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

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:

Trademarks:

Abstract

WLM's Algorithms - How WLM Works

The WLM component of the z/OS operating system 'wakes up' on a regular basis to adjust system settings to meet goals. What is happening when it 'wakes up'? During this presentation Peter Enrico will explain how WLM works, and flow of the WLM algorithm decision making process. This is an interesting presentation, and the attendee is sure to learn a great deal about WLM.

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
- MSU Reductions
  - Engagements designed specifically to reduce your MSU consumption
- 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!

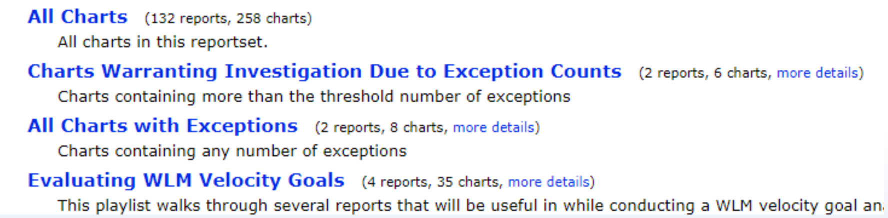
- Essential z/OS Performance Tuning
  - October 3-7, 2022
- WLM Performance and Re-evaluating Goals
  - September 12-16, 2022
- Parallel Sysplex and z/OS Performance Tuning
  - February 7-8, 2023
- Also… please make sure you are signed up for our free monthly z/OS educational webinars! (email contact@epstrategies.com)

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

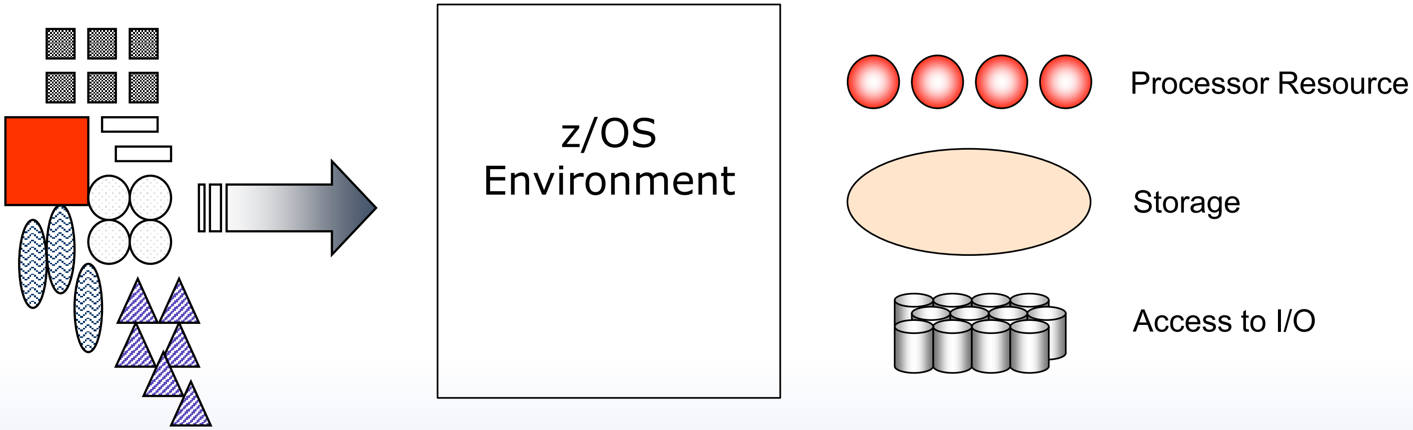
EPS presentations this week

What	Who	When	Where
PSP: z/OS Performance Tuning - Some Top Things You May Not Know	Peter Enrico Scott Chapman	Tue 1:15	Delaware A
Planning Your Next Mainframe Processor Upgrade	Scott Chapman	Tue 2:45	Franklin C
z/OS Performance Risk Management: Easy Things To Do To Reduce the Risk of Bad Performance	Scott Chapman	Wed 10:30	Franklin C
Pinpointing Transient Performance Problems with SMF 98 & 99	Peter Enrico	Thu 8:00	Franklin A
WLM's Algorithms - How WLM Works	Peter Enrico	Thu 1:15	Franklin C

System Resources

- A z/OS system has a finite amount of resources
  - CPU capacity, Storage capacity, I/O subsystem, logical resources, etc..

- All *knobs* and *dials* under WLM's control are dynamically set by the WLM to meet the goals of the work being managing



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!

Key WLM Design Point:

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

SRM Relationship to WLM

- SRM still exists when system is in goal mode
  - SRM controls access to system resources
    - Back in MVS/ESA Days there were a 'static set of controls' in IPS/ICS/OPT
    - With WLM such controls were replaced by WLM algorithms

- WLM component has two primary functions
  - Provides externals interfaces to allow system administrators and applications to interface to WLM externals
    - SRM is then hidden
  - Policy adjustment and resource adjustment algorithms
    - To set controls

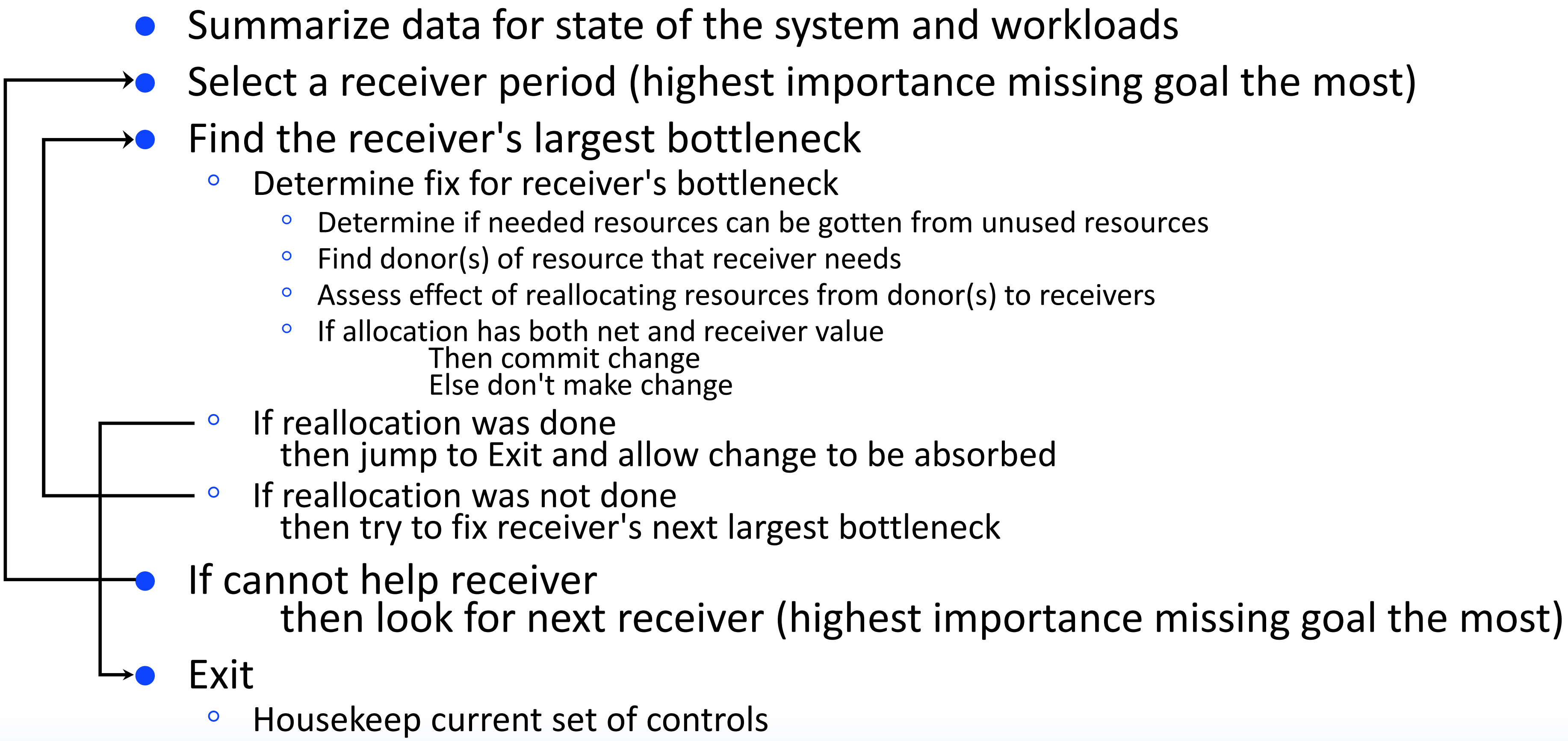
WLM Algorithm Phases

- 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

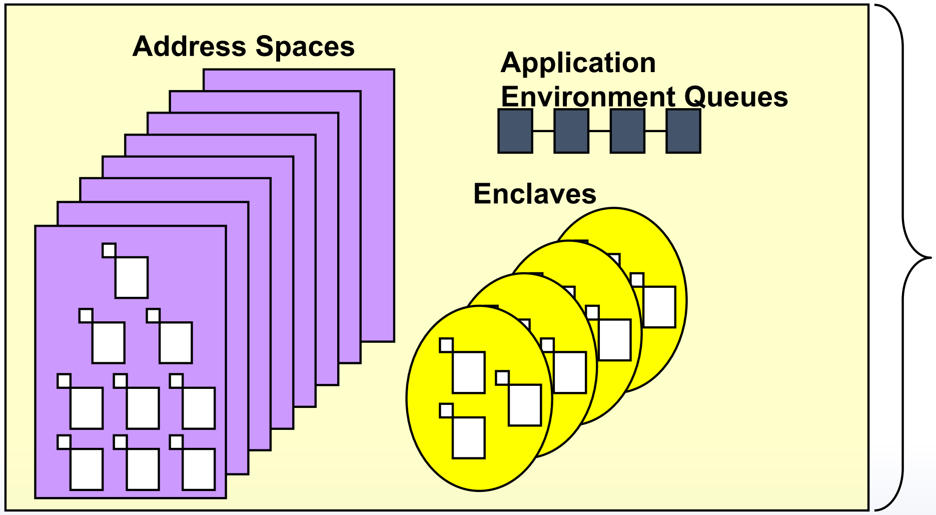
Summarize data for state of the system and workloads

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

Example: Resources

- WLM collects lots of measurements on what periods, address spaces, and enclaves are using which resources
  - CPU – CPU consumption, rate of consumption, etc.
  - Storage occupancy
  - Devices used by a particular service class
  - Etc.



- Period MPL
- Period CPU usage
- Period I/O information
- Address space data
- Enclave data
- Application environment Queue length data

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
State Sampled	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
	Using		Non-Storage Delays						Storage Delays				

What are the different sampled states?

- Every 1/4 second the WLM state sampler ripples through the system to sample a variety of states
  - Sampling used rather than actual measured Using and Delays since precise measuring too expensive (although some states are measured and converted to samples)

- The states that WLM samples can be broadly categorized as follows:
  - Using states
  - Delay states – Processor
  - Delay states – non-Processor, non-Storage
  - Delay states – Storage for goal honored address spaces
  - Delay states – Storage for goal ignored address spaces
  - Other states

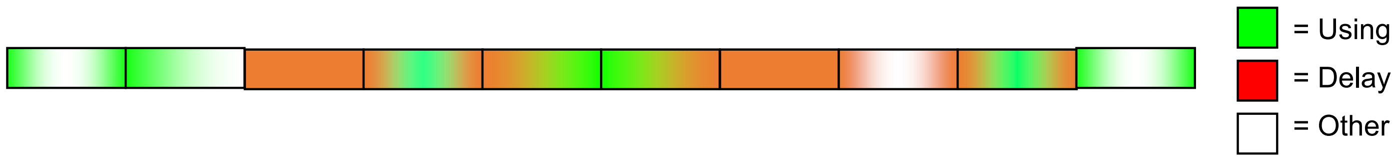
- A high level summarization of the using and delay states are shows on the next several foils

- WLM also samples other states related to work managers (such as CICS, IMS, DB2, WAS, etc.)
  - But the focus of this presentation is WLM state samples used by the WLM algorithms

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

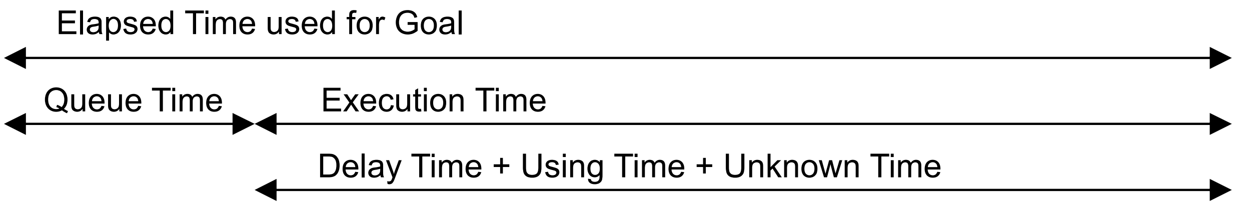
$$\frac{\text{CPU Using} + \text{I/O Using}}{\left(\begin{array}{l}\text{CPU Using} + \text{I/O Using} + \text{CPU Delay} + \text{I/O Delay} \\ + \text{Paging Delays} + \text{MPL Delays} + \text{Queue Delays}\end{array}\right)}$$



- 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



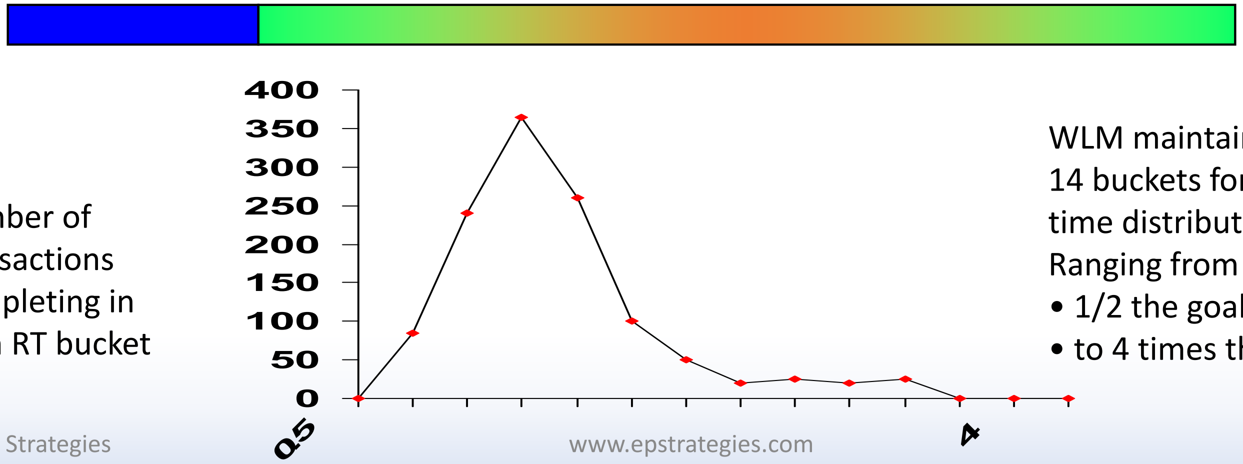
Elapsed Time used for Goal

Queue Time

Execution Time

Delay Time + Using Time + Unknown Time

- Response time distributions



Number of Transactions Completing in Each RT bucket

WLM maintains 14 buckets for response time distribution

Ranging from

- 1/2 the goal
- to 4 times the goal

WLM maintains response time distributions

- WLM maintains a response time distribution for response time goal periods
  - Distribution compose 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%
	0	85	240	365	260	100	50	20	25	20	25	0	0	0

Transaction Count

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

Performance Index (PI) Formulas

- Average Response Time goal periods

$$\text{Average RT Goal PI} = \frac{\text{Actual Average Response Time}}{\text{Average Response Time Goal}}$$

- Velocity goal periods

$$\text{Velocity Goal PI} = \frac{\text{Velocity Goal \%}}{\text{Actual Velocity Achieved}}$$

- Percentile Response Time goal periods

$$\text{Percentile RT Goal PI} = \frac{\text{Actual RT at Percentile}}{\text{Response Time Goal at Percentile}}$$

- Discretionary goal periods
  - Always have PI of .81

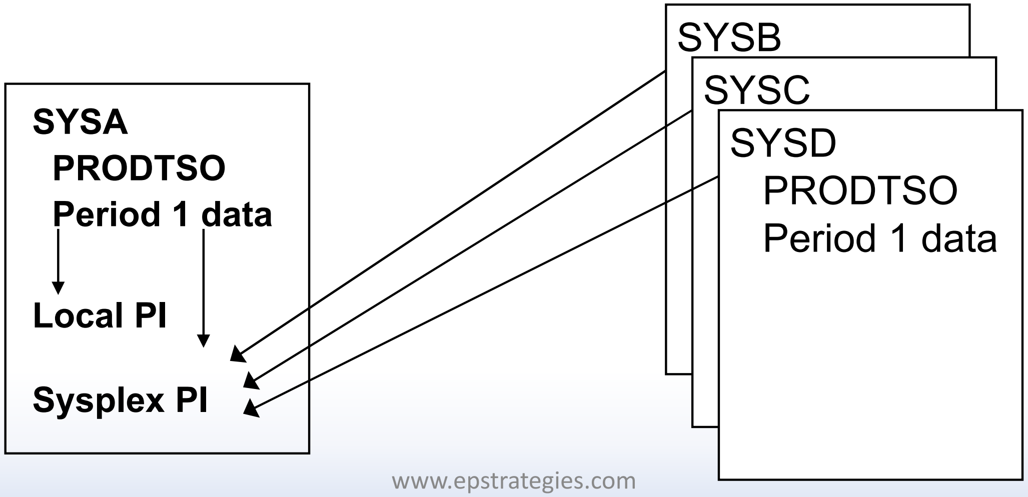
- To calculate the PI for a percentile RT goal we need response time at percentile
  - Determine total number of completed transactions (add all buckets)
  - Using the percentile objective, calculate the percent of total transactions that needed to complete within a certain response time
  - Add buckets 1 to n until you get a transaction count of at least that calculated in step 2
  - Calculate PI by dividing the response time represented by the nth bucket by goal response time

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%
Trans Count	0	85	240	365	260	100	50	20	25	20	25	0	0	0

- Example: Goal = 85% within 2.0 seconds
  - Total completed transaction (in above distribution) = 1190
  - 85% of 1190 = 1011.5
  - When add buckets 1 to n we find the 6th bucket brings us to 1050 (just past goal value)
  - The 6th bucket represents 100% of goal of 2 seconds or 2.0 seconds
  - PI = 2.0 seconds / 2 seconds = 1.0

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

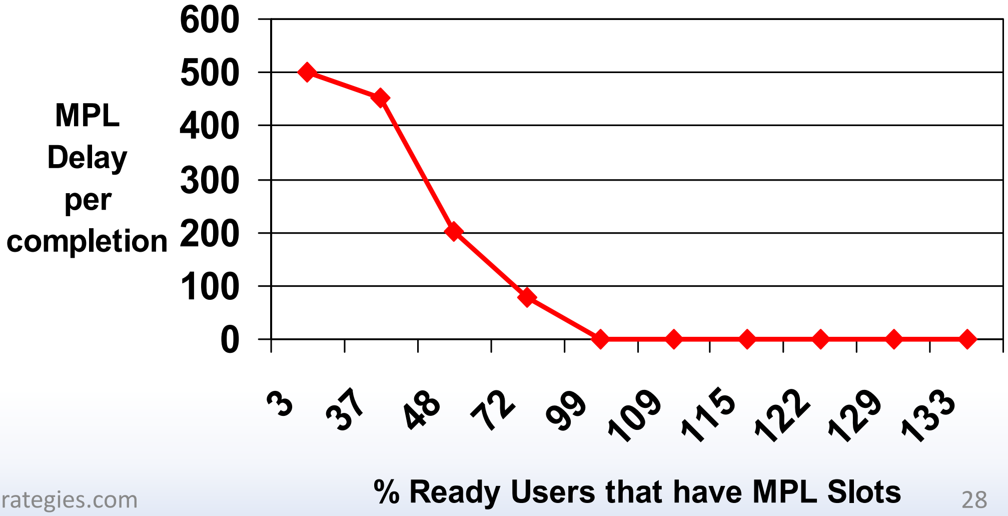


WLM maintains a series of plots

- 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

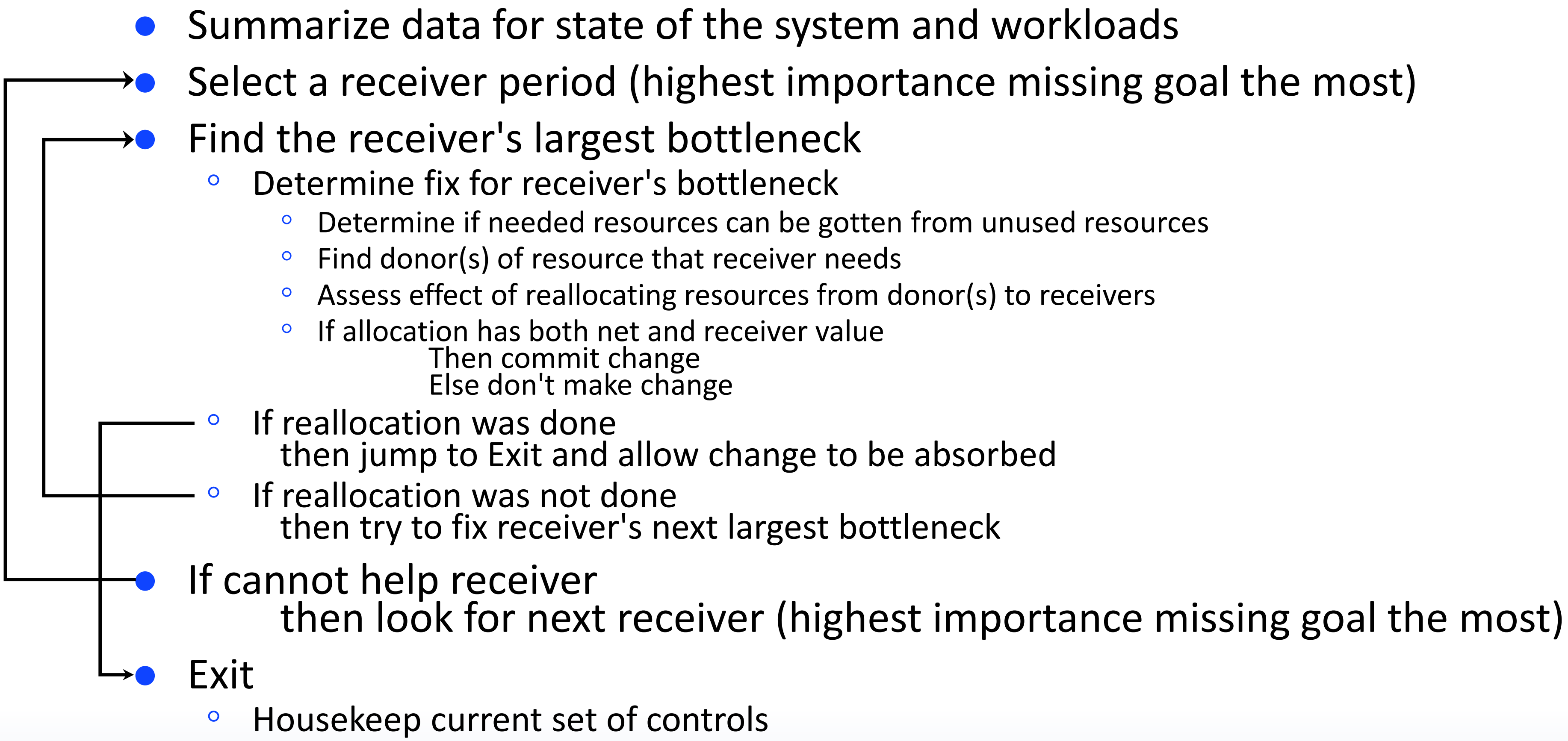
- Histories used to examine data over a period of time
  - Allows WLM to have a controlled way to go back in time for enough representative data points for accurate decisions
  - For example, a history *may* have the following structure

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

Figuring out what periods need help, and which periods can be taken from

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

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

PA Loop: WLM selects a Receiver

- Select a service class period to help in the following order:
  - By importance
  - By PI
    - More emphasis is placed on sysplex PI than on local PI
    - Only selects periods with PI > 0.9
  - Periods in resource groups below service minimum are also considered

- Basically WLM tries to help the most important work missing its goal the most
  - Tries to meet Sysplex goal at importance, then tries to meet local goal

- Note: Many factors are involved here and it is not a simple selection:
  - If 'critical controls' are in effect this process becomes more complicated
  - Even more complicated if resource group maximums and minimums are being utilized

PA Loop: Selecting a Receiver

- Items to note about selecting receivers
  - When selecting a receiver (or a donor), WLM takes both the local and Sysplex PI into consideration
  - Will first consider periods not meeting goal before considering periods meeting goals
  - Takes into consideration if the period is below resource group minimum,
  - Takes into consideration discretionary goals below resource minimum
  - If goals are being met, then takes into consideration periods meeting goals (including discretionary periods)

- Search order is complicated, but at a very high level (assuming no resource groups or critical controls) the selection algorithm is similar to the following:
  - Search importance 1 periods based on Sysplex PI
  - Search importance 1 periods based on local PI
  - Search importance 2 periods based on Sysplex PI
  - Search importance 2 periods based on local PI
  - Search importance 3 periods based on Sysplex PI
  - Search importance 3 periods based on local PI
  - Etc.

PA Loop: WLM finds Receiver’s bottleneck

- Find the receiver's largest delay
- Assumptions include:
  - Sampled delays are contributing to why the receiver is missing goals
  - Largest delay of a receiver indicates receivers biggest inhibitor to meeting goal
  - An action to decrease delay samples should improve period performance
  - If period is being served, find largest delay impacting a supporting period
  - Only delays that WLM can manage are considered and addressed

- Delay data
  - Standard Delays:
    - CPU, I/O, Capping due to max, swap in, MPL, QMPL
  - Auxiliary Storage Delays:
    - Private area paging, Common area paging, Cross memory paging, VIO, standard Hiperspace, ESO Hiperspace
  - Server Delays:
    - MPL, Private area paging, VIO, standard Hiperspace

- Note: WLM will not try to help 'unknown' or states that it does not know about

PA Loop: WLM finds a potential Donor

- Select potential donors of required resource
  - Generally, the reverse order of receiver selection
  - Discretionary resources first considered
  - Discretionary goal periods are considered next
  - Periods meeting goals are then considered to the best donors
    - Then select from lowest importance to highest importance
  - Select work that holds resource in need (i.e. storage, higher DP)
  - Both resource group minimums and PI are considered
  - Multiple donors may donate multiple resources to help a receiver during one policy adjustment interval
  - Resource may come out of discretionary resources
    - Those resources that can be reallocated with little or no effect on the system's ability to meet goals

WLM Algorithms

This section is a very high-level overview, and does not cover all areas

WLM attempts to fix Receiver's Bottleneck

- WLM directs the setting of the following SRM controls
  - CPU Dispatching priorities
  - I/O Dispatching priorities for UCB queuing, Parallel Access Volumes, etc…
  - Target multi-programming levels
    - MPL in-target
    - MPL out-target
  - Storage targets (proactive and reactive)
    - Restrictive targets
    - Protective targets
  - Capping
  - Number of server address spaces serving an application environment queue
  - Batch initiators
  - Etc.

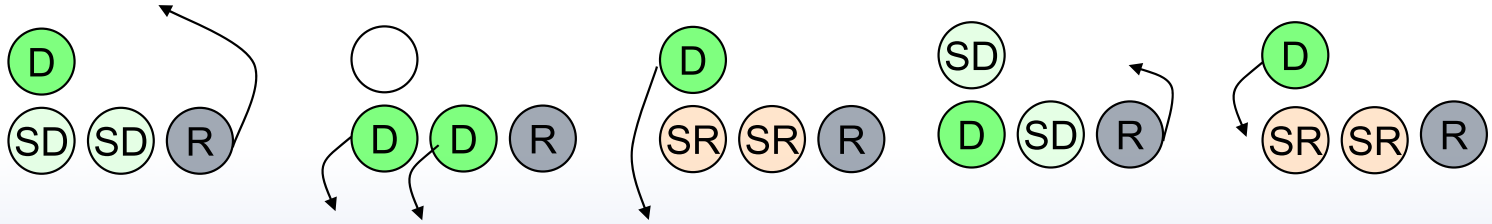
Example: WLM Possible WLM Actions - CPU

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

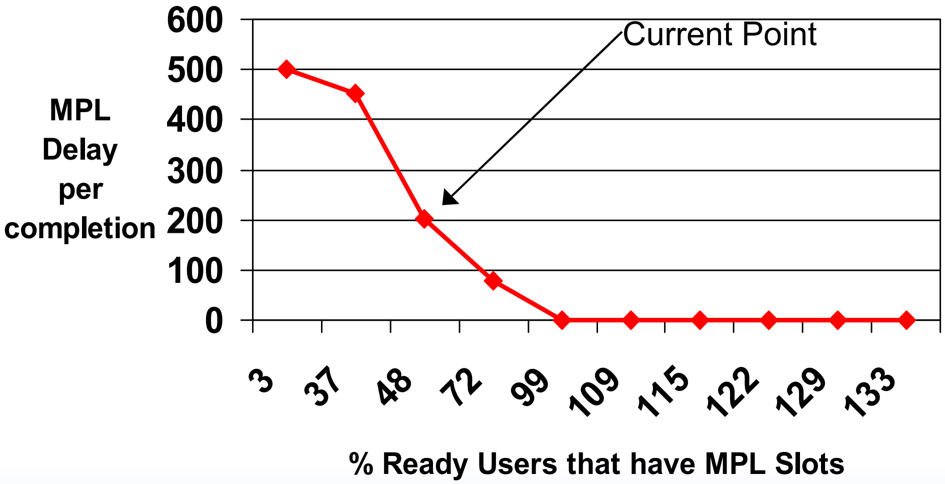
Example: WLM Possible WLM Actions - CPU

- 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



Example: WLM Possible WLM Actions - MPL

- MPL Targets
  - Receiver may have MPL targets increased to alleviate MPL delay
  - Donor may have MPL targets lowered or donate storage
  - WLM adjusts the MLP in-target and out-target for the receiver period



MPL Plot Example:

- Shows how response time may improve by increasing MPL slots
- Shows how response time may degrade by reducing MPL slots
- Also needs the Ready User average plot when considering decreasing MPL

Individual Address Space Monitoring

- The granularity of most WLM controls are at the service class period level
  - But some address spaces may have individual storage policies
  - That is, an address space may have a storage policy separate and different from all other address spaces in its own period

- For an address space to have an individual address space storage policy it must be monitored separately for a period of time by WLM
  - WLM only monitors address space if finds 'interesting'
    - Example: if an address space is using a lot of storage

- Address spaces eligible for monitoring and individual storage policies include
  - Address spaces assigned a velocity goal
  - Address space assigned a discretionary goal
  - Address space is found to be a server
    - Example: CICS transaction management turned on, so a CICS address space is eligible
  - Address space is assigned a response time goal of greater than 20 seconds

Review of WLM Storage Controls

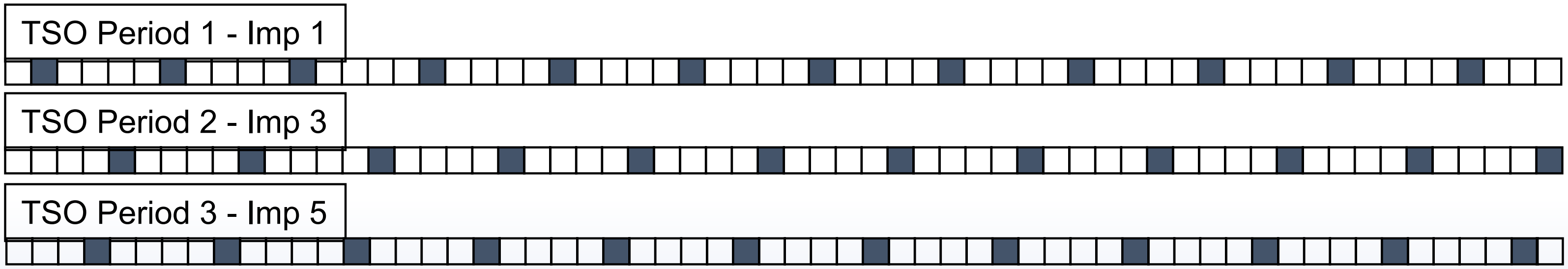
- Currently WLM policy adjustment algorithms exist to alleviate or prevent paging, and to manage access to processor storage
  - Storage isolation targets are the most common way:
    - Individual storage targets (placed on individual address spaces)
    - Period wide storage targets (placed on all address spaces in a period)
    - Common area storage targets (placed on common areas such as CSA)

- Types of storage targets include:
  - Protective processor
    - Storage target to protect some number of frames in processor storage
    - A way to ensure an address space has the storage it needs to avoid paging
    - Protective targets are especially useful for server subsystems like CICS, IMS, DB2, Server address spaces
  - Restrictive processor
    - Storage target to preferentially migrate number of frames down to the target
    - A way to limit the amount of processor storage an address space occupies to prevent other address spaces from paging
    - Restrictive targets are useful for large batch workloads that want to consume more storage than is available

WLM PA Actions - Group Caps

- Cap Slices
  - Used to enforce resource group maximums
  - 1 cap slice = 1/256 of an SRM second
  - Cap slices spread evenly for each period

TSO Period	CPU/SEC	PI	CPU/SEC	PI	CPU/SEC	PI
1	883.2	0.3	728.0	0.3	505.4	0.4
2	307.6	0.6	245.4	0.7	159.1	1.0
3	586.4	0.4	517.7	12.2	298.2	33.0
Total	1777.2		1491.1		963.7	
Group Max	2000		1500		1000	



WLM PA Actions - I/O Priority

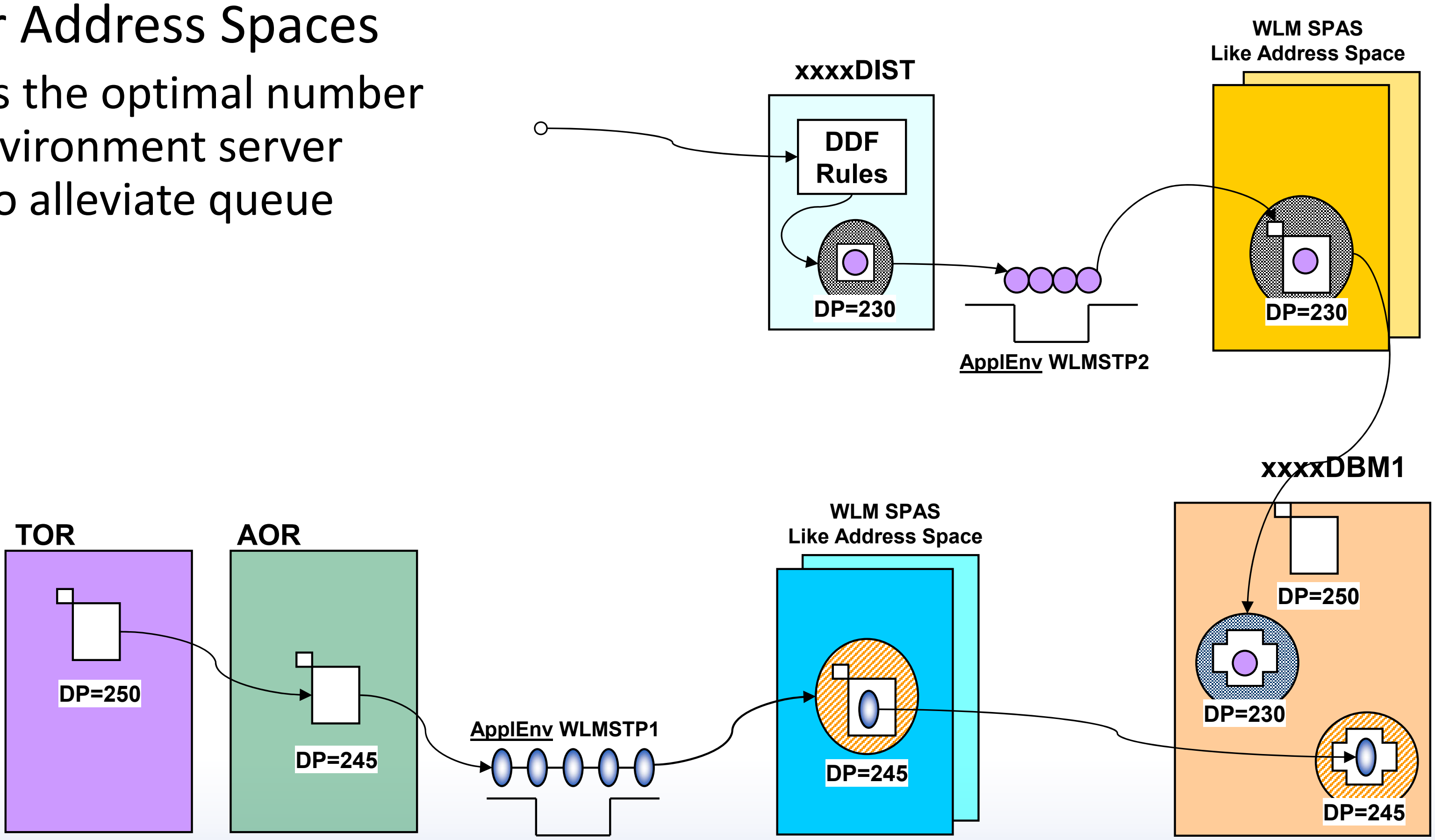
- I/O Priority
  - Works similar to WLM CPU dispatch priority algorithms
  - Donor must be competing with receiver for some of devices so action has effect
  - Device Clustering
    - WLM needs to be aware of periods competing for same devices
    - Device clustering is used to determine this relationship
      - Each class associated with a single cluster

Device Clusters:
- Cluster 1 = 1,2,3
- Cluster 2 = 4,5

Service Class	Dev 200	Dev 201	Dev 202	Dev 500	Dev 501	Dev 502	Dev 503
Class 1	100	150	150	0	0	0	0
Class 2	0	90	100	0	0	0	0
Class 3	0	100	100	5	0	0	0
Class 4	0	0	0	100	100	100	100
Class 5	0	0	0	0	150	0	150

WLM PA Actions - Server Spaces

- Number of Server Address Spaces
  - WLM determines the optimal number of application environment server address spaces to alleviate queue delays



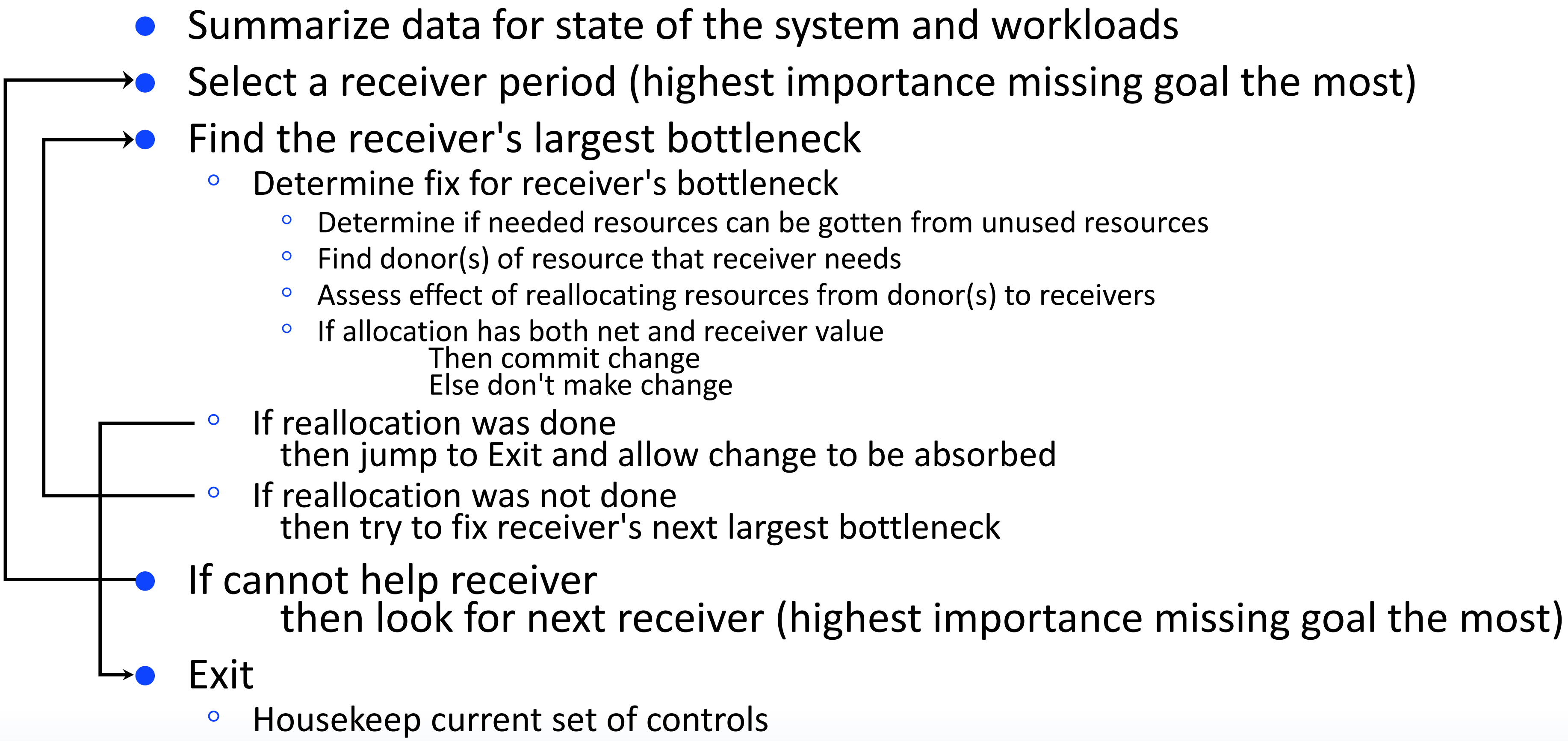
WLM Discretionary Goal Management

- Discretionary Goal Management
  - When certain types of work are over-achieving their goals, they may have their service capped in order to give discretionary work a better chance to run
  - When no Discretionary Goal Management
    - On saturated systems, discretionary work can be shut out
      - even when other work is over-achieving goals
    - Some customers move work to importance 5
- With Discretionary Goal Management
  - Cap over-achieving work to allow discretionary to run
  - Uses ‘internal resource group’ maximums to manage capping
    - Capping begins when PI drops below .7
    - Capping stops when PI rises above .81
  - Criteria for donors avoids impact to online work
    - R/T goal greater than 1 minute
    - Velocity goal less than or equal to 30%
    - Later periods capped first

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

Up to this point, nothing has changed…
Should a possible change be permitted?

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

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

- Receiver is only helped by a specific donor if there is projected to be sufficient *net value*
  - Designed to reject changes that will harm the donor more than the projected improvement to the receiver
  - Allows WLM to assess taking from other donors

- All external service policy specifications are considered for both primary and secondary donors
  - goals
  - importance
  - resource group minimums and maximums

- Example
  - PRODBAT PI = 4.0
  - WLM algorithms suggest improvements can bring PI to 3.0
  - Change hurts donor more then helps receiver

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 PA Loop: Commit Change

- If change has both receiver and net value then the change is committed
  - At most 1 period is help every PA interval
- If change is not committed
  - Try fixing another bottleneck for the receiver
  - If receiver does not have another bottleneck, find another receiver
- Periods are sometimes ‘ignored’ when help is given
  - There is a skip counter set when a service class period is chosen as a receiver for help and help has been given.
    - In this case the skip counter is set
    - In order to be marked you must have been chosen as a receiver first

WLM PA Loop: Housekeeping

- Housekeeping storage controls that may be out of date or ineffective
  - Storage targets are reduced
    - if doing so will not effect goals
    - WLM reduces targets by 'nibbling' away at them
  - Allows WLM to react to changes in the workload as time passes
    - If target is later needed, it can always be increased again by policy adjustment
    - Proactive approach to resource management

- Reverse Housekeeping anticipates the need for higher targets
  - If an address space or period owns a lot of storage, and targets are low, then increase the targets
    - Provides work with needed protection in case of a sudden increase in demand for resources by other work
    - Proactive approach to resource management

WLM Resource Adjustment

- How WLM keeps system resources effectively utilized
  - Runs every 2 seconds

- Includes:
  - Detecting and addressing under-utilized, over-utilized, and shortage conditions

- All resource adjustment decisions are within the constraints set by policy adjustment algorithms
  - Not done at the expense of goals!

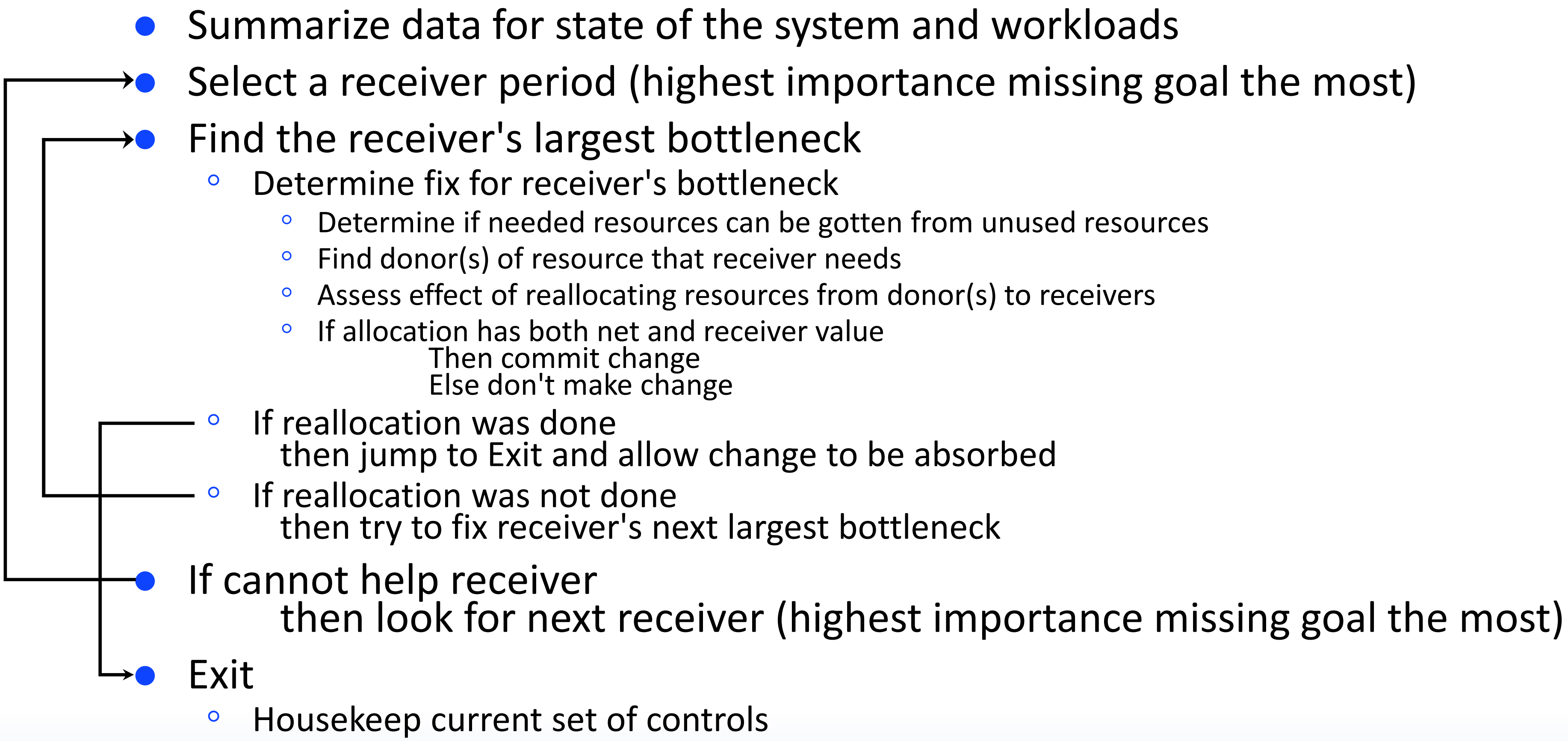
WLM Resource Adjustment Actions

- If system resources are over utilized
  - Get rid of work via lower MPL, etc.

- If system resources are under utilized
  - Allow more work to come into system via higher MPL, etc.

- If unmanaged paging too high
  - Usually when an unmanaged address space is causing paging
  - Look for address spaces that are paging and manage them

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?