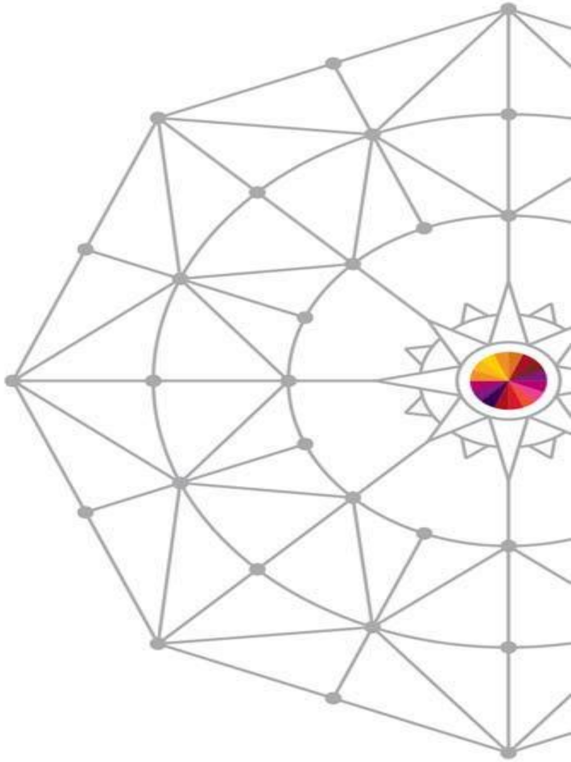
z/OS WLM – Are You Set Up to Fail?

Bradley Snyder

IBM

Summer Share - Pittsburgh

August 7, 2014



#SHAREorg

SHARE in Pittsburgh 2014

Agenda

- Setting the right goal
- IBM Recommendations not being followed
  - Service Definition Coefficients
  - Classifying new work and SYSOTHER
  - SYSSTC Observations and recommendation
- CICS/IMS Velocity vs. Response Time Goals
- Common mistakes
  - Bad Service Class Example

Priority Number One - Setting the Right Goal!!

- Resource allocation is guided by the Workload Manage Policy (WLM)
- Decisions by WLM are done specifically for how well a service class period is doing relative to the goal specified
  - Examples:
    - CPU dispatching Priority
    - MPL
    - Partition Weight (with Intelligent Resource Director – IRD)
- Unexpected/Unwanted results can happen when goal is set inappropriately
- All goals should be achievable and realistic
- Review Workload Activity Report (Session tomorrow!)
- WLM Monitors the Performance Index (PI) to determine what actions should be taken
  - PI of 1.0 means workload is exactly meeting the goal specified
  - PI of greater than 1.0 means workload is missing its goal
  - PI of less than 1.0 means workload is beating its goal

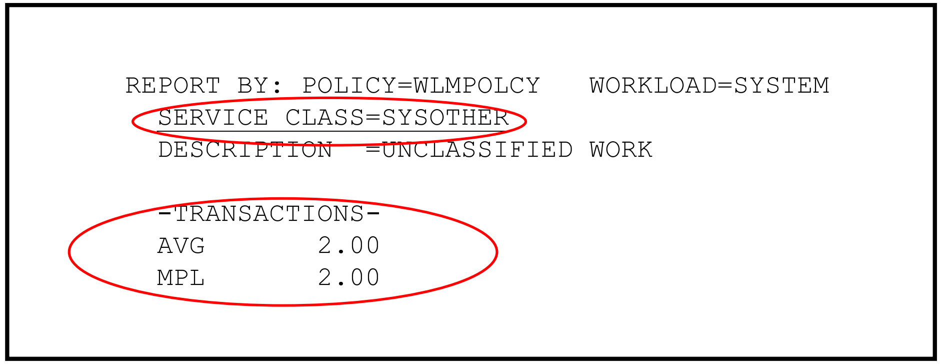
- Periodically review workloads during peak periods to determine if goals are set appropriately

Service Definition Coefficients

- Following are typical values
  - CPU – 1.0, SRB – 1.0, IOC – 0.1, MSO – 0.0000
  - MSO needs to be 0, long running address spaces (CICS, IMS, DB2) will accumulate large amounts of service, the reports will be skewed
  - MSO is a calculated value
    - Storage used while CPU Cycles are being used
    - With no storage contention, old frame pages may stay for multiple days
  - Many times see IOC set to 0.5
    - Site preference, how much weight to give I/O to period aging values
    - With value of 0.1, easier to compare units to CPU/SRB

- When changing values, need to evaluate impact on multi-period service classes

Set Proper Defaults for New Work



```
REPORT BY: POLICY=WLMPOLCY   WORKLOAD=SYSTEM
  SERVICE CLASS=SYSOTHER
  DESCRIPTION  =UNCLASSIFIED WORK

  -TRANSACTIONS-
  AVG        2.00
  MPL        2.00
```

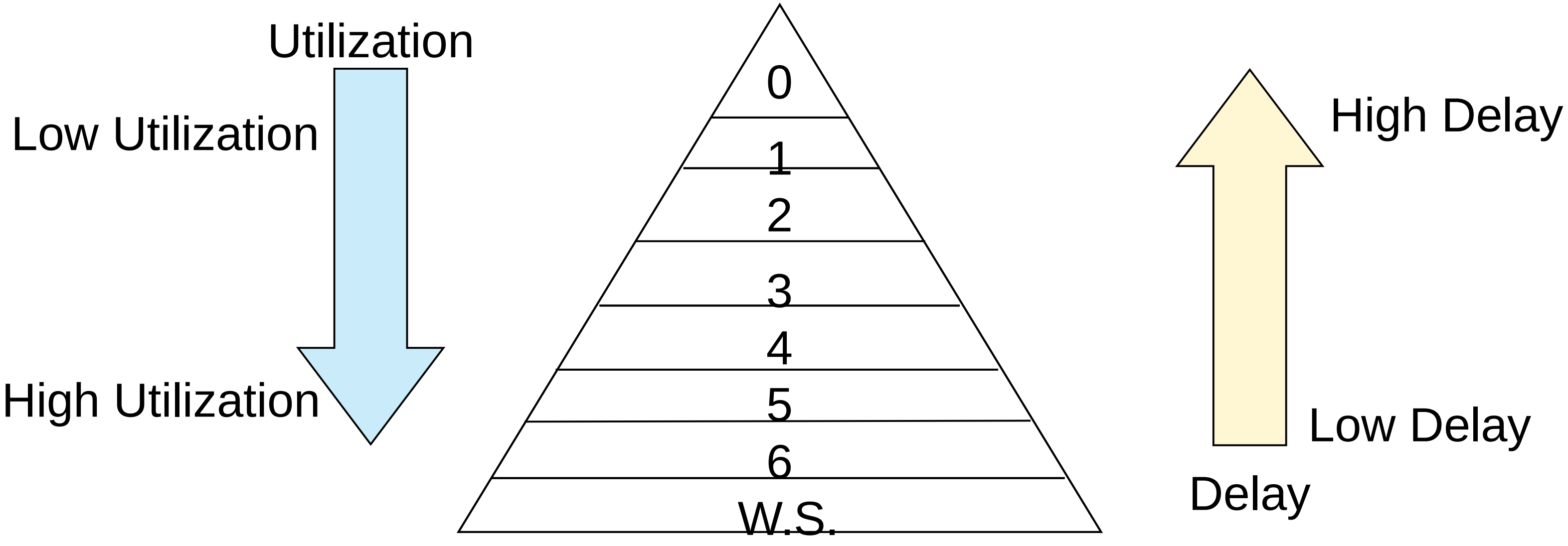
- Unclassified work will default to one of two places
- Started Tasks default to SYSSTC
  - New started tasks may dominate the system
- All other work defaults to SYSOTHER
  - New work will get no service in busy system
- Recommendation:
  - Under Started Task Rules, have ‘NEWWORK’ as your default service class to prevent new tasks from running in SYSSTC
    - Give NEWWORK a medium importance and velocity
  - In many areas it is a good idea to have a default for all classification rules
    - Should only be active if work is otherwise unclassified
    - Ie. CICSDFLT, BATDFLT, etc.
  - Monitor default service classes for any activity
  - If defaults have activity, or even have resident transactions, work to classify work as soon as possible
    - Many times see TCPIP Enclaves in SYSOTHER

User Work in SYSSTC

- Many times we have seen work classified SYSSTC that should not be there
  - DB2MSTR, DB2WLM, some CICS address spaces, MQ MSTR and CHIN address spaces, etc.
  - System Programmers like their TSO IDs to be here
- SYSSTC has second highest dispatching priority. Any long running tasks that end up here will block any other work
- Recommendations for SYSSTC
  - DB2IRLM and IMS IRLM – Lock manager needs high dispatching priority in order to let work flow properly through the system
  - “Emergency” TSO ID – Only one TSO ID should be defined to SYSSTC
    - All other TSO IDs should be grouped together, no special high priority service class for system programmers or management

Importance Levels

- Many customers not effectively using importance levels
- Policies with over half of all workload at IMP=1
  - Best recommendation is to use all five importance levels in order to differentiate work
- Policy must be clear enough so that in times of contention you know which workloads will get delay
  - There will be some importance level where delays are not acceptable



Use of CPU Critical

- Limit use of CPU critical
  - Intended to be used when rapid workload shifts happen regularly and WLM will not be fast enough in adjusting priorities
- CPU Critical only protects that work from lower importance work, no protection from work at same or higher importance, better to have the right goal
  - A service class with importance of 2 and CPU critical set to YES is not treated as more important than other IMP=2 work
- When running CICS/IMS with response time goals, and CPU critical is necessary, designate both regions and transactions as CPU critical
  - Handles idle periods and restarts

Other Issues That Can Cause Problems

- Use of Average Response time Goals instead of Percentile
  - Use of percentile goals negates impact of outliers
  - Work should have even distribution for average response time goals to work well
- Unachievable/Unrealistic velocity goals – ie. goal of 90
  - Check velocities of SYSTEM and SYSSTC to determine highest achievable velocities
  - Smaller n-way partitions will necessitate lower velocity goals
- Do not want some regions doing region management, and some transaction management
  - In workload activity report, see service class SERVER serving CICSPRD and SERVER service classes
- Server service classes should be separated from other service classes

CICS and IMS – R.T. or Velocity Goal?

- Which is the better way to manage online work?
- Remember, WLM will set dispatching priority for the region
  - Need to have the CICS and IMS Regions dispatched properly
  - CICS and IMS have their own internal routines to decide which to run within their regions
  - If transactions 0101 and PRD1 both run in AOR1, CICS will decide which to dispatch, **NOT** Workload Manager
- So the ‘right’ goal depends on your environment

Velocity Goals for CICS and IMS

- Velocity goals are acceptable for environments with only one partition, or sysplexes with similar sized partitions and on the same kind of hardware
  - A sysplex with a 4-way and a 20-way may not be a good candidate
  - Want to put enough work into each service class that WLM sampling gives a good view of usage
- Can be used when the nature of online transactions does not make classification of transactions goals reasonable
  - Vastly different types of transactions would skew response time distribution data
    - Especially if transaction types change over time
  - Two transactions service classes in same region will get same dispatching priority
- Velocity goals do need to be monitored and may need to be adjusted during any processor changes
  - Processor upgrades, LPAR definition changes, etc.

Response Time Goals for CICS and IMS

- 3 major advantages of response time goals
  - Easier to understand and can be set to a business SLA
  - Normally no need to change when environment is changed
  - Can use same goal across entire parallel sysplex, regardless of individual partition size/speed
- Too many policies have too many response time goals defined in policy
  - Okay only if each region only runs one type of transaction
  - Keep it Simple!
- Recommendation:
  - Strive for 1 to 2 response time service classes (Fast, Other)
  - Set goal for dominate transaction(s)
  - Manage to stable population
    - If you know 10% or transactions will never meet goal, take that into account when setting the goal,

Velocity Goals – Understand Distribution

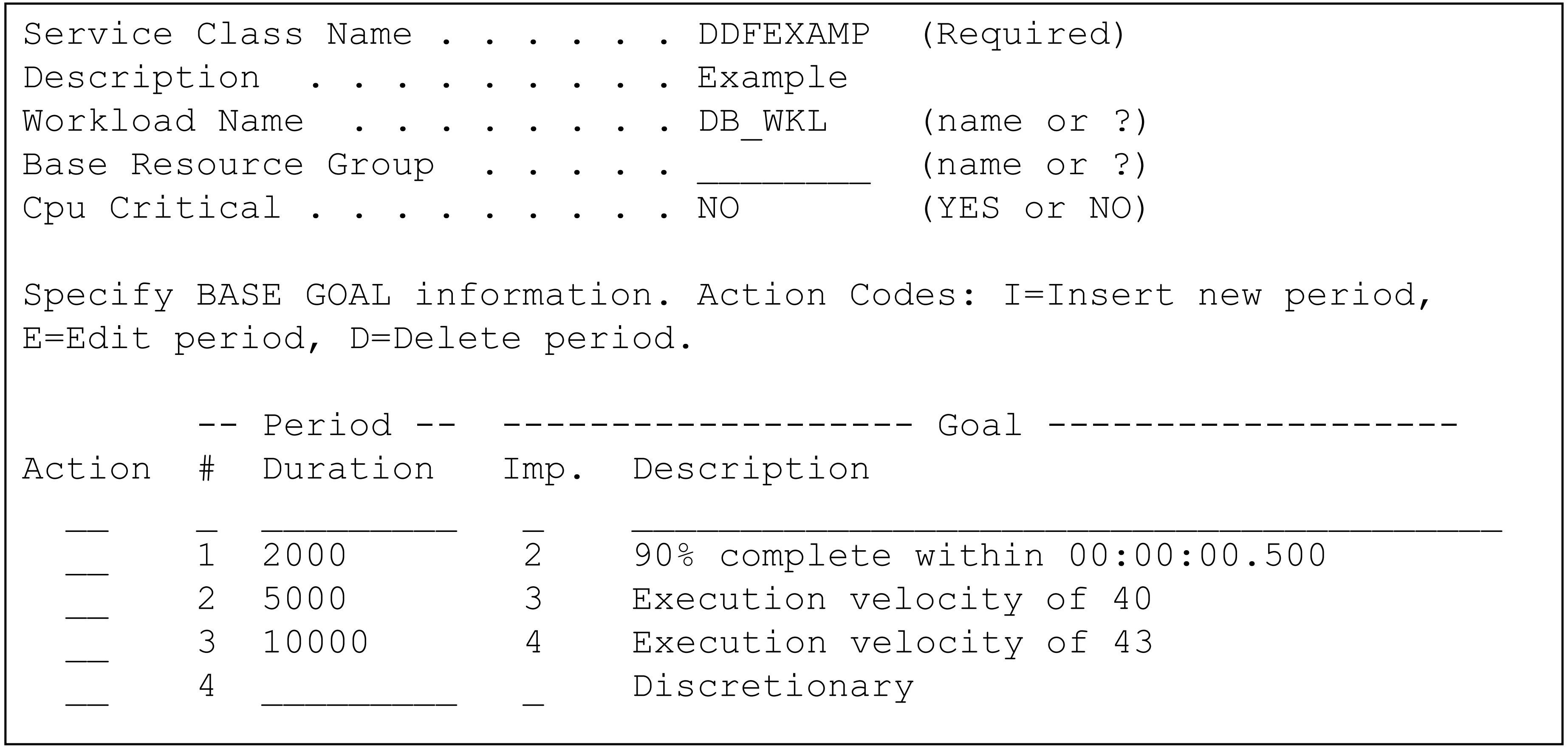
- When running with velocity goals, for online work create report classes or use new functionality in z/OS V1R13 support
  - Will give transaction level information and statistics
- Advantages
  - Get ended transaction rate
  - Average transaction time
  - Response time breakdown buckets (only in V1R13)
- Review service classes to determine possibility of migrating if desired

Setting Service Class Definitions

- Use service class called DDFEXAMP to highlight many common issues
- Issues common to multi-period service classes
  - DDF
  - TSO
  - Batch
  - Enclaves

The DDFEXAMP Service Class

- Four period service class with response time and velocity goals



```
Service Class Name . . . . . . DDFEXAMP  (Required)
Description  . . . . . . . . . Example
Workload Name  . . . . . . . . DB_WKL    (name or ?)
Base Resource Group  . . . . . ________  (name or ?)
Cpu Critical . . . . . . . . . NO        (YES or NO)

Specify BASE GOAL information. Action Codes: I=Insert new period,
E=Edit period, D=Delete period.

        -- Period --  ------------------- Goal -------------------
Action  #  Duration   Imp.  Description

  __    _  _________   _    ________________________________________
  __    1  2000        2    90% complete within 00:00:00.500
  __    2  5000        3    Execution velocity of 40
  __    3  10000       4    Execution velocity of 43
  __    4  _________   _    Discretionary
```

DDFEXAMP Issue Number One

- Multiple Service Class Periods
  - Recommendation is to use as few multi-period service classes as possible
  - Keep within the rule of thumb of 25-35 active service class periods
  - Each period needs to have sufficient samplings and ended transactions to give accurate view of workload
- Typical assumption is longer running work is less important
  - Low importance work can hold resources needed by high importance work
  - If work drops into lower periods, less access to CPU.
- For Batch and Websphere work
  - Batch and Websphere queue time delay only accumulates in first period
  - May have more success with single period workloads
- Proper use of multiple periods
  - For some work, many times unable to have all work in single period due to mixture
  - Attempt to keep number of periods to a max of 2
  - Check Standard Deviation of response time in Workload Activity Report

Multiple Period Considerations

- Workload Manager makes more statistically valid decisions when there are more samples in a service class period
- Review RMF Workload Activity Report for service class utilization by period
- If one period of a multi-period service class is always much smaller than the other periods, consider consolidation
- For example, typical utilization pattern of three period service class
  - SCLAS Period 1 – APPL% =   71.1
  - SCLAS Period 2 – APPL% =   0.37
  - SCLAS Period 3 – APPL% = 138.0
- In this case, period 2 should either be combined with 1 or 3

DDFEXAMP Issue Number Two

- Comparing periods 2 and 3, velocity goals are too close
  - Period 2 Velocity of 40, Period 3 Velocity of 43
  - Indication of trying to micro-manage goals
- Many times customers will have two or more service classes with velocity goals that are too close together
- Workload Manager does not manage a velocity, it adjusts a dispatching priority and observes the resulting velocity
  - Different dispatching priorities can result in wide variety of achieved velocities
- Velocity goals should be set with a difference of at least 10 to be effective and meaningful
  - Any service classes with goals closer than 10 should be evaluated to be combined into one service class

DDFEXAMP Issue Number Three

- Discretionary goal used
- When discretionary goals are active, discretionary goal management may affect other production service classes
  - Can see capping on other service classes with velocity less of 30 or less or response time goal over 1 minute, and PI less than 0.71
- Work that holds resources should be in managed service class
  - Some customers may be okay with longer running batch in discretionary
  - Discretionary is first work to see delay, should have no SLA associated for work with a discretionary goal
- Discretionary work and specialty processors
  - The ‘Needs Help’ algorithm will not cause discretionary work to run on general purpose CPs
- Only discretionary goals get mean time to wait

DDFEXAMP Issue Number Four

- Duration Values are not set appropriately
- Can only be checked by reviewing RMF Data
  - DDFEXAMP Period 1 duration of 2,000
  - From RMF:

```
REPORT BY:  POLICY=POL01      WORKLOAD=DB2         SERVICE CLASS=DDFEXAMP

  -TRANSACTIONS-   TRANS-TIME HHH.MM.SS.TTT  --DASD I/O--   ---SERVICE---
  AVG        78.97  ACTUAL                201  SSCHRT   1466  IOC            0
  MPL        78.97  EXECUTION             200  RESP      7.2  CPU      29852K
  ENDED     349080  QUEUED                  0  CONN      0.4  MSO            0
  END/S     387.87  R/S AFFIN               0  DISC      6.0  SRB            0
  #SWAPS         0  INELIGIBLE              0  Q+PEND    0.8  TOT      29852K
  EXCTD          0  CONVERSION              0  IOSQ      0.0  /SEC      33169
```

  - Divide total service by total ended transactions
    - Or service/sec divided by end/s
  - On average, transactions ended in period 1 used 85.5 service units
- Many times duration values were not adjusted when SDC were changed to current recommended values
  - CPU and SRB changed from 10.0 to 1.0

DDFEXAMP Issue Number Four

- When multiple periods are necessary, usually better for first period to have a more uniform set of transactions
  - Easier to set a proper response time goal if desired
- DDFEXAMP Period 1 has a goal of 90% of transactions completing in 0.5 seconds
  - Adjusting duration from 2,000 to 200 will allow for better management of short running transactions with a tighter goal
- This leads us too….

DDFEXAMP Issue Number Five

- PI of Service Class is 0.5. Indicates goal may be too loose.
- All goals must be checked to determine if they are set appropriately for business goals and average system performance
- Changes in technology and applications may change achievable performance level, and goal should be adjusted accordingly
- Note: In RMF, percentile response time PI will range from 0.5 to 4.0
  - Response time buckets range from half of goal to four times goal

```
----------RESPONSE TIME DISTRIBUTION----------
     ----TIME----    --NUMBER OF TRANSACTIONS--    -------PERCENT-------  0    10   20   30   40   50   60   70   80   90   100
     HH.MM.SS.TTT    CUM TOTAL         IN BUCKET    CUM TOTAL    IN BUCKET |....|....|....|....|....|....|....|....|....|....|
  <  00.00.00.250       349080            348929          100        100.0 >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
  <= 00.00.00.300       349080                 0          100          0.0 >
  <= 00.00.00.350       349080                 0          100          0.0 >
  <= 00.00.00.400       349080                 0          100          0.0 >
  <= 00.00.00.450       349080                 1          100          0.0 >
  <= 00.00.00.500       349080                 0          100          0.0 >
  <= 00.00.00.550       349080                 3          100          0.0 >
  <= 00.00.00.600       349080                 0          100          0.0 >
  <= 00.00.00.650       349080                 0          100          0.0 >
  <= 00.00.00.700       349080                 2          100          0.0 >
  <= 00.00.00.750       349080                 5          100          0.0 >
  <= 00.00.01.000       349080                 0          100          0.0 >
  <= 00.00.02.000       349080                 0          100          0.0 >
  >  00.00.02.000       349080               151          100          0.0 >
```

Review

- Follow recommendations on SDC
- In debate of Response Time goals vs. Velocity goals, understand and use what is best for your environment
- Server service classes need appropriate even when transactions are running
- Separate Velocity goals by at least 10 each
- Use multiple periods sparingly
- Monitor PIs of all service classes
  - PI for Response time goals have range of 0.5 to 4.0
- For CICS/IMS transaction goals, have only 1 or 2

- And as always, keep number of active service class periods to a range of 25 to 35!!!