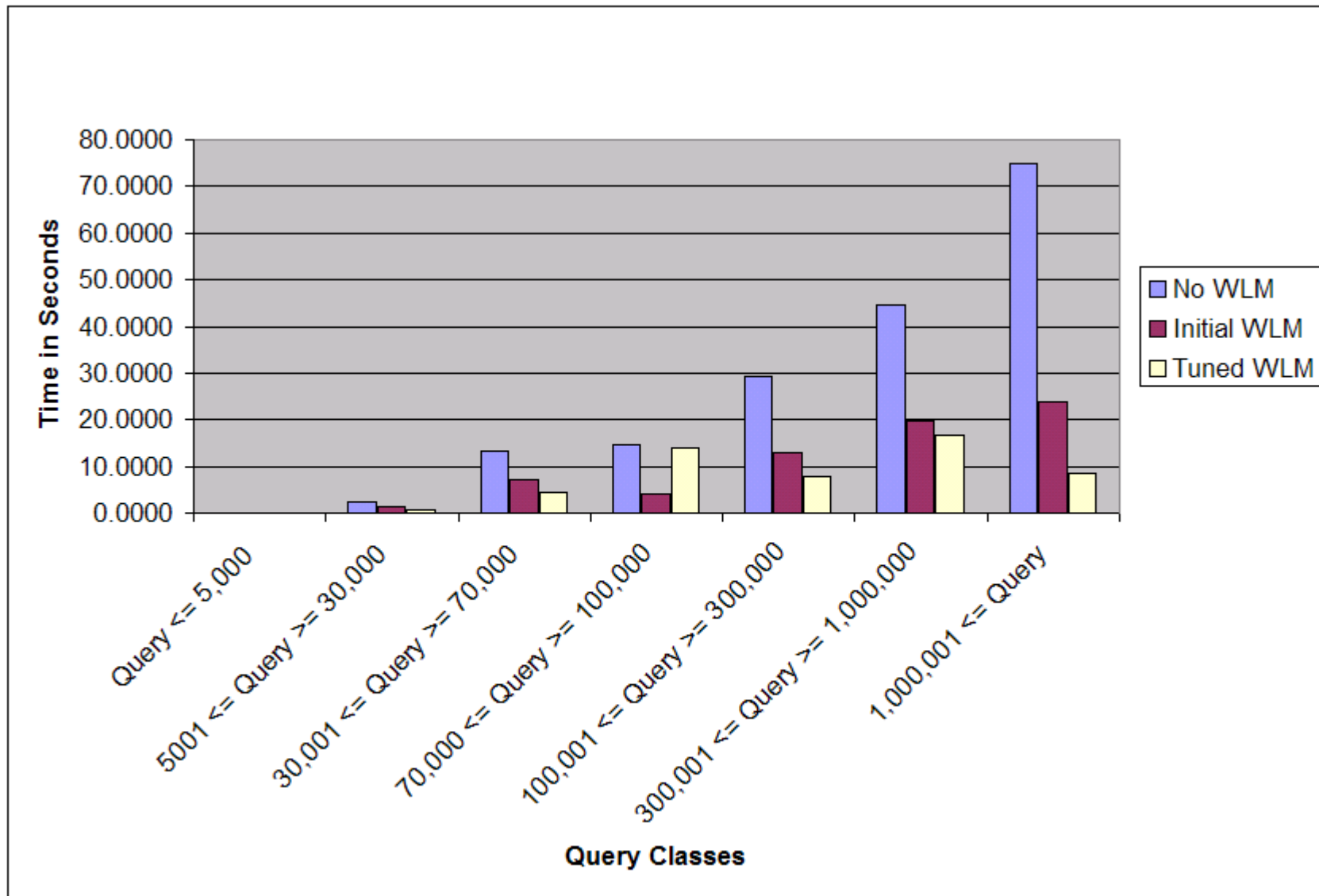
Best Practices Roadmap

- **Divide incoming work into categories**
  - Categorizing work is safe (no change in behavior)

- **Monitor to validate work is properly categorized**

- **Apply controls to categories**
  - Limit concurrency of complex queries
  - Monitor to validate configuration and troubleshoot performance
  - Impose limits to reduce wasted resources and enforce policies
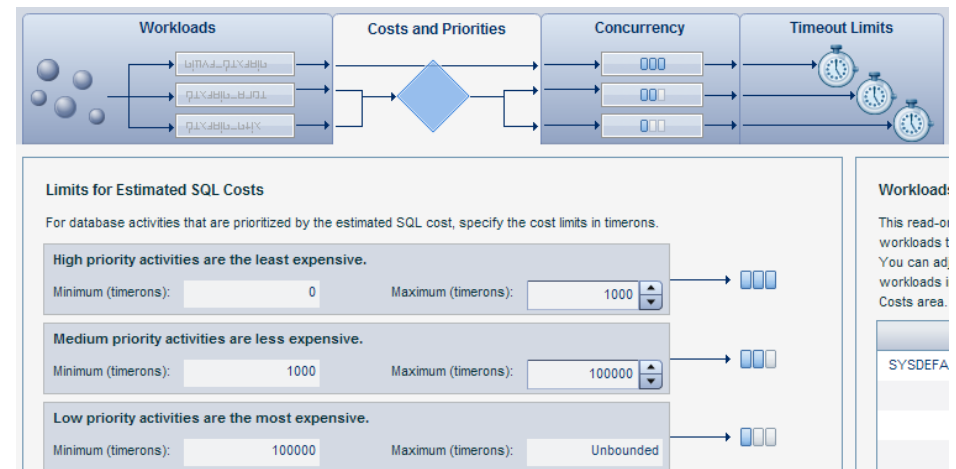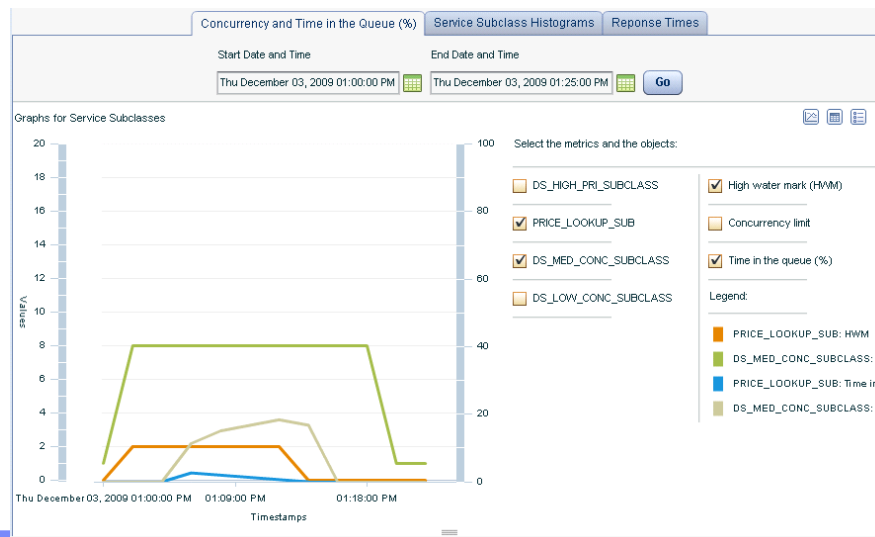 - Prioritize work to match business needs

Simplified Configuration via Web Based Tooling

- **Optim Performance Manager 4.1**
  - Integrated WLM monitoring and configuration
  - Choice of solution templates
  - Works for DB2 LUW v9.5 or v9.7

- **Admin Console in InfoSphere Warehouse v9.7**
  - Template configuration
 - Based on 3 service subclasses
  - Adjustable parameters

Categorizing Work – Available Criteria

- **Connection attributes**
  - User or group
  - Application
  - Tags inserted by middleware
  - Etc.
- **Type of activity**
  - DML
    - READ (SQL Queries)
    - WRITE (insert, update and delete)
  - DDL
  - LOAD
  - Etc.
- **Estimated cost (SQL queries only)**

Categorizing Work - Mechanisms

WLM uses a two step evaluation to categorize activities

1. Workloads categorize connections
2. Work action sets further categorize by activity type and/or cost
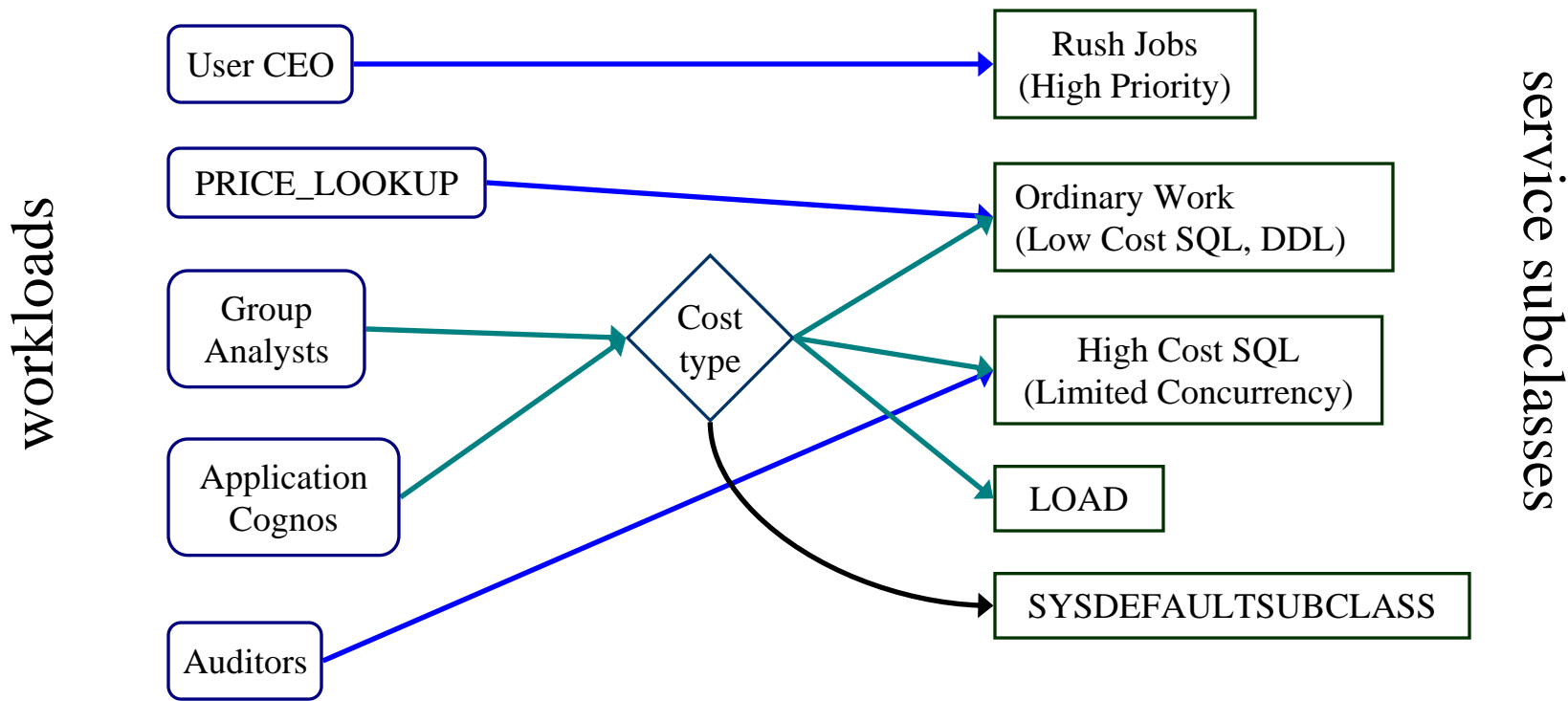
Step 1 – Define Workloads

- **Define a workload for each interesting source of work**
  - Application
    - Allows you to treat some applications differently
    - LOAD is not an application
  - Individual user, group or role
    - Treat activities differently based on **who** submitted them
  - DB2 client information fields
    - Some middleware can tag work by filling in client info fields
  - Combination of any of the above
    - Queries from SAS treated differently if submitted by analyst versus other users

Reasons to define a workload

- **Identify work to be controlled**
- **Monitor by workload**
- **Label activities for troubleshooting**
- **Categorize work (map to a service class)**
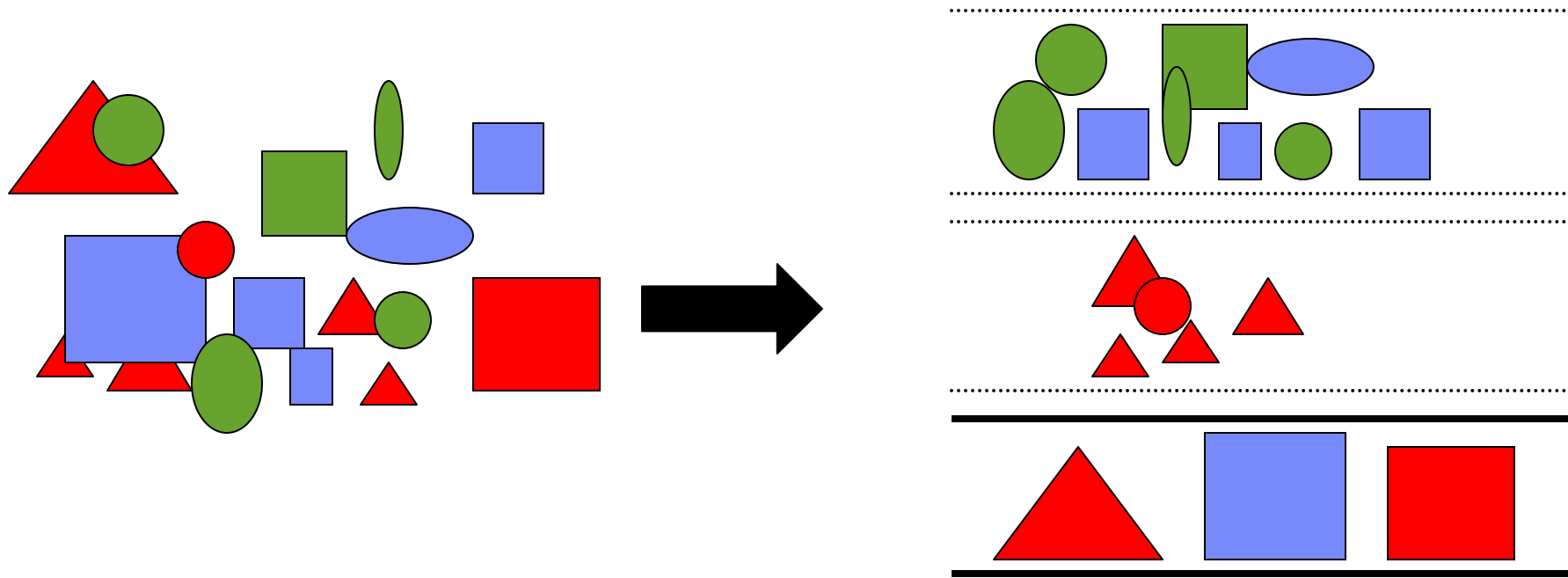
Discovering Connection Attributes

- **Optim Performance Manager (OPM) displays connection attributes**
- **View Current Activities in WLM configuration**

View Current Activities

Use the connection attribute values of the current activities to help you define the connection attributes of you

Workload Name	User ID	Application Name	Client Application Name	Group ID ▲	Client Accounting String	Client Workstation
SYSDEFAUI	DB2ADMIN	db2bp.exe		DB2ADMNS, DB	bi	
SYSDEFAUI	DB2ADMIN	db2jcc_application		DB2ADMNS, DB		lava
SYSDEFAUI	DB2ADMIN	db2jcc_application	DS_WLM_CONFIG	DB2ADMNS, DB		localhost
SYSDEFAUI	DB2ADMIN	db2bp.exe		DB2ADMNS, DB	oltp	

Connection Attributes via Extended Insight

- **Extended Insight clusters activities by connection attributes**
- **Browse history of activities**

| | Activate... | Deactivate... | New... | Edit... | Copy... | Reset | Delete |

Workload Cluster Group/Workload Cluster	Average End-to-End Response	Maximum Inflight Elapsed Time	Maximum End-to-End Response	A g D S ..
▼ ◆ WLMTEST4	0.015	0.015	0.485	◄
▼ ◆ Client user IDs	0.015	0.015	0.485	◄
◆ robin	0.016	0	0.485	◄
◆ barney	0.015	0.015	0.109	◄
◆ marshal	0.015	0	0.047	◄
◆ ted	0.015	0	0.078	◄
◆ lily	0.015	0	0.032	◄
▼ ◆ Client application names	0.015	0.015	0.485	◄
◆ engineering	0.015	0	0.485	◄
◆ payroll	0.015	0	0.109	◄
◆ accounting	0.015	0	0.328	◄
◆ r&d	0.015	0.015	0.109	◄
▼ ◆ Host names/IP addresses	0.015	0.015	0.485	◄
◆ 9.30.64.155	0.015	0.015	0.485	◄
▼ ◆ Application Types	0.015	0.015	0.485	◄

Connection Attributes via WLM Table Functions

- ```
  SELECT workload_name, application_name, …
  FROM table(wlm_get_service_class_workload_occurrences('', '',
  -2));
  ```

- **As shown, displays connections attributes for all current connections**

- **Filter as necessary to find what you are looking for**

Real Life Example: Cognos

Set properties - EAPPS

General   **Connection**   Permissions

Specify the parameters for the child connections of this data source.

Commands:
Specify the commands that the database executes when certain events occur.

Entries: 1   –   4

	Name	Value		Delete child values
☐	Open connection commands	(None)	Set...	☐
☐	Open session commands	(None)	Set...	☐
☐	Close session commands	(None)	Set...	☐
☐	Close connection commands	(None)	Set...	☐

Clear

```
<commandBlock>
  <commands>
    <sqlCommand>
      <sql> CALL SYSPROC.WLM_SET_CLIENT_INFO(
        #$account.personalInfo.userName#, 'MachineName',
        #$account.parameters.var1#, 'ApplicationName',
  'AUTOMATIC')
      </sql>
    </sqlCommand>
  </commands>
</commandBlock>
```

Step 2 - Create a Service Superclass

- **A service superclass is a prerequisite for:**
  - A work action set to map activities by type and estimated cost
  - Service subclasses

- **One service superclass is sufficient for basic use cases**

- **Redirect user defined workloads to new service superclass**

- **SQL syntax**
  - CREATE SERVICE CLASS main_super;
  - ALTER WORKLOAD cognos_wl SERVICE CLASS main_super;

Step 3 – Create Service Subclasses

Rush Jobs
(High Priority)

Ordinary Work
(Low Cost SQL, DDL)

High Cost SQL
(Limited
Concurrency)

LOAD

- **High priority, rush jobs**
  - Predefine a place for work that needs to run **now** at high priority
- **Ordinary work, low or medium cost SQL**
  - This includes the bulk of work in a warehouse
  - Default priority, no concurrency limits
- **Long running queries, high cost SQL**
  - Limited concurrency
- **LOAD activities**
  - Limit number of concurrent LOAD activities

- **Tip: collect at least base monitoring data COLLECT AGGREGATE ACTIVITY DATA BASE**

Service Subclasses in OPM Template

- **Service subclasses created by WLM configuration in OPM**

Service Subclass	Min. Cost (timerons)	Max. Cost (timerons)	Agent Priority	Concurrency Limit	
DS_HIGH_PRI_SUBCLASS	Not applicable	Not applicable	Default ▼	Unlimited	
DS_MED_CONC_SUBCLASS	0	100000 ▲▼	Default ▼	90 ▲▼	
DS_LOW_CONC_SUBCLASS	100000	Unbounded	Default ▼	8 ▲▼	
DS_LOAD_SUBCLASS	Not applicable	Not applicable	Default ▼	2 ▲▼	

Step 4 – Create a Work Action Set

- **Map activities to subclasses based on type or cost**
- **Anything not mapped falls into SYSDEFAULTSUBCLASS**
- **Evaluation order matters**
- **Template mappings:**
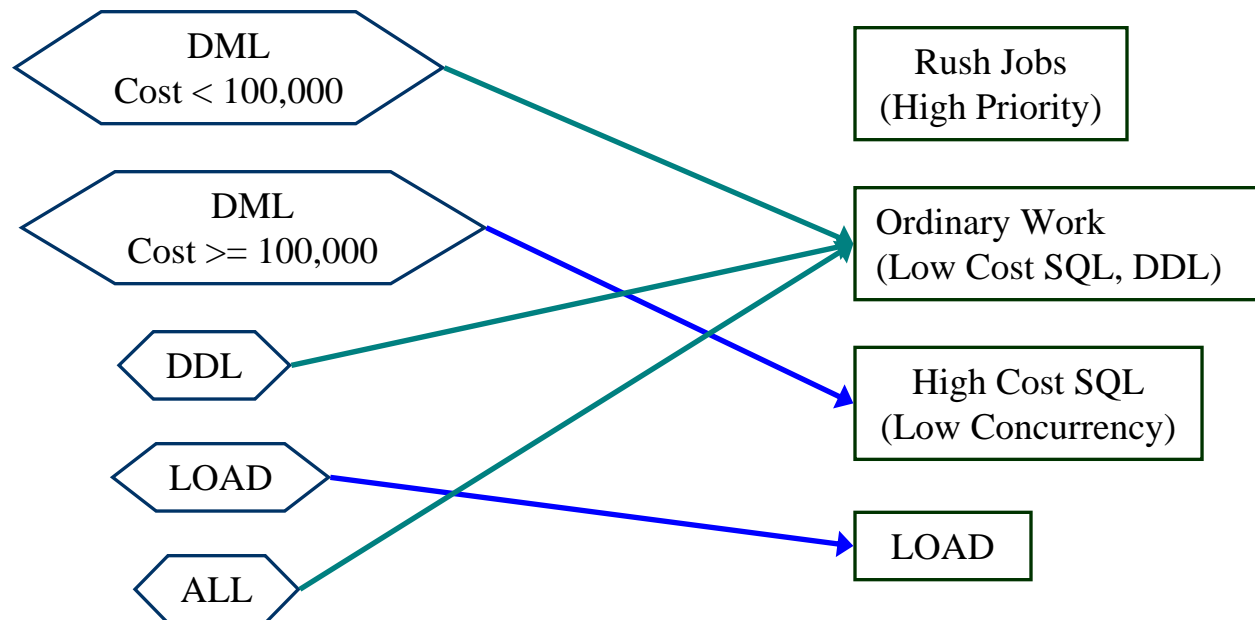
Template DDL for Work Class Set + Work Action Set

- CREATE WORK CLASS SET "DS_AUTOMGMTSU_WORK_CLASS_SET" (
  WORK CLASS "DS_LOW_COST_DML_WC" WORK TYPE DML
   FOR TIMERONCOST FROM 0.0 TO 100000.0 POSITION AT 1,
  WORK CLASS "DS_HIGH_COST_DML_WC" WORK TYPE DML
   FOR TIMERONCOST FROM 100000.0 TO UNBOUNDED POSITION AT 2,
  WORK CLASS "DS_DDL_WC" WORK TYPE DDL POSITION AT 3,
  WORK CLASS "DS_LOAD_WC" WORK TYPE LOAD POSITION AT 4,
  WORK CLASS "DS_OTHER_WC" WORK TYPE ALL POSITION AT 5);

- CREATE WORK ACTION SET "DS_AUTOMGMTSU_WORK_ACTION_SET"
    FOR SERVICE CLASS "DS_AUTO_MGMT_SUPER"
    USING WORK CLASS SET "DS_AUTOMGMTSU_WORK_CLASS_SET" (
    WORK ACTION "DS_MAP_LOW_COST_DML_WA" ON WORK CLASS
  "DS_LOW_COST_DML_WC"
      MAP ACTIVITY WITHOUT NESTED TO "DS_MED_CONC_SUBCLASS",
    WORK ACTION "DS_MAP_HIGH_COST_DML_WA" ON WORK CLASS
  "DS_HIGH_COST_DML_WC"
      MAP ACTIVITY WITHOUT NESTED TO "DS_LOW_CONC_SUBCLASS",
    WORK ACTION "DS_MAP_DDL_WA" ON WORK CLASS "DS_DDL_WC"
      MAP ACTIVITY WITHOUT NESTED TO "DS_MED_CONC_SUBCLASS",
    WORK ACTION "DS_MAP_LOAD_WA" ON WORK CLASS "DS_LOAD_WC"
      MAP ACTIVITY WITHOUT NESTED TO "DS_LOAD_SUBCLASS",
    WORK ACTION "DS_MAP_OTHER_WA" ON WORK CLASS "DS_OTHER_WC"
      MAP ACTIVITY WITHOUT NESTED TO "DS_LOW_CONC_SUBCLASS");

Use the Tooling to Save Work

SET WORKLOAD TO SYSDEFAULTADMWORKLOAD;
CREATE SERVICE CLASS "DS_AUTO_MGMT_SUPER";
CREATE SERVICE CLASS "DS_HIGH_PRI_SUBCLASS" UNDER "DS_AUTO_MGMT_SUPER" COLLECT AGGREGATE ACTIVITY
    DATA BASE;
CREATE SERVICE CLASS "DS_MED_CONC_SUBCLASS" UNDER "DS_AUTO_MGMT_SUPER" COLLECT AGGREGATE
    ACTIVITY DATA BASE;
CREATE SERVICE CLASS "DS_LOW_CONC_SUBCLASS" UNDER "DS_AUTO_MGMT_SUPER" COLLECT AGGREGATE
    ACTIVITY DATA BASE;
CREATE SERVICE CLASS "DS_LOAD_SUBCLASS" UNDER "DS_AUTO_MGMT_SUPER" COLLECT AGGREGATE ACTIVITY
    DATA BASE;
CREATE WORK CLASS SET "DS_AUTOMGMTSU_1263546069031_WORK_CLASS_SET" (WORK CLASS
    "DS_LOW_COST_DML_WC" WORK TYPE DML FOR TIMERONCOST FROM 0.0 TO 100000.0 POSITION AT 1, WORK CLASS
    "DS_HIGH_COST_DML_WC" WORK TYPE DML FOR TIMERONCOST FROM 100000.0 TO UNBOUNDED POSITION AT 2,
    WORK CLASS "DS_DDL_WC" WORK TYPE DDL POSITION AT 3, WORK CLASS "DS_LOAD_WC" WORK TYPE LOAD
    POSITION AT 4, WORK CLASS "DS_OTHER_WC" WORK TYPE ALL POSITION AT 5);

    ▪

    ▪

    ▪

ALTER WORKLOAD "SYSDEFAULTUSERWORKLOAD" SERVICE CLASS "DS_AUTO_MGMT_SUPER" COLLECT AGGREGATE
    ACTIVITY DATA BASE ;
CREATE WORK ACTION SET "DS_AUTOMGMTSU_1263546069031_WORK_ACTION_SET" FOR SERVICE CLASS
    "DS_AUTO_MGMT_SUPER" USING WORK CLASS SET "DS_AUTOMGMTSU_1263546069031_WORK_CLASS_SET" (
    WORK ACTION "DS_MAP_LOW_COST_DML_WA" ON WORK CLASS "DS_LOW_COST_DML_WC" MAP ACTIVITY
    WITHOUT NESTED TO "DS_MED_CONC_SUBCLASS", WORK ACTION "DS_MAP_HIGH_COST_DML_WA" ON WORK
    CLASS "DS_HIGH_COST_DML_WC" MAP ACTIVITY WITHOUT NESTED TO "DS_LOW_CONC_SUBCLASS", WORK
    ACTION "DS_MAP_DDL_WA" ON WORK CLASS "DS_DDL_WC" MAP ACTIVITY WITHOUT NESTED TO
    "DS_MED_CONC_SUBCLASS", WORK ACTION "DS_MAP_LOAD_WA" ON WORK CLASS "DS_LOAD_WC" MAP ACTIVITY
    WITHOUT NESTED TO "DS_LOAD_SUBCLASS", WORK ACTION "DS_MAP_OTHER_WA" ON WORK CLASS
    "DS_OTHER_WC" MAP ACTIVITY WITHOUT NESTED TO "DS_LOW_CONC_SUBCLASS") ;

No queries were harmed during the making of this movie.

- **No controls imposed yet**
  - All configuration to this point only *categorizes* work
  - These changes are safe in a production DB

- **Next steps**
  - Monitor to validate work is properly categorized

Working Iteratively

- **Configure WLM iteratively**
  – Categorize work
  – Monitor to validate categories
  – Apply controls
  – Monitor to validate controls

- **Make one change at a time**

- **Keep monitoring data for future comparison**
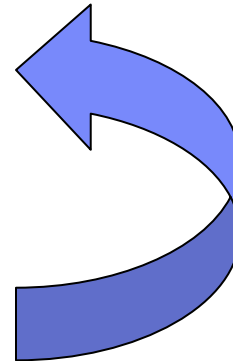
Step 5 - Baseline Monitoring

- **Create all WLM related event monitors**
  - Activity Event Monitor
    - Allows capture of details about activities in a workload or service class
  - Statistics Event Monitor
    - Captures histograms, counts and high water marks
  - Threshold Event Monitor
    - Allows capture or details about threshold violations
- **Choose an appropriate tablespace**
  - Spans all partitions
  - Suitable for heavy IO activity
- **No overhead for unused WLM event monitors**
  - No events captured by these unless requested
  - Configure individual workloads, service classes, work actions to capture only events of interest

Turn on WLM Related Monitoring in OPM

Step 2 of 4: Configure monitoring profiles

Define the type of monitoring data that is collected by enabling the corresponding monitoring profiles. If you selected Use predefined template or Configure like on the previous page, then the associated profiles are enabled.

Selected configuration: **Use existing configuration**

Monitoring settings

Retention times and sampling intervals	🖉
DB2 event monitor configuration	🖉

Monitoring profiles

Inflight performance, reporting, or Workload Manager

These profiles collect performance statistics for the data server, which are shown in the inflight dashboards, in Workload Manager, or in the reports.

✔ Basic	
☐ Locking	🖉
☐ Active SQL and Connections	🖉
☐ I/O and Disk Space	🖉
✔ Workload Manager	🖉
☐ Dynamic SQL	🖉

Creating WLM Event Monitors

- **Look at** sqllib/misc/wlmevmon.ddl
  - Modify the script to specify an appropriate tablespace
    event monitors can consume substantial space
- **Activate the WLM event monitors**
  - `SET EVENT MONITOR <name> STATE 1;`
- **Set the collection interval**
  - OPM default is 5 minutes
  - Syntax for configuring interval manually
    ```
    UPDATE DATABASE CONFIGURATION FOR <dbname> USING
    WLM_COLLECT_INT 5;
    ```

Turn on Low Overhead Monitoring

- **Service subclasses**
  - Aggregate activity (basic)
  - Low overhead
  - RECOMMENDATION: turn this on permanently
  - `ALTER SERVICE CLASS <subclass-name> UNDER <superclass-name>  COLLECT AGGREGATE ACTIVITY DATA BASE ;`

- **Workloads**
  - High water marks for workloads are always collected if statistics event monitor is enabled

Viewing Histograms in OPM

- **Shown in OPM**
  - ActivityTotalTime
  - QueueTime
  - EstimatedCost
    - **New in v4.1.0.1**

- **Aggregated for selected period**
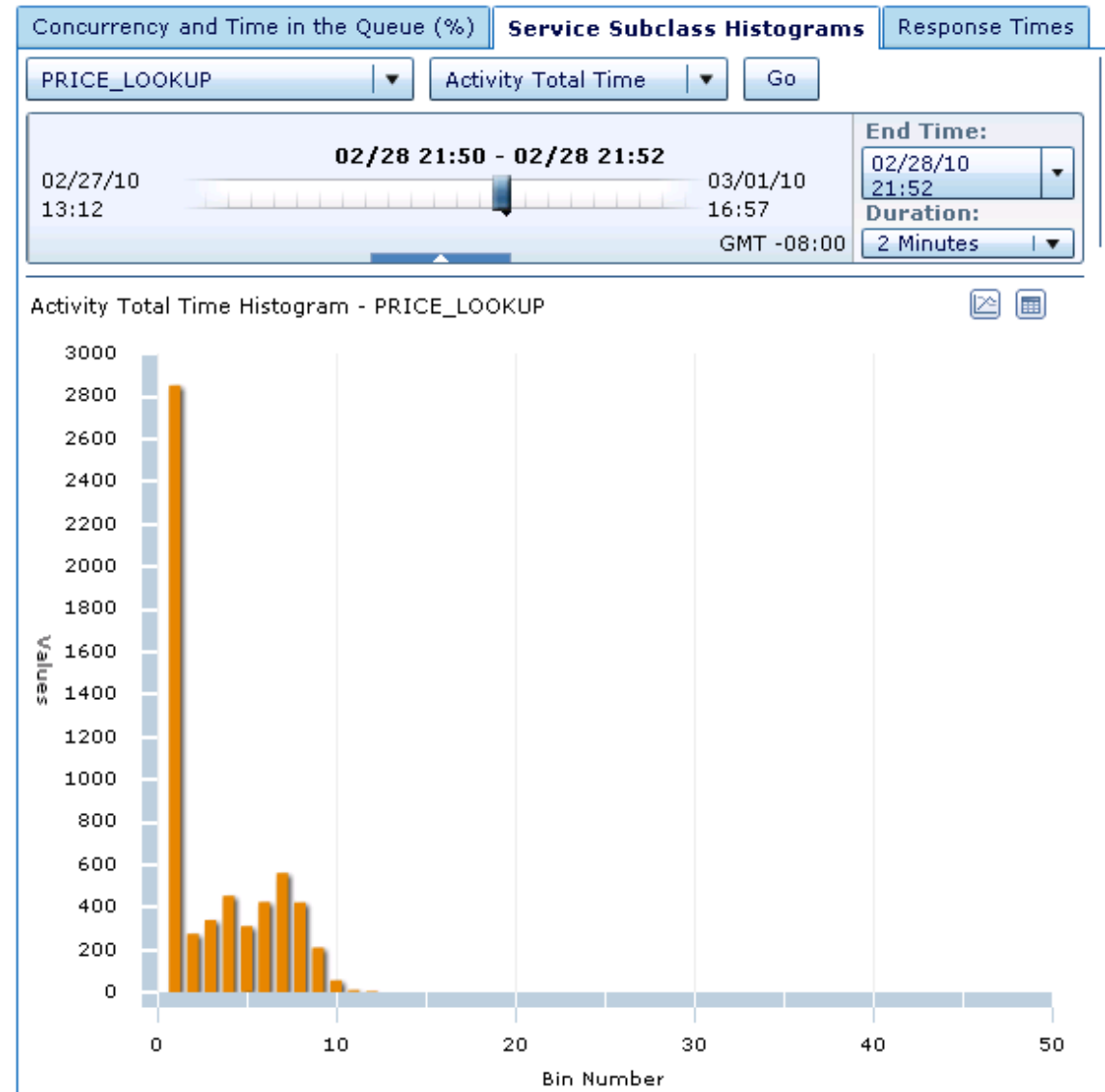
Viewing Histograms via SQL

- **You can use SQL to view histograms**

- **SQL for viewing a histogram**
  - SELECT TOP/1000 AS TIME_seconds, SUM(NUMBER_IN_BIN) AS #EXECUTIONS FROM HISTOGRAMBIN_DB2STATISTICS WHERE HISTOGRAM_TYPE = 'CoordActLifetime' GROUP BY TOP/1000 order by TOP/1000;

Step 6 - Determine Cost Boundaries

- **Adjust cost boundaries to distribute long running activities to the intended service subclass**

- **Rules of thumb, typical values**

SUBCLASS	MAX EST COST (TIMERONS)	Percent of resources consumed	Typical concurrency
Unlimited	5 K	5 %	(unlimited)
Trivial	30 K	45 %	50
Simple	300 K	25 %	15
Medium	5 M	15 %	10
Complex	(unlimited)	10 %	4

Validate Mapping of Large Queries

- **Compare histograms of ActivityTotalTime for subclasses**
- **Verify that longer running queries are routed as desired**
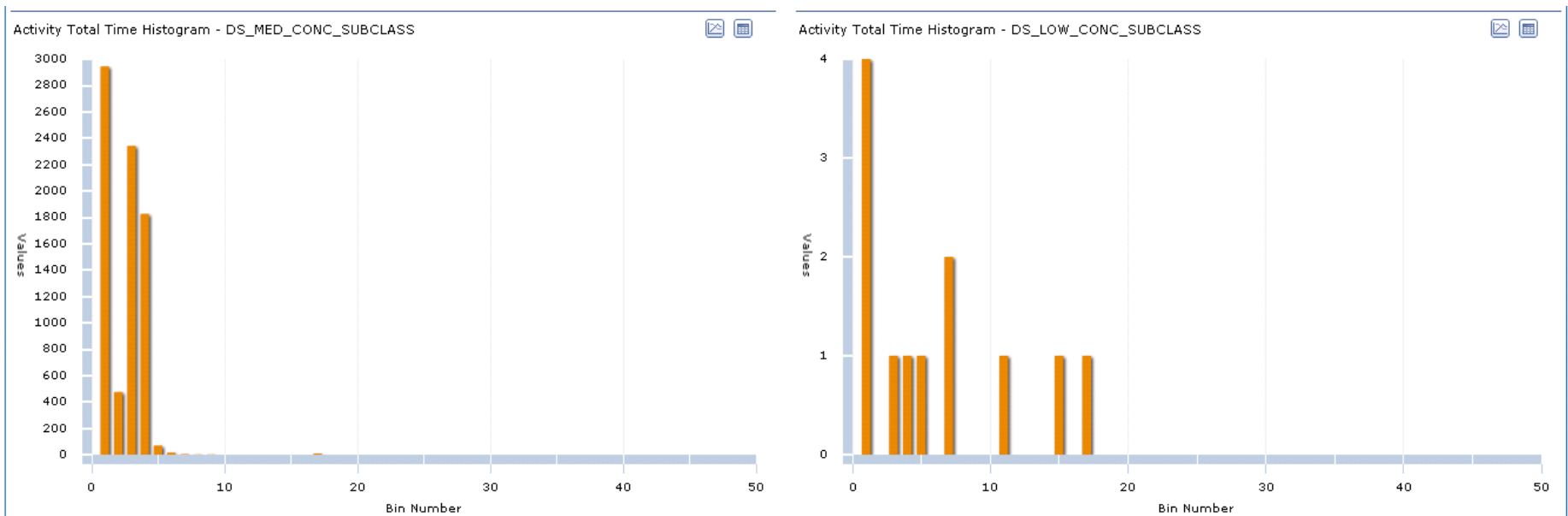
Step 7 – Limit Concurrency

- **Use the ConcurrentDBCoordActivities threshold**
- **Start by limiting concurrency for LOAD, large queries**
- **Generally get substantially improved performance, stability by**

Step 8 – Set Priorities for Subclasses (Optional)

- **For data warehouse type workloads, consider adjusting prefetch priority in subclasses**
  - Applicable only for scans
  - Has no effect on page reads, such as index lookups

- **For OLTP workloads, consider adjusting buffer pool priority**
  - This is a v9.7 only feature

- **Use sparingly / with caution: agent priority**
  - Higher agent priority for rush jobs can be useful
  - Agent priority for SYSDEFAULTSYSTEMCLASS must be at least as high as any user defined service subclass
  - Lower priority for long running queries makes things worse

Step 9 - Protect Against Rogue Queries

- **Reactive activity thresholds**
  - ActivityTotalTime
  - CPUTime**(New in v9.7)**
  - SQLRowsReturned
  - SQLRowsRead   **(New in v9.7)**
  - SQLTempSpace
  - ConnectionIdleTime
- **Predictive activity threshold**
  - EstimatedSQLCost

Configure Thresholds Iteratively

- **WLM policies cannot be verified outside of production DB**
  - Verify by deploying monitoring only controls
  - After reviewing monitoring data, alter to enforced controls
- **WLM policies must be informed by baseline monitoring**
  - Browse history of how long queries run
  - Create thresholds based on historical trends and data

Create/Refine Thresholds

Monitoring via Event Monitors

Validate

Observe Activity

Monitoring via Event Monitors

Apply Controls
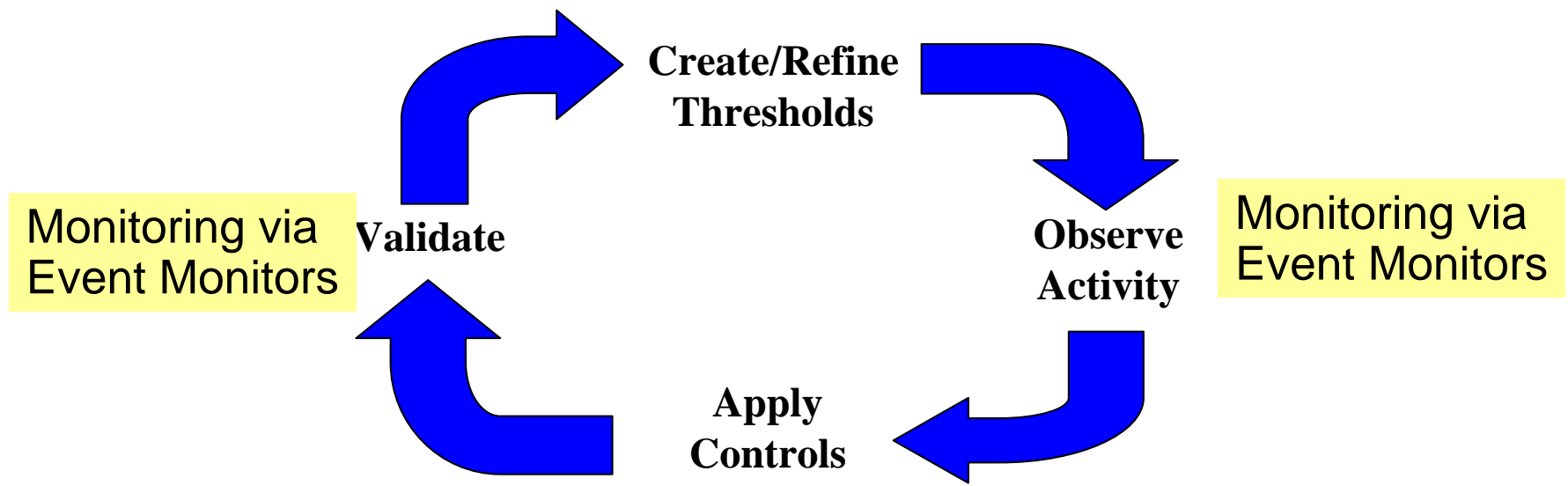
Threshold Configuration in OPM

Service Subclasses

Each row in the table represents a service superclass and service subclass combination for which you can define additional thresholds.

Service Superclass	Service Subclass
MAIN_SUPER	DS_HIGH_PRI_SUBCLASS
MAIN_SUPER	UNLIMITED_SUBCLASS
MAIN_SUPER	TRIVIAL_SUBCLASS
MAIN_SUPER	SIMPLE_SUBCLASS
MAIN_SUPER	MEDIUM_SUBCLASS
MAIN_SUPER	COMPLEX_SUBCLASS
MAIN_SUPER	DS_LOAD_SUBCLASS

Thresholds of the Service Subclass

Enable any thresholds that you want to enforce.

Time Limits **Row Limits** Temp Space Limits

Service superclass: MAIN_SUPER

Service subclass: SIMPLE_SUBCLASS

Threshold type: SQL rows returned

Detects and controls activities that return an excessive number of rows.

Limit for SQL rows returned (Number of rows): 100000

☑ Monitor the activities that exceed the limit

☐ Stop the activities that exceed the limit

☑ Enable threshold

Autonomic Performance Objectives
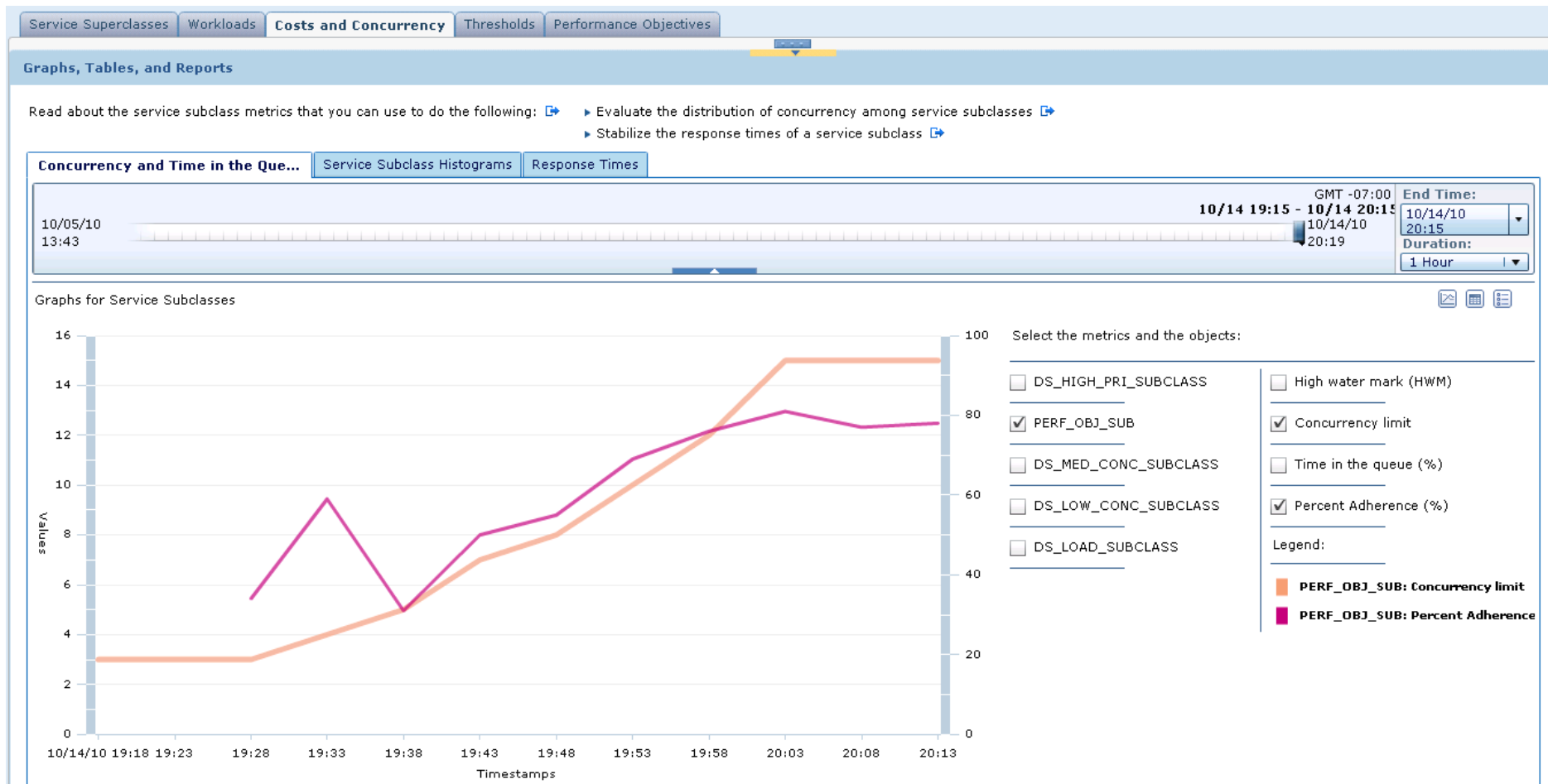
Additional Scenarios
and
Special Cases

Alternate Solution - Priority Aging

- **"Easy button"**
  - Work started at appropriate priority based on cost or workload
  - Automatically remap long running work to lower priority
  - Customization is possible, but not necessary
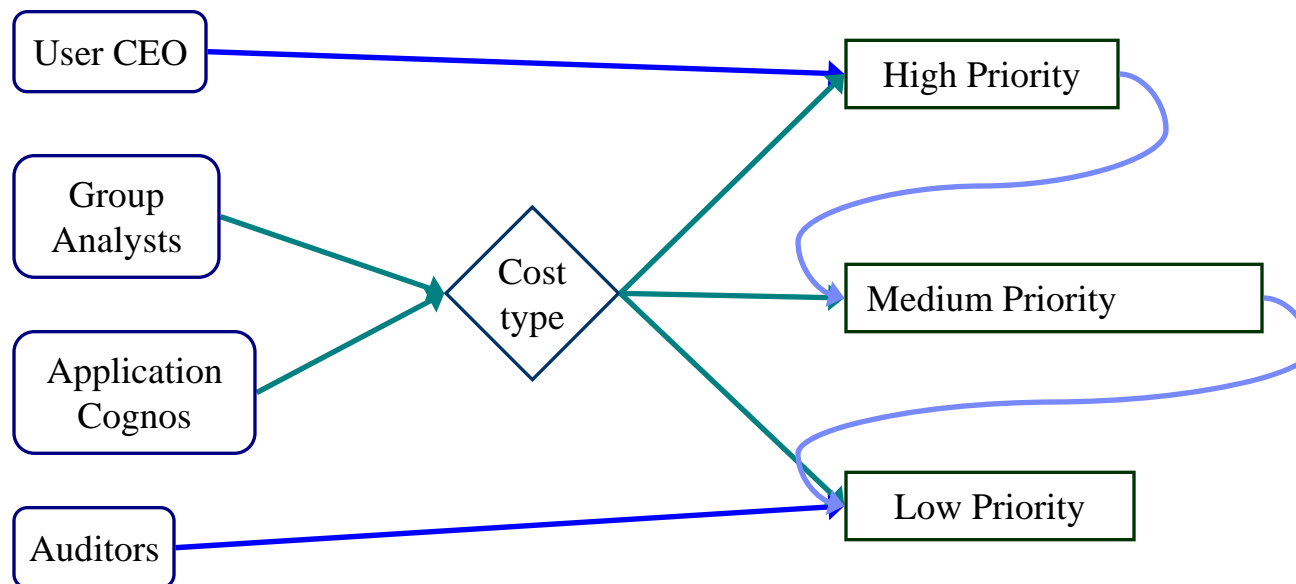
Mixing Priority Aging with Concurrency

- **Recommendations**
  - Use with caution
  - Consider alternate approach with concurrency thresholds on workloads

- **Potential pitfalls**
  - Activities continue to hold concurrency tickets from originating service class after they are mapped
  - Activities mapped into a service class do not queue for entry

Waiting in Queues
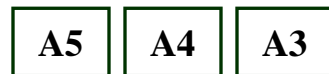
Concurrency limited to **2** for each subclass

3 waiting | A5 | A4 | A3 | | A2 | A1 |
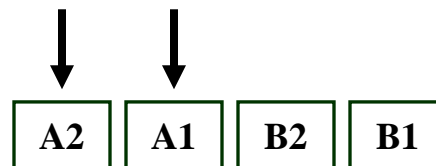
Activities remapped from A block activities waiting in Q A

4 waiting | B6 | B5 | B4 | B3 | | A2 | A1 | B2 | B1 |

Activities remapped to B exceed the intended limit

Alternative for Combining Priority Aging, Concurrency

- **Concurrency on** workloads
- **Priority aging on** service subclasses
- Requires v9.7 features

workloads

| User CEO |
| Group Analysts |
| Application Cognos |
| Auditors |

Cost type

| High Priority |
| Medium Priority |
| Low Priority |

service subclasses

Concurrency at **workload** level

Priority aging on **subclasses**
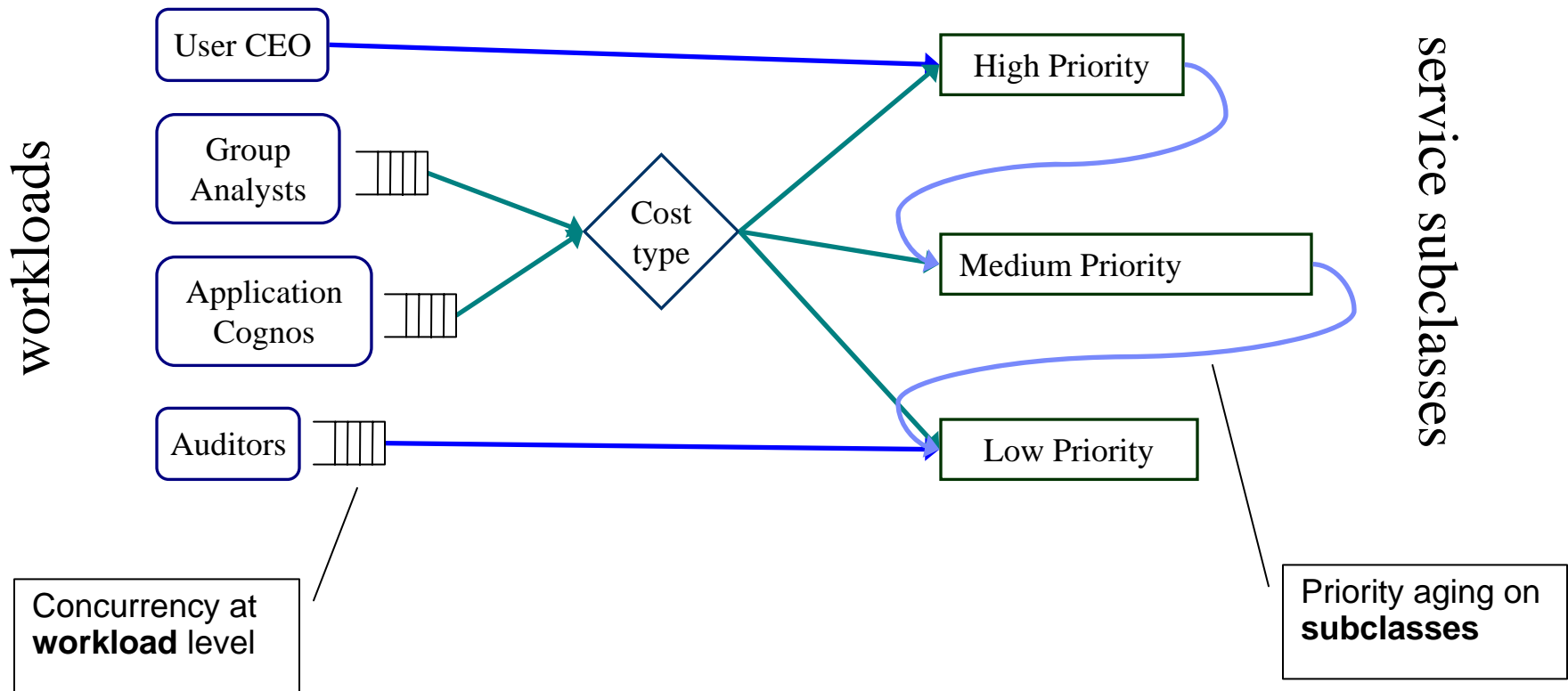
Scenarios for Multiple Service Superclasses

- **Divide resources**
  - Separate superclasses for departments or applications
  - Divide total available system resources among superclasses

- **Different rule sets**
  - Separate superclass for ETL
    - Up to 4 concurrent LOAD activities
    - Huge queries allowed, but at low priority

SALES_SUPER

SALES_PRICE_LOOKUP

CPU Priority HIGH

SALES_ACTIVITIES

SALES_REPORTS

Activities > 10
⇨ Queue

ETL_SUPER

ETL_LOAD

Activities > 4
⇨ Queue

ETL_QUERIES

CPU Priority LOW
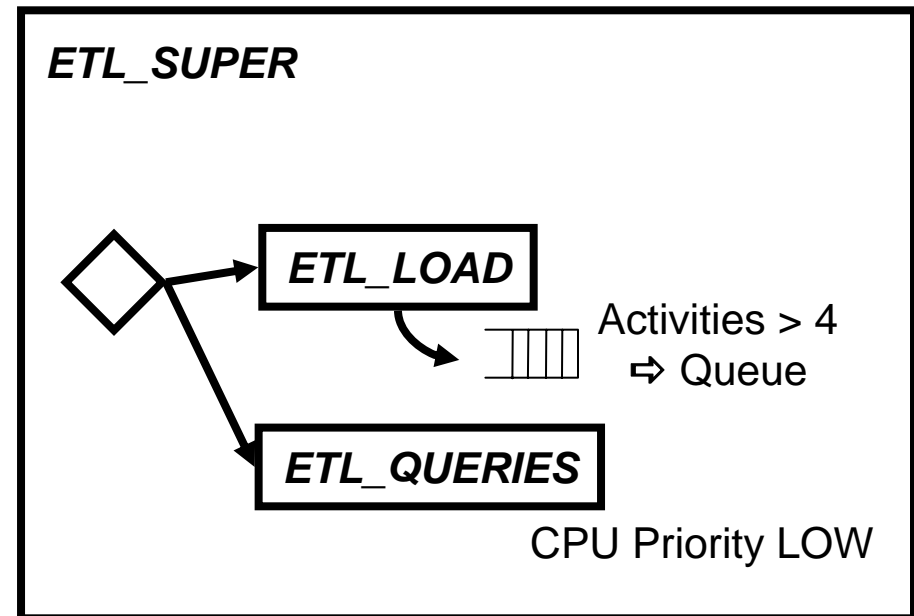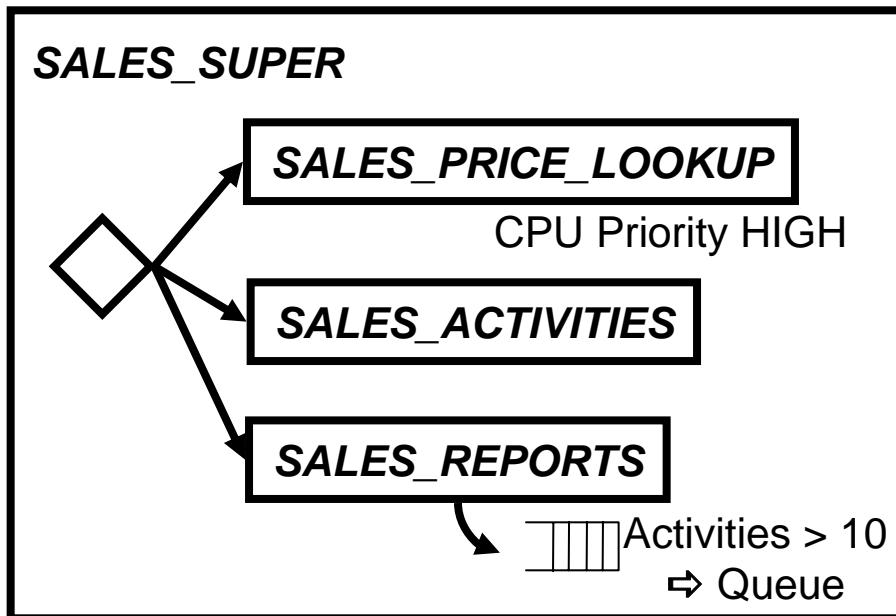
Use AIX WLM for Strict Division of CPU Resources

DB2

AIX

User Requests

| workload: SALES_WL | → | DB2 superclass: SALES_SUPER | → | AIX superclass: SALES_OS |

| workload: MARK_WL | | |
| workload: CAMPAIGN_WL | → | DB2 superclass: MARK_SUPER | → | AIX superclass: MARK_OS analysis |

| workload: Default user requests | → | DB2 superclass: Default user requests | → | AIX superclass: Default user requests |
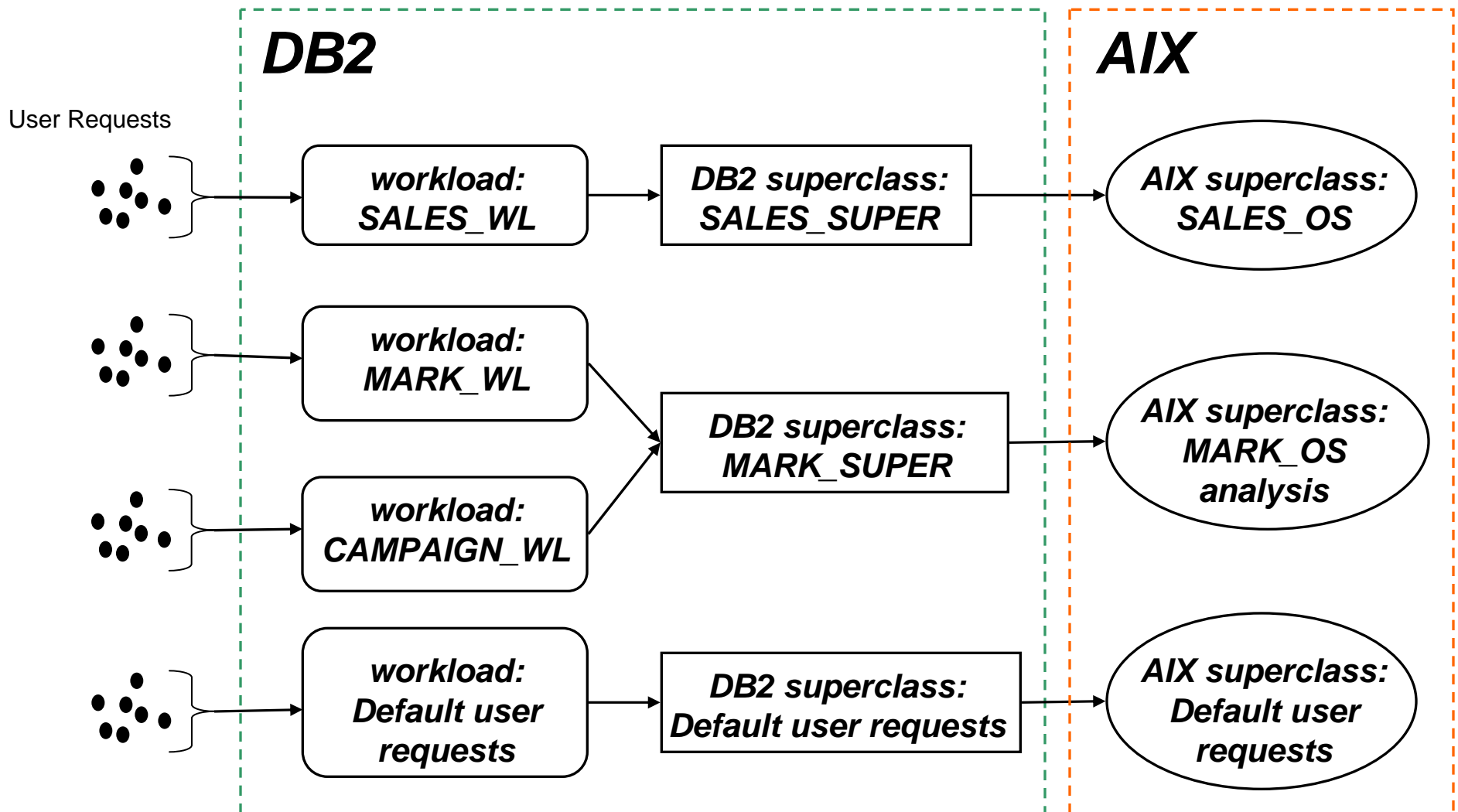
Integrating DB2 WLM with AIX WLM

- **HARDMAX caps CPU usage of a service class**
  - Applicable in scenarios where concurrency limits are insufficient in reducing priority of a service class
- **AIX WLM**
  - Use HARDMAX to cap CPU usage of a service class
  - AIX WLM provides additional statistics
  - Other features of AIX WLM applicable only for systems which are heavily resource constrained
- **Refer to WLM Best Practices white paper for details**
  - Example script for dynamically adjusting HARDMAX

Production Shifts

PEAK HOURS

SALES_SUPER

High Cost SQL

Activities > 2
⇨ Queue

<else> → Ordinary Work

CPU Priority HIGH

ETL_SUPER

ETL LOAD

Activities > 4
⇨ Queue

<else> → Ordinary ETL Work

CPU Priority HIGH

OFF HOURS

SALES_SUPER

High Cost SQL

Activities > 10
⇨ Queue

<else> → Ordinary Work

CPU Priority LOW

ETL_SUPER

ETL LOAD

Activities > 4
⇨ Queue

<else> → Ordinary ETL Work

CPU Priority HIGH
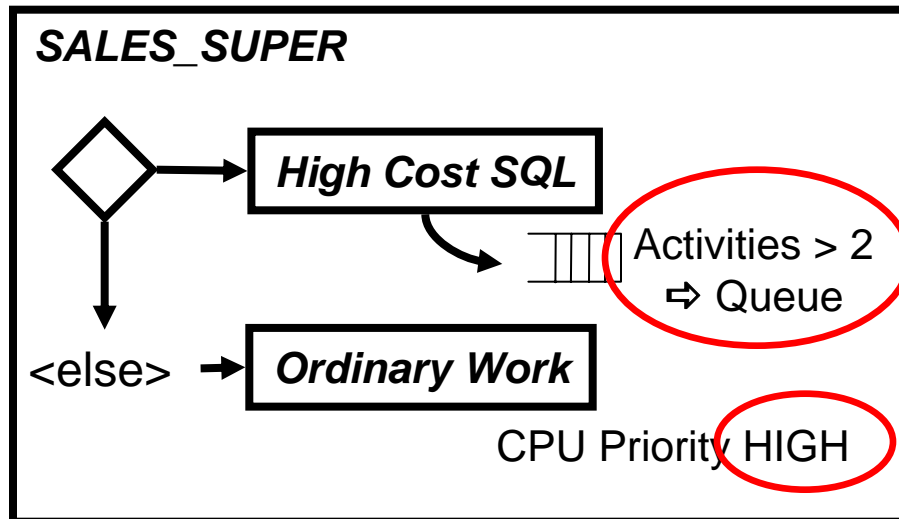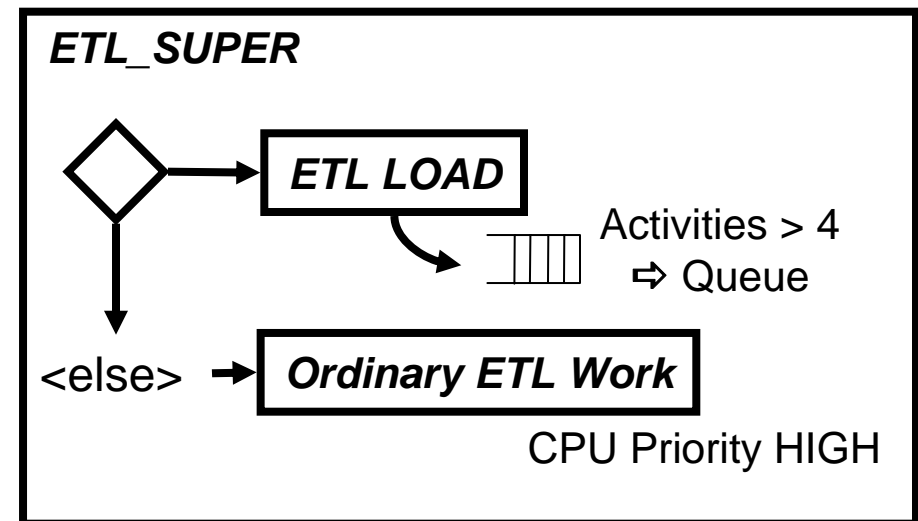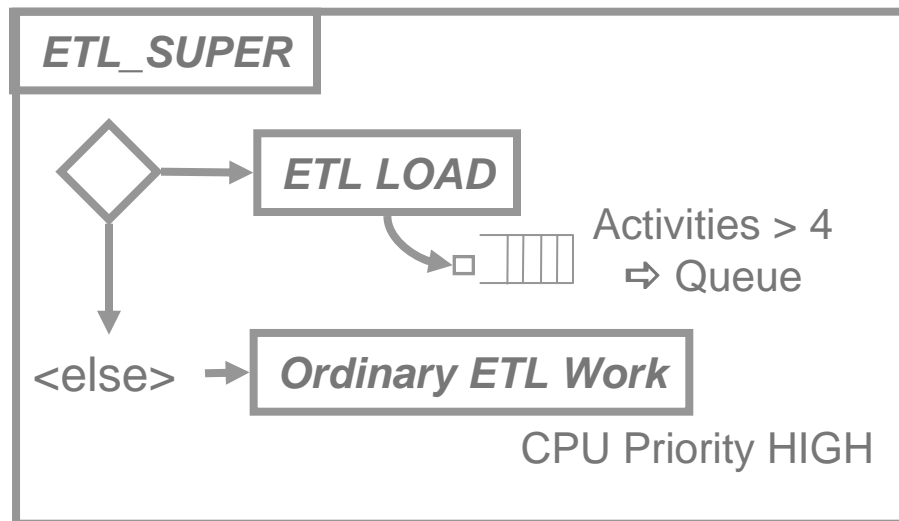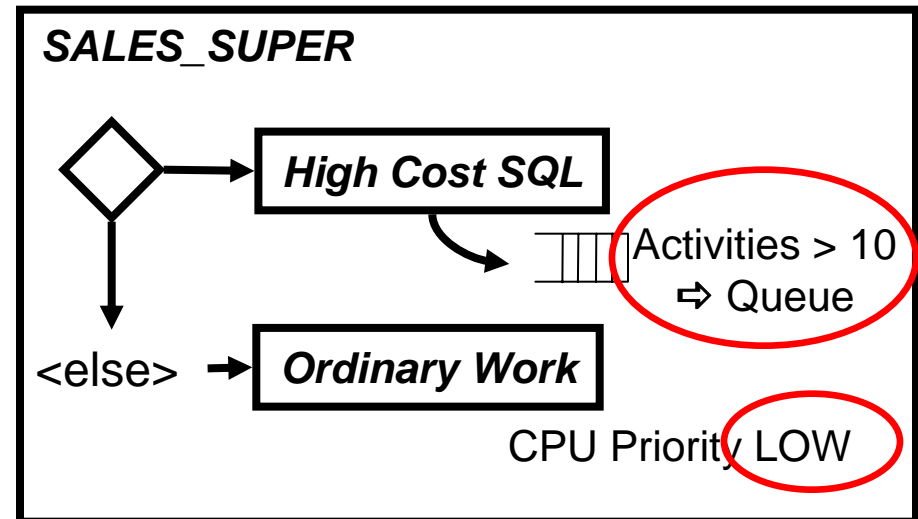
Recommendations for Production Shifts

- **Avoid drop / create of workloads and service classes**
    - Drop requires you first disable, then wait for activities to drain

- **To block a workload from running**
    - ALTER WORKLOAD … DISALLOW DB ACCESS;
    - Matching connections will error out

- **Alter is robust and online**
    - Alter priority of a service class
    - Alter mapping of a workload
    - Alter limits for a threshold
    - Alter a threshold to disable

Migration from QP / Governor

- **QP can provide insights about activities in your DB**
- **WLM in DB2 offers additional controls**
  - Thresholds
  - Controls for activities other than DML, such as DDL, LOAD
  - CPU and prefetch priorities

- **Concurrency controls in WLM are slightly different**

- **Migration script available to partly automate migration**
  - sqllib/samples/perl/qpwlmmig.pl
  - Recommend revisiting policies during migration
  - OPM recognizes output of this script

Summary

Stepwise Methodology

1. **Workloads**
   - Categorize connections
2. **Service superclass**
   - Container for service subclasses and work action set
3. **Service subclasses**
   - Apply controls to service subclasses
4. **Work action set**
   - Maps activities based on type or cost
5. **Baseline monitoring**
   - Work iteratively
   - Validate after each change
6. **Set cost boundaries**
7. **Limit concurrency for expensive activities**
8. **(Optional) priorities on subclasses**
9. **Thresholds**
   - Protect against rogue queries

Alterative Approaches

- **Priority aging**
  - Long running activities automatically lowered in priority
  - Easy deployment
    - No tuning necessary
    - Low risk solution
- **Configuring concurrency thresholds on workloads**
- **Integration with AIX or Linux WLM**
  - Provides sandboxing, or strict caps on resource usage
  - Allows global management across multiple databases

Additional Information

- ## DB2 and OPM Information
  - **IBM DB2 Database for Linux, UNIX, and Windows Information Center:**
    - http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp

  - **Tutorial for Workload Management**
    - http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.wlm.doc/doc/c0053139.html

  - **IBM Optim Performance Manager Information Center:**
    - http://publib.boulder.ibm.com/infocenter/idm/v2r2/topic/com.ibm.datatools.perfmgmt.overview.doc/topics/helpindex_opm.html

- ## Feedback
  - Presentation format and contents
  - Additional DB2 topics you are interested
  - Follow on questions for the presentation

- ## Contact: fning@ca.ibm.com