



WLM's Algorithms - How WLM Works



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



Creators of Pivotor®

Peter Enrico

Email: Peter.Enrico@EPStrategies.com

Enterprise Performance Strategies, Inc.

3457-53rd Avenue North, #145

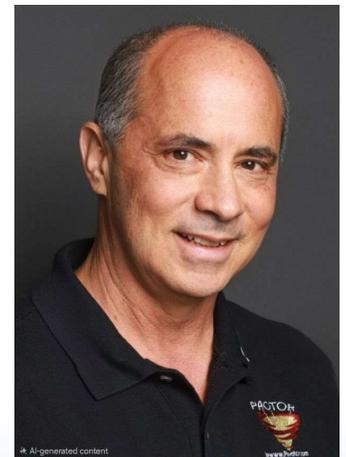
Bradenton, FL 34210

<http://www.epstrategies.com>

<http://www.pivotor.com>

Voice: 813-435-2297

Mobile: 941-685-6789



www.epstrategies.com

Contact, Copyright, and Trademarks



Questions?

Send email to performance.questions@EPStrategies.com, or visit our website at <https://www.epstrategies.com> or <http://www.pivotor.com>.

Copyright Notice:

© Enterprise Performance Strategies, Inc. All rights reserved. No part of this material may be reproduced, distributed, stored in a retrieval system, transmitted, displayed, published or broadcast in any form or by any means, electronic, mechanical, photocopy, recording, or otherwise, without the prior written permission of Enterprise Performance Strategies. To obtain written permission please contact Enterprise Performance Strategies, Inc. Contact information can be obtained by visiting <http://www.epstrategies.com>.

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's Algorithms - How WLM Works

Here at EPS, we regularly get questions along the lines of:

- How does WLM make decisions?
- What does WLM use to make its decisions?
- How can I tell what decisions WLM is making?
- How can these decisions be influenced by a WLM administrator?

During this webinar, *Peter Enrico*, an original WLM designer and developer, will discuss each of these questions, and he will provide the inside-scoop necessary to gain a better understanding of WLM. You are sure to not only to learn more about WLM in this webinar, but also how you, as a WLM administrator, can gain both insights and influence over WLM decision making.

EPS: We do z/OS performance...



- We are z/OS performance!
- **Pivotor**
 - Performance reporting and analysis of your z/OS measurements
 - Example: SMF, DCOLLECT, other, etc.
 - Not just reporting, but cost-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
- **MSU reductions**
 - Application and MSU reduction, as well as contract reviews
- **Information**
 - We present around the world, bi-weekly webinars, and participate in online forums

z/OS Performance workshops available



During these workshops you will be analyzing your own data!

- Essential z/OS Performance Tuning
 - March 30 – April 3, 2026 (4 days, excl Wednesday the 1st)
- WLM Performance and Re-evaluating Goals
 - June 22 – 26, 2026 (4 days, excl Wednesday the 24th)
- Parallel Sysplex and z/OS Performance Tuning
 - May 12-13 2026
- Also... please make sure you are signed up for our free monthly z/OS educational webinars! (email contact@epstrategies.com)

Like what you see? Winter / Spring 2026 Webinars



- Free z/OS Performance Educational webinars!
 - The titles for our Winter/Spring 2026 webinars are as follows:
 - ✓ *New Year's Resolutions for z/OS Performance and Capacity People*
 - ✓ *How WLM Makes Decisions*
 - *What I Learned about VSAM RLS SMF Data*
 - *z/OS Performance Spotlight: Some Top Things You May Not Know*
 - *Building a Strong Foundation When You're New to z/OS Performance*
 - *Wait...Do We Need to Re-evaluate our WLM Goals?*
 - *z15 to z16 to z17 – What has changed?*
 - *Evaluating in the Mainframe Environment*
 - *Managing Workload Manager: Multiple Sysplexes and Asymmetric Sysplexes*
 - *Introduction to z Processor Measurements*
 - *(more to be announced)*
- 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>

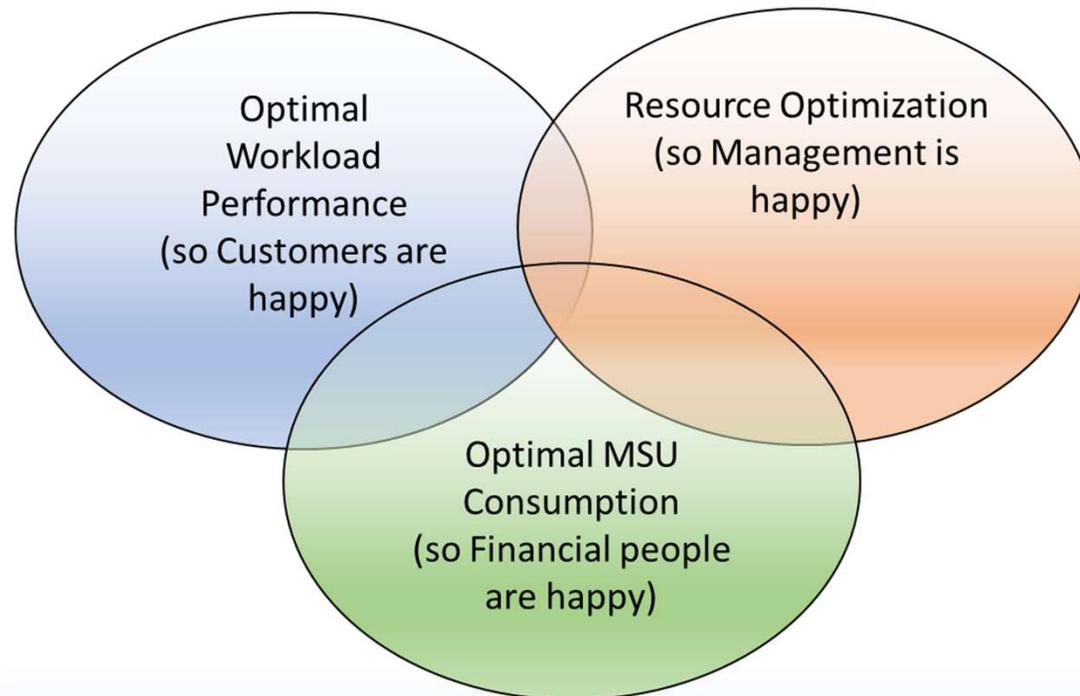


How does WLM make decisions?

The Performance Balancing Act



- Performance on z/OS is about finding an optimal balance among 3 areas



WLM Approach to Workload Management



- WLM dynamic workload management decisions based on
 - Degradation analysis
 - Dynamic workload characterization
- Input into decisions includes understanding of
 - Resource requirements of workloads (based on state sampled data)
 - Current state and usage of system resources (CPU, storage, I/O, etc.)
 - How well all work in system is meeting goals?
- Projection of the effect a change will have
 - WLM does not use **Rules of Thumb**
 - Models decisions using plots and histories of past performance
- Primary approach
 - Don't do anything stupid!

WLM Algorithm Phases



- There are two primary phases of WLM algorithms
- Policy Adjustment (PA)
 - Done approximately every 10 seconds (AKA 'PA interval')
 - Objectives include:
 - Summarize state of system and resources
 - Help work meet goals by setting resource controls
 - Housekeep resource controls that may be out of date
- Resource Adjustment (RA)
 - Done approximately every 2 seconds (AKA 'RA interval')
 - Objectives include:
 - improve efficiency of system resources
 - avoided if at the expense of goals

WLM Policy Adjustment – 'The Loop'



- Summarize data for state of the system and workloads
- Select a receiver period (highest importance missing goal the most)
- Find the receiver's largest bottleneck
 - Determine fix for receiver's bottleneck
 - Determine if needed resources can be gotten from unused resources
 - Find donor(s) of resource that receiver needs
 - Assess effect of reallocating resources from donor(s) to receivers
 - If allocation has both net and receiver value
 - Then commit change
 - Else don't make change
 - If reallocation was done then jump to Exit and allow change to be absorbed
 - If reallocation was not done then try to fix receiver's next largest bottleneck
- If cannot help receiver then look for next receiver (highest importance missing goal the most)
- Exit
 - Housekeep current set of controls



Key WLM Design Point:

The best predictors of the future
are the behaviors of the past.



What does WLM use to make its decisions?

Summarize State of System & Workloads



- Like any performance analyst, WLM collects and uses a variety of measurements as input to its algorithm decisions
 - System related measurements
 - CPU, I/O, Storage, logical resources such as batch initiators, application environments, etc.
 - Service Class period measurements
 - Using and Delay state samples for all work in period
 - Resource usage
 - Transaction throughput, response times, and velocities
 - How well goal is being met
 - Application environment queue lengths
 - Sysplex related measurements
 - WLM maintains an 'awareness' on local system of how well work on other systems in Sysplex are doing
 - Resource usage for Intelligent Resource Director

WLM collects Using and Delay samples



- The WLM state sampler collects Using and Delay state data
 - Sampling is the primary function of the WLM address space
 - The sampler allows WLM to gain insight into why work is delayed, and what work is using the resources that may be causing the delays
- State sampling runs once per 1/4 second
 - Samples are amassed into histories to give WLM insight into state samples over time
- It is very important to note that WLM is most interested in samples of the Using and Delay states of work for resources that WLM has control over

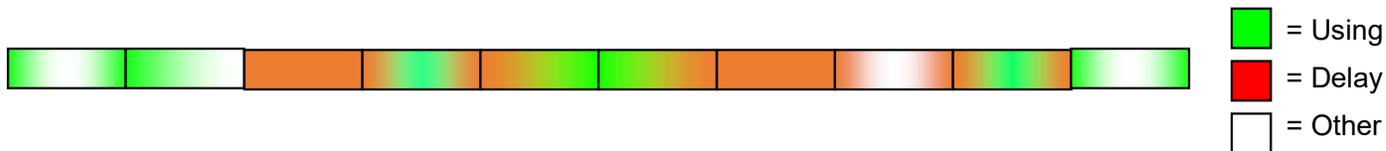
Number of Samples	140	38	75	30	0	0	200	0	5	0	0	0	0
	CPU Using	I/O Using	CPU Delay	I/O Delay	CAP Delay	Swap In Delay	MPL Delay	QMPL Delay	Private Area Paging	Common Area Paging	Xmem Area Paging	VIO Area Paging	HSP Area Paging
State Sampled	Using		Non-Storage Delays					Storage Delays					

WLM uses the samples to calculate velocities



- The WLM velocity formula is based on known using and delay samples
 - Using: Processor, and optionally I/O
 - Delay: Processor, Storage, MPL, Queue, and optionally I/O

$$\frac{\text{CPU Using} + \text{I/O Using}}{(\text{CPU Using} + \text{I/O Using} + \text{CPU Delay} + \text{I/O Delay}) + \text{Paging Delays} + \text{MPL Delays} + \text{Queue Delays}}$$

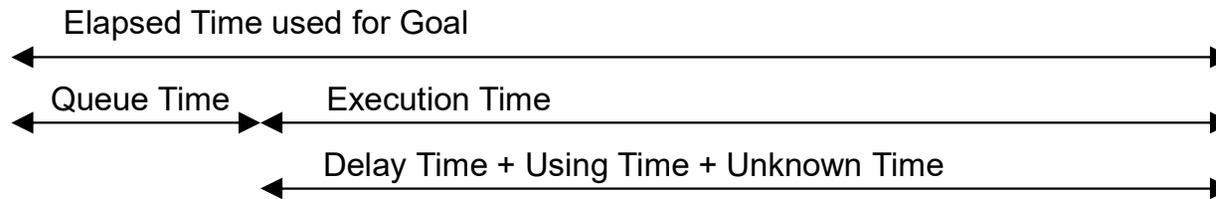


- Based on samples rather than measured results
 - May not always reflect actual behavior of work
 - Measuring actual using and delay would cost too much
 - An attempt is made to relate samples to response times for RT goal work

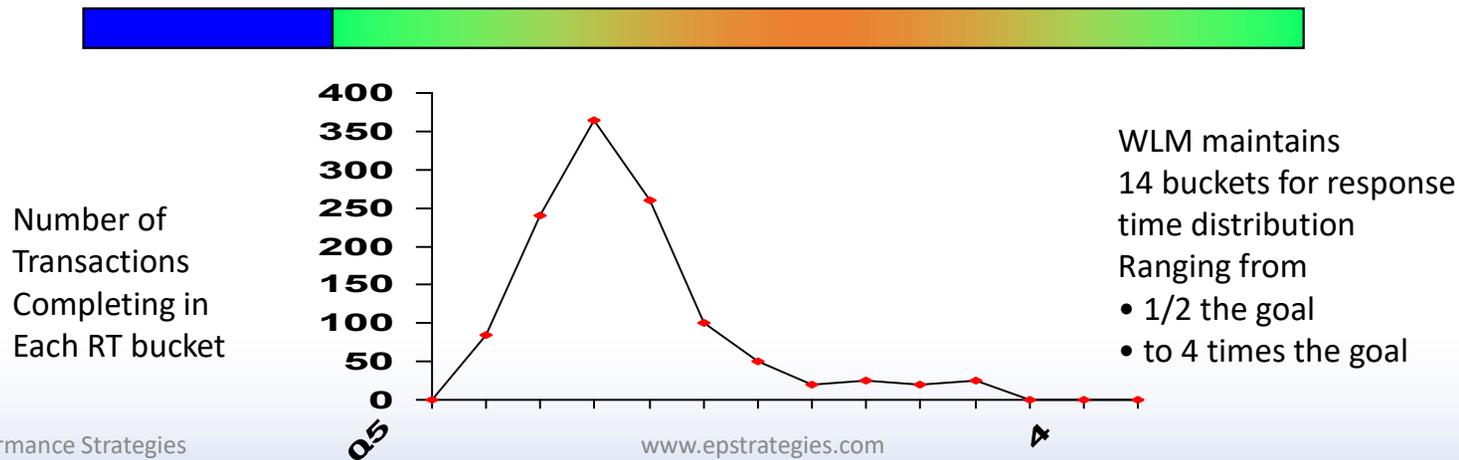
WLM collects completion information



- Transaction throughput
- Transaction response times



- Response time distributions



WLM maintains response time distributions



- WLM maintains a response time distribution for response time goal periods
 - Distribution composed of 14 buckets
 - Each bucket represents a count of transactions that completed within a certain percentage of the assigned goal value
 - Examples:
 - Bucket 4 represents count of all transactions completing between 70% and 80% of the goal value
 - Bucket 6 represents count of all transactions completing between 90% and exactly the goal value
 - Bucket 12 represents count of all transactions that complete between 1.5 and twice the goal value
 - Bucket 13 represents count of all transactions that complete between twice and 4 times goal value

Bucket	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Width	<=50%	60%	70%	80%	90%	100%	110%	120%	130%	140%	150%	200%	400%	>400%
Transaction Count	0	85	240	365	260	100	50	20	25	20	25	0	0	0

WLM calculates Performance Index (aka PI)

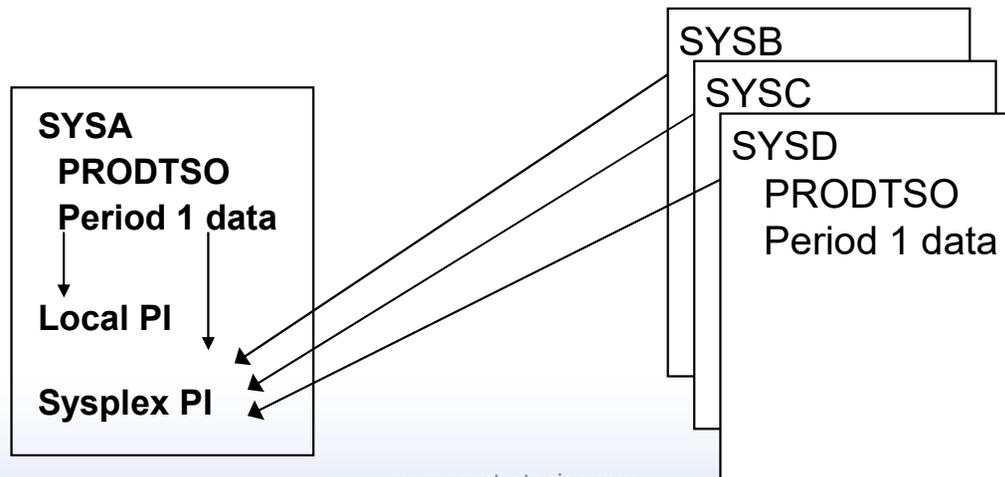


- Now that WLM knows velocities, completions, and response time, it is able to calculate PI for every service class period
 - PI is an indicator of how well a service class period is achieving its goal
 - Allows for comparison of unlike goals for unlike work
- PI < 1 indicates that a goal is being exceeded
 - example: PI = .5 means that work is achieving twice goal
- PI = 1 indicates that a goal is exactly being met
- PI > 1 indicates that a goal is being missed
 - example: PI = 3 means goal is being missed by 3 times

WLM calculates both Local PI & Sysplex PI



- Each local system calculates two Performance Indexes for each goal period
 - Local PI
 - Indicates how well goal period is doing on local z/OS image
 - Based on goal period data just from local z/OS image
 - Sysplex PI
 - Indicates how well goal period is doing globally throughout the Sysplex
 - Based on period data from all z/OS images in goal mode in Sysplex



WLM maintains a series of plots



- Plots used to track how well work is being processed

- Some of the plots include

- Period Paging Rate Plot
- Period MPL Delay Plot
- Period Ready User Average Plot
- Period Swap Delay Plot
- Proportionate Aggregate Speed Plot
- Queue delay Plot
- Queue ready user average Plot
- Active server instance Plot
- Others...

MPL Plot Example:

- shows how response time may improve by increasing MPL slots
- shows how response time may degrade by reducing MPL slots



WLM maintains histories



- Histories used to examine data over a period of time
 - Allows WLM to have a controlled way to go back in time for enough representative data points for accurate decisions
 - For example, a history *may* have the following structure

Last 10 Seconds
Last 30 seconds
Last 60 seconds
Last 5 minutes
Last 15 minutes
Previous 15 minutes

- New data put in row one
- After some number of intervals, the data is rolled forward
- Each successive row represents data that was collected further in the past and over a longer period of time



How can I tell what decisions WLM is making?

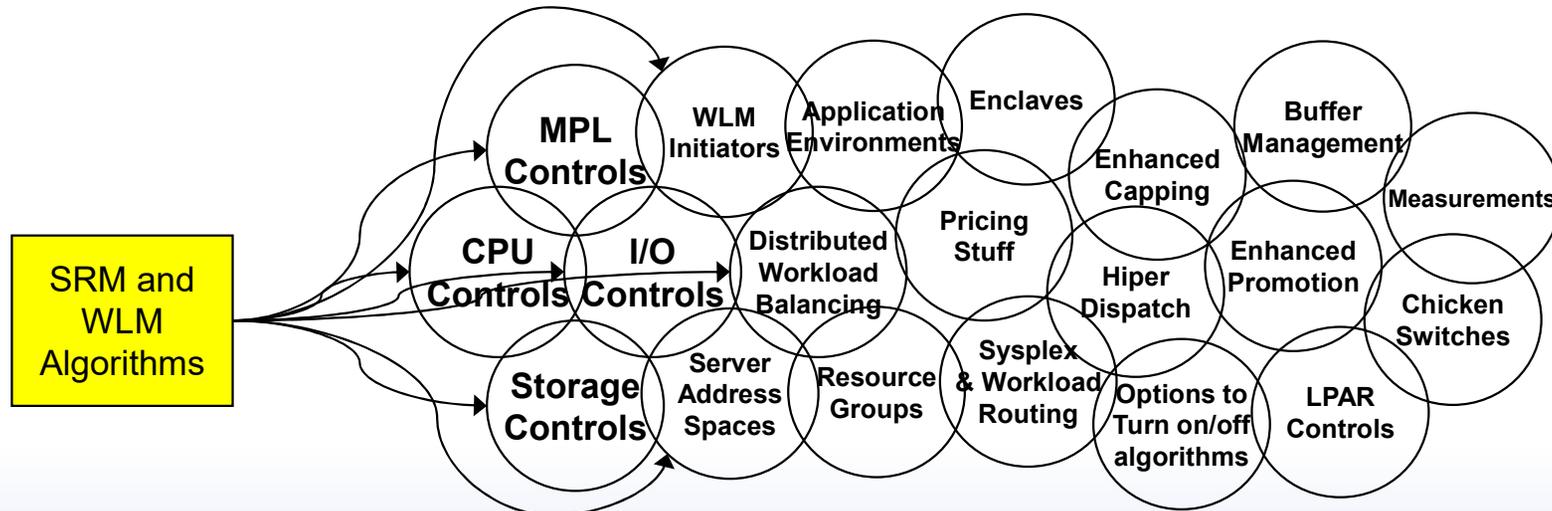
Workload Manager Algorithms



- Internally to the z/OS workload manager are a series of algorithms that are used to ensure the installations performance objectives are met

Policy adjustment – algorithms to help the workloads meet their assigned goals

Resource adjustment – algorithms to help optimize the use of system resources



How can I tell what decisions WLM is making?



- It is tough to tell exactly what decisions WLM is making
 - Easy to notice:
 - Were there changes to CPU dispatching priorities?
 - Were there changes to WLM managed initiators?
 - Were there changes to HiperDispatch pooling and parking / unparking of engines?
 - Resource group controls and capping
 - Harder to notice:
 - Modifications to MPL controls
 - Modifications to storage controls
 - Proactive active or resource optimization controls
 - Workload distribution
 - Sysplex management

WLM chooses Receivers and Donors



- Receiver – a service class period to help

- WLM will help only one receiver during each policy adjustment interval

- Goal Receiver

- Period with goal that needs help

- Resource Receiver

- Period to whom WLM will give the resources in order to help the 'goal receiver'

- Secondary Receiver

- Period that is helped indirectly due to an action to help the goal receiver

- Donor – a service class period to potentially 'donate' resources to help receiver

- WLM may take from multiple donors during each policy adjustment interval

- Goal Donor

- Period whose goals may be impacted by resource donation

- Resource Donor

- Period to donate resources

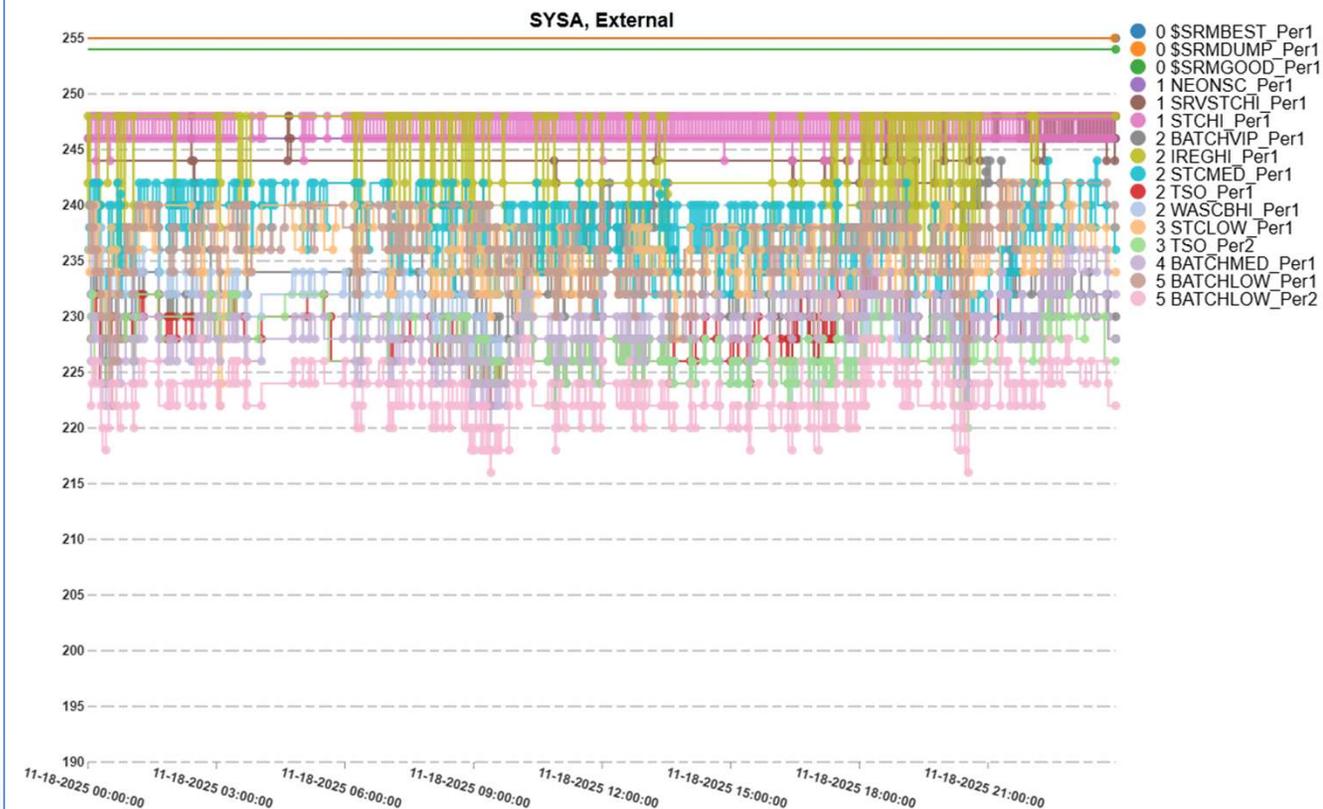
- Secondary Donor

- Period that donates indirectly when receiver is helped

Did WLM make any CPU DP decisions?



WLM SMF 99.6 - CPU Dispatching Priority



The following Pivotor chart shows then changes to the CPU dispatching priorities for customer defined WLM service class periods (also known as External Periods).

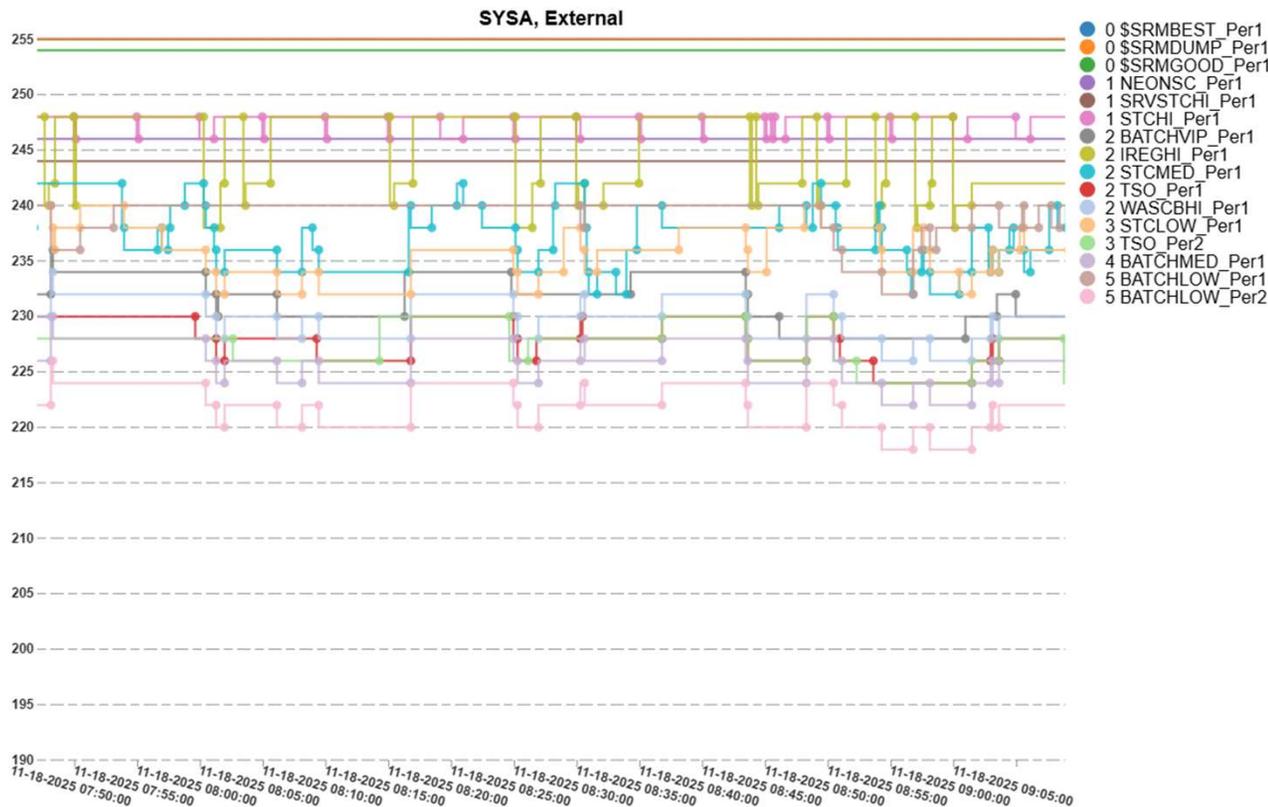
This charge shows the entire day.

The chart clearly shows WLM has been making CPU dispatching priority changes.

Did WLM make any CPU DP decisions?



WLM SMF 99.6 - CPU Dispatching Priority



The following Pivotor chart shows then changes to the CPU dispatching priorities for customer defined WLM service class periods (also known as External Periods).

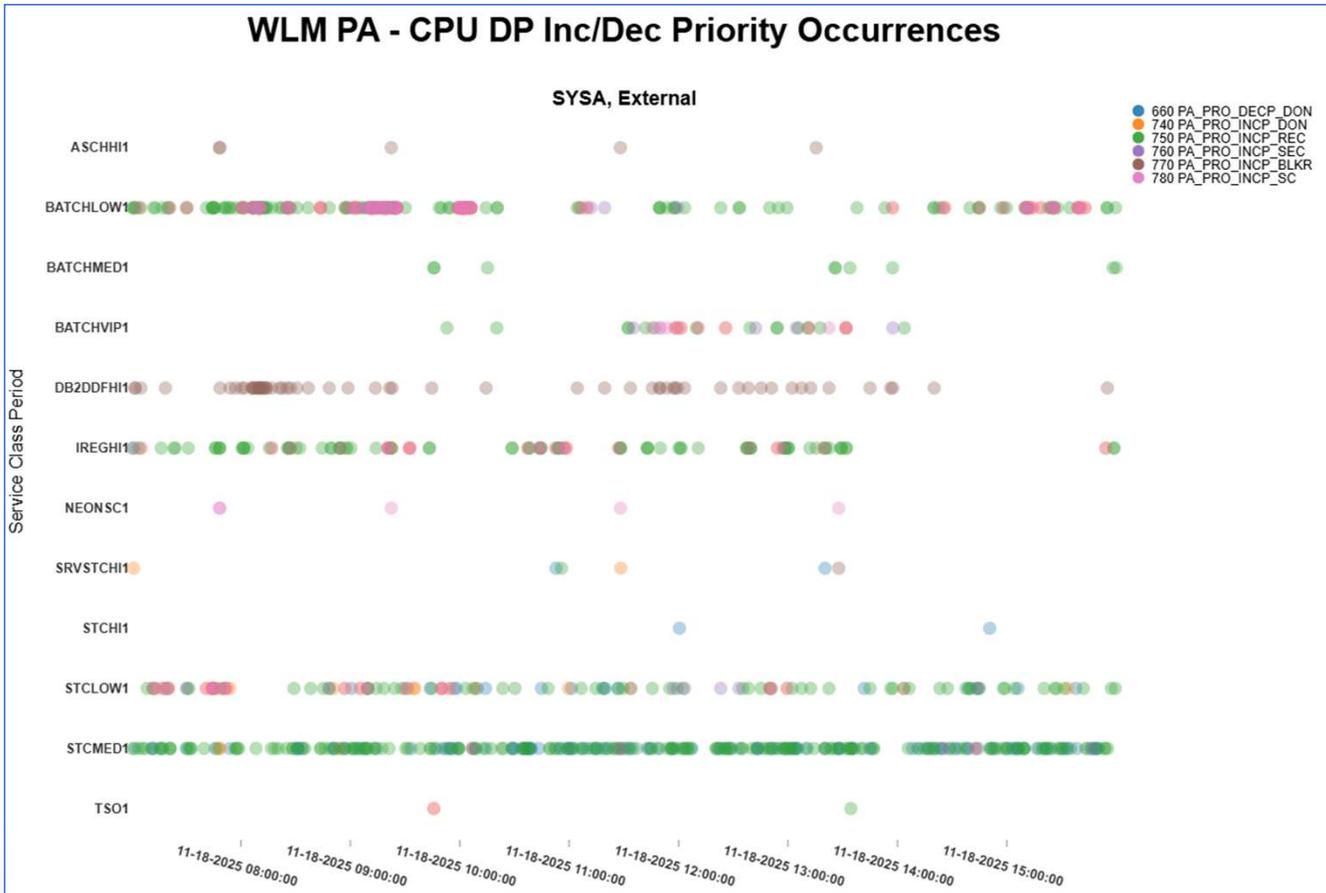
This particular chart focuses on the 8:00am hour.

The chart clearly shows WLM has been making CPU dispatching priority changes.

Did WLM make any CPU DP decisions?



WLM PA - CPU DP Inc/Dec Priority Occurrences



The following Pivotor chart shows a different way that we can tell WLM has made changes to the CPU dispatching priorities for customer defined WLM service class periods (also known as External Periods).

This particular chart focuses on a full day.

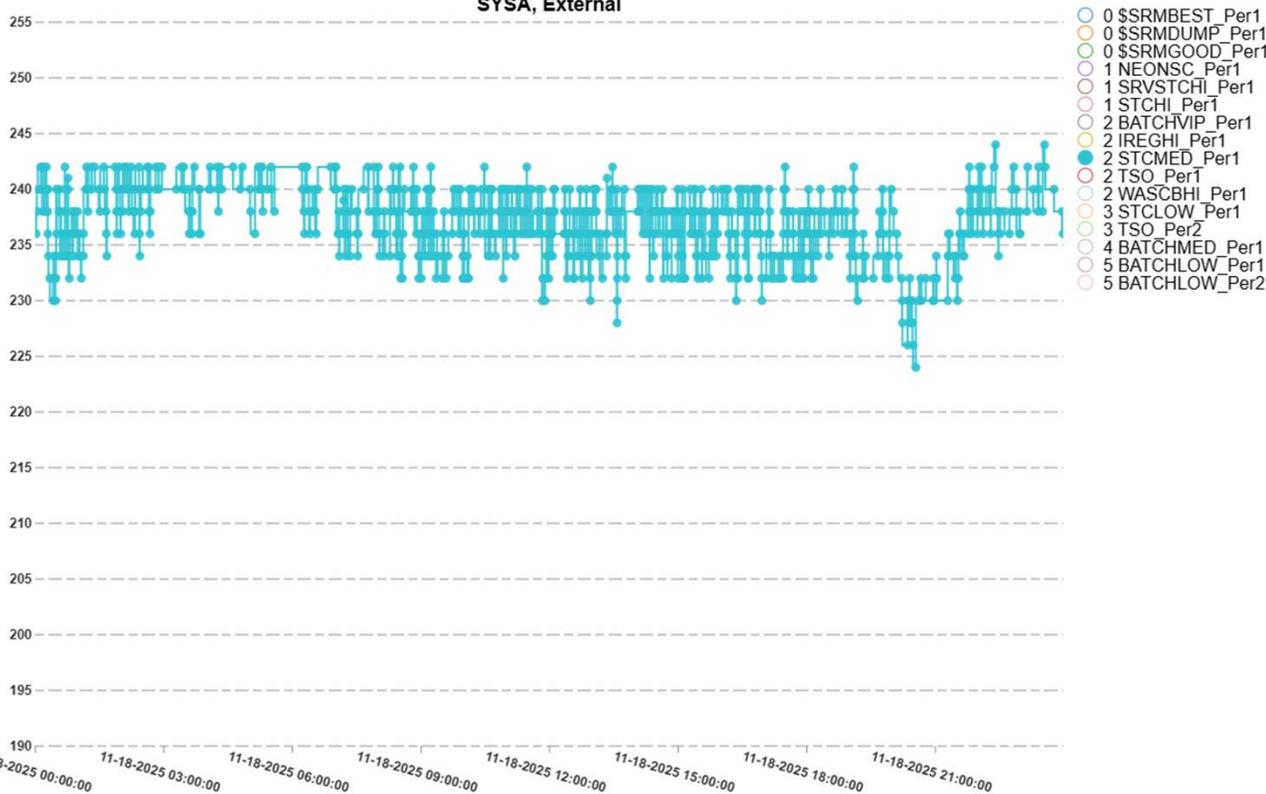
The chart shows which periods had CPU movement actions and why

Did WLM make any CPU DP decisions?



WLM SMF 99.6 - CPU Dispatching Priority

SYSA, External



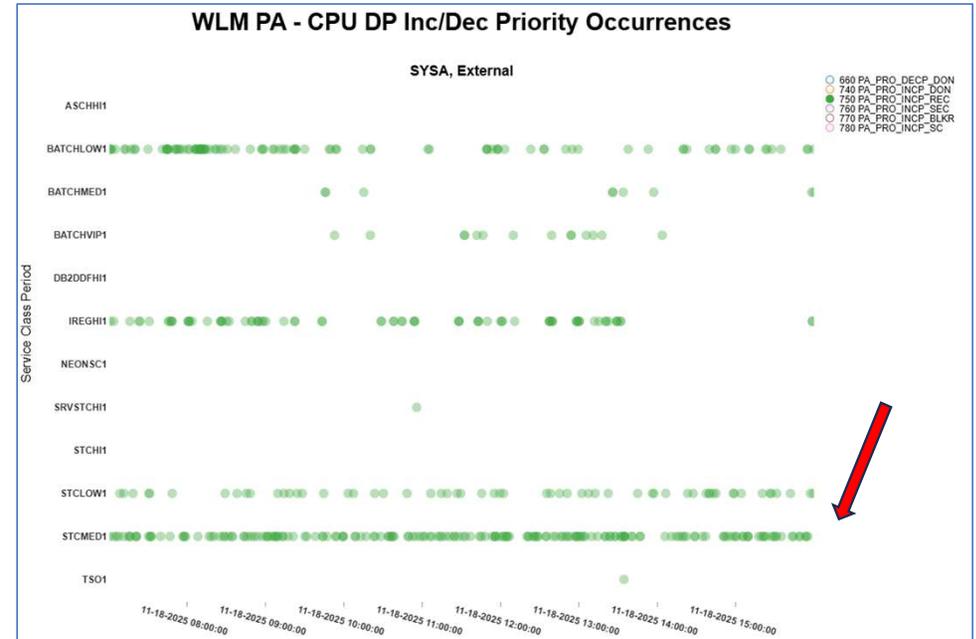
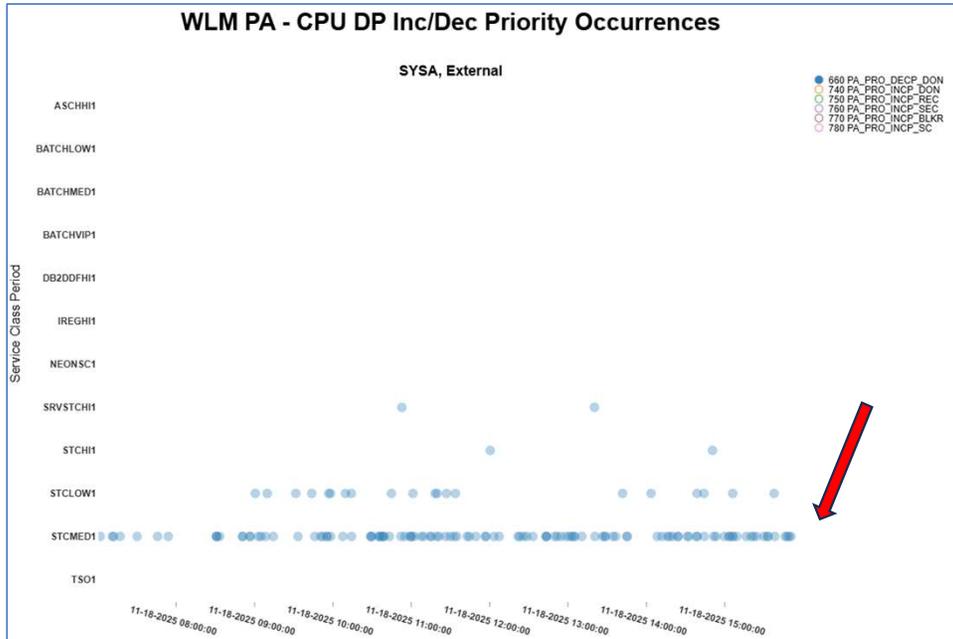
The following Pivotor chart shows the CPU dispatching priority movement for a period named STCMED. Notice the regularly increasing and decreasing of the priorities.

Did WLM make any CPU DP decisions?



Decrease Priority of the Donor

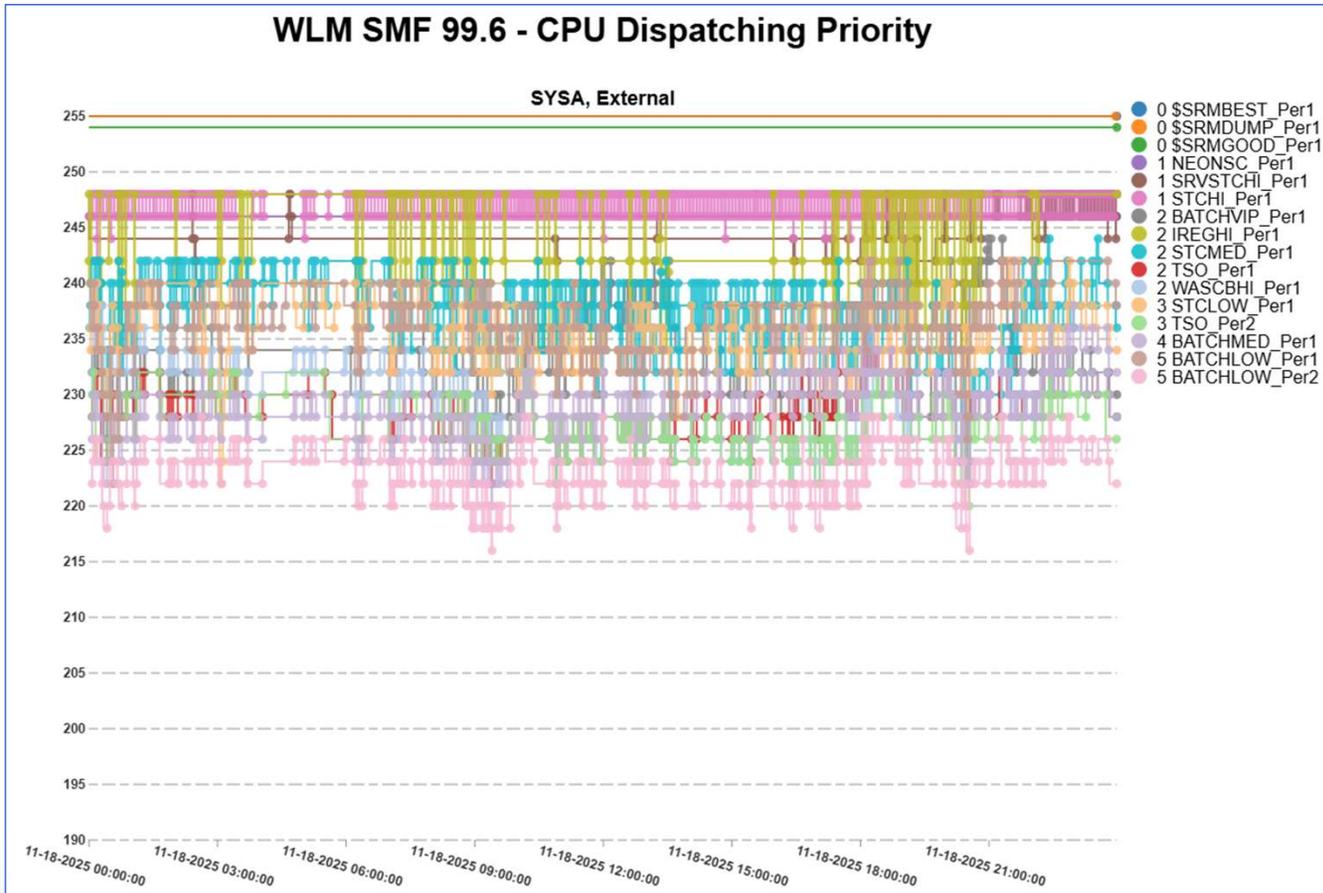
Increase Priority of Receiver



Easy to tell if WLM changed CPU DPs



WLM SMF 99.6 - CPU Dispatching Priority



The following Pivotor chart shows then changes to the CPU dispatching priorities for customer defined WLM service class periods (also known as External Periods).

The chart clearly shows WLM has been making CPU dispatching priority changes.



How can these decisions be influenced
by a WLM administrator?

Example: WLM Possible WLM Actions - CPU



● Dispatching Priority

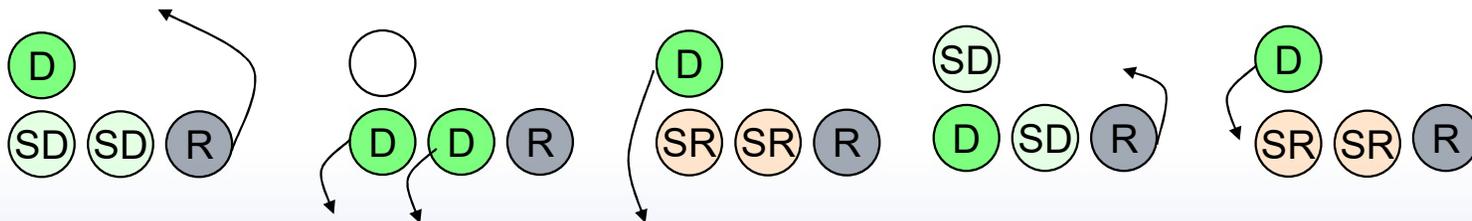
- Priority adjustment for
 - Periods with goals or server period
 - Discretionary periods in a resource group
- Small consumer
 - For periods that use very little CPU
 - Gets these periods 'out of the way' of critical adjustments
- Actions include:
 - Increase Receiver's priority
 - Decrease Donor's priority
 - Decreased service consumption and/or increased wait-to-using ratio
 - Both

255	SYSTEM
254	SYSSTC
253	'Unused' (SYSSTC1-5)
249	
248	Small Consumer
247	Priorities Used for RT or Velocity Periods (i.e. Imp 1 – 5)
203	
202	Unused
201	Discretionary (MTTW)
192	
191	Quiesce

Example: WLM Possible WLM Actions - CPU



- WLM will model and project effects of dispatching priority adjustments
 - Objective: Increase Receiver's CPU using, or decrease Receiver's CPU delay
 - Interesting concepts:
 - Wait-to-Using ratio - ratio of CPU delay samples to CPU using samples (change in ratio used to determine change in CPU delay)
 - Maximum demand
 - Theoretical maximum percentage of total processor time a period can consume if it had no CPU delay
 - Achievable maximum demand
 - Percentage of total processor time a service period is projected to consume, taking into account demand of all higher work
 - Some possible actions

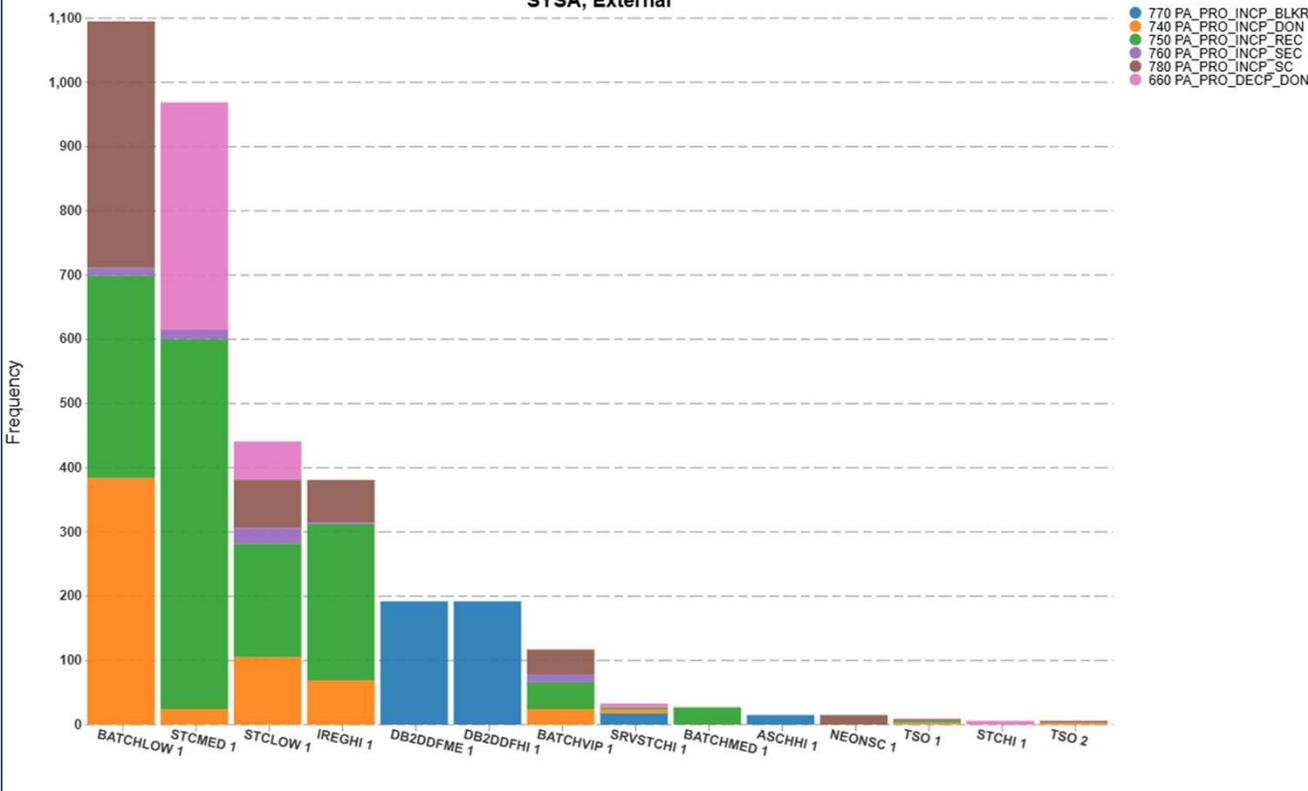


Easy to tell if WLM changed CPU DPs



WLM PA - Top CPU DP Movement Periods
(Policy Adjustment Decisions)

SYSA, External



The following Pivotor chart shows which customer defined WLM service class periods (also known as External Periods) had the most CPU dispatching priority changes.

Changes are not bad or good: They are what they are.

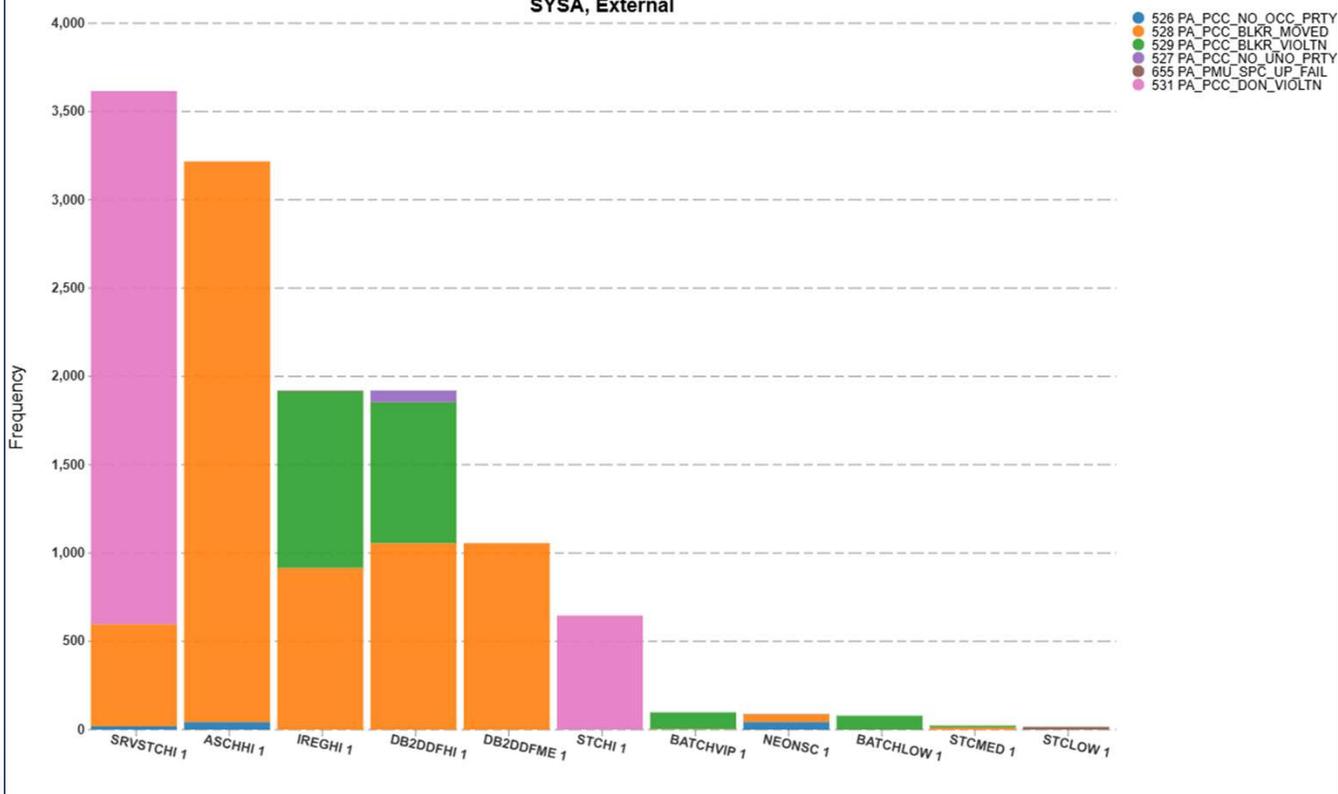
However, when you see a high importance period regularly as a donor, then receiver, then a donor then a receiver then a donor then receiver... etc... Means that the goal and importance should be revisited

Easy to tell if CPU Critical Influenced DPs



WLM PA - Top CPU Critical Period Actions (Policy Adjustment Decisions)

SYSA, External



The following Pivotor chart shows which periods were affected by CPU critical restrictions.

In this example, SRVSTCHI wants to move down on a regular basis, but this violates the CPU critical control since they period is marked as CPU critical.

So, an example of a period marked a CPU critical, WLM wants to take CPU away by decreasing priority, but cannot.

WLM admin should address this.

WLM PA Loop: Receiver Value Check



- Receiver Value

- Receiver helped only if there is projected to be sufficient *receiver value*
 - Designed to reject 'small or marginal improvements'
 - Allows WLM to get on to addressing larger problems for other periods
- Minimum projected improvement to make change worth the effort
 - Projected PI improvement
 - or projected minimum group service increase
 - or some other projected minimum criteria

- Guideline:

- Projected PI improvement is the larger of (10% of the PI change to meet goal) or (0.05)
- Or Reduction in delay samples is at least half of the largest delay

- Example:

- PRODTSO period 1 PI = 3.5
- WLM algorithms suggest improvements can bring PI to 3.46
- Don't take action

WLM PA Loop: Net Value Check



- Net Value Check - Very high level logic
 - If receiver is more important than donor
 - Make the move if receiver is missing goal
 - If receiver is less important than donor
 - Never make the move if donor is missing goal
 - or is projected to miss goal
 - If receiver and donor are equal importance
 - Receiver's PI benefit is more than donor's loss, and
 - less disparity in projected PIs
- Resource group minimums and maximums make this even more complicated

WLM Policy Adjustment – 'The Loop'



- Summarize data for state of the system and workloads
- Select a receiver period (highest importance missing goal the most)
- Find the receiver's largest bottleneck
 - Determine fix for receiver's bottleneck
 - Determine if needed resources can be gotten from unused resources
 - Find donor(s) of resource that receiver needs
 - Assess effect of reallocating resources from donor(s) to receivers
 - If allocation has both net and receiver value
 - Then commit change
 - Else don't make change
 - If reallocation was done then jump to Exit and allow change to be absorbed
 - If reallocation was not done then try to fix receiver's next largest bottleneck
- If cannot help receiver then look for next receiver (highest importance missing goal the most)
- Exit
 - Housekeep current set of controls



Questions?

WLM Policy Adjustment – 'The Loop'



- Summarize data for state of the system and workloads
- Select a receiver period (highest importance missing goal the most)
- Find the receiver's largest bottleneck
 - Determine fix for receiver's bottleneck
 - Determine if needed resources can be gotten from unused resources
 - Find donor(s) of resource that receiver needs
 - Assess effect of reallocating resources from donor(s) to receivers
 - If allocation has both net and receiver value
 - Then commit change
 - Else don't make change
 - If reallocation was done then jump to Exit and allow change to be absorbed
 - If reallocation was not done then try to fix receiver's next largest bottleneck
- If cannot help receiver then look for next receiver (highest importance missing goal the most)
- Exit
 - Housekeep current set of controls