

# Introduction to the WLM

Scott Chapman  
Enterprise Performance Strategies, Inc.  
Scott.chapman@EPStrategies.com



# Contact, Copyright, and Trademarks



## Questions?

Send email to [performance.questions@EPStrategies.com](mailto: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<sup>®</sup>, Reductions<sup>®</sup>, Pivotor<sup>®</sup>**

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

Other trademarks and registered trademarks may exist in this presentation

# Abstract



- This presentation will provide an overview of the z/OS Workload Manager (WLM). The attendee will learn the basic constructs of WLM, how WLM manages workloads and optimizes resources in the z/OS environment. The attendee will walk out of this presentation with not just a working knowledge of WLM, but also a few hints, tips, and exercises they can use back at their installation.

# Agenda



- Introduction
- WLM Terminology and Concepts
- Goals and Importances
- WLM Algorithms
- Top 3 WLM Problems

# EPS: We do z/OS performance...



- Pivotor - Reporting and analysis software and services
  - Not just reporting, but analysis-based reporting based on our expertise
- Education and instruction
  - We have taught our z/OS performance workshops all over the world
- Consulting
  - Performance war rooms: concentrated, highly productive group discussions and analysis
- Information
  - We present around the world 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
  - October 3-7, 2022
- WLM Performance and Re-evaluating Goals
  - September 12-16, 2022
- Parallel Sysplex and z/OS Performance Tuning
  - August 8-12, 2022
- Also... please make sure you are signed up for our free monthly z/OS educational webinars! (email [contact@epstrategies.com](mailto:contact@epstrategies.com))

# Like what you see?



- The z/OS Performance Graphs you see here come from Pivotor™
- If you don't see them in your performance reporting tool, or you just want a free cursory performance 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>
- We also have a **free** Pivotor offering available as well
  - 1 System, SMF 70-72 only, 7 Day retention
  - That still encompasses over 100 reports!

**All Charts** (132 reports, 258 charts)

All charts in this reportset.

**Charts Warranting Investigation Due to Exception Counts** (2 reports, 6 charts, [more details](#))

Charts containing more than the threshold number of exceptions

**All Charts with Exceptions** (2 reports, 8 charts, [more details](#))

Charts containing any number of exceptions

**Evaluating WLM Velocity Goals** (4 reports, 35 charts, [more details](#))

This playlist walks through several reports that will be useful in while conducting a WLM velocity goal an.

# EPS presentations this week



What	Who	When	Where
Introduction to z Processor Measurements	Scott Chapman	Mon 10:30	Cumberland L
Introduction to WLM Management of CICS and IMS Workloads	Peter Enrico	Mon 14:15	Cumberland K
Introduction to the WLM	Scott Chapman	Tue 13:00	Cumberland AB
z/OS WLM – Revisiting Goals over Time	Peter Enrico	Wed 8:00	Cumberland L
Top WLM Mistakes and Questions	Peter Enrico Scott Chapman	Wed 13:00	Cumberland AB



# Introduction

# In the olden days...



- Before WLM, IPS and ICS parmlib members were used to control work performance
- Dispatching priorities basically fixed
- Most other platforms still work on fixed dispatching priorities
  - Sometimes (usually? often?) defaulting to a single priority
- Fixed dispatch priorities may be fine:
  - There's limited types of work on the system
  - The system is usually run at low utilization levels
- Neither describes the modern mainframe!

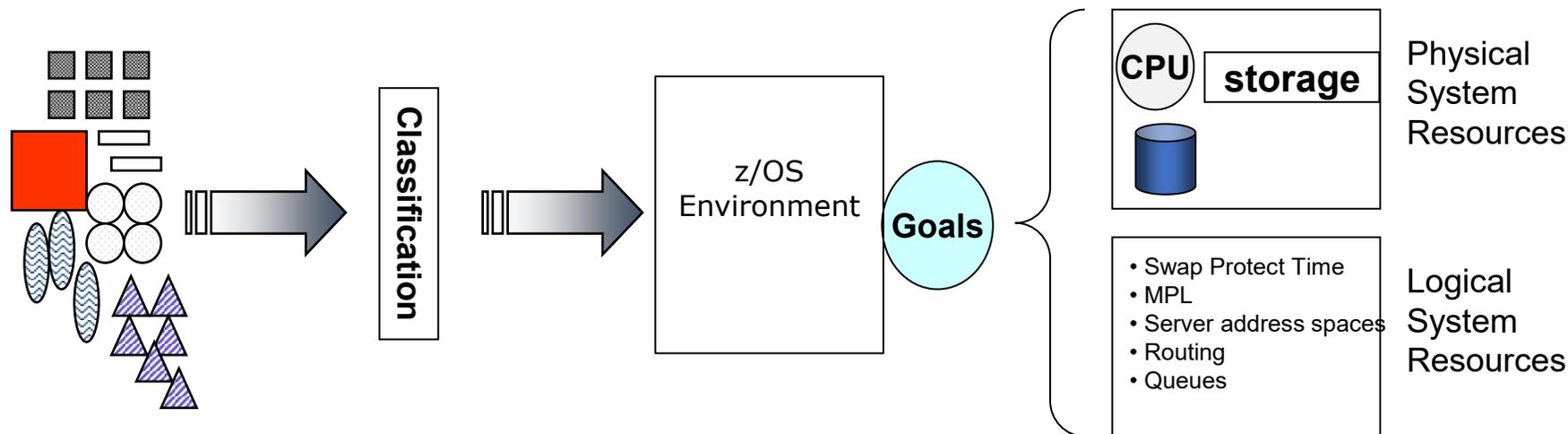
PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
20	0	4239600	404008	15724	S	113.5	1.3	0:33.62	/usr/lib/jvm/java
20	0	4239600	400080	15732	S	105.3	1.3	0:25.32	/usr/lib/jvm/java
20	0	4239600	400080	15732	S	103.0	1.2	0:35.04	/usr/lib/jvm/java
20	0	4239600	400080	15732	S	103.0	1.2	0:43.44	/usr/lib/jvm/java
20	0	4239600	400080	15732	S	100.7	1.2	0:27.73	/usr/lib/jvm/java
20	0	4239600	400080	15732	S	100.3	1.1	0:34.13	/usr/lib/jvm/java
20	0	4239600	400080	15732	S	100.3	1.2	0:44.94	/usr/lib/jvm/java
20	0	4239600	400080	15732	S	98.7	1.1	0:29.67	/usr/lib/jvm/java
20	0	4239600	366904	15716	S	98.3	1.2	0:42.16	/usr/lib/jvm/java
20	0	4239600	335896	15732	S	98.0	1.1	7:52.13	/usr/lib/jvm/java
20	0	4239600	354156	15724	S	97.7	1.1	0:35.42	/usr/lib/jvm/java
20	0	4239600	374860	15720	S	95.7	1.2	0:34.45	/usr/lib/jvm/java
20	0	10.7g	74624	12716	S	75.9	0.2	0:02.30	/usr/bin/jjs -scr
25	5	121092	9388	820	R	67.7	0.0	181:51.04	xz -z -c -l
20	0	11.3g	70024	13500	S	39.9	0.2	0:01.21	java -jar /var/li
20	0	10.5g	43112	12400	S	29.7	0.1	0:00.90	/usr/bin/jjs -scr
25	5	126664	3080	2084	R	26.7	0.0	74:21.32	mysqldump --max_a
20	0	10.7g	75016	12760	S	10.9	0.2	0:02.21	/usr/bin/jjs -scr
20	0	13.1g	2.4g	12296	S	8.9	8.1	111:59.11	java -Djava.secur
20	0	0	0	0	S	1.3	0.0	6:05.99	[kswapd0]
20	0	1132780	34264	6616	S	0.7	0.1	152:42.94	/usr/sbin/collect
20	0	11.5g	78124	12740	S	0.7	0.2	0:02.86	/usr/bin/jjs -scr
20	0	162380	2756	1672	R	0.7	0.0	0:00.46	top -c
20	0	0	0	0	S	0.3	0.0	2:33.25	[rcu_sched]
20	0	4481752	74744	14780	S	0.3	0.2	0:02.49	/bin/java -Xms8M
20	0	4481752	73112	14840	S	0.3	0.2	0:02.36	/bin/java -Xms8M
20	0	10.7g	71316	12720	S	0.3	0.2	0:02.38	/usr/bin/jjs -scr
20	0	4481752	74360	14816	S	0.3	0.2	0:02.40	/bin/java -Xms8M
20	0	10.7g	66196	12720	S	0.3	0.2	0:02.22	/usr/bin/jjs -scr
20	0	6559208	293760	12148	S	0.3	0.9	47:50.17	/usr/lib/jvm/java
20	0	11.5g	137104	13592	S	0.3	0.4	0:03.84	java -jar /var/li
20	0	2610124	31816	14940	S	0.3	0.1	0:00.18	/opt/pivotor/prod
20	0	4481752	77152	14828	S	0.3	0.2	0:02.32	/bin/java -Xms8M
20	0	4481752	76732	14808	S	0.3	0.2	0:02.44	/bin/java -Xms8M
20	0	55824	4920	2752	S	0.0	0.0	2:13.33	/usr/lib/systemd/
20	0	0	0	0	S	0.0	0.0	0:00.75	[kthreadd]
20	0	0	0	0	S	0.0	0.0	0:02.74	[ksofirqd/0]

Almost everything is the same priority!

# z/OS Workload Management Objectives



- Simplify system performance administration
  - Eliminate manual micromanagement of static dispatching priorities
- Indicate to the system what the desired expectation of the work is rather than specific priorities
  - Dynamically respond to how work is running
- Integrate system feedback to performance monitors and reporters





# WLM Terminology & Concepts

# WLM Service Definition



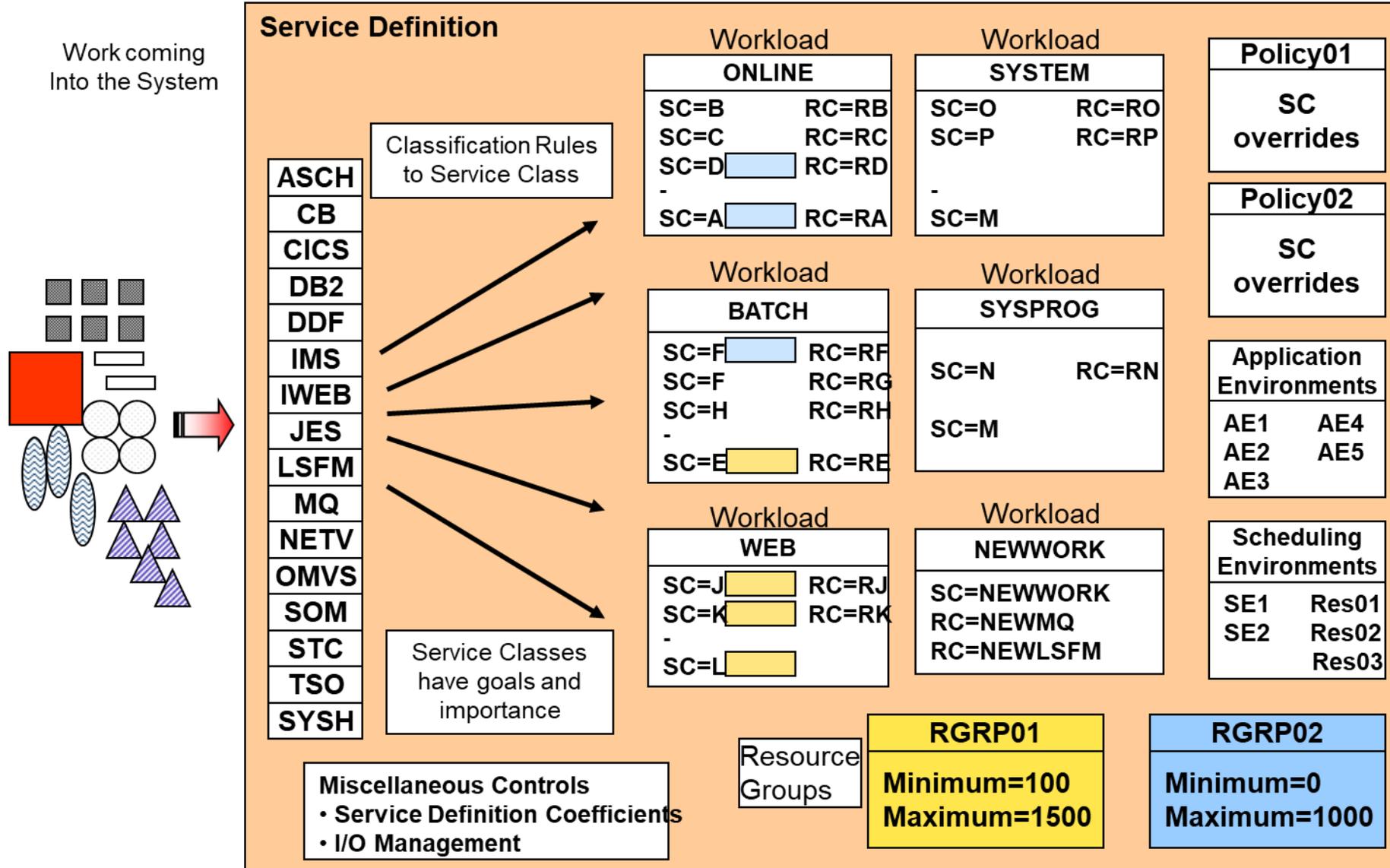
- WLM Service Definition contains all the information needed by WLM to manage the work
- Single active service definition per sysplex
  - Active service definition stored in the WLM couple dataset
  - You can't share service definitions across sysplexes but you can copy one to another
- Working copy can be stored in ISPF tables ~~in a PDS or~~ in XML format
  - XML format is current default and recommended approach
  - Can be used to copy SD between sysplexes if you have a need for that
  - May be easier on the humans if the service definitions are similar between plexes
    - Almost always will have some things specific to a sysplex or even system
    - Personally, I at least want the names of similar things to be the same

# What's in a Service Definition?



Service Policies	Named sets of overrides to defined goals
Workloads	Groups of service classes for reporting purposes
Service Classes	Groups of work with similar performance goals, business importance, and resource requirements for reporting and management purposes
Report Classes	Groups of work for more granular reporting purposes
Resource Groups	Limits or guarantees access to processor or memory resources
Classification Rules	Assign incoming work to a service class and/or report class
Application Environments	Groups of application functions that execute in server address spaces and can be requested by a client
Scheduling Environments	Lists of named resources along with their required state
Service Coefficients & Options	Miscellaneous settings for WLM controls

We obviously can't cover all of this in an hour!



Peter's high-level overview

# WLM ONE-SHEET GUIDE

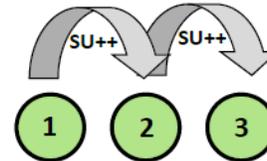


SCPs have a goal and importance

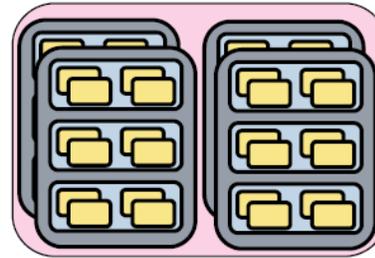
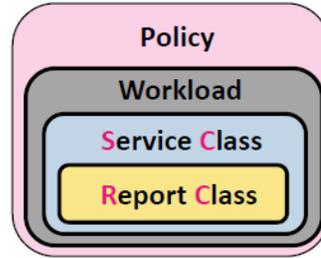


Service Classes have 1 to 8 Periods

More than 3 periods discouraged, rarely need more than 2 periods



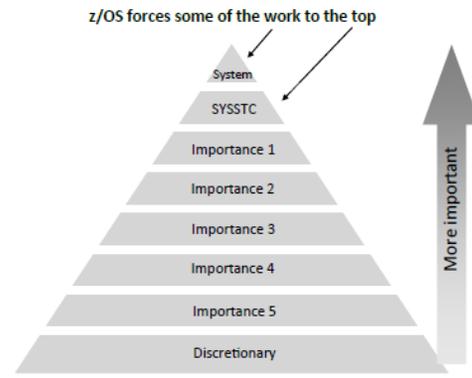
Work switches periods as it consumes more resources



RCs should not cross SCs!



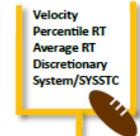
Generally, each system should have no more than about 25-30 active service class periods at any given time



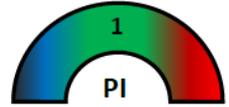
z/OS forces some of the work to the top

Every SCP is assigned an importance

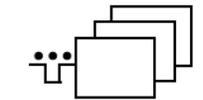
Velocity =  $(U / (U+D)) * 100$   
All goals should be periodically reviewed but velocity goals must be reviewed after hardware changes



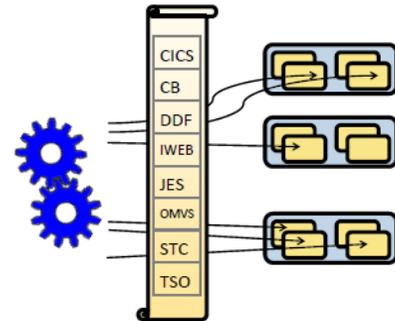
Aim for Goldilocks goals



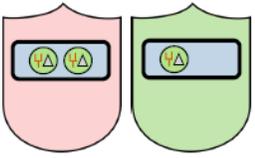
Performance Indexes are calculated per SCP on both a sysplex and system scope



Application Environments allow WLM to create and manage server address spaces for work to run in



Work is classified by various qualifiers, depending on source: user id, transaction name, job class, many others



Service classes may belong to Resource Groups, which may limit or guarantee resources

Use RGs sparingly and carefully



Scheduling Environments allow work to be directed to systems that have particular resources

Scott's one-sheet WLM high-level overview.

# Classification Rules



- Classification rules are the rules WLM uses to associate incoming work with:
  - Service Class
  - Reporting Class (optional, but highly recommended!)
- Classification rules are specified in terms of transaction qualifiers
  - Attributes of incoming work are compared to these qualifiers and, if there is a match, then the rule is used to assign a service class to the work
  - Subsystems also have a default service class for those units of work that do not match any rules
- Many different qualifiers, such as userid, transaction name, subsystem name, IP address, system name, etc.

# Service Classes (SC)



- Within a WLM Workload, Service Classes define work with similar:
  - Work types
  - Performance goals
  - Resource requirements
  - Business importance to the installation
- A service class consists of:
  - Service class name
  - Service class description
  - Period(s)
    - Performance goal and importance
    - Durations
  - Resource group name
- Service class can only be associated with one workload
- Can define up to 100 service classes

**COWPBAT Service Class**

Period 1  
Goal = Velocity 15  
Importance 4  
RGRP = FENCED

**PRODTSO Service Class**

Period 1 – 500 SU  
Goal = RT 0.5 sec, 95%  
Importance 2  
RGRP =

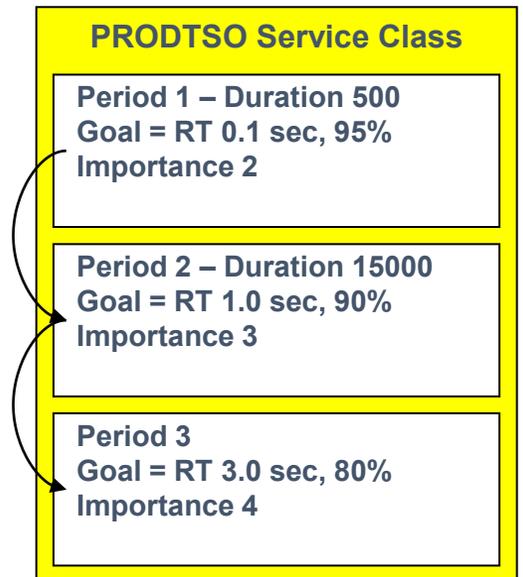
Period 2 – 1500 SU  
Goal = RT 1.5 sec, 90%  
Importance 3  
RGRP =

Period 3  
Goal = Velocity 31  
Importance 4  
RGRP =

# Service Class Periods (SCP)



- WLM manages work at the Service Class Period level
- Service Classes have 1 to 8 periods
  - Most SCs will be single-period
  - Each period has its own goal, importance, and duration
- Duration: Amount of resource consumed, measured in service units
  - As consumption exceeds period duration work moves to the next period
- Why use multiple periods in a SC?
  - Protect and promote “trivial” transactions
  - Prevent resource-intensive transactions from impacting other work
    - Don’t let the hogs trample the humming birds!



# Workloads



- A Workload is a grouping of work used for reporting/monitoring purposes
  - Groups of service classes
  - Workload definitions have no effect on policy or resource adjustment algorithms
  - Strictly for reporting purposes

ONLINE Workload
SC Name = CICSHIGH • Critical CICS Transactions
SC Name = CICSMED • Regular CICS Transactions
SC Name = CICSLOW • Low important CICS Transactions
SC Name = CICSREGS • CICS Regions

PRODWEB Workload
SC Name = WEBHIGH • Critical Web Transactions
SC Name = WEBHTML • Regular HTML Requests
SC Name = WEBSERV • Low important Web Transactions
SC Name = WEBREGS • WebSphere Regions

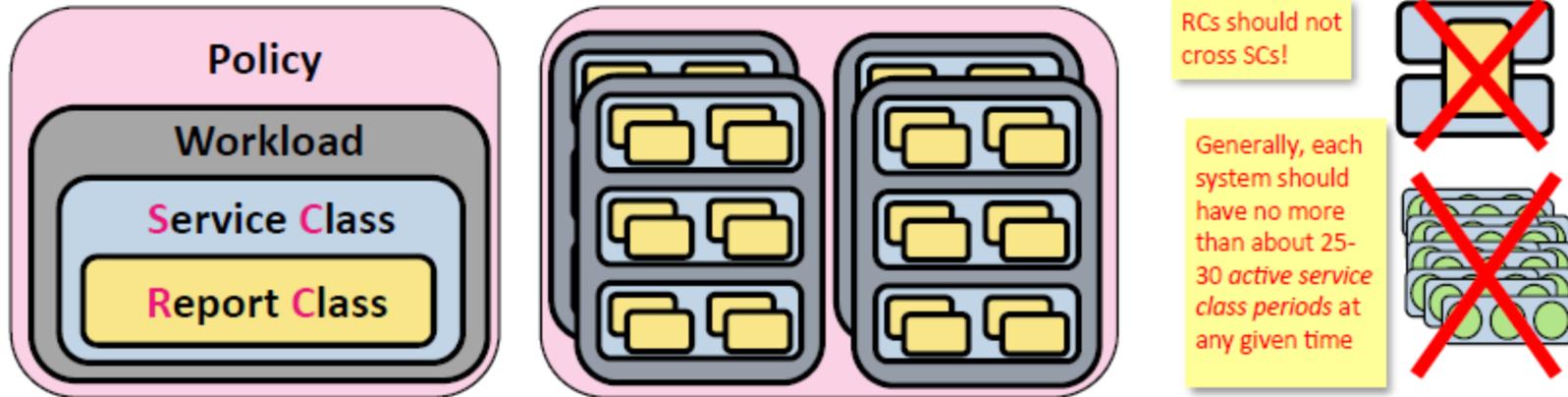
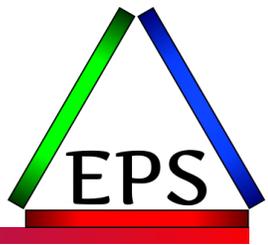
SYSPROG Workload
SC Name = PRODTSO • Production TSO
SC Name = PRODUSS • Production USS Users
SC Name = PRODBAT • Production Batch

# Report Classes (RC)



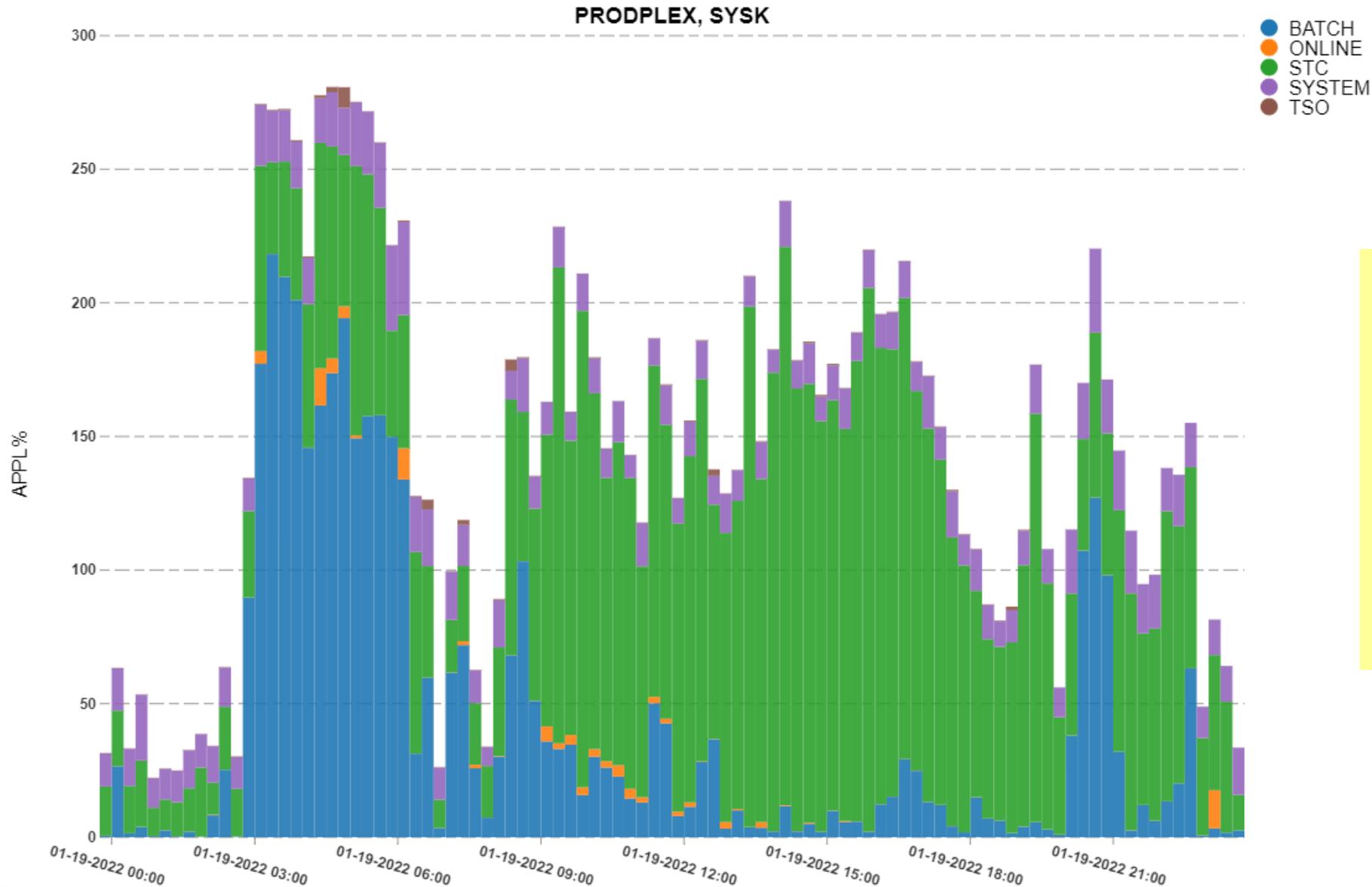
- Subsets of work can be assigned to a report class for reporting purposes
  - Homogenous RC: all the work in a RC is for the same SC (do this!)
    - Ideally: that SC is only used in one subsystem as well
  - Heterogenous RC: work in the RC is for more than one SC (don't do this!)
    - In this case some measurements can be at the very least misleading
- Only measurable overhead for RCs is the volume of SMF 72 records cut
- Report classes are technically optional, but please don't consider them so!
  - Good use of RCs can let you easily do things like application-level reporting
  - RCs have no bearing on WLM management of the work, so change at will
- Can define up to 2047 RCs

# Workloads vs. Report Classes



- Workloads and Report Classes are strictly for reporting/monitoring purposes
- Workloads encompass multiple service classes
- Service classes (should) encompass multiple report classes

# WLM CPU - CP CPU APPL% by Workload (CP + zAAP on CP + zIIP on CP)

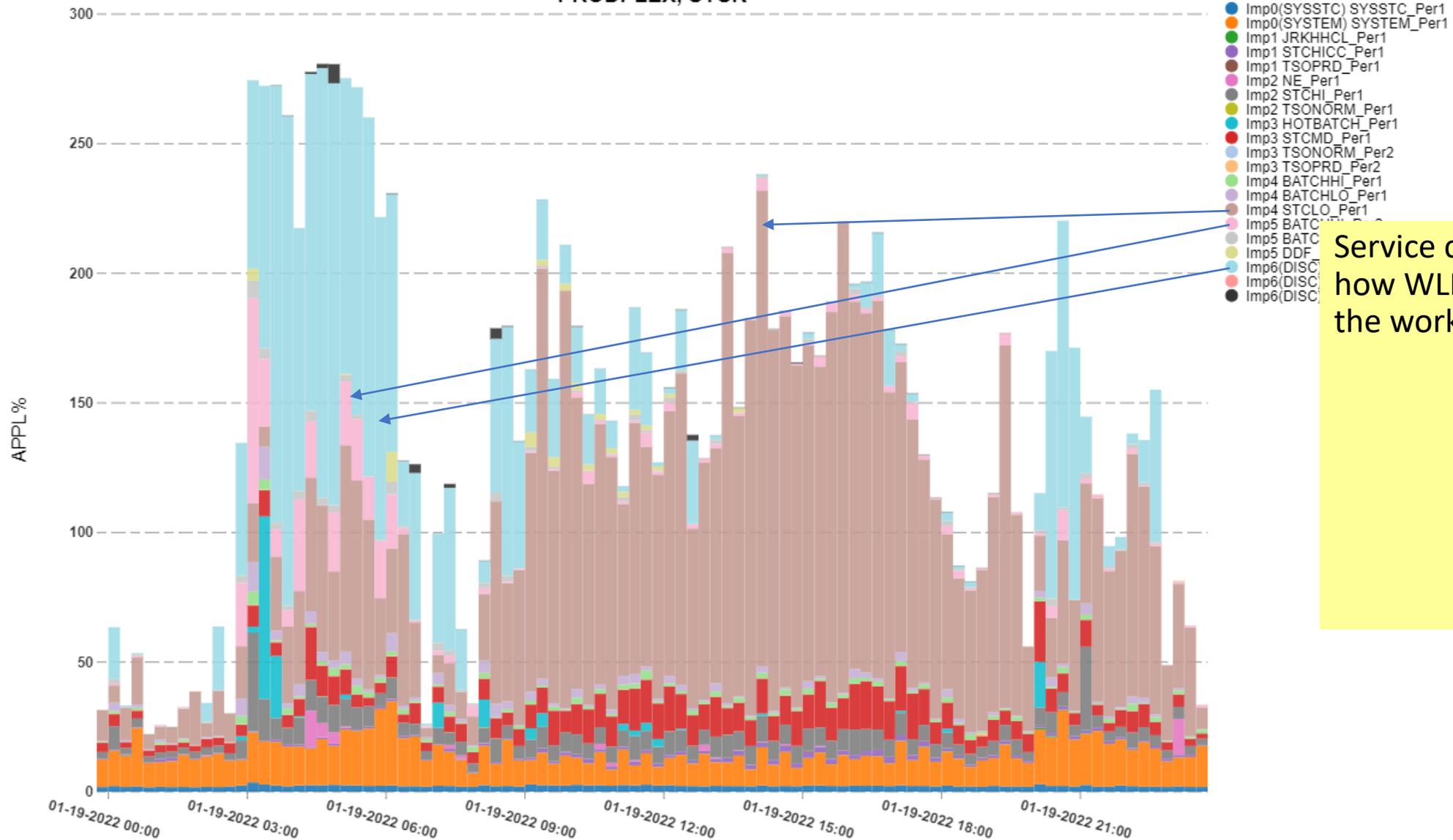


Workloads give you a very high level view of what's consuming the CPU

# WLM CPU - CP APPL% by Service Class

(CP + zAAP on CP + zIIP on CP)

PRODPLEX, SYSK

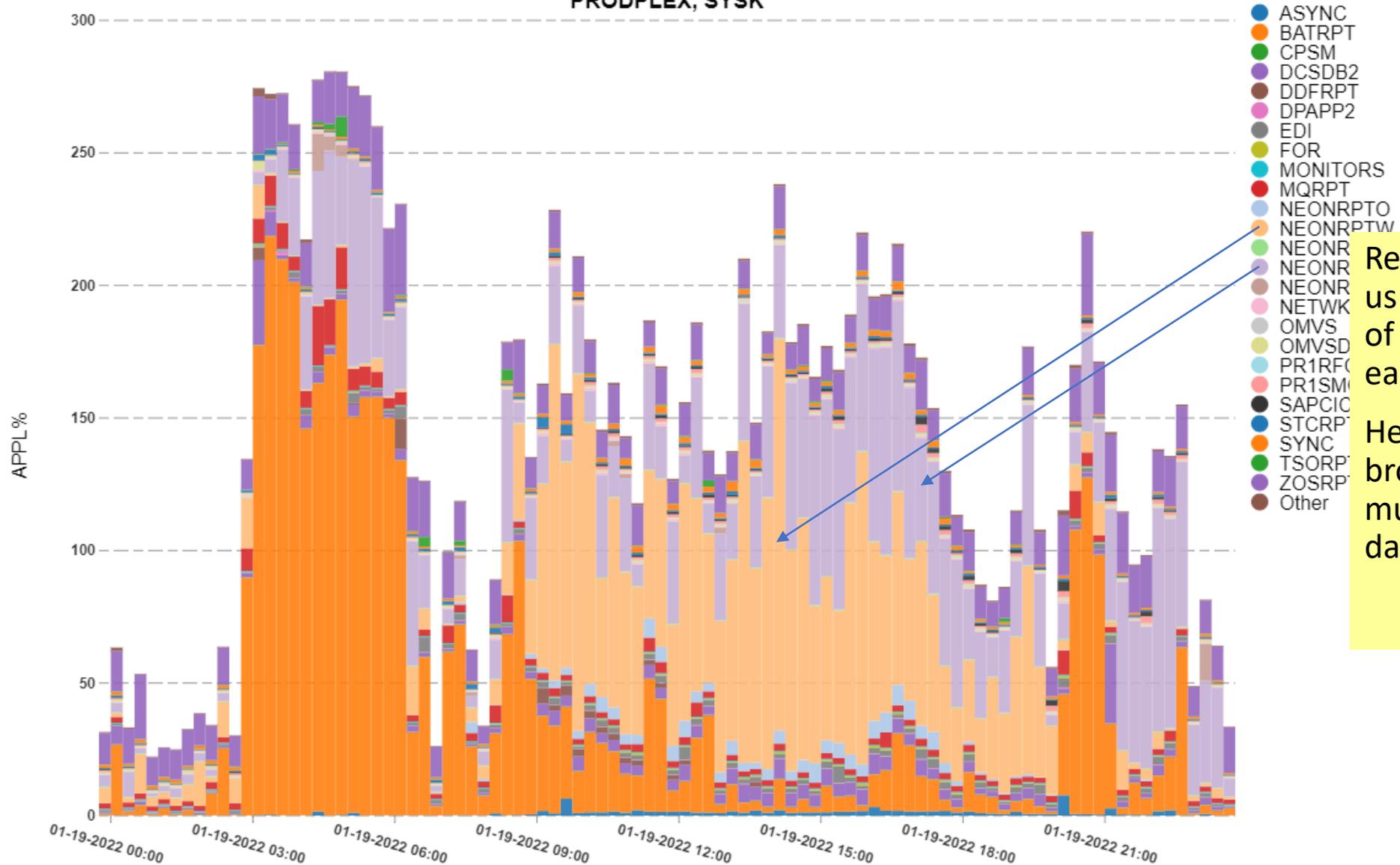


Service class periods are how WLM is managing the work.

# WLM CPU - CP APPL% for Top Report Classes

(CP + zAAP on CP + zIIP on CP)

PRODPLEX, SYSK



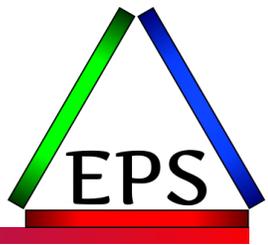
Report classes can give us a better breakdown of what is running in each service class.

Here it appears STCLO is broken down into multiple RCs during the day.



# Goals and Importances

# Importance Levels



- Helps WLM to prioritize work relative to other work
  - WLM attempts to meet higher importance goals before trying to meet lower importance goals
  - Higher importance work tends to receive resources to help it meet its goals
  - Lower importance work tends to have resources taken away before higher importance work
- All work assigned a velocity or response time goal is also assigned a relative importance level
  - 1 - highest
  - 2 - high
  - 3 - medium
  - 4 - low
  - 5 - lowest
- Work assigned a WLM goal does not have an importance
- Although not official importance levels, for reporting sometimes:
  - 0 = SYSTEM/SYSSTC work (fixed, highest dispatching priorities in the system)
  - 6 = Discretionary

Importance does not necessarily translate to dispatching priorities!

# SYSTEM and SYSSTC



- There are WLM defined service classes without explicit goals but rather managed with specific dispatching priorities
- Considered Importance 0 (i.e. higher than importance 1)
- SYSTEM Service Class
  - Highest CPU & I/O dispatching priority: x'FF' or 255
  - Address spaces in SYSTEM are not managed by WLM (except for XMEM delays)
  - Examples of address spaces in SYSTEM include
    - MASTER, GRS, DUMPSRV, CATALOG, RASP, XCFAS, SMXC, CONSOLE, IOSAS, others
- SYSSTC Service Class
  - Second highest CPU & I/O dispatching priority: x'FE' or 254
  - Address spaces in SYSSTC not usually managed by WLM (except for XMEM delays and Servers)
  - Contains started tasks not explicitly classified to another service class
  - Work of the following subsystem types may go into SYSSTC
    - ASCH, JES, OMVS, STC, TSO

# Performance Goals



- Performance goals are assigned to each period in a service class
  - All service classes have at least one period and each period has a goal
- There are four types of goals:
  - Response time goal
    - Indicates how quickly you want work to be processed
    - Can be based on average or percentiles
  - Execution velocity goal
    - Indicates the speed (or acceptable delay) for work
  - Discretionary goal
    - For low priority work for which you do not have any particular performance goal
  - WLM defined goals
    - Implied objectives of work WLM determines as needing special requirements
- WLM algorithms manages all work to one of these goal types

# Response Time Goals



- Response Time = time on z/OS (including queue, execution, delay times)
- Average RT Goal = average RT of all ended transactions
  - Outlier transactions can have a significant impact on the average
- Percentile RT Goal = x% under y seconds
  - Outliers effectively ignored
- Whether to use Averages or Percentiles depends on the workload characteristics and what you're managing to
  - Many prefer percentiles due to the potential for outliers skewing the average
  - But there are workloads where the outliers are important
  - My current thinking is that if you don't have regular outliers that you want to ignore, averages may be better

# Velocity Goals



- Velocity is a measure of acceptable delay
- Request to WLM that states a percentage of 'time' when the work wants to run, it is able to run, and is not delayed for lack of WLM managed resources
- But not really 'time' – really a percentage of known samples

$$\text{Velocity} = \frac{\text{Using Samples}}{\text{Using Samples} + \text{WLM Managed Delay Samples}}$$

- Example: Velocity goal of 25 states that it is acceptable that 25% of known samples show when worked wants to run it has the resources to run
  - Thus: Velocity goal of 25 states that 25% of samples are using samples

# Velocity Issues



- Achievable velocity for a given workload depends on lots of factors
  - Units of work in the system (more = more samples = usually more delay samples)
  - Number of CPUs
  - Speed of CPUs
  - Amount of other work in the system
  - I/O Priority Management WLM Option
- Achieved velocities will vary and velocity goals need to be regularly revisited
- Be wary of very high velocity goals (e.g.  $\geq 85$ )
  - If achieving them, could be a result of having I/O Priority Management enabled, and it may be preventing WLM from effectively managing the work
  - (Look for my Revisiting WLM Options presentation)

# So What Type of Goal Should be Used?



- Response Time:

- Wherever you can, assuming:
  - The goal is in the seconds range (or sub-second)
  - There are enough transactions completing: at least 10 in 20 minutes, preferably more
- Great for transactional work
- Gives you response time reporting for your transactions (i.e. CICS)
- Easy to understand

- Discretionary:

- Work that can wait until other work is done
- Last “penalty” period for certain workloads
- Note that if the system is constrained, discretionary work will be mostly delayed

- Velocity:

- Use for everything else (most STCs, most batch) that shouldn't be in SYSSTC or SYSTEM

# What about SYSTEM & SYSSTC?



- Reserve SYSTEM for what automatically goes there
  - “HIPRI” attribute on when calling ASCRE to create new address space
  - May still want to classify such work though so you can assign a report class
  - MASTER, GRS, DUMPSRV, CATALOG, RASP, XFSAS, CONSOLE, etc.
- Every product thinks it should be in SYSSTC, but not everything should be
  - SYSTEM and SYSSTC don’t get proactive storage management
  - Not everything needs to run at dispatching priority 254 to get good service
  - If everything is #1, nothing is #1
  - Misbehaving (e.g. looping) address spaces in SYSTEM/SYSSTC can have a significant negative impact on all other work on the system

# SYSSTC Good & Bad Candidates



- Started tasks that *are* good candidates for SYSSTC:
  - Critical system tasks that shouldn't be in SYSTEM
  - Most monitors
  - IRLMs or other lock managers
  - High importance STCs that are predictable and don't use much CPU
- Started tasks that are *not* good candidates for SYSSTC:
  - Storage intensive or storage critical address spaces
  - Address spaces getting WLM server management
    - Can see this in SDSF where SERVER=YES
  - Most subsystem work managers and databases:
    - CICS, IMS, DB2, WebSphere, ADABAS, IDMS, etc.
  - Address spaces where CPU consumption is not predictable and could be high
    - Or you just don't know yet



# WLM Algorithms

# WLM Algorithms



- Policy Adjustment (PA)

- Done 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

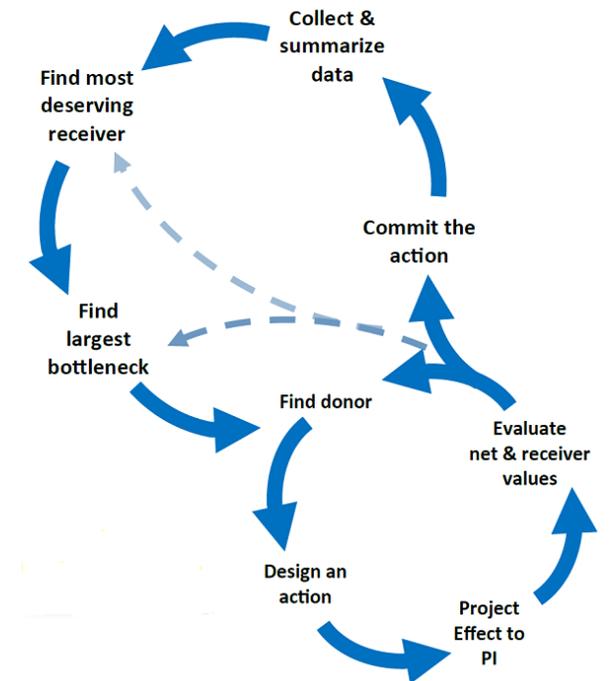
- Resource (RA)

- Done approximately every 2 seconds (AKA 'RA interval')
- Objectives include:
  - improve efficiency of system resources
  - avoided if at the expense of goals

# 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 significant 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



# Policy Adjustment Loop and SCPs



- Note that:
  - WLM will only attempt to help one SCP per PA interval
  - WLM will attempt to help highest importance SCP missing it's goals
  - PA Interval is 10 seconds
- So things don't change too quickly!
- It may take a while to get to lower-importance SCPs
- Hence, the rule of thumb:

You should have no more than 25-30 *active service class periods* per system at times of interest

# Performance Index (PI)



- During Policy Adjustment summarization WLM calculates the 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
- **Just because your work shows  $PI \leq 1$  doesn't mean your goals are good!**



# Top 3 Common WLM Issues

# Too easy goals



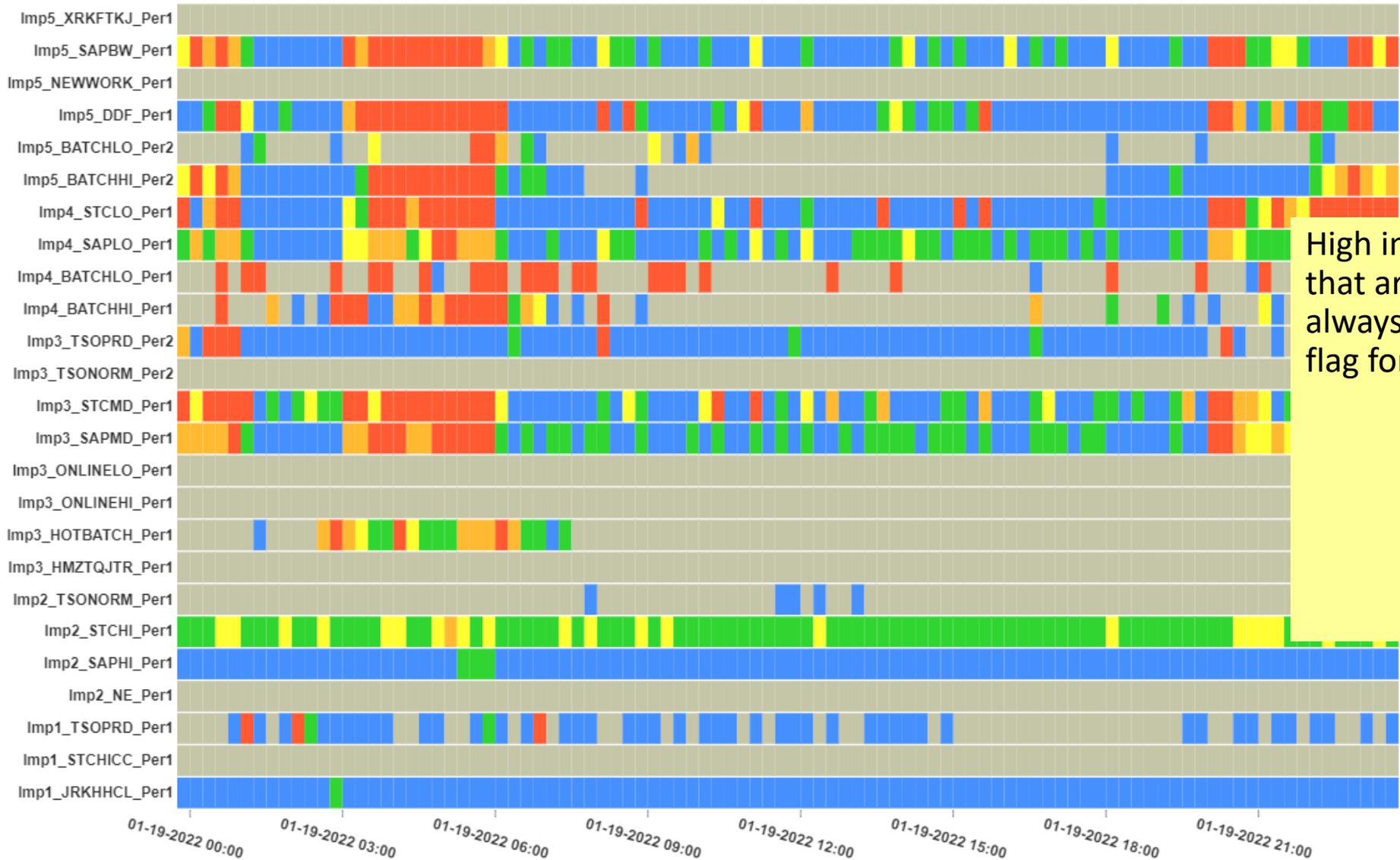
- When goals are too easy, the work essentially becomes unmanaged
- A PI hovering around a little more than 1 is goodness
- Opposite is also true: goals that are too difficult also results in WLM effectively ignoring the work

# WLM PI - PI Heat Chart for Service Class Periods



- ≤ 0: Zero
- ≤ 0.81: Over Achieving
- ≤ 1.1: Met
- ≤ 1.4: Fair
- ≤ 1.99: Warning
- higher: Severe

PRODPLEX, SYSL



High importance SCPs that are always blue are always (ironically) a red flag for me.

# RT Goals for the wrong work



- To effectively use response time goals, the work being managed should have a regular completion of work
  - Transactions per second = good fit!
  - Transactions per minute = maybe, but think about it
  - <1 Transaction per minute = probably not a good fit

# Ended Transactions Over Time

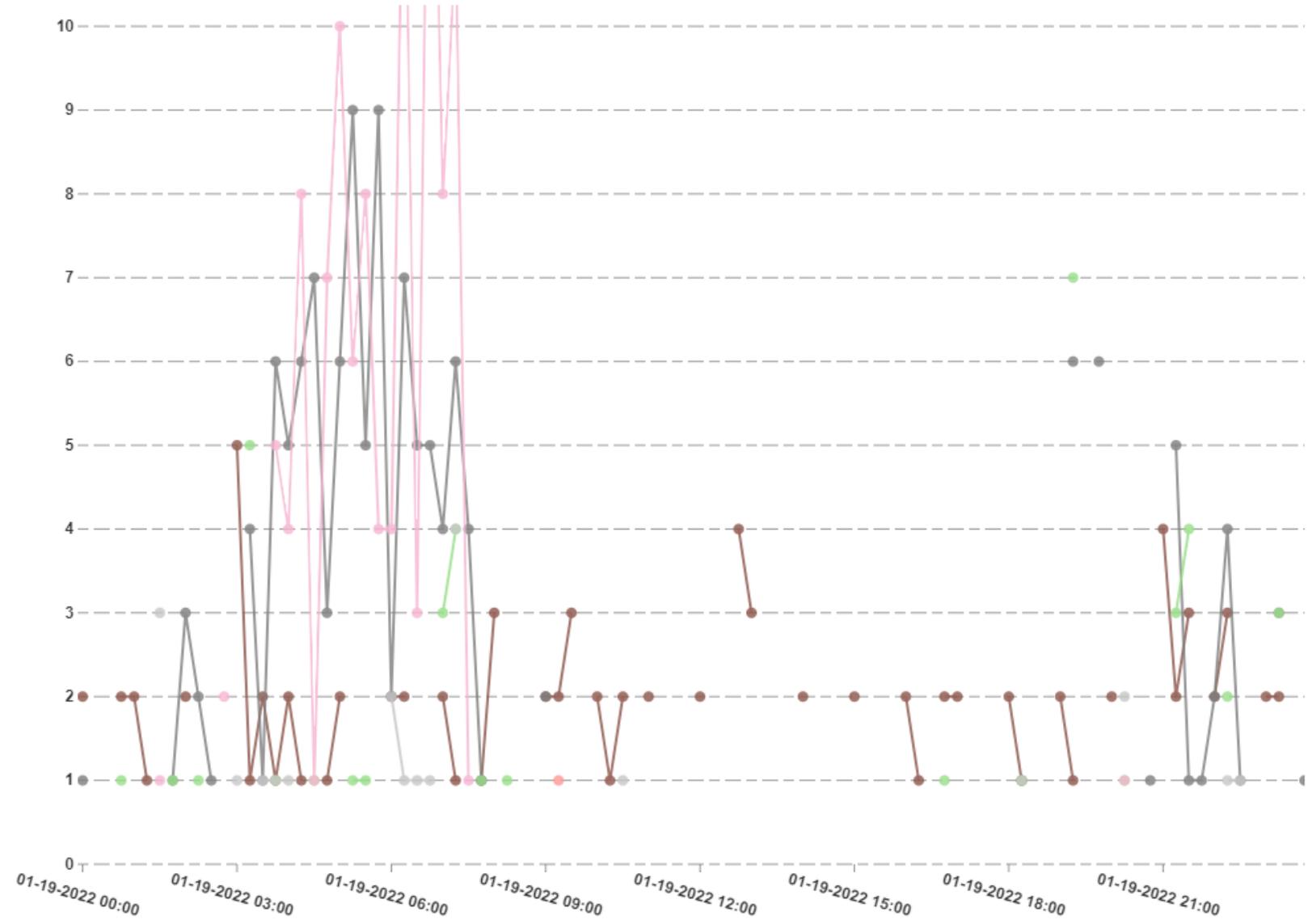
(Top Periods With Greater Than 0 Ended Transactions)



PRODPLEX, SYSL

- Imp2 NE\_Per1
- Imp2 SAPHI\_Per1
- Imp2 STCHI\_Per1
- Imp3 SAPMD\_Per1
- Imp4 BATCHLO\_Per1
- Imp4 SAPLO\_Per1
- Imp5 BATCHHI\_Per2
- Imp5 DDF\_Per1
- Imp5 SAPBW\_Per1
- Imp3 STCMD\_Per1
- Imp4 STCLO\_Per1
- Imp0(SYSSTC) SYSSTC\_Per1
- Imp4 BATCHHI\_Per1
- Imp1 TSOPRD\_Per1
- Imp3 TSOPRD\_Per2
- Imp3 HOTE
- Imp6(DISC)
- Imp2 TSON
- Imp6(DISC)
- Imp5 BATC

These are total ended transactions (jobs) per 15 minute interval for some batch service classes. Probably not good candidates for RT goals.

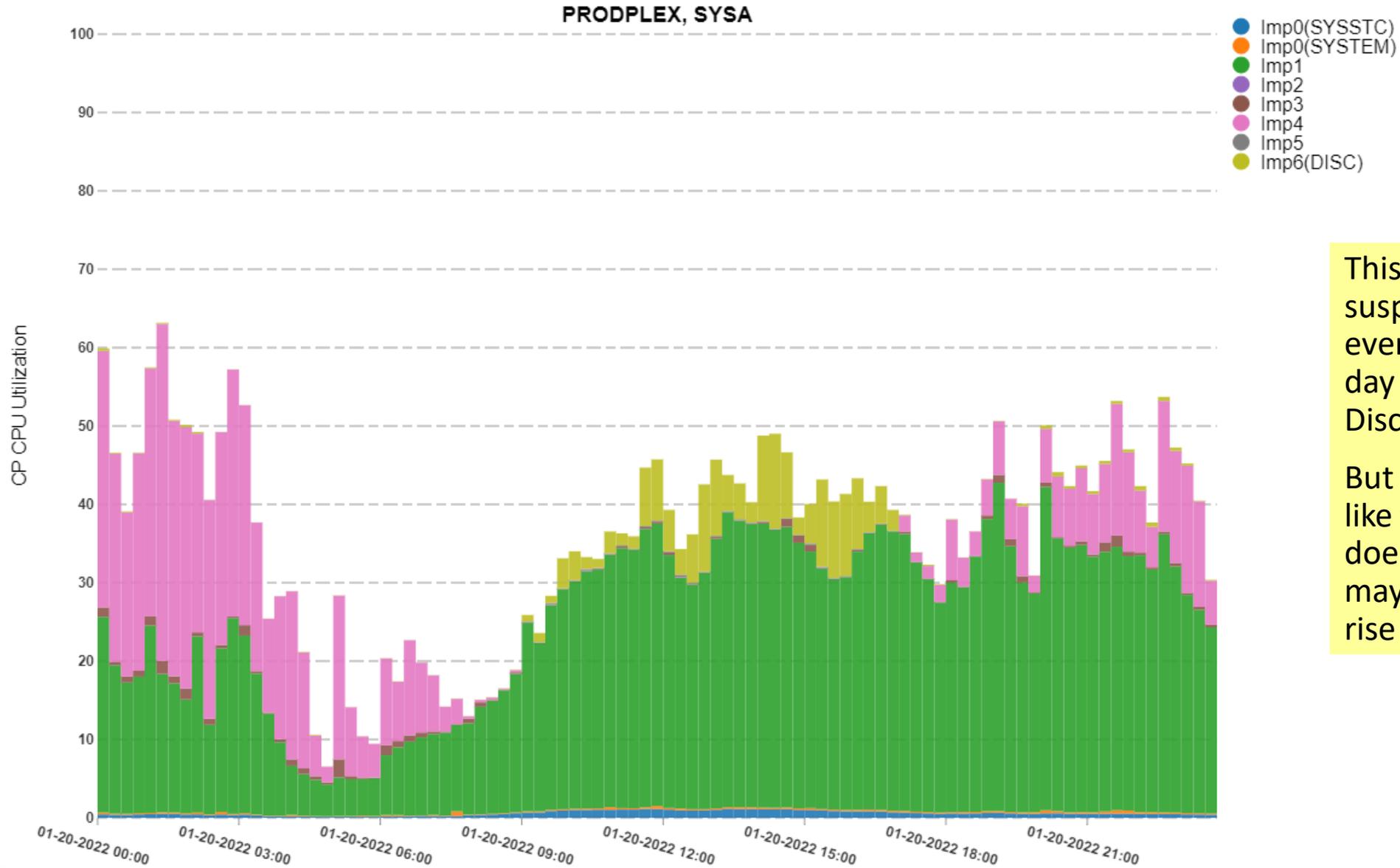


# Importance Bunching



- If everything is most important nothing is most important
- We regularly see customers who have the majority of their work running in one or maybe two importance levels, this is usually a problem
  - May not be a problem if the system truly does do only one type of work
    - E.G. this LPAR provides DB2 DDF access to just one type of transaction
  - Even if the LPAR is “production online” only, most likely you have both a transaction manager and database there and those can and (and often should) run at different importance
  - Usually there will be variety of transactions too and it can be beneficial to classify transactions to different importances as well (where possible)

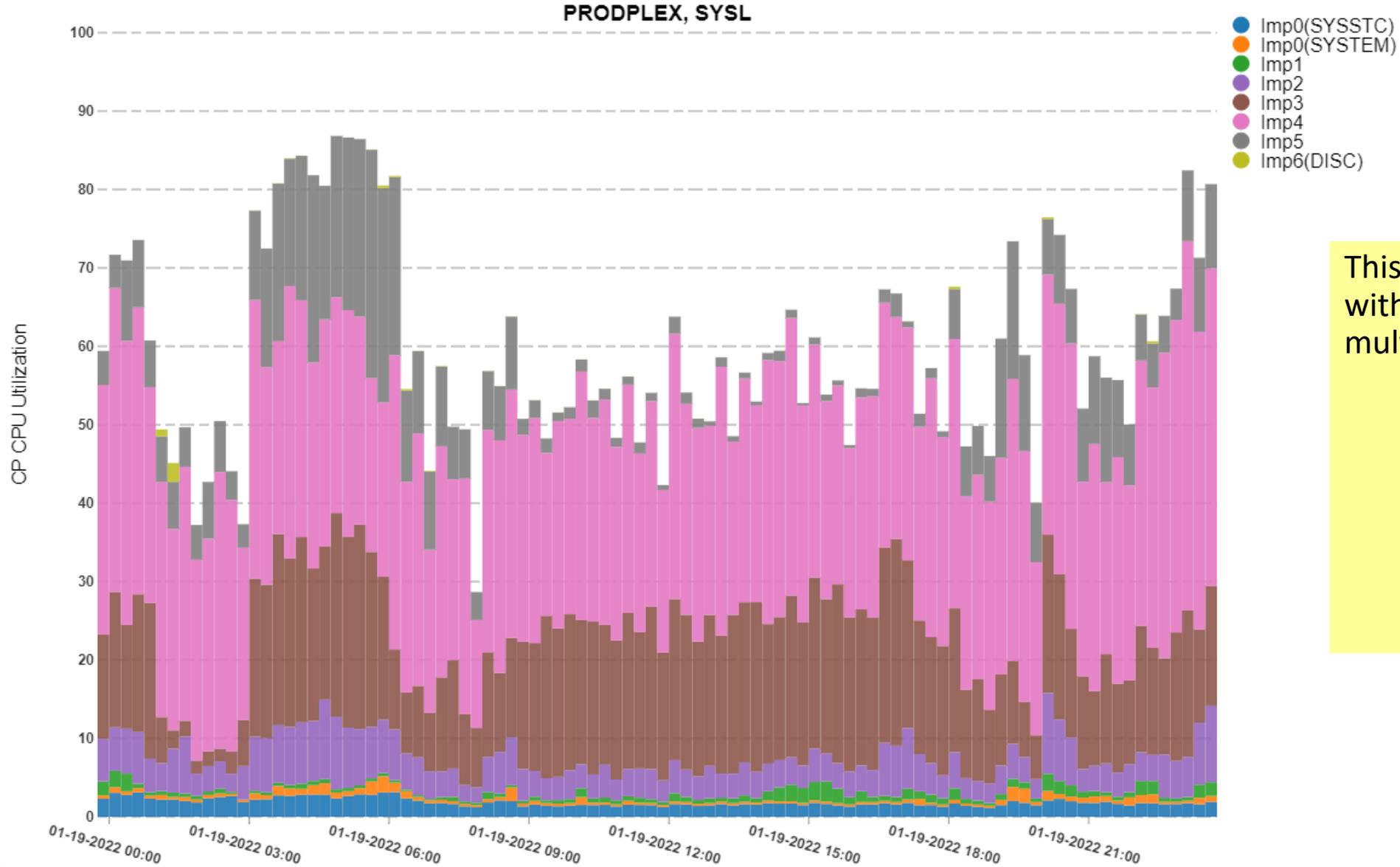
# WLM Importance - CP CPU Utilization (CP + zAAP on CP + zIIP on CP)



This is somewhat suspect: pretty much everything during the day is importance 1 or Discretionary.

But at low utilizations like this it probably doesn't much matter. It may when utilizations rise though!

# WLM Importance - CP CPU Utilization (CP + zAAP on CP + zIIP on CP)



This looks pretty good with work spread across multiple importances.



# Summary

One-Sheet WLM

# Complete your session evaluations for a chance at daily prizes!

To complete, visit  
[www.share.org/evaluation](http://www.share.org/evaluation)  
and see your progress on the  
leaderboard!

PERSON

PROGRAM

PROJECT