

WLM Velocity Goals in the Year 2017

Peter Enrico

Email: Peter.Enrico@EPStrategies.com

Enterprise Performance Strategies, Inc.
3457-53rd Avenue North, #145
Bradenton, FL 34210

<http://www.epstrategies.com>

<http://www.pivotor.com>

Voice: 813-435-2297

Mobile: 941-685-6789



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



Creators of Pivotor®

© Enterprise Performance Strategies, Inc.

Contact, Copyright, and Trademark Notices

Questions?

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

Copyright Notice:

© 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 Velocity Goals in the Year 2017
 - Velocity goals have been around since the inception of the z/OS Workload Manager. Velocity is the goal that is assigned to longer running work when that work needs to make progress and there is no short response time objective. Unfortunately, achieving velocity goals are so heavily dependent on a variety of processor related factors such as capacity of the processor, number of processors, HiperDispatch, capping, defined capacity maximums, weight enforcements, specialty engines, SMT, and so much more.
 - During this session Peter Enrico will explain velocity goals and the latest influences in the modern day era.

2017 z/OS Performance Workshop Schedule

During these workshops you will be analyzing your own data!

□ Parallel Sysplex and z/OS Performance Tuning

Instructors: Peter Enrico and Scott Chapman

- August 29-31, 2017 **online**

□ WLM Performance and Re-evaluating of Goals

Instructors: Peter Enrico and Scott Chapman

- June 12-16, 2017 in Dallas, TX
- September 25-29, 2017 in Minneapolis, MN

□ Essential z/OS Performance Tuning Workshop

Instructors: Scott Chapman and Peter Enrico

- Fall, 2017 in Columbus Ohio

WLM Measurement Reports Processing/Discussion Offer !!!

❑ Special Reports Offer!

- See your WLM SMF 99 records in chart and table format
- Please contact me, Peter Enrico for instructions for sending raw SMF data
 - ❑ Send an email to peter.enrico@epstrategies.com
- Deliverable: Dozens of WLM based reports (charts and tables)
 - ❑ Period setup
 - ❑ Performance Index analysis
 - ❑ Velocity goal analysis
 - ❑ Response time goal analysis
 - ❑ Multi-period analysis
 - ❑ WLM workload analysis – batch, started tasks, DDF, etc.
 - ❑ WLM resource analysis (CPU / storage / I/O)
 - ❑ And much more!
- ❑ One-on-one phone call to explain your measurements



www.pivotor.com

Presentation Overview

- Introduction to WLM velocity goals
- A review of using and delay samples
- Factors that influence achieved velocities
- Performing a velocity goal analysis
 - And recommendations!

Introduction to Velocity Goals

Introducing 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 it 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 60 states that it is acceptable that 60% of known samples show when worked wants to run it has the resources to run
 - Thus: Velocity goal of 60 states that 60% of samples are using samples

Velocity Calculation

- Based on WLM managed using and delay samples
 - Using: CPU and optionally I/O
 - Delay: CPU, Paging, MPL, queuing, and optionally I/O

$$Velocity = \frac{CPU\ Using + I/O\ Using}{CPU\ Using + I/O\ Using + CPU\ Delay + I/O\ Delay + Storage\ Delays + Queue\ Delays}$$



- = Using
- = Delay
- = Unknown

Velocity is NOT speed!

- Velocity is not a goal or measure of speed for how fast the work should run or is running
 - It is merely a measure of the relationship of progress and delay

- Velocity goals are in place for work for which regular 'progress' is designed
 - Say a fast motorbike can run at a speed 120 mph
 - If it can travel at 120mph, does travel at 120mph and has zero delay
 - Then is traveling at velocity of 100
 - Say a bicycle can run at a speed of 20 mph
 - If it can travel at 20 mph, does travel at 20 mph, and has zero delay
 - Then is traveling at velocity of 100



Let's talk about samples

What is WLM state sampling?

- 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				

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

- Note: 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 Using States

- CPU using
 - Dispatchable units found using the general CP processors
 - Includes AAP or IIP using

- AAP Using
 - Dispatchable units found using the zAAP processors

- AAP on CP Using
 - zAAP eligible dispatchable units found using the general CP processors

- IIP Using
 - zIIP eligible dispatchable units found using the zIIP processors

- IIP on CP Using
 - zIIP eligible dispatchable units found using the general CP processors

- I/O using
 - Connect time for DASD I/O converted to samples

WLM Delay States - Processor

- CPU delay
 - A dispatchable unit found waiting to be dispatched, or a TCB waiting for a local lock

- AAP delay
 - zAAP eligible dispatchable units found waiting for a zAAP processor

- IIP delay
 - zIIP eligible dispatchable units found waiting for a zIIP processor

- CAP delay
 - Work is marked non dispatchable due to the enforcement of a resource group maximum

 - Work is marked non dispatchable due to the enforcement of discretionary goal management
 - Should only show up for Velocity goals ≤ 30 , or response time goals > 1 minute

 - Note: other types of capping (such as LPAR capping) will be seen as CPU delays

WLM Delay States – Non-Processor, Non-Storage

- I/O delay
 - Work delayed during a DASD I/O and encountered PEND or IOSQ

- Swap delay
 - Address space going into central storage but delayed waiting to be swapped-into memory. Sampled once per address space.
 - Note: Physical swapping no longer supported on z/OS

- MPL
 - A swapped-out and ready address space was delayed and could not come back in due to lack of MPL

- QMPL
 - Work is waiting for a batch initiator or on an application environment queue and waiting for a scalable server address space

WLM Delay States – Storage for Goal Honored Spaces

□ Auxiliary Storage Paging Delay Samples

■ PRIV

- Address space was delayed due to a page-in from auxiliary storage for a private area page

■ COMM

- Address space was delayed due to a page-in from auxiliary storage for a common area page

■ XMEM

- Address space was delayed due to cross memory paging
- The address space delayed was doing a cross memory access, and the target address space took a page fault and was waiting for a page in.

■ VIO

- Address space was delayed due to a page-in from auxiliary storage for a VIO page

■ STD HSP

- Address space was delayed due to a page in from auxiliary storage for a scroll (standard) Hiperspace

■ ESO HSP

- Address space was delayed due to a read miss for an expanded storage only (ESO or cache) Hiperspace
 - The delay is an estimate of the time it would take to read data from a database if the buffer had been removed from the ESO Hiperspace to result in the read miss

WLM Delay States – Storage for Goal Ignored Spaces

- Delay sample experienced by server address spaces while serving transactions
 - SERV PRIV
 - Server address space was delayed due to a page-in from auxiliary storage for a private area page
 - SERV VIO
 - Address space was delayed due to a page-in from auxiliary storage for a VIO page
 - SERV SHSP
 - Address space was delayed due to a page in from auxiliary storage for a scroll (standard) Hiperspace
 - SERV MPL
 - A swapped-out and ready server address space was delayed because it could not come back in due to lack of MPL

Now let's talk about the shortfalls of velocity goals

Weakness in the Velocity Concept

- Velocity does not incorporate any measure of demand
 - Work assigned a velocity goal has no end time objective
 - Instead, velocity goals are an objective to help work regularly make progress

- Velocities only consider sampled states under the control of WLM
 - WLM can help increase / decrease velocities by increasing / decreasing CPU using / delay samples by changing CPU dispatching priorities
 - But WLM cannot do the same for unknown states, or states outside of WLM's control
 - So work could be making great progress, but WLM thinks otherwise

- Based on samples rather than measured results
 - So achieved velocities do not always reflect actual behavior of work
 - CPU and I/O usings / delays, paging delays are counted per dispatchable unit
 - Other types of samples, such as Unknown and Idle are once per address space
 - Enclaves do not incur storage delays. Instead the address spaces in which the enclaves run are attributed those

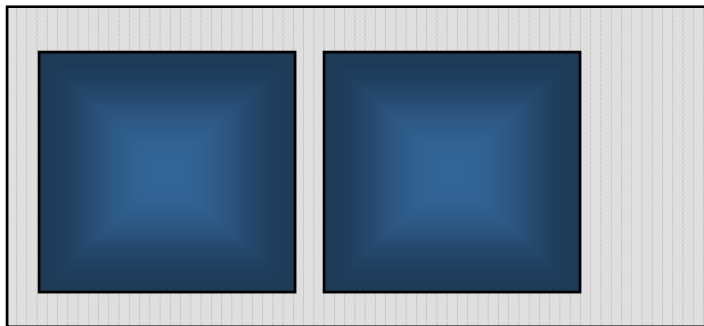
Factors Affecting Execution Velocity cont...

- Amount of work contributing to samples in velocity formula
 - Velocities are calculated on a service class period basis

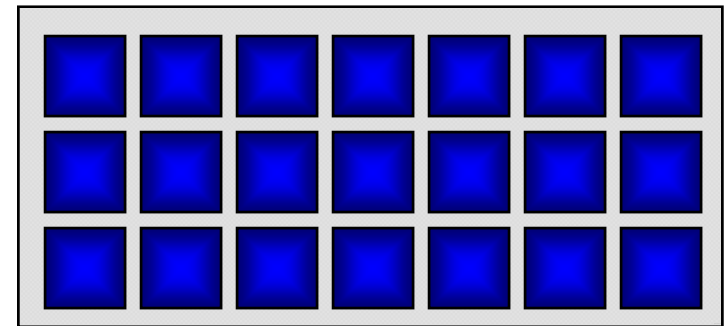
 - Number of address spaces in period will influence:
 - number of samples collected
 - fewer address spaces mean fewer samples
 - the calculated velocity
 - fewer samples forces WLM to go further back in history to understand current work

So Velocity based on $\frac{5}{5+10}$ same as Velocity based on $\frac{5000}{5000+10000} = 33\%$

Server Class = BIGBATCH, period 1



Server Class = IMSMPRS, period 1

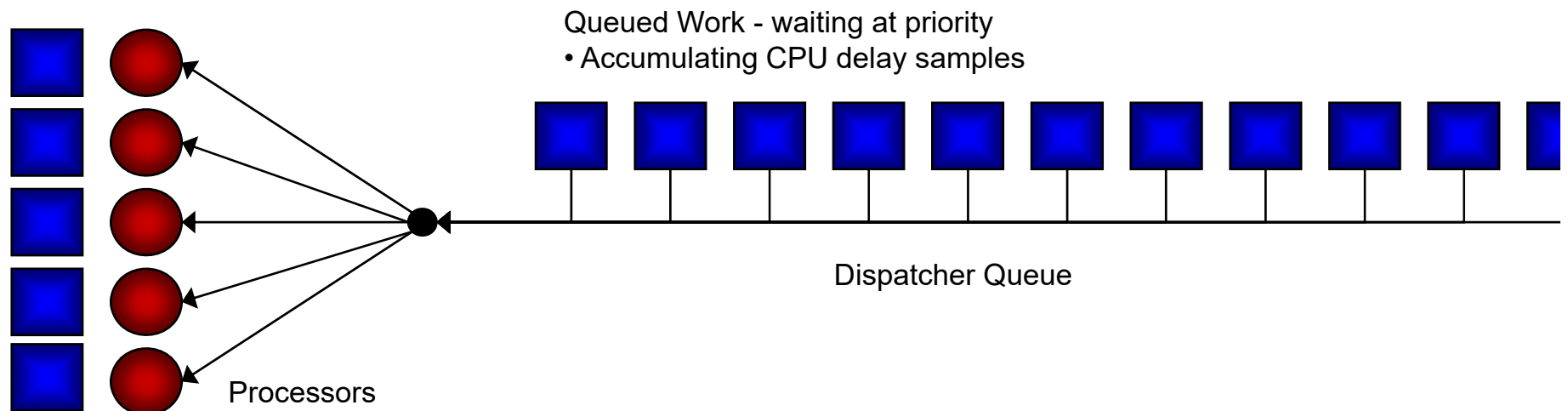


Factors Affecting Execution Velocity cont...

- Periods with more address spaces or enclaves (dispatchable units) tend to have more CPU queuing
 - Usually results in lower velocities
- Example:
 - Say we have a period with 100 active IMS MPRs that all want CPU
 - A Velocity goal of 60 may be impossible
 - Velocity goal of 10 more typical
 - Lesson: Just because this may be critical work does not mean a low velocity goal is a bad objective

Dispatched Work

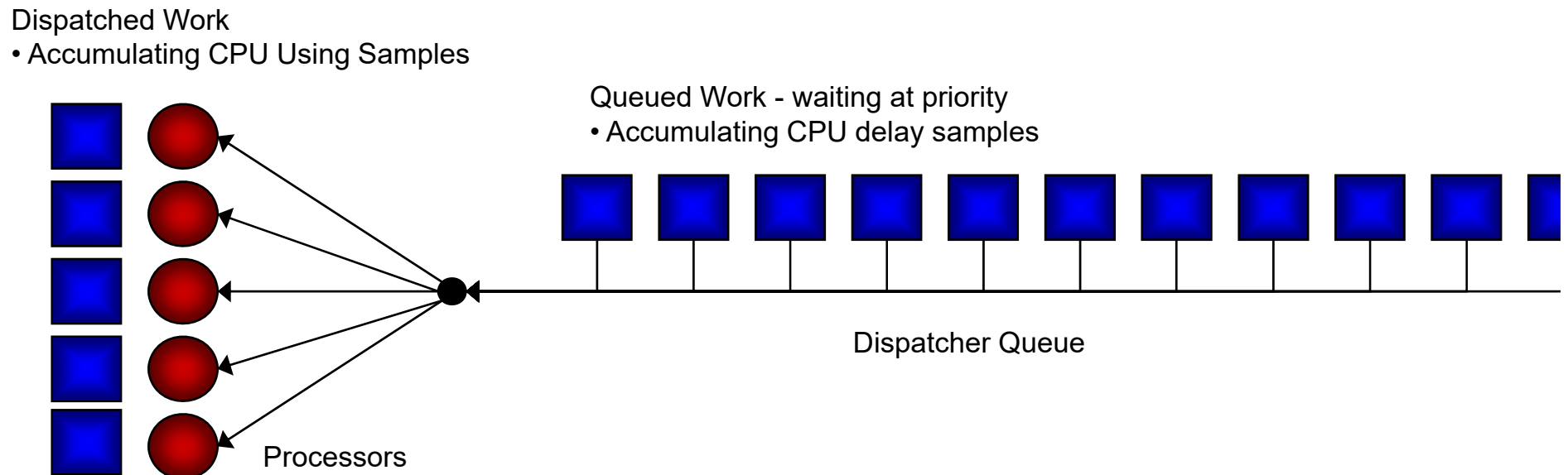
• Accumulating CPU Using Samples



Example:

How Processor Number and Speeds Can Affect Velocity

- Velocity is influenced by number of CPUs
 - If number of processor goes down, velocities tend to go down
 - If number of processor goes up, velocities tend to go up
 - Faster engines
 - May cause using samples of quick running work to be missed and result in sporadic velocities for intermittent or bursty work.



Factors Affecting Execution Velocity cont...

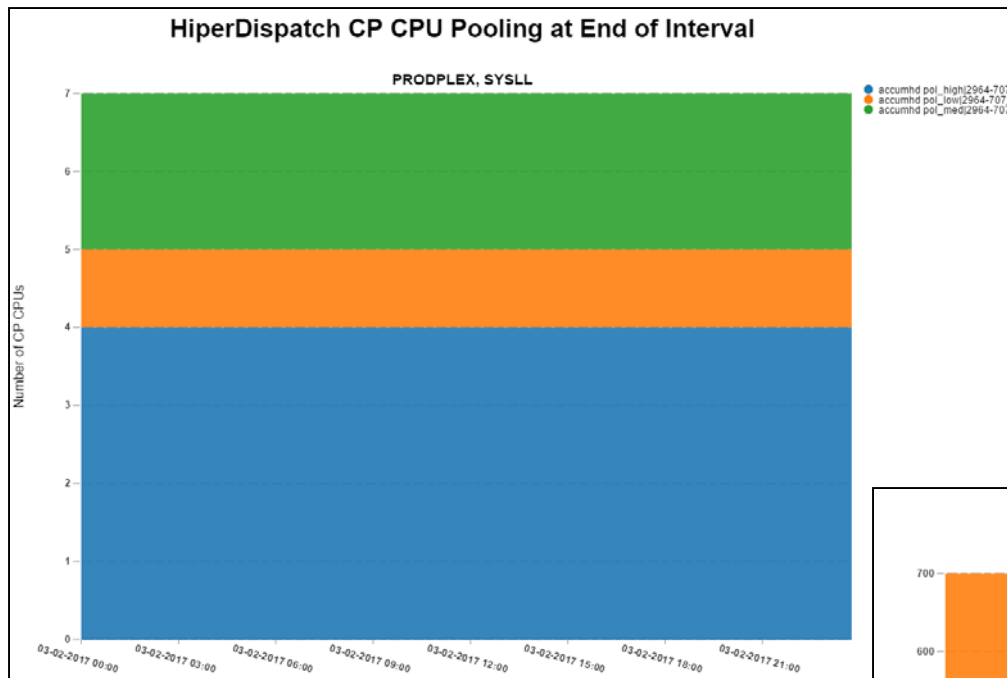
- Sampled states have varied meaning cont...
 - At similar CPU utilization, multitasking address spaces will have different CPU queue lengths
 - Higher probability that all processors are busy on 1-way

- Example

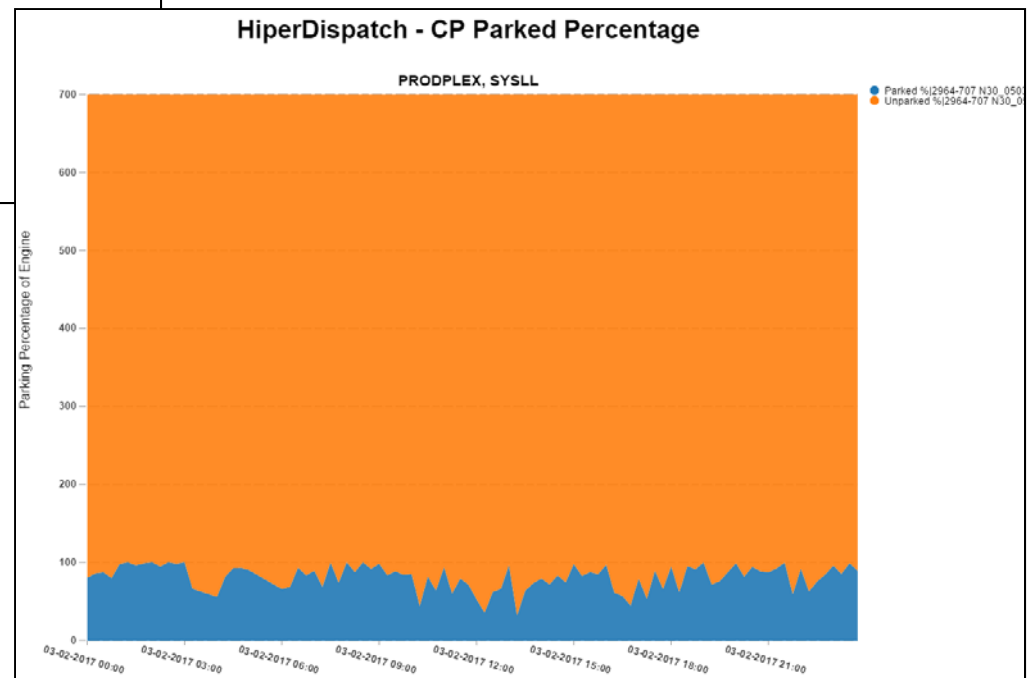
n-way affect on DB2 velocity		
Number of CPUs	1	5
DB2 Achieved Velocity	13.3	74.9
SYSSTC Achieved Velocity	17.4	74.7
CPU Busy %	89.75	88.60

* Table source from Arwe velocity paper mentioned in bibliography

HiperDispatch – Pooling and Parking

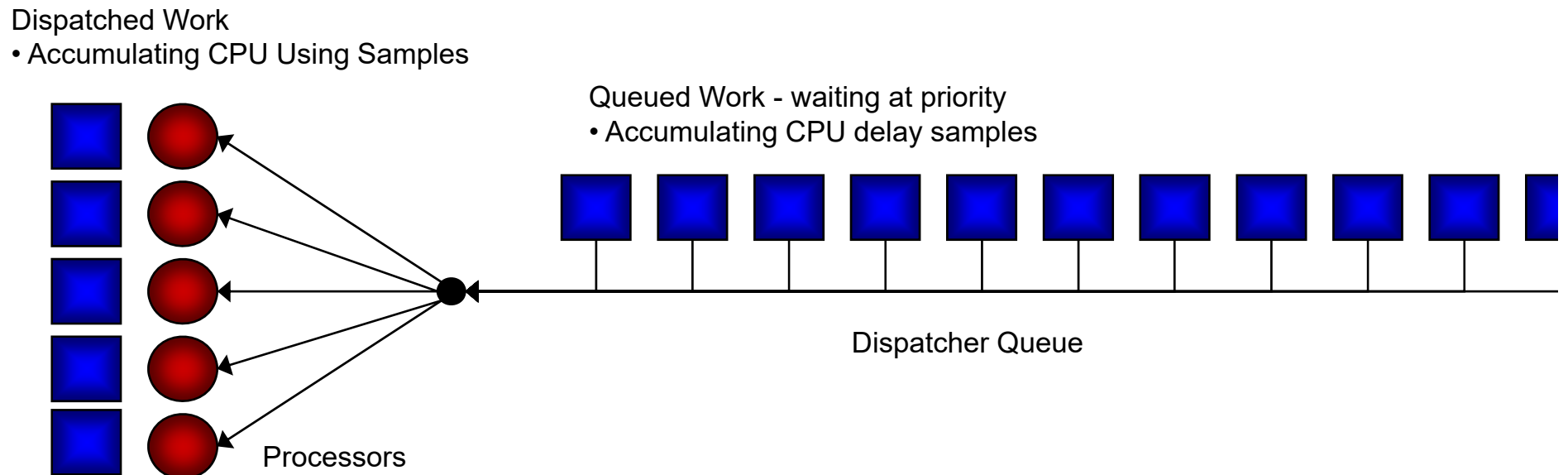


- Is this a 7 way LPAR?
 - Since configured with 7 CPs
- Is this a 5 way LPAR?
 - Since Low Pool has 2 CPs
- Is this a 6 way LPAR?
 - Since sometimes a low pool is unparked



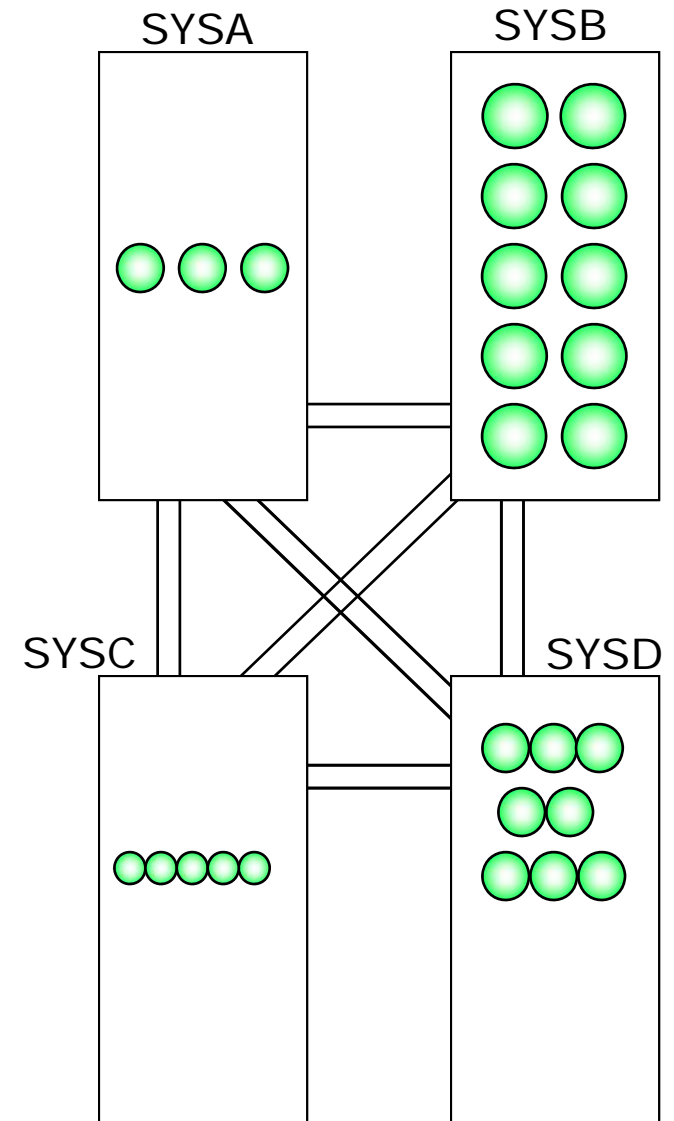
Other Velocity Factors

- Other causes of delay include
 - Reduced preemption
 - Results in all work incurring CPU delayed from time to time
 - Fair share dispatching
 - Eliminates the need for work to be secluded to a period for micromanagement of access to the CPU



Planning an Asymmetrical Sysplex

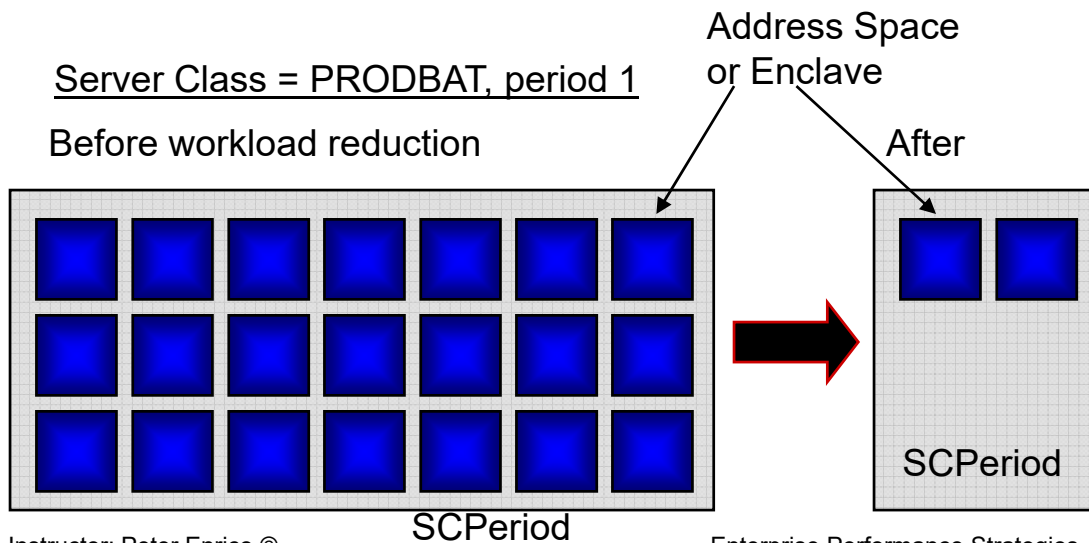
- **Goals are Sysplex wide**
 - How do you assign a velocity goal in an asymmetrical environment?
 - Example: CICSREG class assigned velocity 60 goal
 - Too aggressive for SYSA
 - Too easy for SYSB
 - Just right for SYSD
 - If really a problem, then use system classification
 - If SYSA then classify to velocity 20 goal period
 - If SYSB then classify to velocity 60 goal period
 - If SYSD then classify to velocity 40 goal period
 - The system name qualifier is supported for address spaces whose execution system is known at classification time.
 - Note that JES is not eligible for this qualifier, as the system on which classification occurs may not be the system on which the job is run.



Velocity May Be Affected By Workload Reduction

- Amount of work contributing to samples affects calculated velocity
 - Velocities are calculated on a service class (goal) period basis
 - Number of address spaces or enclaves in period will influence:
 - Number of using and delay samples collected
 - Fewer address spaces or enclaves mean fewer contributing samples
 - Calculated velocities (will be more sensitive and erratic)

Velocity based on $\frac{5000}{5000+10000}$ same as Velocity based on $\frac{5}{5+10} = 33\%$



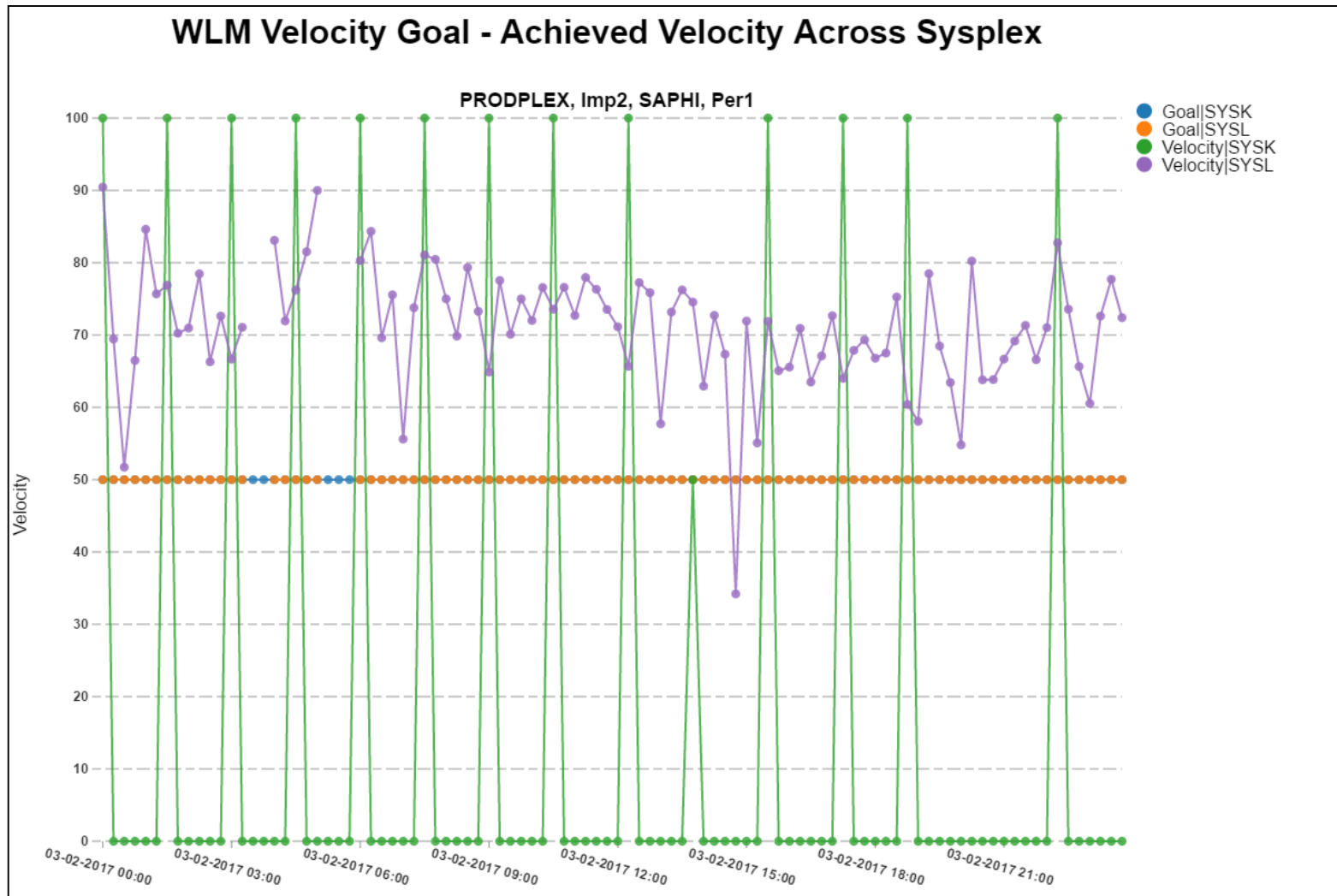
Example of how sensitive velocity calculation is when based on few samples

$$\frac{4}{4+11} = \text{Velocity } 26\%$$

$$\frac{5}{5+10} = \text{Velocity } 33\%$$

$$\frac{6}{6+9} = \text{Velocity } 40\%$$

WLM Velocity Goal - Achieved Velocity Across Sysplex



Turn on I/O Priority Management

- Most installations today do not have storage delays
 - This results in CPU using and delay states being the single most contributors to achieved velocities
 - By enabling I/O priority management the I/O using and delay states tend to 'smooth' out achieved velocities

$$\left(\frac{\text{CPUUsing}}{\text{CPUUsing} + \text{CPUDelay} + \text{StorageDelays}} \right) \text{ versus } \left(\frac{\text{CPUUsing} + \text{I/O Using}}{\text{CPUUsing} + \text{CPUDelay} + \text{I/O Using} + \text{I/O Delay} + \text{StorageDelays}} \right)$$

$$\left(\frac{257}{257 + 126 + 14} \right) = \text{vel } 65\% \qquad \left(\frac{257 + 289}{257 + 126 + 289 + 455 + 14} \right) = \text{vel } 48\%$$

Performing a Velocity Goal Analysis and Recommendations

Evaluating Velocity Goal Periods

- Some key objectives of this evaluation include the following:
 - Ensure the work assigned a velocity goal is velocity goal oriented
 - Online and database regions such as CICS, IMS, and DB2
 - Long running or intermittent batch
 - Started tasks not in SYSSTC and not discretionary
 - Any long running work
 - Evaluate the velocity goal value
 - Where do the PIs hover? Between 0.5 and 1.2 is usually OK
 - Determine if the goal is too easy or too hard
 - Are the PIs either always well above 4 or 5?
 - Remember, too difficult of a goal makes the work less likely to be helped
 - Are the PIs always below 0.81 or lower?
 - Remember, too easy of a goal makes the work vulnerable to lower CPU DPs
 - When tuning velocities goals (and goals in general), pay more attention to higher importance periods than lower

Don't Use Velocity As Substitute for Importance

- How many WLM Service Definitions have you seen with something like this?
 - Questions: Which is the more aggressive goal?

Service Class	Period	Imp	Goals
CICSHIGH	1	1	Velocity 50
CICSMED	1	1	Velocity 20

When NOT to Use a Velocity Goal

- Combine work into small number of velocity periods
 - Example: low=10, medium=30, high=60
 - Combining like work is good

- Relative velocities do not equate to relative dispatching priorities
 - A high velocity goal period could be given a low CPU or I/O dispatch priority
 - Less worry since MVS dispatcher now has a 'fair share' algorithm

- Be careful of very high velocities
 - Means you want to give a CPU to the work
 - Remember, high velocities are more sensitive to processor differences
 - Make sure the high velocity can be obtained on a smaller processor

- Don't micromanage velocities
 - Little to no difference between velocity 40 and velocity 43

Velocity Goal Questions

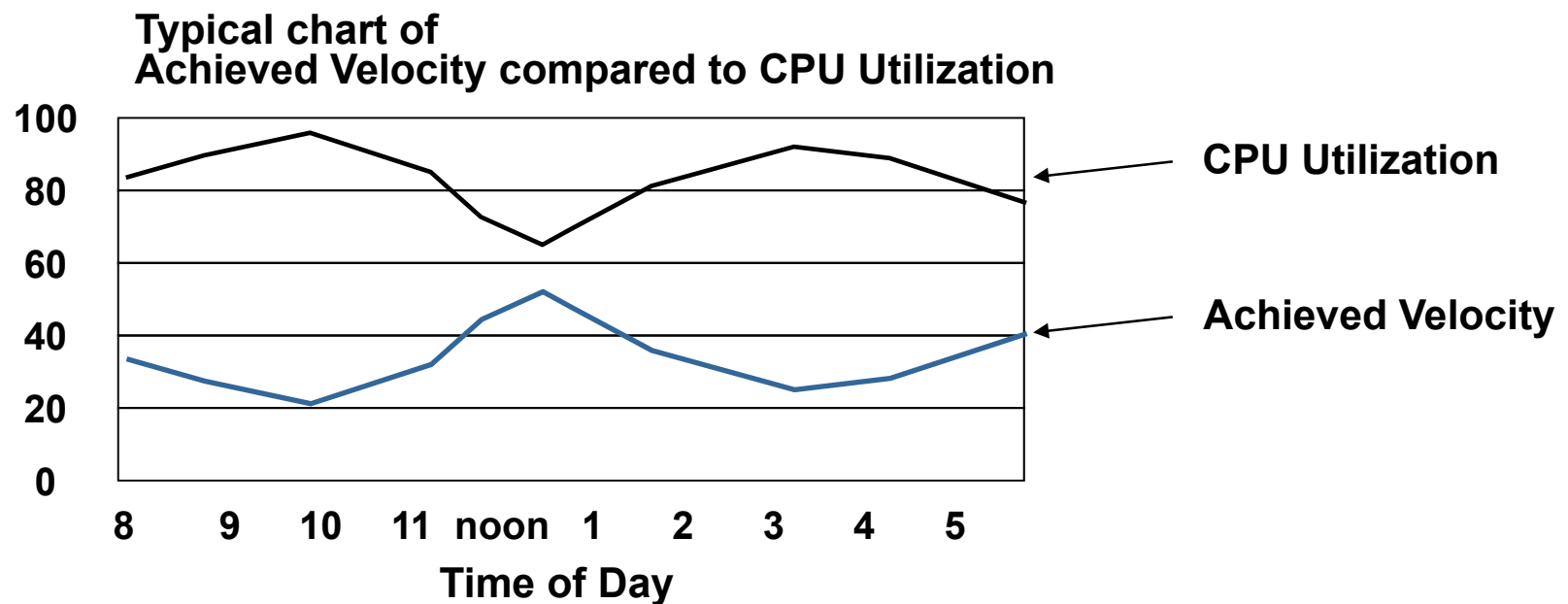
- What are the velocities regularly being achieved?
 - Are the achieved velocities for a period regular or erratic?
 - Are the achieved velocities for a period regularly very high or very low?
 - Does there appear to be a correlation between achieved velocities and the processor utilization for the LPAR and machine?

- Are there any assigned velocity goals greater than 90? If so, can these goals really be explained and justified?

- How much work is regularly running in the velocity goal period?
 - Relative to the number of active address spaces or enclaves in the period.
 - Relative to the amount of resource consumption of the work in the period.

Choosing Velocity Goals

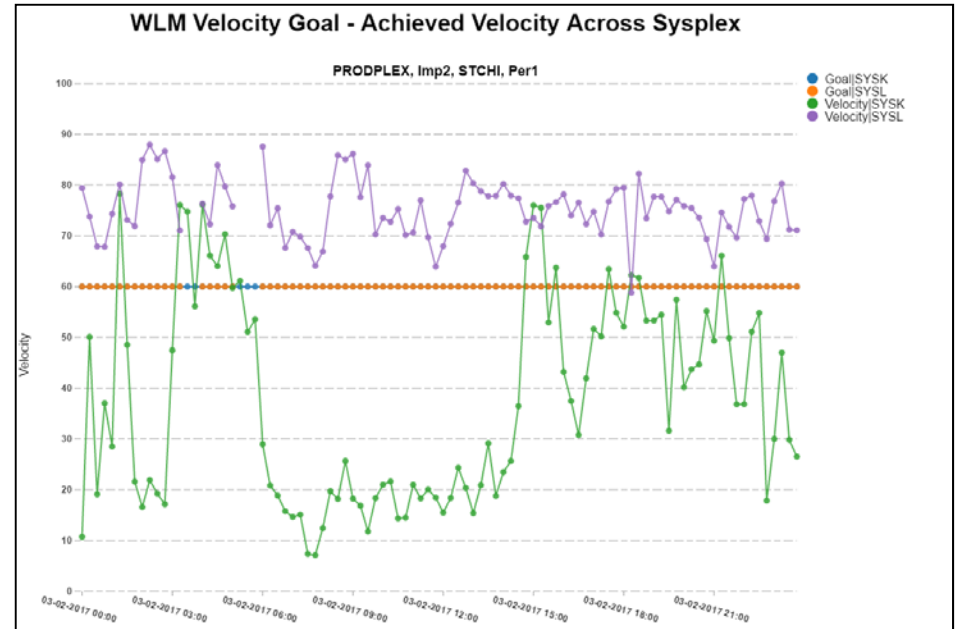
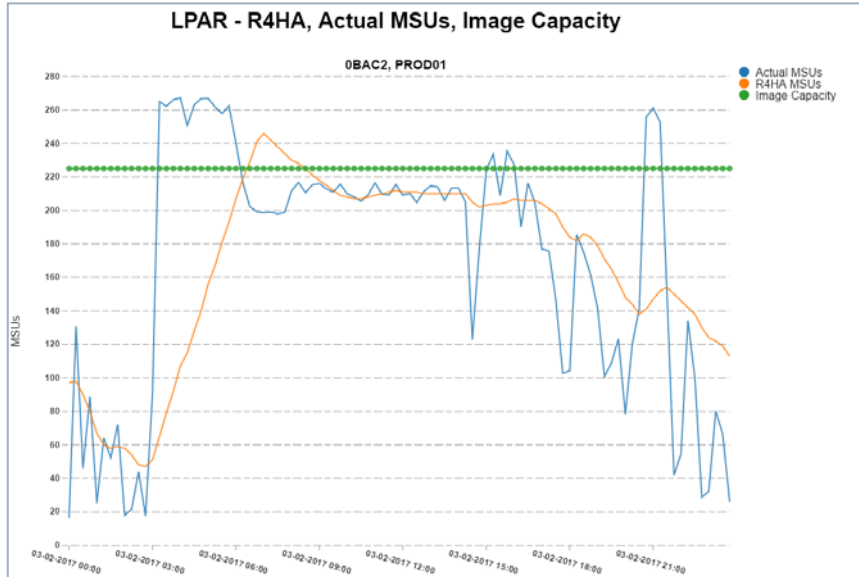
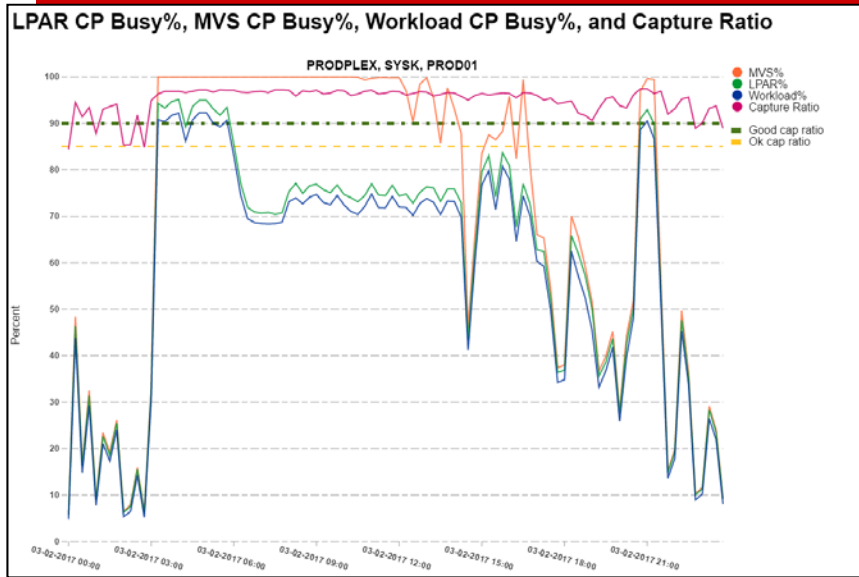
- Use data from peak periods to set and tune goals
 - When using historical data to determine goals for work
 - Gives the highest response time goal with acceptable service
 - Gives the lowest velocity goal with acceptable service



Tune Velocities During Peak Periods

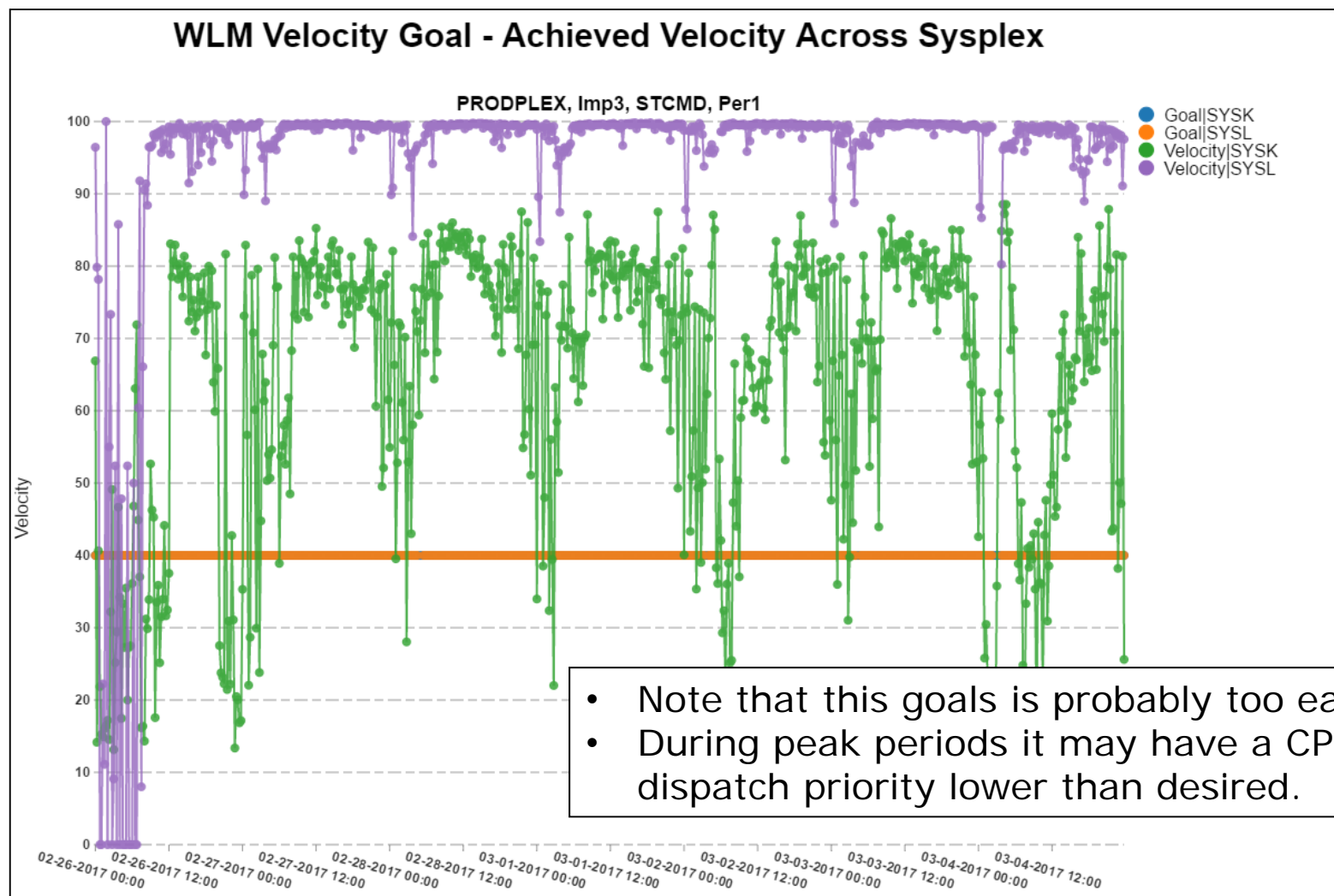
Peak is defined as:

- When processor is busy
- or When capping is occurring
- or When work needs to run

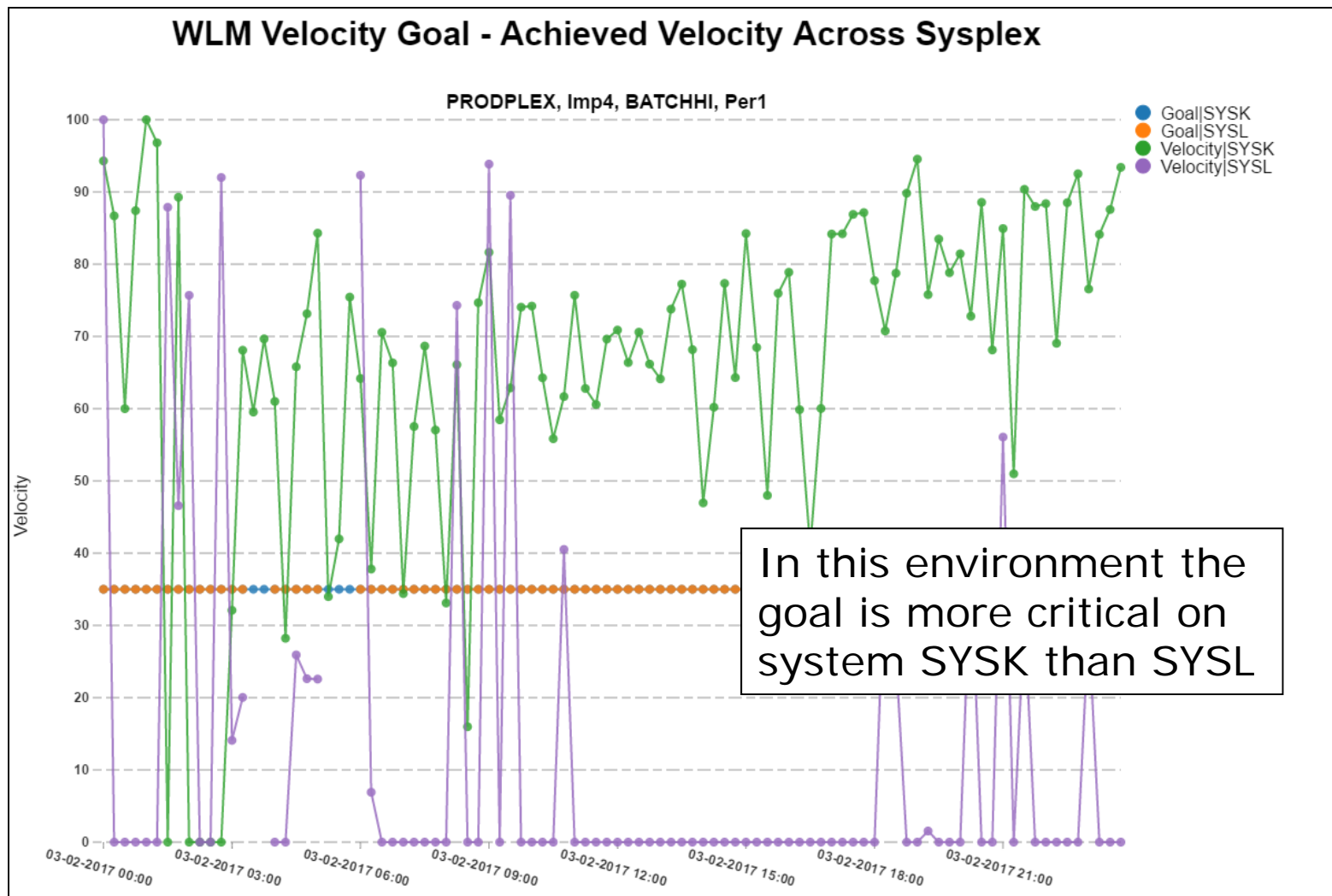


In this case, goal may be too difficult during peak CPU busy which is during capping

Measure Velocities Over Time



Measure Velocities on each system



Avoid monitoring percentage of samples

- Most monitors report samples are percentages of total samples

W O R K L O A D A C T I V I T Y

PAGE 26

z/OS V2R2 SYSPLEX SEBPLEXP DATE 07/01/2016 INTERVAL 15.00.171 MODE = GOAL
 RPT VERSION V2R2 RMF TIME 11.45.00

POLICY ACTIVATION DATE/TIME 07/01/2016 09.00.01

----- SERVICE CLASS PERIODS

REPORT BY: POLICY=PRIME WORKLOAD=ONLINE SERVICE CLASS=ONL_HI RESOURCE GROUP=*NONE PERIOD=1 IMPORTANCE=1
 CRITICAL =CPU I/O PRIORITY GROUP=HIGH

-TRANSACTIONS-	TRANS-TIME	HHH.MM.SS.TTT	--DASD	I/O--	---SERVICE---	SERVICE TIME	---APPL %---	--PROMOTED--	----STORAGE----						
AVG	16.47	ACTUAL	0	SSCHRT	26.4	IOC	12915	CPU	5.161	CP	0.64	BLK	0.000	AVG	36844.95
MPL	16.47	EXECUTION	0	RESP	0.7	CPU	271627	SRB	1.060	AAPCP	0.00	ENQ	0.000	TOTAL	606919.2
ENDED	0	QUEUED	0	CONN	0.2	MSO	0	RCT	0.071	IIPCP	0.00	CRM	0.000	SHARED	65.87
END/S	0.00	R/S AFFIN	0	DISC	0.5	SRB	55804	IIT	0.114			LCK	0.087		
#SWAPS	716	INELIGIBLE	0	Q+PEND	0.0	TOT	340346	HST	0.015	AAP	N/A	SUP	0.000	-PAGE-IN RATES-	
EXCTD	0	CONVERSION	0	IOSQ	0.0	/SEC	378	AAP	N/A	IIP	0.05			SINGLE	0.0
AVG ENC	0.00	STD DEV	0					IIP	0.441					BLOCK	0.0
REM ENC	0.00					ABSRPTN	23							SHARED	0.0
MS ENC	0.00					TRX SERV	23							HSP	0.0

TRANSACTION APPL% : TOTAL : CP 0.64 AAP/IIP ON CP 0.00 AAP/IIP 0.05
 MOBILE : CP 0.00 AAP/IIP ON CP 0.00 AAP/IIP 0.00

GOAL: EXECUTION VELOCITY 55.0% VELOCITY MIGRATION: I/O MGMT 81.2% INIT MGMT 81.2%

SYSTEM	RESPONSE TIME	EX	PERF	AVG	--EXEC USING%--				EXEC DELAYS %				-USING%-		--- DELAY % ---		%			
	VEL%	INDX	ADRSP	CPU	AAP	IIP	I/O	TOT					CRY	CNT	UNK	IDL	CRY	CNT	QUI	
PROD	--N/A--	81.2	0.7	22.0	0.0	N/A	0.0	0.1	0.0					0.0	0.0	18	82	0.0	0.0	0.0

What is a better way to analyze samples?

- Looking at percentage of samples is helpful, but absolute values are more helpful

Sample Counts for BIGBATCH

Number of Samples	100	25	255	30	0	0	0	100	0	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					

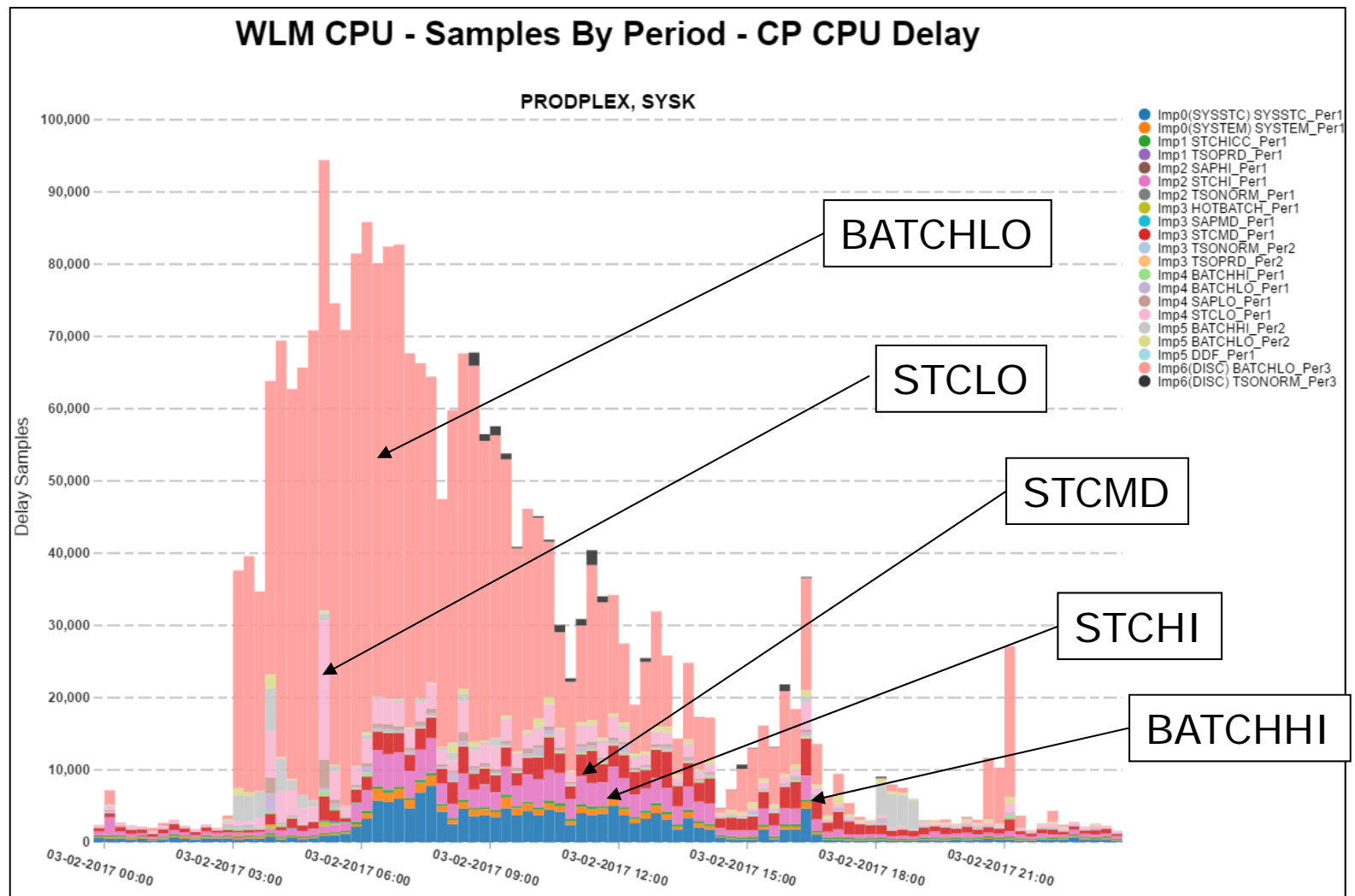
Sample Counts for OMVSPROD

Number of Samples	10	5	2	5	0	0	0	0	0	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					

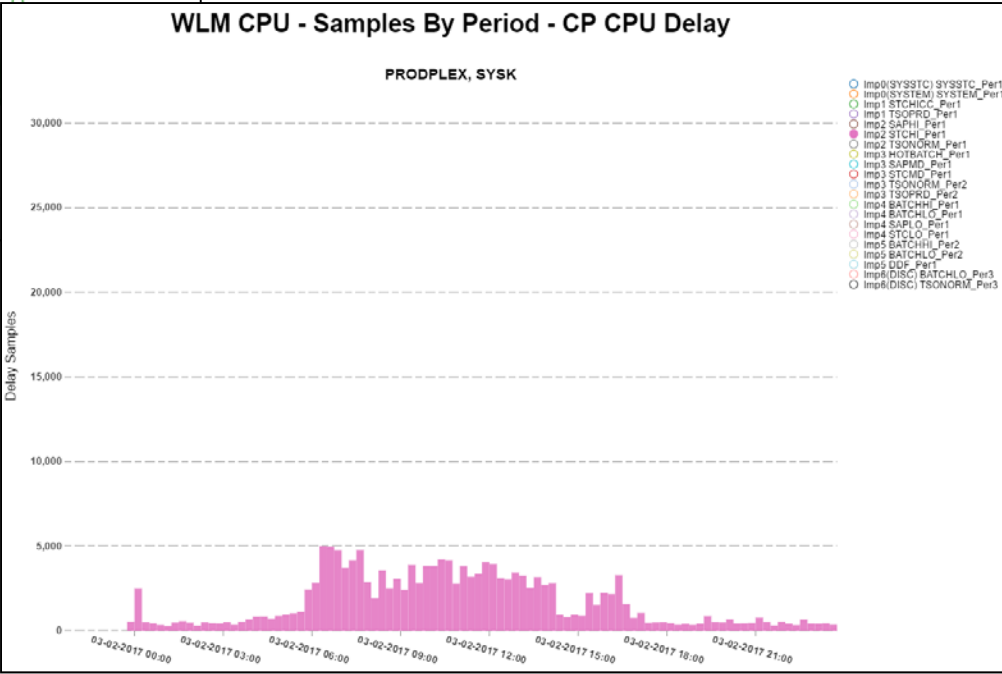
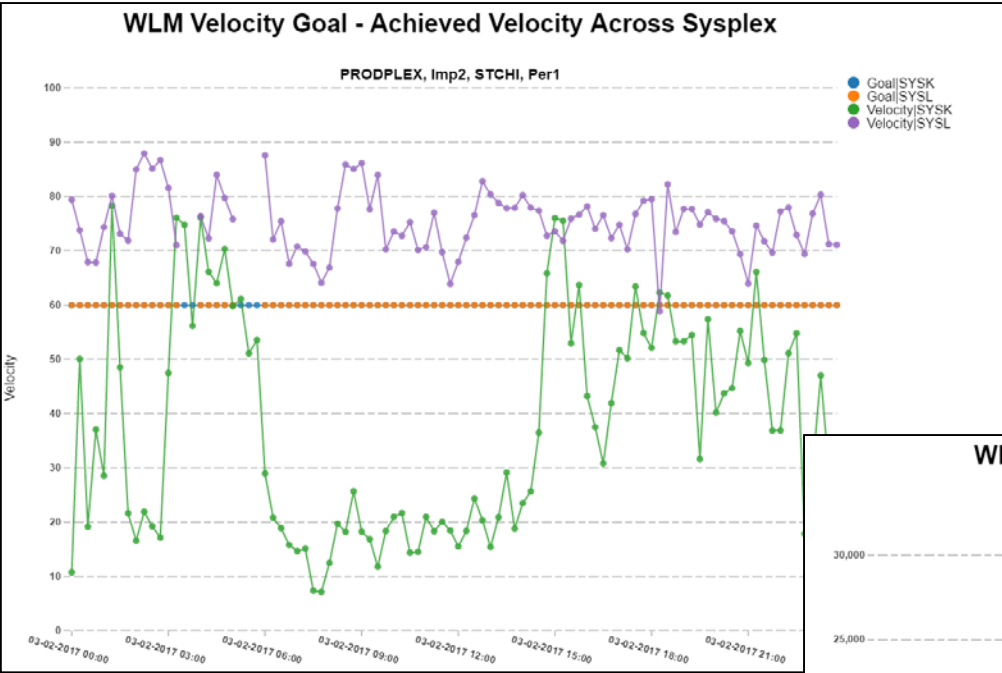
Sample Counts for IMSMPRS

Number of Samples	10,000	2,500	25,500	13,000	0	0	0	0	0	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					

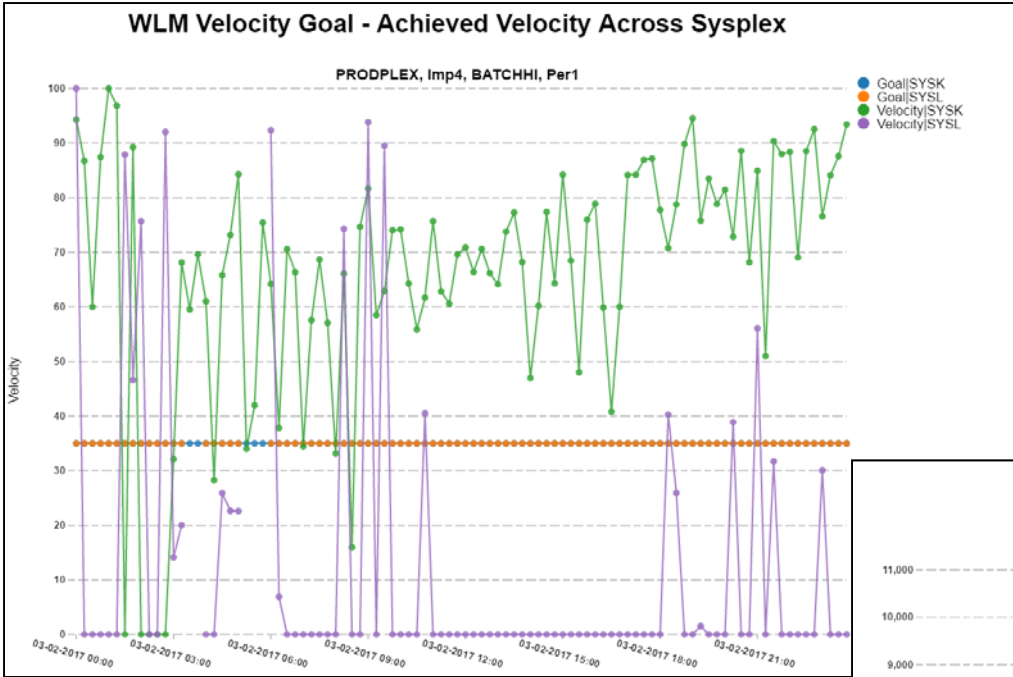
Evaluate velocities relative to number of samples



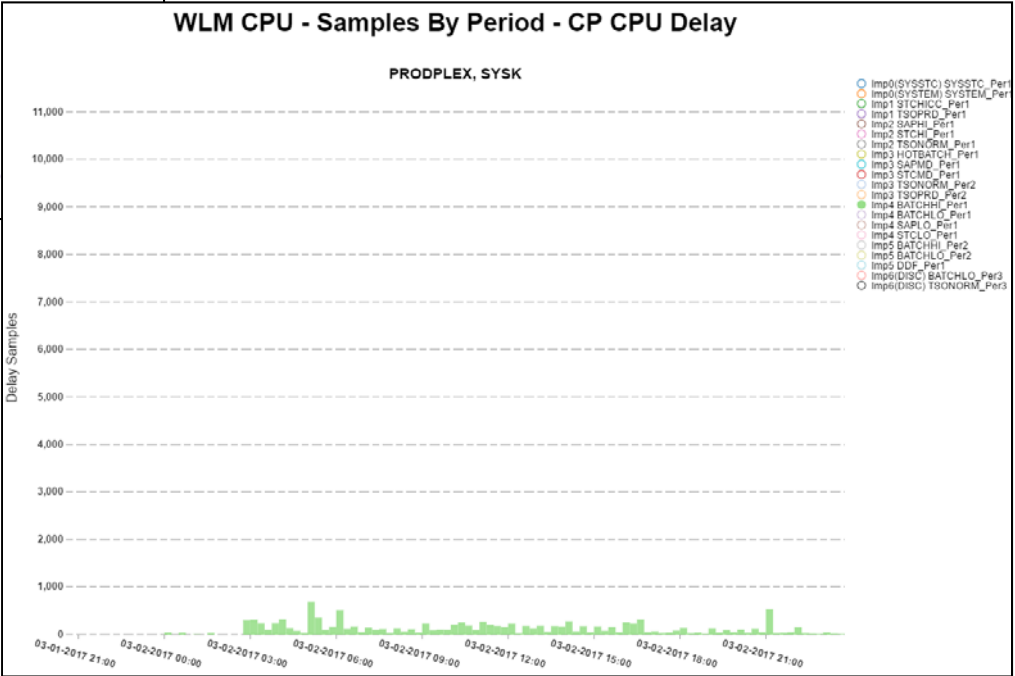
Notice Velocity independent of number of samples



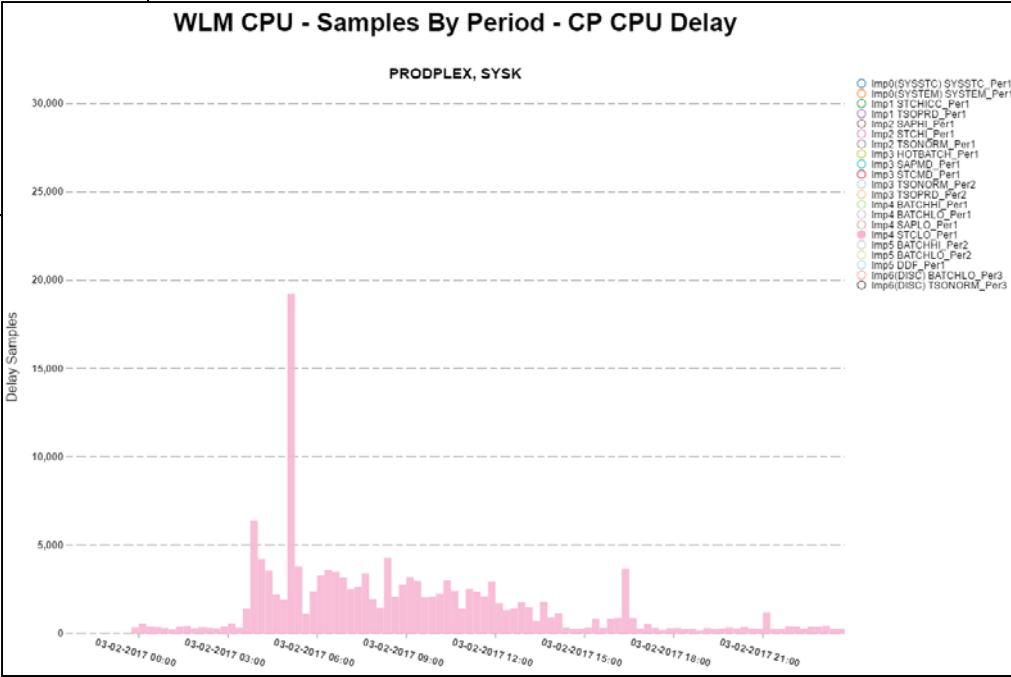
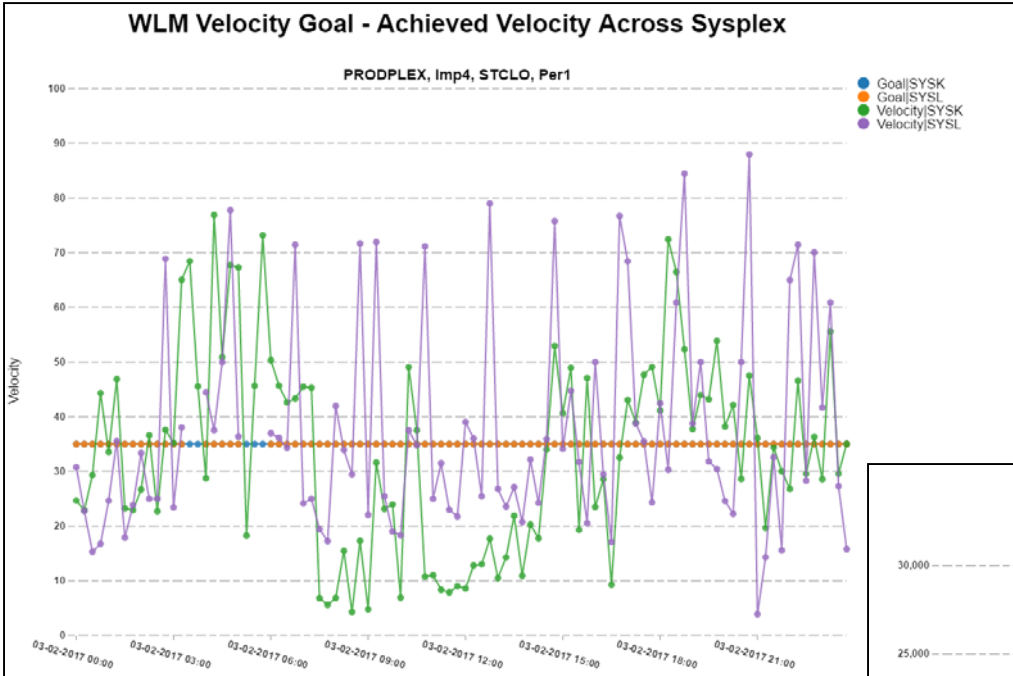
Notice Velocity independent of number of samples



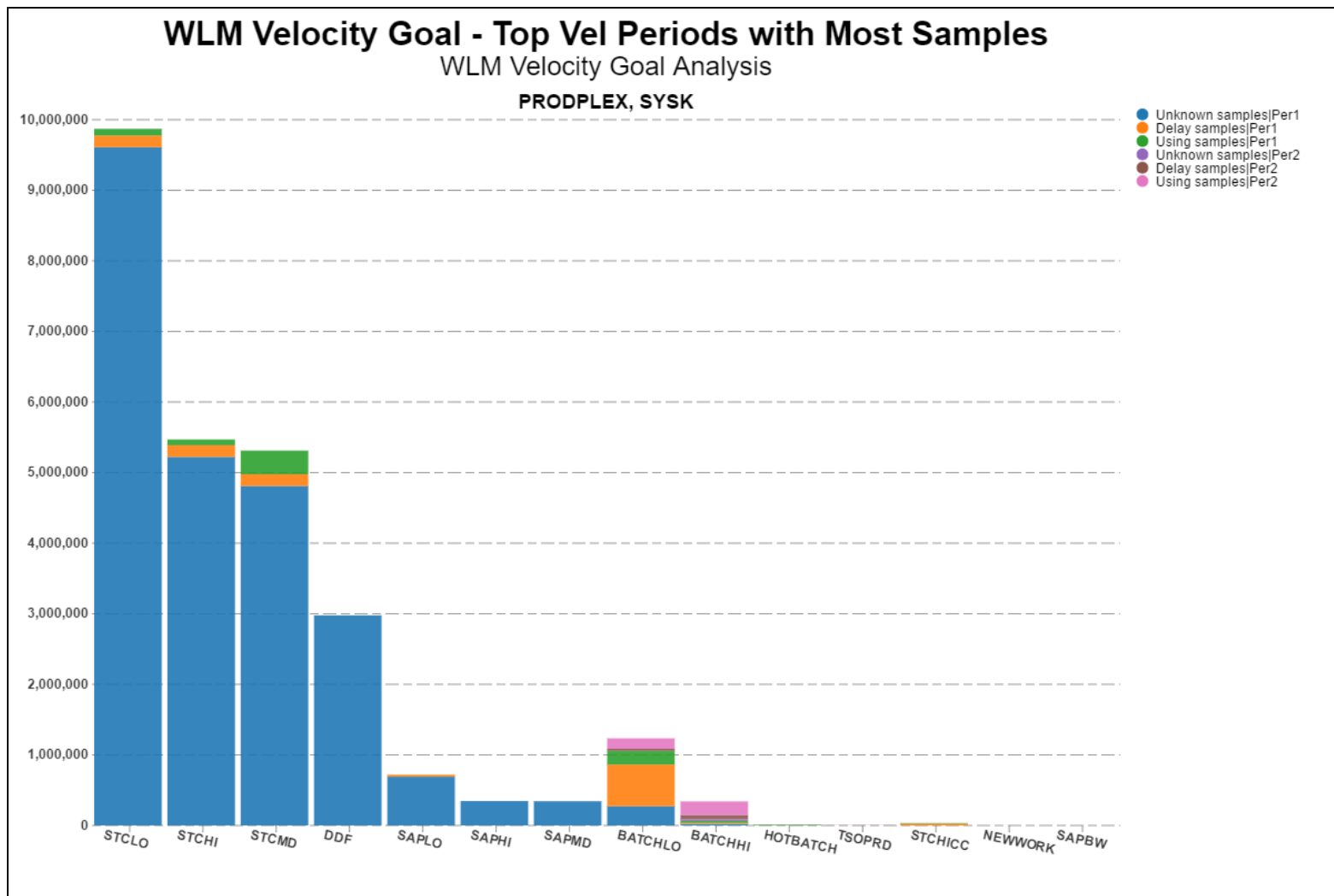
BATCHHI has very few samples



Notice Velocity independent of number of samples



Measure Delay, Using, and Unknown samples for period of study



Velocity Summary

- Of all the WLM goal types, execution velocity is the most difficult one to assign and maintain
 - Seems like a simple concept until you need to use it

- Velocity will be most appropriate for many service classes
 - So you need to use it

- Make sure you examine a sufficient set of data when assigning velocities
 - Make sure you look at data from multiple systems and multiple times of the day