

May 7-11, 2006

Tampa Convention Center

Tampa, Florida, USA

Session: A03

IDUG® 2006

North America

# Workload Manager for the DB2 Professional

Tom Moulder  
TREX Associates, Inc.

Monday, May 8, 2006 • 04:00 p.m. – 05:10 p.m.

Platform: DB2 for z/OS

GoFurther



## Topics for Today

- Workload Manager Terminology
- Workload Manager's Hierarchical Design
- Goal Mode Choices and their Consequences
- Recommendations for Performance
- Recommendations for Ease of Maintenance
- Summary

## Service Definitions

- A Set of Objectives
- Used to Manage System Resources
- Defined as Rules with Varying Importance
- Based on Identifiable Workloads
- Hopefully with a good naming convention

3

Service Definitions can be viewed as the z/OS parameters that are specified in a Policy (to be explained on the next slide) that set the operational control parameters for z/OS. In this particular case, talking about Workload Manager, these definitions would control the scheduling of all workloads in a system. The parameters that are specified instruct z/OS in exactly how to identify each particular workload. This can be much easier depending upon the naming conventions that are used for all the varying workloads that execute on z/OS. As for DB2, this is made easier by good naming conventions for CICS and IMS transactions and the associated plans and packages they invoke within DB2. When you talk about multi-platform distributed workload then you get into more of a good naming convention for Stored Procedures these days as well.

## Policy

- A collection of Service Definitions
- Specific to a Defined Interval of Time
  - Weekdays
  - Weekends
- One Policy in effect at a time
  - Therefore, it must be all inclusive

4

z/OS has duplexed Policy data sets than contain a large number of operational parameter definitions that control the individual choices of z/OS. These choices in turn determine the efficiency of a system. Multiple Policies can be defined and changed dynamically with a single operator command. However, the system will only allow one Policy to be in effect at a time. Most z/OS shops where I have worked have only one Policy and do not switch that Policy for any reason. Perhaps they have been fortunate enough to have the ability to specify how to adequately handle all workloads at all times in one Policy.

## Service Class

- A Unique Workload
  - Easily Identifiable
    - Monitors
    - Batch jobs
    - TSO Users
    - CICS/IMS Transactions
- With the Same Performance Requirements

5

Each unique identifiable workload is assigned a Service Class in the Policy definition. A Service Class must be assigned to everything that executes in z/OS. If the Policy Definitions do not result in a match for assigning a Service Class, then z/OS will assign a default class that is called “Discretionary”. Any workload in the “Discretionary” Service Class will be dispatched when there is nothing else available to be dispatched. This could seriously affect the response time of a workload in this Service Class.

CICS and IMS transactions may fall into multiple Service Classes as each application gets a business priority that is associated with the “Goal” as defined in technical terms to Workload Manager.

## Report Class

- A Unique Workload
- Purely used for Reporting
- Allows Grouping and Aggregation
- No Impact on
  - Actual Performance
  - Resource Allocation

6

In addition to Service Classes, Workload Manager has Report Classes. They are purely used for reporting and have no connection to Service Classes. If a business application has multiple priorities that must be defined by a variety of Service Classes, then a Report Class could be defined to group all of the Service Classes together into a Report Class for the Application.

There are many possibilities for the use of Report Classes.

## Resource Groups

- System Resources
- Provides Service
- Some Examples
  - Initiators for Batch jobs
  - Minimum and Maximum TSO user limits
  - Stored Procedure Address Spaces

7

Resource Groups are used to define system resources that should be managed by Workload Manager. For this particular Group (IDUG), the most important example would be Stored Procedure Address Spaces. The Policy can then have definitional parameters that control the use of the resource. For DB2 this could simply be in terms of a minimum and maximum number of address spaces for a particular purpose.

## Periods

- A Method for Changing Requirements
- Usually over a period of time
- Provides a means to
  - Lower/Raise Priority

8

A Period allows for Workload Manager to change the priority of an entity after some duration has occurred. The duration could be some period of time, it could be simply the use of some number of system resources as determined by z/OS.

Primarily this is used to reduce the priority of an entity after it has used some amount of elapsed or CPU time. However, it could be used to raise the priority under the assumption that every job should ultimately complete and perhaps if a job is in the system for a very long period of time something should be done to simply get this job out of the system.

## Goals

- Two Types
  - Response Time Goals
    - Average or Percentile
    - Expressed in Seconds or Portions of a Second
  - Velocity
    - Represents a Percentage of Available CPU Time

9

Performance goals within WLM can be either response time goals or velocity goals.

Response time goals are either based on average or some percentile. Average response time goals require a minimum of ten transactions with a ten second assessment period of WLM. It is a true average and thus the smaller the number of transactions the more a single abnormal response time can affect the average. Percentiles can be expressed as any percentage. WLM then calculates the percentage of transactions that are less than the response time goal specified. This method is more popular as it provides a hedge against periods of smaller transaction volume when abnormal response times can affect the average.

Velocity goals are used to assign a percentage of available CPU time to a task. This is usually assigned to tasks that will have a small number of transactions per ten second assessment interval. Monitors in the system fall into this category. Other examples will be given in later slides.

## Performance Index

- WLM Calculated Value
- Determines State of a Particular Unit of Work
- Possible States
  - = 1 UOW is Meeting Goal
  - > 1 UOW is Better than Goal
  - < 1 UOW is Not Meeting Goal

10

Workload Manager calculates a Performance Index for every task in the system that is ready to execute. If it calculates out to exactly 1, then the task is achieving its “Goal” as defined to Workload Manager. If it is more than 1, then the task is doing better than its defined “Goal” to Workload Manager. If it is below 1, then the task is not doing as good as its “Goal” as defined to Workload Manager. This index provides Workload Manager with a list of tasks that should be helped by increasing dispatching priority every time WLM evaluates the system to see if changes should be made. This evaluation occurs every 10 seconds and at evaluation time only one task can be modified. This Index is used in conjunction with priority to determine that task z/OS should assist with an increased dispatching priority.

## Application Environment

- Group of Programs that provide Similar Services in a Common Address Space or Spaces
- Once given a Name in WLM, Controls are put in Place to Manage Resource Utilization
  - Typically Address Spaces
- Used by Websphere and DB2
- Priority Assignment is Controlled by Service Class, not Application Environment

11

An Application Environment is used by WLM to manage resources such as Address Spaces within z/OS. Application Environment has nothing to do with dispatching priorities. What it provides is the definitional parameters to control resource utilization of address spaces. As far as DB2 is concerned, these Application Environments are used to control the number of address spaces that can be used a common set of stored procedures.

## Enclaves

- Territory Surrounded by Another's Territory
- For z/OS
  - Any combination of SRB and/or TCB Tasks
  - Represent a "Business" Unit of Work to WLM
  - Can Cross Address Spaces
  - Assigned to a single Service Class in WLM
  - Maintain same Priority across Tasks

12

An Enclave as far as z/OS is concerned is a task that operates within another task at a dispatching priority that is totally separate from the surrounding task. Specifically, the best example of this is the Stored procedures Address Spaces for DB2. You can direct multiple completely different Stored Procedures into the same address space and have each Stored Procedure be dispatched by z/OS at multiple different dispatching priorities. Many refer to these as "Enclave SRBS". The fact is that an "Enclave" can be either an SRB or a TCB. It refers to any task that has its dispatching priority managed separate from the Address Space where it resides.

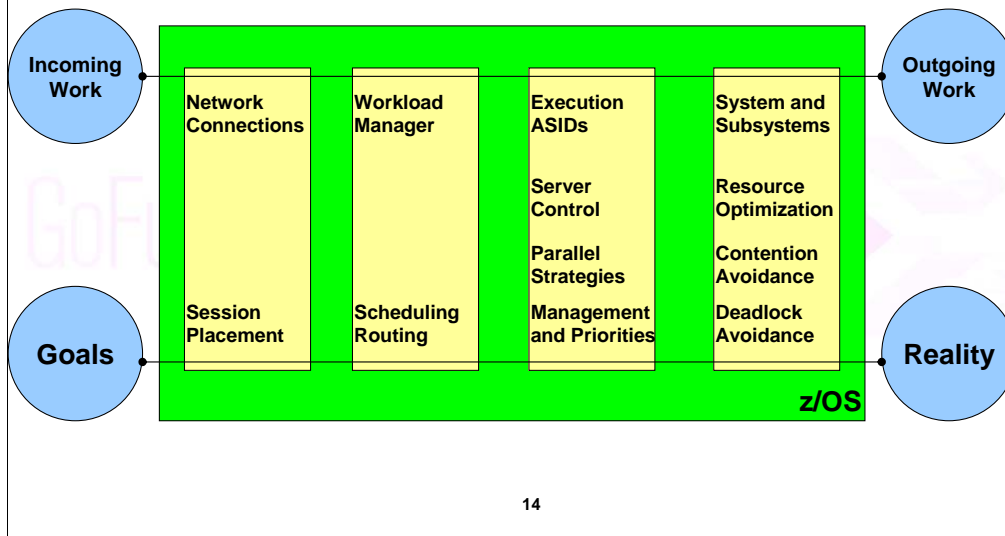
## Objectives of WLM

- Simplify Performance Requirements
  - Definition
  - Control
  - Reporting
- Goal oriented Resource Management
- Tied to Business Importance
- Affects CPU and Memory Resources
- Coordinates across members of a sysplex

13

WLM was designed to simplify the specification of performance management requirements to z/OS in such a fashion that it could be easily tied to business importance. Many will argue that it failed to achieve that objective. One thing almost everyone will agree on is that the parameter definitions are completely different from SRM (its predecessor). My belief is that many have resisted the conversion to WLM because they are totally unfamiliar with the definitional parameters and they are so different from anything they have ever dealt with before. WLM also has suffered from many companies who have had problems with the initial implementation.

## The Big Picture



14

WLM manages work inside z/OS regardless of its origination. Through whatever subsystems might be involved and down to the execution unit which would be a TCB or an SRB. It does this through the parameter definitions that we have just defined. Customers usually take the time to monitor the reality of priorities to insure that the theory of the parameters matched the reality at execution time. Changes are often made as workloads change and as new applications are added to the system.

## A Little Bit Deeper

- z/OS Dispatching
  - Accomplished by Dispatching Priority
    - One Byte
    - Zero to 255
    - Highest is better
  - Tasks are dispatched by task type
    - Global SRB
    - Local SRB
    - TCB and Enclaves

15

z/OS manages performance for each task in the system by means of a dispatching priority. This is a one byte field that can thus have values from 0 to 255. Tasks are dispatched when they reach the highest dispatching priority of all tasks ready for execution. Tasks are also picked for execution by type of task. Global SRB and Local SRB tasks are always scheduled first before any TCB or “Enclave” tasks.

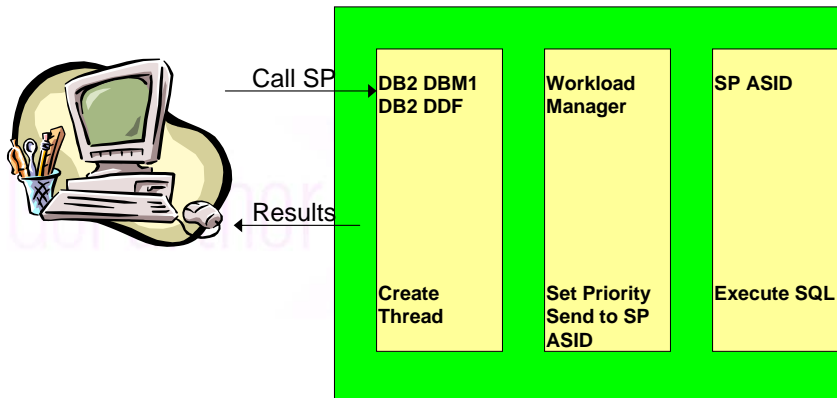
## WLM Interface to z/OS Dispatcher

- WLM in Goal Mode controls Dispatching Priority
- Priority is set by Service Class in Policy
- Priority of one task may be adjusted by WLM
  - Assessed every 10 seconds
  - Based on Performance Index of Tasks
  - Based on Importance of Tasks
- Objective is Gradual Adjustments Not Mood Swings Back and Forth

16

WLM controls the dispatching priority of all tasks in the system. The Policy determines the Service Class which in turn determines the dispatching priority. WLM makes a system assessment every ten seconds at which time it make a change to one task in the system. The change is to increase the dispatching priority for a task that is not meeting its “Goal” as defined in the Policy. Ten seconds may seem like a long period of time for z/OS, but more frequent changes could result in the system thrashing as it just moved tasks in and out for execution.

## DB2 Example

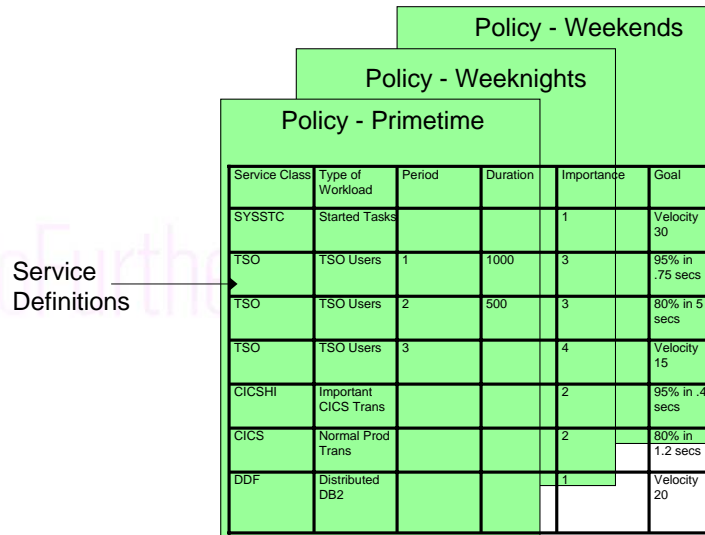


17

The easiest DB2 example is to use a Stored Procedure example. A call is made to a Stored Procedure. WLM gets invoked to determine the Service Class for the Stored Procedure which sets the dispatching priority. Also, WLM determines the Application Environment which determines the Stored procedure Address Space that will be used to schedule execution. If additional Address Spaces are required for this execution, then they are created by WLM.

Similarly, when a CICS/IMS transaction invokes a DB2 service a thread is created or used from a pool. The CICS/IMS transaction was classified by WLM when it entered the system. When a DB2 service is invoked, WLM assigns the same dispatching priority of the CICS/IMS transaction to the DB2 work that must be performed on behalf of the transaction.

## Workload Manager's Hierarchical Design



18

The graphic describes the multi dimensional aspects to WLM and the parameters that are used to control performance with z/OS.

## Structure Explanation

- Importance
  - Lower the Number – Higher the Importance
- Duration
  - Length of the Period in Service Units
- Service Class to Period Relationship
  - One to Many
- Goal
  - Velocity guarantees CPU if needed up to percent
  - Response Time Goal is internal response

19

This is a summary of all the definitions that have been discussed to this point. Hopefully, this will pull all of these definitions together.

## Sample Spreadsheet

- Available from Speaker
- Compiled Over Time from Various Sources
- A Good start
- Goal of this sample –
  - Reduce the complexity
  - Get on one worksheet
  - More easily see relationships of classes
  - Get a system perspective

20

Based on discussions with various customers, a sample spreadsheet is available from the speaker to document all the different parameters in one place. The intent is to make it a little easier to visualize the relationships between Service Classes and easily determine the ones with the highest priority goals within WLM.

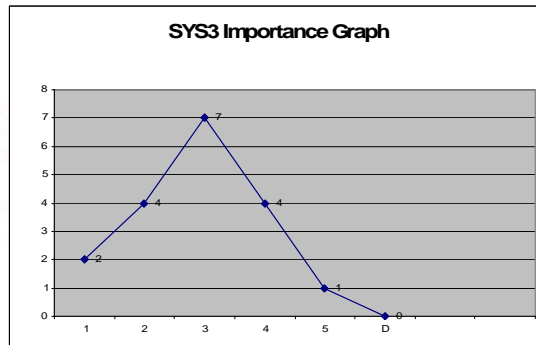
## Sample Spreadsheet

- Row for each Service Class/Period
- Columns
  - Service Class
  - Description
  - Workload
  - Resource Group if it applies
  - Period
  - Duration
  - Importance
  - Goal

The spreadsheet contains a row for each Service Class period and columns to document that Class. You can enter a description along with the basic definitional parameters for the Service Class.

## Sample Spreadsheet

- Chart Importance
- Look for Overloads
- Strive for Bell Curve
- Notice this is Lopsided
- No One Wants to be Discretionary Work



22

The spreadsheet includes a graph of the importance assignments in the Policy.

It is often an inclination to define too many classes as a high importance. My recommendation is to seek for a bell curve. This results in many application workloads in the middle competing for CPU within z/OS.

## Importance Choices and Their Consequences

- Competing for CPU and Memory
  - Equality within an Importance Number
  - Lower Numbers beat Higher Numbers
- When Goals are being met
  - Higher Numbers given more Access to Resources
- When Goals are Not Being Met
  - Lower Number get More access to Reach Goals
- Assessment every 10 seconds
- Only One Change each Assessment

23

z/OS calculations to determine which task to assist at an assessment interval takes into consideration the Importance Level. Competition is within an importance level. Therefore, if z/OS determines that an importance level 1 task needs assistance, then it is the one that is picked. Even if the Service Class of the task would represent a task with a small dispatching priority. As long as all Importance 1 tasks are meeting their goals, then z/OS goes on to Importance 2 and so forth until discretionary workloads are assisted after all other workloads are making their goals.

This makes it important in z/OS to define realistic goals for all workloads. If the goals can never be met, then z/OS will continually raise the dispatching priority of the task as high as the policy will allow in an attempt to help the workload reach its goal. This will be done at the expense of lower importance work that might benefit much more from the assistance.

## Recommendations for Importance 1

- Reserved for System Tasks
  - Started Tasks
  - Monitors
  - IRLM
  - DDF
- Usually Velocity Controlled
- Response Time Requires 10 transactions per 10 seconds for response time goals

24

Importance 1 should be reserved strictly for system tasks. These tasks usually have a small number of transactions because WLM only sees the one address space which doesn't change in the interval so it represents only one transaction to WLM. Therefore, these tasks are best managed by velocity goals. A goal of 30 for these tasks would allow any one of them to use up to 30% of the CPU to do whatever needs to be done. The only time this would cause a problem for z/OS is in the event of an error in any one of these tasks and that error if it drives CPU utilization could take a large percentage of the available CPU thus causing problems for other tasks in the system. The good news is that most of the tasks are well behaved and in the event of an error they can be removed from the system by a single command to request the task terminate.

## CICS Recommendations

- CICS TOR
  - Control with Velocity – few transactions
- CICS AOR
  - Response time goals preferred
  - Multiple Service Classes by Transaction Names
  - Choices Limited by CICS TCB Structure
- CICS FOR
  - Velocity Controls

25

CICS has a number of different address spaces types to perform its functions. The TOR and FOR regions usually have a small number of transactions and work best with velocity controls.

I would set the velocity at any particular value and monitor performance. If performance does not meet expectations, then I would raise the velocity level until performance matches expectations. The only exception to this is if there are other workloads in the system that are more important that begin to fail meeting expectations because of the change to this particular CICS velocity. If performance meets expectations, then I would lower the velocity until the performance does not meet expectations. At that point, the velocity can be moved back up to the previous value.

CICS AOR (Application owning Regions) behave differently, because the transactions actually perform the majority of their work in this address space. In this case response time goals are used by the majority of companies. Percentile goals work better when the number of transactions is small enough that a few transactions can affect the average response time calculation.

## DB2 Considerations

- IRLM and DDF already mentioned
- Stored Procedures
  - Classified in many ways
    - Plan, Package, Client Address, Authid
  - Executed in WLM controlled ASID
  - Multiple TCBS per ASID—DSNTIPX DSNZPARM
  - Enclaves – Separate priorities
  - Flow through Application Environments

26

DB2 stored procedures are managed through a combination of Application Environments and Service Class assignments. The DSNTIPX dsnzparm entry controls the number of TCBS allowed in each Stored procedure Address Space. The number of address spaces can either be set at one or will be controlled by WLM and ability of z/OS to create new address spaces for the workload. Many fear this and limit Application Environments to one address space.

## Recommendations - AE

- Application Environments
  - How Many ?
    - Add for all Necessary Controls
    - Affected by the NUMTCB parameter
    - Application Single Threading Issues
  - SSID in the name?
    - Increases the Number
    - Is Workload Significantly Different across SSIDs?
    - Does it help Categorize the Workloads for z/OS?

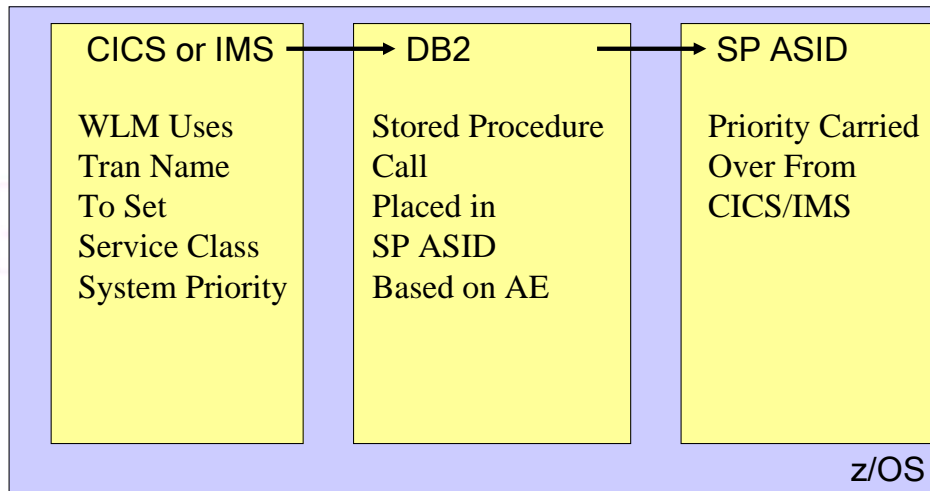
27

I recommend as few as possible for the number of Application Environments. I recommend as many as you need to provide for the varying workloads in the system.

There are necessary controls that must be addressed. So it is very difficult to determine what is the right number of Application Environments. Single Threaded requirements will increase the number of Application Environments because each will require its own Address Space.

I have seen some include the DB2 Subsystem ID in the name of the Application Environment. I would not recommend this unless the subsystem has some particular influence on the performance of Stored Procedures that execute on the system.

## DB2 Transaction Flow



28

WLM also controls the priority of DB2 work that comes into the system through CICS or IMS. WLM classifies the work by transaction typically in CICS or IMS. Any DB2 work on behalf of that transaction operates in the same service class as the original transaction regardless of the address space where it is actually dispatched.

## Recommendations - Monitoring

- Support for Enclaves is important
- Easily Drill from Familiar Thread Display
- Show Enclave Dispatching Priority
- Ensure Policy Definitions are Correct

How can you monitor such an environment?

Make sure that your monitor provides all the support you need to monitor enclaves. Make sure it is easy to identify enclaves as a separate workload and to monitor the performance of this workload to make sure it is meeting expectations.

## Recommendations - Setting Goals

- Started Tasks
  - Usually set as Velocity goals
  - High Velocity says ...
    - Dispatch this task whenever it wants the CPU
  - Low Velocity says ...
    - Let this task plod along so it will eventually finish

30

Started Tasks will only have a single transaction as far as WLM is concerned at each ten second assessment of the system. Therefore, these should be managed with a velocity goal. Set the velocity goal by ranking the tasks based on business priorities and let that determine the actual velocity setting.

## Recommendations - Response Goals

- Short versus Long Goals
  - Dividing Line is 10 seconds
- Percentile versus Average
  - Percentile are preferred
  - More consistent
  - Not affected by aberrations as easily
- WLM has a binary perspective
  - Meeting or Missing Goals

31

Response time goals should be realistic. This turns out to be important because of the binary perspective of WLM. A task is either meeting its goal or not. Based on priority WLM will try to help the application meet its goal. Percentile goals work better for workloads with smaller number of transactions because they are not affected by aberrations in response time.

## Recommendations - Service Classes

- Maximum of 99 Allowed
- 15 – 30 is the recommended number
- Separate out
  - Distributed DB2 Work
  - Open Edition (USS) Work
- Ensure a Service Class has similar workload
  - Long and short transactions together is trouble

32

Service Classes are actually used for determining dispatching priority in z/OS. The larger the number of classes, the more CPU time z/OS will expend in deciding what to do at each assessment interval. Workloads should be classified so that very similar work is in the same class. This makes it much easier to set response time goals that are realistic and achievable.

## Recommendations - DDF Goals

- If Threads=Inactive AND Release(commit)
  - One Enclave created per active interval
  - Response time values do not include Think Time
  - Use Response Time Goals
- If Threads=Active OR Release(Deallocate)
  - One Enclave for life of thread
  - Response Time goals include Think Time
  - Use Velocity Goals

33

I hope this slide is self explanatory for the recommendations. The key principle is that the smaller the number of potential transactions the more you want to go with velocity goals. That is the point of this recommendation.

## Summary

- WLM Advantages
  - Responsive to system changes
  - Response based on business goals
  - Strives to utilize all resources
- WLM Drawbacks
  - Complex definitions
  - Hard to understand calculations
  - Many view as a black box

My goal was to take the mystery out of WLM. I hope you will agree that has been accomplished in this presentation.

Session A03

Workload Manager for the DB2 Professional

**Tom Moulder**

TREX Associates, Inc.

tom.moulder@trexassociates.com

GoFurther

