WLM Period Durations and Prepping for z/OS 2.5 SDC IOC=0

Peter Enrico

Email: Peter.Enrico@EPStrategies.com

Enterprise Performance Strategies, Inc.
3457-53rd Avenue North, #145
Bradenton, FL 34210
http://www.epstrategies.com
http://www.pivotor.com

Voice: 813-435-2297
Mobile: 941-685-6789

z/OS Performance Education, Software, and Managed Service Providers

Creators of Pivotor®

Enterprise Performance Strategies, Inc. ©

Contact, Copyright, and Trademark Notices

Questions?

Send email to Peter at Peter.Enrico@EPStrategies.com, or visit our website at http://www.epstrategies.com or http://www.pivotor.com.

Copyright Notice:

Trademarks:

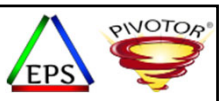
Enterprise Performance Strategies, Inc. presentation materials contain trademarks and registered trademarks of several companies.

The following are trademarks of Enterprise Performance Strategies, Inc.: **Health Check®, Reductions®, Pivotor®**

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries: IBM®, z/OS®, zSeries® WebSphere®, CICS®, DB2®, S390®, WebSphere Application Server®, and many others.

Other trademarks and registered trademarks may exist in this presentation

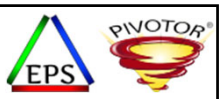
Abstract



- **WLM Period Durations and Prepping for z/OS 2.5 SDC IOC=0**
  - For z/OS 2.5, the WLM Service Definitions Coefficients (SDCs) have finally been eliminated. For years, we have been recommending CPU and SRB set to 1, and MSO set to 0. As of z/OS 2.5, the hardcoded values will be CPU and SRB to 1, MSO and IOC to 0. For some installations, this will require tweaking of the durations of their multiple period service classes.
  - During this webinar, Peter Enrico will discuss this change in more detail. He will also provide an simple exercise so that will help you prepare for z/OS 2.5 by adjusting your period durations now. So, join this webinar knowing your current service definition coefficients and period durations. This will be instructional.

3

EPS: We do z/OS performance…



- We love to bring you these complimentary webinars and to teach you how to do what we do!
- But remember, we would love to work with you!
  - ***Pivotor - Performance reporting and analysis of your z/OS measurements (i.e. SMF, etc.)***
    - Not just reporting, but coset effective analysis-based reporting based on our expertise
  - ***Performance Educational Workshops (while analyzing your own data)***
    - Essential z/OS Performance Tuning
    - Parallel Sysplex and z/OS Performance Tuning
    - WLM Performance and Re-evaluating Goals
  - ***Performance War Rooms***
    - Concentrated, highly productive group discussions and analysis

4

Like what you see?

- Free z/OS Performance Educational webinars!
  - The titles for our Fall 2020 webinars are as follows:
    - ✓ *Advantages of Multiple Period Service Classes*
    - ✓ *Refresher of SMF 113 Processor Cache Counters and Concepts*
    - ✓ *WLM SYSTEM / SYSSTC*
    - ✓ *Using Long Term Reporting: Pivotor Past Perfect*
    - ✓ *Catching Up with Bob Rogers*
      - *Specific Topic to be decided? Suggestions?*
    - ✓ *System Recovery Boost (SRB): The Turbo Button for z/OS*
    - ✓ *WLM Period Durations and Prepping for z/OS 2.5 SDC IOC=0*
    - *ETR vs ITR*
    - *Inventory Those Managed Resources and Workloads*
    - *Data Visualization - Pivotor People Pontificate*
  - Let us know if you want to be on our mailing list for these webinars
- If you want a free cursory review of your environment, let us know!
  - We're always happy to process a day's worth of data and show you the results
  - See also: http://pivotor.com/cursoryReview.html

5

Quick Refresh of Multiple Period Service Classes

6

Purpose of WLM Service Classes

- Why Service Classes?
  - To group work with similar performance characteristics for WLM management
  - … but this also means that service classes are used to separate unlike work away from each other
- Examples of similar characteristics includes:
  - Work types
  - Resource requirements
  - Workload objectives
  - Business Requirements

STCHIGH Service Class Period 1 Goal = Velocity 50 Importance 1 RGRP = none	**DB2PRD Service Class** Period 1 Goal = Velocity 60 Importance 1 RGRP = none
SAPHIGH Service Class Period 1 Goal = Velocity 50 Importance 1 RGRP = none	**SAPMED Service Class** Period 1 Goal = Velocity 60 Importance 2 RGRP = none
CICSTORS Service Class Period 1 Goal = Velocity 60 Importance 1 RGRP = none	**CICSAORS Service Class** Period 1 Goal = Velocity 60 Importance 2 RGRP = none

7

But separation of work is not always possible

- As mentioned, work with similar performance characteristics is grouped into a Service class
- But at WLM classification, sometimes not enough is known about the work to separate the work into different service classes
  - Yet these different types of work should be managed separately
- Example: when the transaction starts it may not be known:
  - … if the transaction will have a long or short response time
  - … if transaction will be a large resource consumer or not
  - … if the transaction will absorb service very quickly or if uses service more intermittently

PRODBAT Service Class
- Long or short?
- Big CPU consumer?
- CPU intensive or I/O intensive?

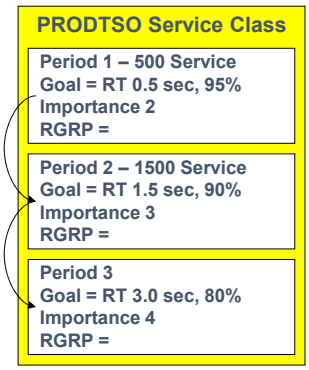
DDFPROD Service Class
- Long or short?
- Big CPU consumer?
- CPU intensive or I/O intensive?

TSOPROD Service Class
- Long or short?
- Big CPU consumer?
- CPU intensive or I/O intensive?

8

Reason for Multiple Period Service Classes

- Because some work may have variable resource requirements, service classes can be defined with multiple periods
  - Periods are a way of defining different goals for work depending on the amount of resources the work consumes
- Typically, periods are used to
  - Give shorter transactions more aggressive goals
  - Give longer transactions less aggressive goals
- Each period consists of
  - Goal and importance
  - Duration (except for last period)
- Durations for period 'aging'
  - The amount of resources, in service, that work consumes
  - As work consumes service and consumption exceeds duration, work is transitioned to the next period and managed to goal of next period
  - Way of aging transactions



9

What is measured for a duration?

- As transactions are processed, they consume system resources
  - The measure of resource consumption is service units
    - CPU and SRB service units (i.e. processor)
    - I/O service units
    - MSO service units (i.e. storage)
- Traditionally, durations are in terms of 'service' and not 'service units'
  - Service is service units weighted by service definition coefficients (SDCs)
  - When duration is set for a period, the service consumed determines period switch
  - Why weight? Historical…

$$\text{Service} = \begin{pmatrix} (\text{CPU SDC} * \text{CPU Service Units}) \\ + (\text{SRB SDC} * \text{SRB Service Units}) \\ + (\text{IOC SDC} * \text{IOC Service Units}) \\ + (\text{MSO SDC} * \text{MSO Service Units}) \end{pmatrix}$$

10

Understanding Service Units

- CPU Service Units
  - Task (TCB) and preemptible SRB execution time multiplied by an SRM constant (SU/Sec) which is CPU model dependent
  - Also includes time used by address spaces in cross memory mode
- SRB Service Units
  - Non preemptible Service Request Block time for both local and global SRBs, multiplied by an SRM constant which is CPU model dependent
  - Also includes time used by address spaces in cross memory mode
- I/O Service Units (also known as IOC)
  - Measurement of individual dataset I/O activity and JES spool reads and writes for all datasets associated with an address space or enclave
  - Calculated using I/O block (EXCP) counts or device connect time
- Storage Service Units (also known as MSO)
  - Measurement of central storage usage, but scaled to attempt to bring in line with CPU capacity
    - Calculated as (Central Page Frames) x (CPU SU) x 1/50
  - Does not include central storage frames used by the a caller while referencing the private area of a target address space

11

Understanding Service Definition Coefficients

- Service Definition Coefficients (also known as SDCs)
  - Set in the WLM service definition
  - Installation defined coefficients used to assign additional weight to one type of service relative to another type of service
  - Allows an installation to specify which type of resource consumption should be emphasized in the calculation for service rates

```
Coefficients/Options  Notes  Options  Help
-------------------------------------------------------------------------------
              Service Coefficient/Service Definition Options
Command ===> ____________________________________________________________

Enter or change the Service Coefficients:

CPU  . . . . . . . . . . . . . . 1.0      (0.0-99.9)
IOC  . . . . . . . . . . . . . . 0.5      (0.0-99.9)
MSO  . . . . . . . . . . . . . . 0.0      (0.0000-99.9999)
SRB  . . . . . . . . . . . . . . 1.0      (0.0-99.9)

Enter or change the service definition options:

I/O priority management  . . . . . . . . YES  (Yes or No)
Enable I/O priority groups . . . . . . . YES  (Yes or No)
Dynamic alias management . . . . . . . . YES  (Yes or No)
```

12

History of Service Definition Recommendations

- Default SDCs since about 197x or 198x
  - CPU=10, SRB=10, IOC=5, MSO=1
  - Remember, once upon a time CPU, memory, and I/O bandwidth were so restricted and limited that you needed to really control large consuming workloads by aging them to lower periods and lower priority controls
- MVS Defaults and commonly used SDCs since about 199x
  - CPU=10, SRB=10, IOC=5, MSO=0
    - Some die-hards would still set MSO=0.0001
  - Central storage became so much more abundant. The mere act of adding central storage to your system would cause workloads to consume more storage, and transactions on systems with MSO non-zero would result in transactions moving to lower periods very rapidly even though the work did not change
    - Setting MSO = 0 eliminated this problem

History of Service Definition Recommendations

- Recommended values since about 2000
  - CPU=1, SRB=1, IOC=-0.5, MSO=0
    - Note: Default values are still CPU=10, SRB=10, IOC=5, MSO=0
    - There were still some die-hards with MSO non-zero such as MSO=0.0001
  - This was due to the rising popularity of Workload License Charges, and using the standard service unit measurement for capping resource group maximums for capping (which were never scaled by SDCs

History of Service Definition Recommendations

- Recommended values by EPS since about 2018 (maybe earlier)
  - CPU=1, SRB=1, IOC=0. MSO=0
  - Summary of reasoning: Aging a transaction based on I/O no longer made much sense since I/O priority management mattered much less due to advent of PAVs, and most I/O processing is also outside the z/OS operating system. So why age a workload based on its I/O characteristics. It is CPU that matters.
- z/OS 2.5 the SDCs go away, and the values will default as follows
  - CPU=1, SRB=1, IOC=0. MSO=0
  - Basically, it is durations are now based on CPU and SRB service units, and not longer based on the concept of 'service'.

Going from CPU=1,SRB=1,IOC=xx,MSO=0

IBM's z/OS 2.5 Migration Step

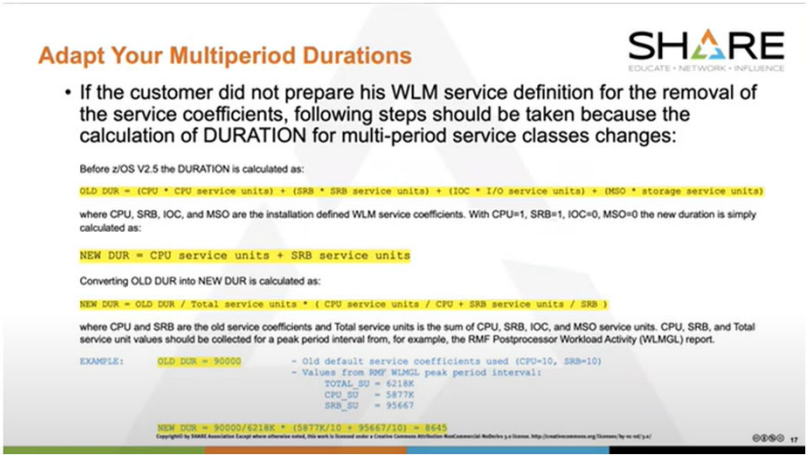
The following is an excerpt from SHARE presentation:
PERFORMANCE INFRASTRUCTURE IMPROVEMENTS IN Z/OS V2.5 WLM
Presenter:
ANDREAS HENICKE (IBM WLM)

Presentation discusses the z/OS 2.5 migration steps suggested to migrate your period durations prior to migrating to z/OS 2.5.

Basically, IBM is suggesting to take CPU and SRB 'service', divide by your current SDCs to convert to 'service units'. Then take the sum of those values and multiple them by the ratio of current duration to service consumed.

Or put a little simpler…
Blah, blah, blah…

Feel free to take this approach, but a bit to complicated for me.



Adapt Your Multiperiod Durations

SHARE
EDUCATE • NETWORK • INFLUENCE

- If the customer did not prepare his WLM service definition for the removal of the service coefficients, following steps should be taken because the calculation of DURATION for multi-period service classes changes:

Before z/OS V2.5 the DURATION is calculated as:

```
OLD_DUR = (CPU * CPU service units) + (SRB * SRB service units) + (IOC * I/O service units) + (MSO * storage service units)
```

where CPU, SRB, IOC, and MSO are the installation defined WLM service coefficients. With CPU=1, SRB=1, IOC=0, MSO=0 the new duration is simply calculated as:

```
NEW_DUR = CPU service units + SRB service units
```

Converting OLD DUR into NEW DUR is calculated as:

```
NEW_DUR = OLD_DUR / Total service units * ( CPU service units / CPU + SRB service units / SRB )
```

where CPU and SRB are the old service coefficients and Total service units is the sum of CPU, SRB, IOC, and MSO service units. CPU, SRB, and Total service unit values should be collected for a peak period interval from, for example, the RMF Postprocessor Workload Activity (WLMGL) report.

```
EXAMPLE:    OLD_DUR = 90000        - Old default service coefficients used (CPU=10, SRB=10)
                                   - Values from RMF WLMGL peak period interval:
                                        TOTAL_SU = 6218K
                                        CPU_SU   = 5877K
                                        SRB_SU   = 95667

NEW_DUR = 90000/6218K * (5877K/10 + 95667/10) = 8645
```


17

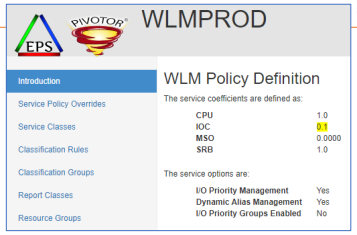
Peter's Approach to Migrating SDCs to New z/OS 2.5

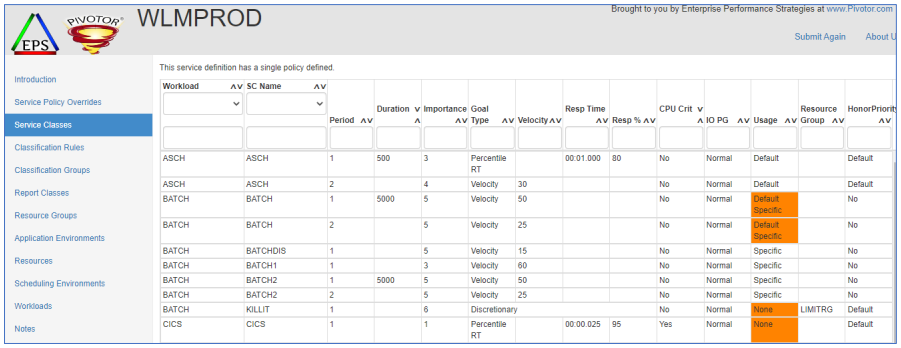
- My general approach is a follows:
  1. Determine your current SDCs
  2. Determine your current multiple period service classes
     - Most likely multiple periods are only being used for the following interactive workloads or certain batch
       - TSO
       - Interactive OMVS
       - DDF
       - WAS CB
       - Batch (sometimes)
  3. Determine which multiple period service classes are consuming I/O service and how much
  4. Then ignore any sort of duration migration exercise for the following enclave workload types since these enclave workloads do not consider I/O service
     - DDF
     - WAS CB
     - So will be left with workloads such as eft with only TSO, interactive OMVS, and Batch,
  5. Revisit duration
     - Either start fresh (which should be done for many periods regardless of this change)
     - Ignore and accept
     - Tweak

18

Steps 1 and 2

1. Determine your current Service Definition Coefficients
2. Determine your current multiple period service classes



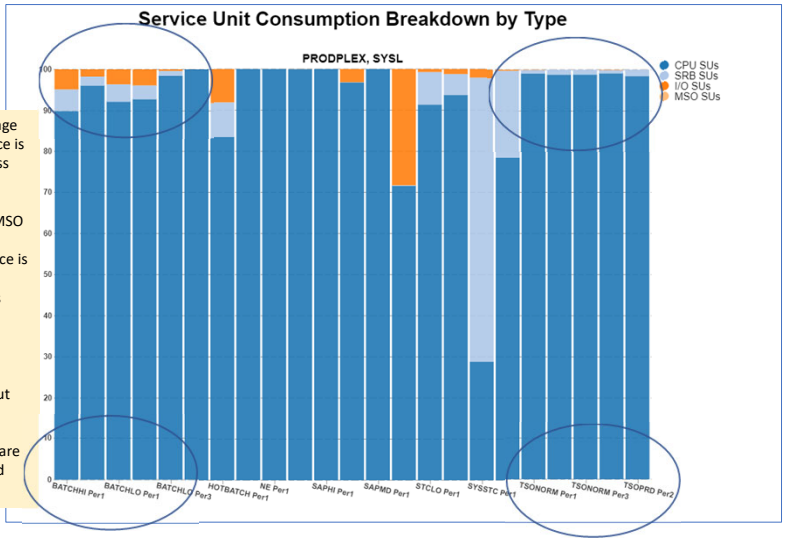


Determine with multiple period service classes are consuming I/O service

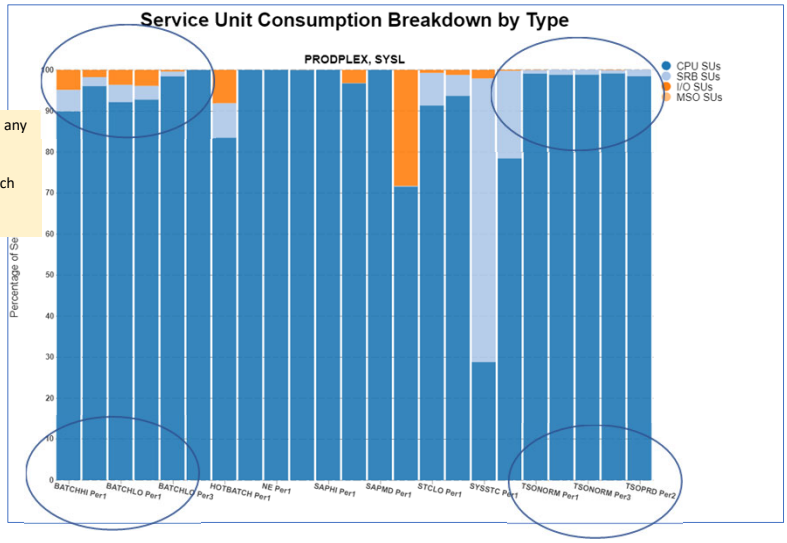
This report shows the percentage breakdown of how much service is consumed by every service class period with activity.

The orange is I/O service and MSO service.

- Chances are your MSO service is zero since MSO=0 has been recommended for 20+ years
- I/O service is what we care about
- Even then we only care about the TSO, DDF, OMVS, Batch periods.
  - Even then, we only care all but the last period



Determine with multiple period service classes are consuming I/O service



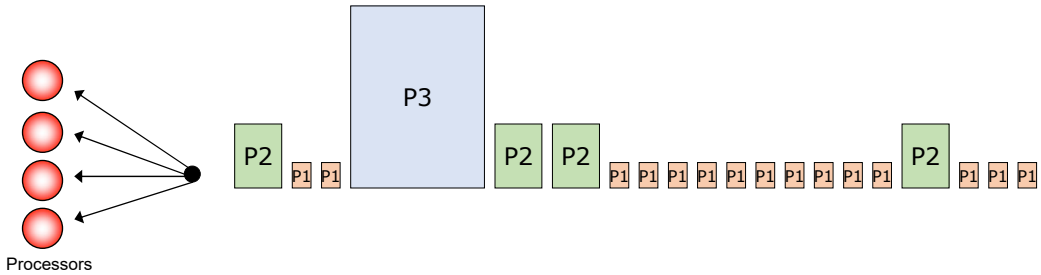
In this example, I would ignore any migration exercise.

Not enough I/O to warrant much of a migration.

21

If still need to migrate, then remember why Multiple Period

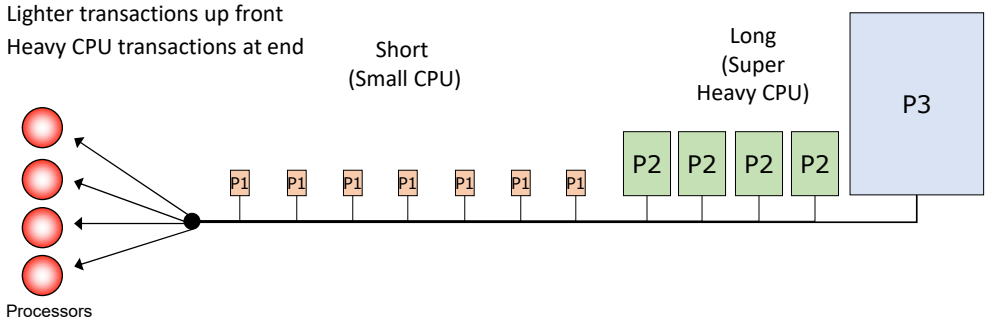
- Some interactive workloads have
  - Typical transaction short running, light CPU usage transactions
  - Atypical long running, CPU intensive transactions
- When a unit of work is dispatched to a processor, it is dispatched for a specific period of time called a dispatch interval
- Units of work are interrupted by pre-emption, and on z/OS there is a reduced preemption algorithm



22

Preferred CPU order of a workload

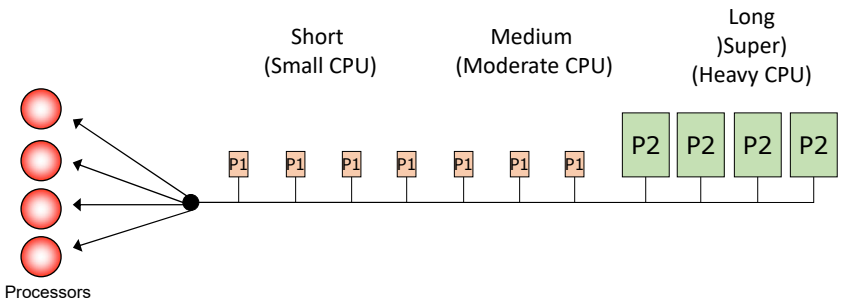
- Some interactive workloads have
  - Typical transaction short running, light CPU usage transactions
  - Atypical long running, CPU intensive transactions
- There is an advantage of recognizing those CPU intensive transactions and manage them at a lower importance level, and towards a more appropriate goal
  - Distribute CPU dispatching priority of transactions based on CPU demands
    - Lighter transactions up front
    - Heavy CPU transactions at end



23

To get the short transaction in and out ASAP

- Sometimes a workload can improve when we let all transactions run a high CPU dispatching priority when they start
  - But anything that is not super quick goes to a lower period
- So lets get the quick / short small CPU transactions in and out right away



24

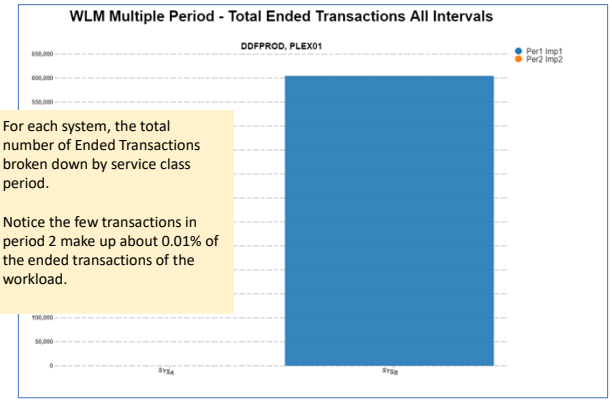
To get the short transaction in and out ASAP

- Sometimes a workload can improve when we let all transactions run a high CPU dispatching priority when they start
  - But anything that is not super quick goes to a lower period
- In this example, say the LPAR has an SU/sec constant of 50,000 SU/sec
- Then duration **250** would represents 0.005 seconds of CPU
- I made the decision that DDF trans run at a higher imp until they use up 5 ms of CPU.

```
Service-Class  Xref  Notes  Options  Help
------------------------------------------------------------------------------
                          Modify a Service Class              Row 1 to 4 of 4
Command ===> ____________________________________________________________

Service Class Name . . . . . : DDFPROD
Description  . . . . . . . . . Production DDF
Workload Name  . . . . . . . . DDFDB2     (name or ?)
Base Resource Group  . . . . . ________    (name or ?)
Cpu Critical . . . . . . . . . NO         (YES or NO)
I/O Priority Group . . . . . . NORMAL (NORMAL or HIGH)
Honor Priority . . . . . . . . NO (DEFAULT 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  250        1      95% complete within 00:00:00.015
   __    2             3      90% complete within 00:00:00.150
******************************* Bottom of data ********************************
```

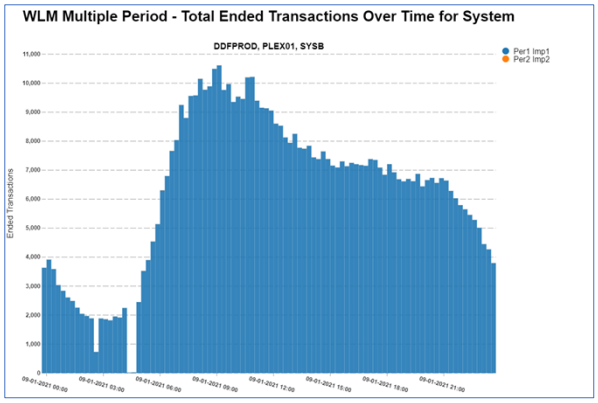
25

DDFPROD – Ended Transactions by Period

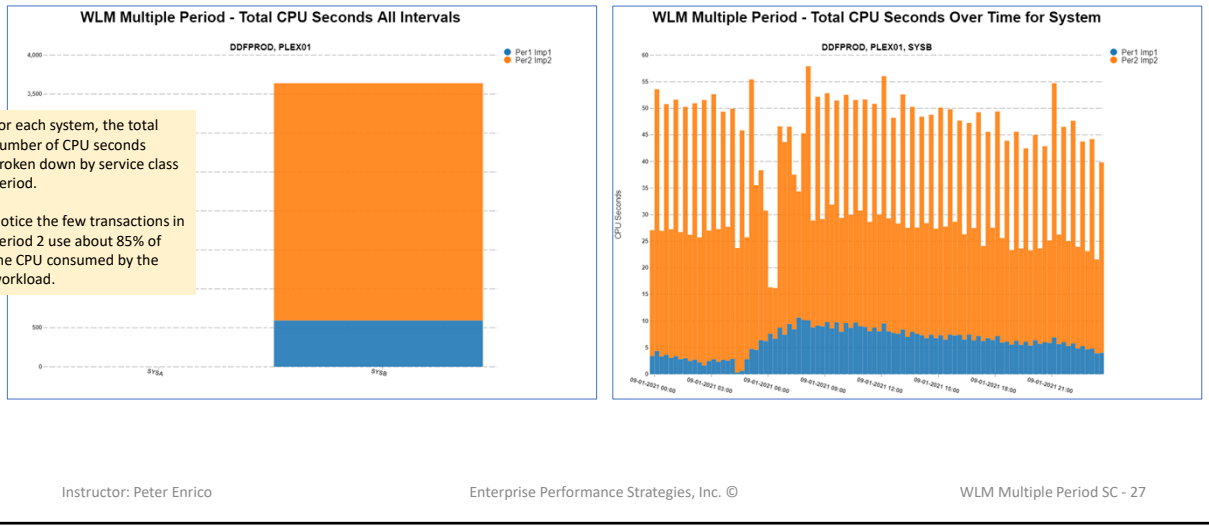


For each system, the total number of Ended Transactions broken down by service class period.

Notice the few transactions in period 2 make up about 0.01% of the ended transactions of the workload.



26



To assist in latent demand management

- Sometimes in heavy CPU constrained environments when there is just too much work running in certain workload periods, it is best to force the distribution of work
- In this example, say the LPAR has an SU/sec constant of 50,000 SU/sec
- I decided the duration
  - 50,000 represents 1 seconds of CPU
  - 200,000 represents 4 seconds of CPU
  - So, anything using more than 5 seconds of CPU will run in period 3

```
Service-Class  Xref  Notes  Options  Help
------------------------------------------------------------------------------
                        Modify a Service Class              Row 1 to 4 of 4
Command ===> ____________________________________________________________

Service Class Name . . . . . : PRDBATCH
Description  . . . . . . . . . Production Nighttime Batch
Workload Name  . . . . . . . . BATCH      (name or ?)
Base Resource Group  . . . . . ________  (name or ?)
Cpu Critical . . . . . . . . . NO        (YES or NO)
I/O Priority Group . . . . . . NORMAL (NORMAL or HIGH)
Honor Priority . . . . . . . . NO (DEFAULT 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  50000       3    Execution velocity of 50
  __    2  200000      4    Execution velocity of 60
  __    3              5    Execution velocity of 60
****************************** Bottom of data ********************************
```

Summary

- Do not over think the migration to

  CPU=1,SRB=1,IOC=xx,MSO=0

- Of little or no concern for DDF and WAS workloads
- Of minor concern to TSO transactions
- Just concentrate on those that use up a lot of I/O service
  - And then revisit durations fresh base on CPU consumed

29