Evaluating Latent Demand in the Mainframe Environment

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®

Contact, Copyright, and Trademark Notices

Questions?

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

Copyright Notice:

Trademarks:

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

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

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

Other trademarks and registered trademarks may exist in this presentation

Abstract

- Evaluating Latent Demand in the Mainframe Environment

In the world of computers, you can think of latent demand as the demand for resources that cannot be met due to constraints. Workloads want to use the resources and have demand for those resources, but the environment does not have the ability to satisfy the demand. During this presentation, Peter Enrico will discuss the measurement and evaluation of latent demand in the mainframe environment. So, if you have a system that is being capped, weight enforced, or if your processor is just out of capacity, you will want to attend this session.

EPS: We do z/OS performance…

- **Pivotor** – z/OS performance reporting and analysis software and services
  - Not just SMF reporting, but analysis-based reporting based on expertise
  - www.pivotor.com

- **Education and instruction**
  - We teach our z/OS performance workshops all over the world
  - Want a workshop in your area? Just contact me.

- **z/OS Performance War Rooms**
  - Intense, concentrated, and highly productive on-site performance group discussions, analysis and education
  - Amazing feedback from dozens of past clients

- **MSU Reduction Exercises**
  - The goal is to reduce the MSU consumption of your applications and environment

- **Information**
  - We present around the world and participate in online forums
  - https://www.pivotor.com/content.html
    https://www.pivotor.com/webinar.html

z/OS Performance workshops available

During these workshops you will be analyzing your own data!

- WLM Performance and Re-evaluating Goals
  - May 12 – 16, 2025 (4 days)

- Parallel Sysplex and z/OS Performance Tuning
  - July 15-16, 2025 (2 days)

- Essential z/OS Performance Tuning
  - September 22-26, 2025 (4 days)

- Also… please make sure you are signed up for our free monthly z/OS educational webinars! (email contact@epstrategies.com)

Like what you see?

- Free z/OS Performance Educational webinars!
  - The titles for our Spring / Summer 2025 webinars are as follows:
    - ✓ *Overseeing z/OS Performance Management With Your Outsourcer*
    - ✓ *Back to basics - Processor Consumption Analysis*
    - ✓ *Pivotor Pointers*
    - *Back to Basics - Evaluating Latent Demand*
    - *Understanding SMF 98 Locking Measurements (with Bob Rogers!)*
    - *Standard Measurements when Monitoring Transactions*
    - *Processor Comparison Discussion*
    - *z/OS Performance Management in an AI World*
    - *Understanding z/Architecture Processor Topologies*
    - *SMF 99 WLM Decision Making Traces*
    - *Understanding SMF 98 Address Space Consumption Measurements*
    - *WLM and CPU Critical Control*

- If you want a free cursory review of your environment, let us know!
  - We're always happy to process a day's worth of data and show you the results
  - See also: http://pivotor.com/cursoryReview.html

Latent Demand

- Latent Demand = work that is waiting to get done, but can't because something in the configuration is preventing it from being dispatched
- We can usually easily tell that there is latent demand
- Understanding how much additional capacity that would require is difficult
  - We usually don't know how much CPU the delayed work units really want
  - In some cases, we can guesstimate (e.g. for batch jobs)
  - It is complicated though:
    - Workloads use more resources than just CPU (so may not be able to fully consume available CPU)
    - Scheduling may prevent workload from shifting
    - Changing workload velocities will cause WLM to make different decisions

The people waiting on the freeway on ramp are latent demand.

Induced Demand

- Induced Demand = work that hasn't come into the system but would if more capacity were available (usually due to user behavior change)
- For example:
  - Ad Hoc Database queries run faster so users create more complicated queries to answer new questions
  - Compiles run faster so programmers are less careful about desk checking and submit more compiles
  - RMF III response time improves so performance analyst pokes through more intervals looking at more data
- Induced demand often may be "good" as it usually means more useful work getting done, but it's even more difficult to predict than latent demand

The people staying at home because the freeway is too crowded are potential induced demand.

When does latent demand build up?

- Latent demand is created when there is some combination of resources (such as CPU) and workload demand:

- Resources:
  - Given a particular set of workload demands, there are not enough resources (such as CPU) to handle the workload's demand

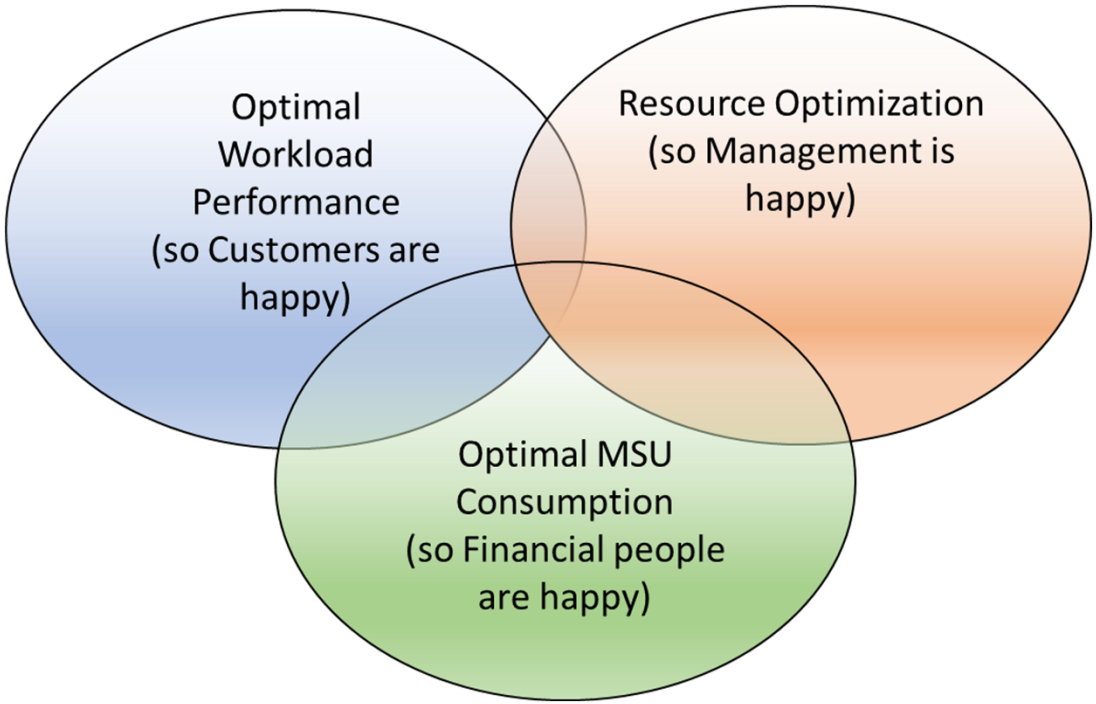
- Workloads:
  - Given a particular amount of capacity, the workloads are placing too much demand on the resources

Why would available resources or workload demand change?

- Resources can become constrained for a wide variety of reasons:
  - Machine level constraints
  - LPAR level constraints
  - Capping Constraints
    - Defined capacity limits, Group capacity limits, Absolute Caps, Group Absolute Cap, etc.
  - Weight enforcement
  - Crossover
  - More…
- Workloads can place additional demand on the resources:
  - Peak periods
    - Nighttime, daytime, seasonal, market open
  - Business grows, new workloads
  - Failover
  - Unexpected ad-hoc activity
  - More…

The Performance Balancing Act

- Performance on z/OS is about finding an optimal balance among 3 areas
  - And dealing with latent demand is no different than any other performance issue



How to alleviate Latent Demand?

- Addressing latent demand is not different than any performance issue:
  - Get more resources
  - Do less work
  - Tuning
  - Take advantage of controls such as capping

- Regardless… the very first exercise is to understand your latent demand!
  - Is there a lot or a little?
  - What are the patterns of activity?
  - What is causing the latent demand?
  - What workloads are suffering?

Did Capping Actually Limit the LPAR?

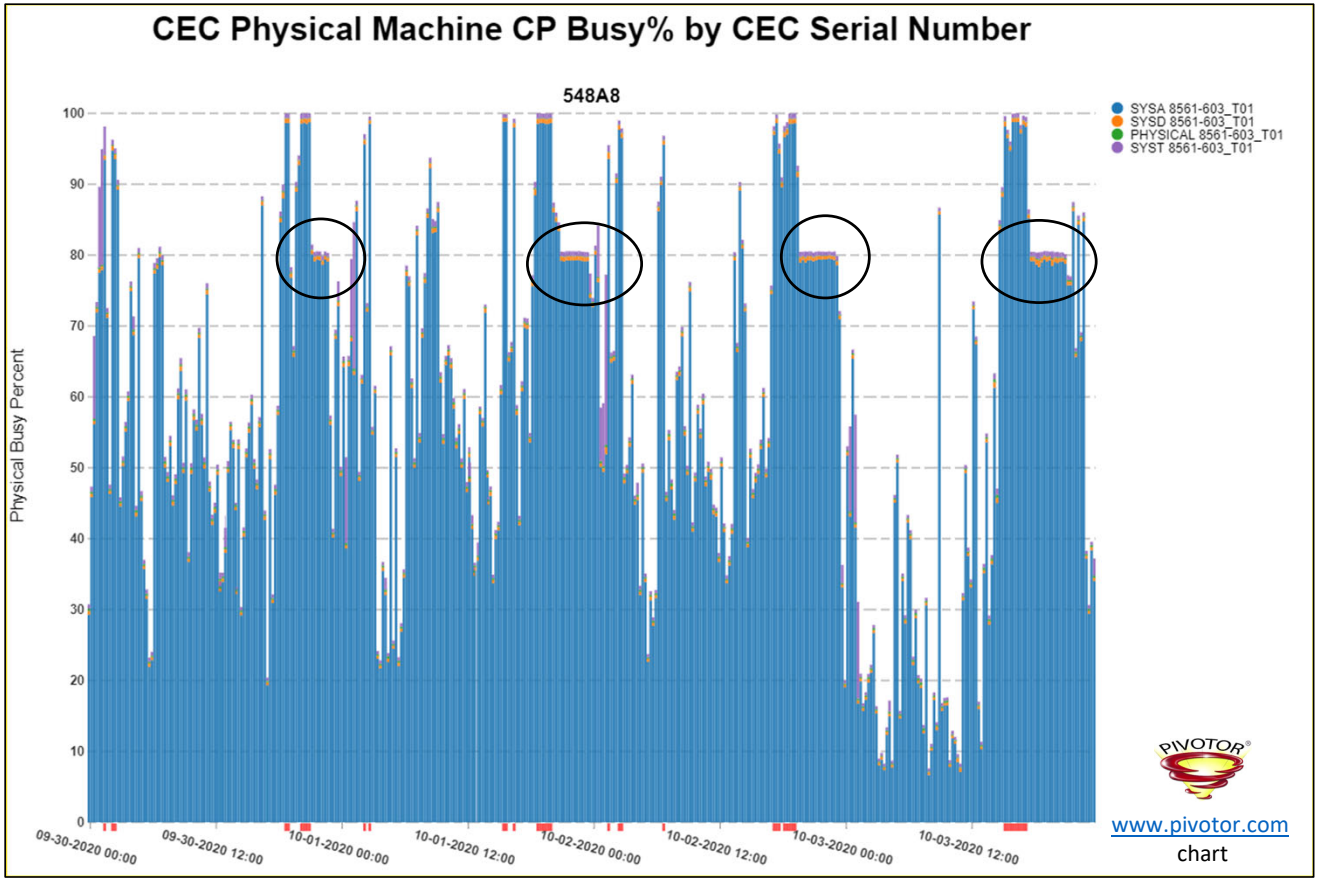
- If demand for CPU is less than the cap, the cap isn't really limiting the LPAR
- RMF records:
  - Samples when the LPAR is considered capped
  - Samples where the cap actually limited the usage of processor resources
- "Considered capped" will usually work out to 100%, except for the first and last intervals when the cap is coming on or off
- "Actually limited" may vary throughout the capping period
  - Lower "actually limited" vs. "considered capped" means capping is causing less latent demand – i.e. capping is causing less delays for work
    - Likely because there's not demand for the full cap amount

Finding likely periods of time when latent demand may be occurring

And what is suffering…

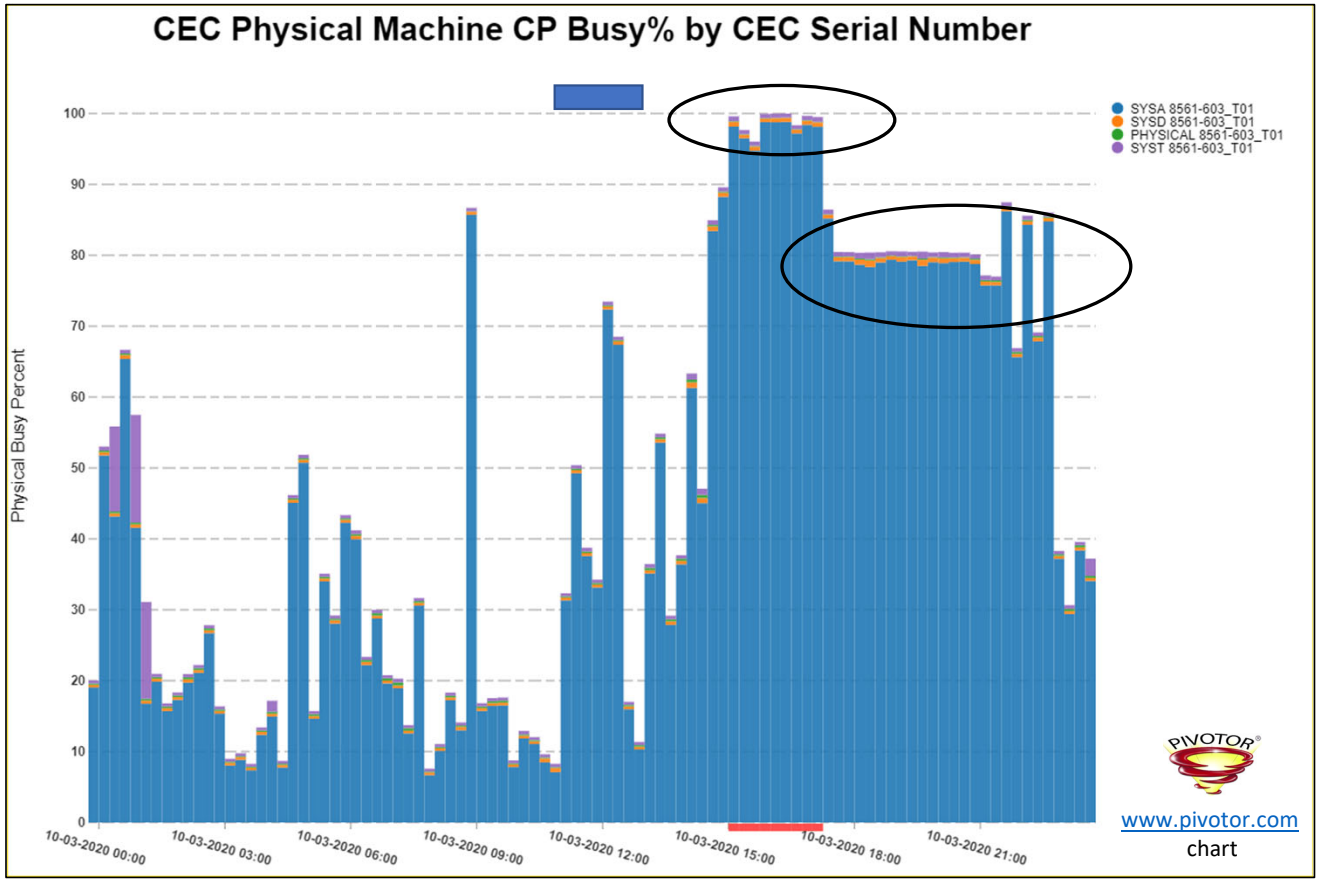
Physical Processor CPU Utilization

- This chart shows a full week of CEC physical processor busy
- Typically, flat areas below 100% physical processor busy hint towards capping
- Flat areas at 100% do not indicate capping, but do indicate resource limits
  - Which have many of the same impacts as capping



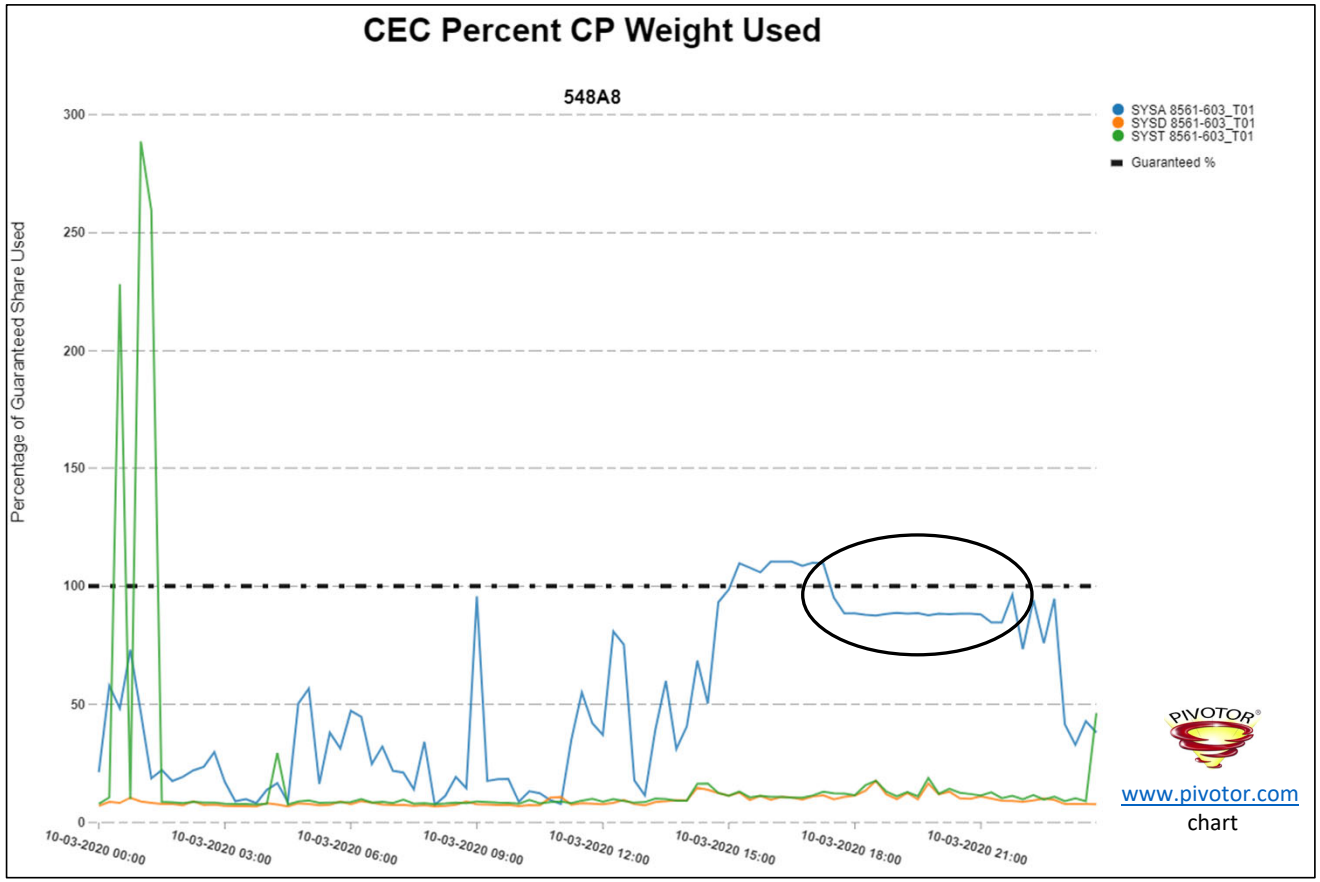
Physical Process CPU Utilization

- This chart just shows a single day
- Note the CPU utilization patterns on this chart
  - The left-hand side is a typical pattern of CPU utilization
  - The right-hand side of the chart shows a plateau
  - Typical indication that there must be some sort capping



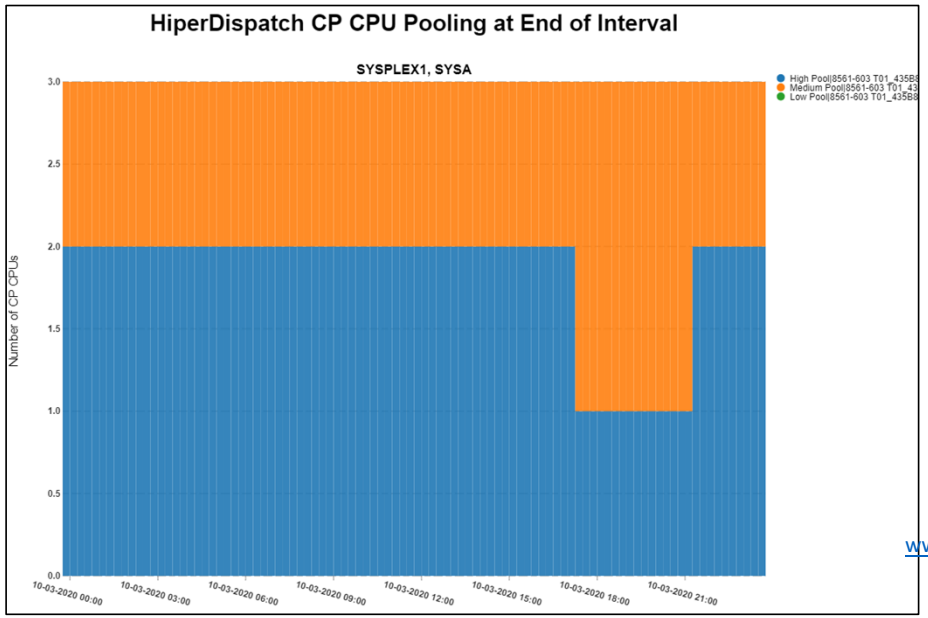
Percentage of Weight Consumed

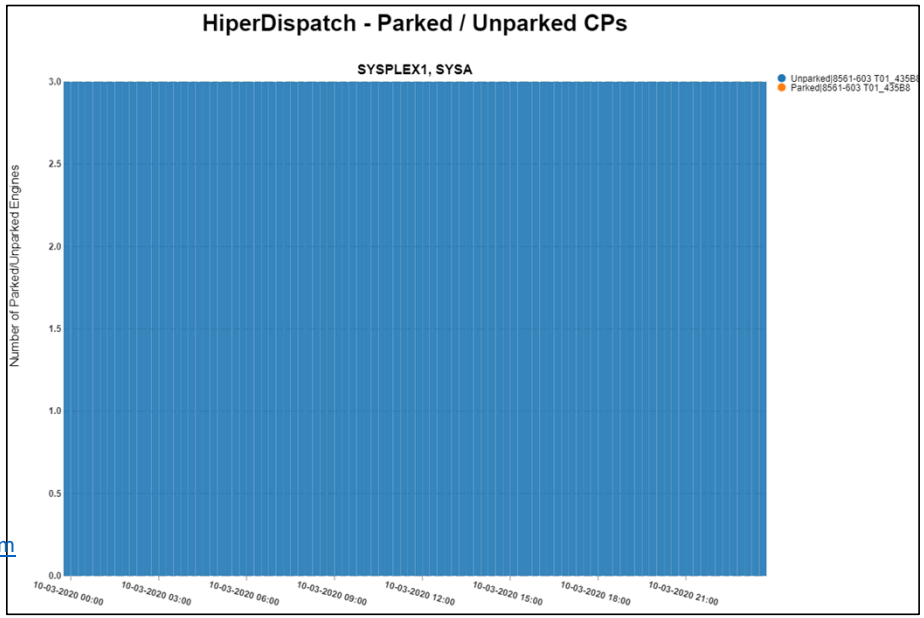
- This chart shows the percentage of the weight consumed
- 100% is an interesting number since it indicates the possible threshold for hardware capping when demand is greater than capacity of machine
  - > 100% indicates LPAR used more than its guaranteed share
  - < 100% indicates LPAR used less than its guaranteed share
- A flat line area, as circled, that is below 100% and usually indicates a different type of capping other than weight enforcement
  - Example: defined capacity limits



Capping and Weight Enforcement affect HiperDispatch Pooling

- When capping is enforced, it is very possible that HiperDispatch changes the pooling of the CPUs
  - In this example, 2 high and 1 mediums turn into 1 highs, 2 mediums
  - There are no low pool processors, so all are un-parked.
  - This matters since high pool CPUs tend to have longer dispatch intervals than medium pool CPUs

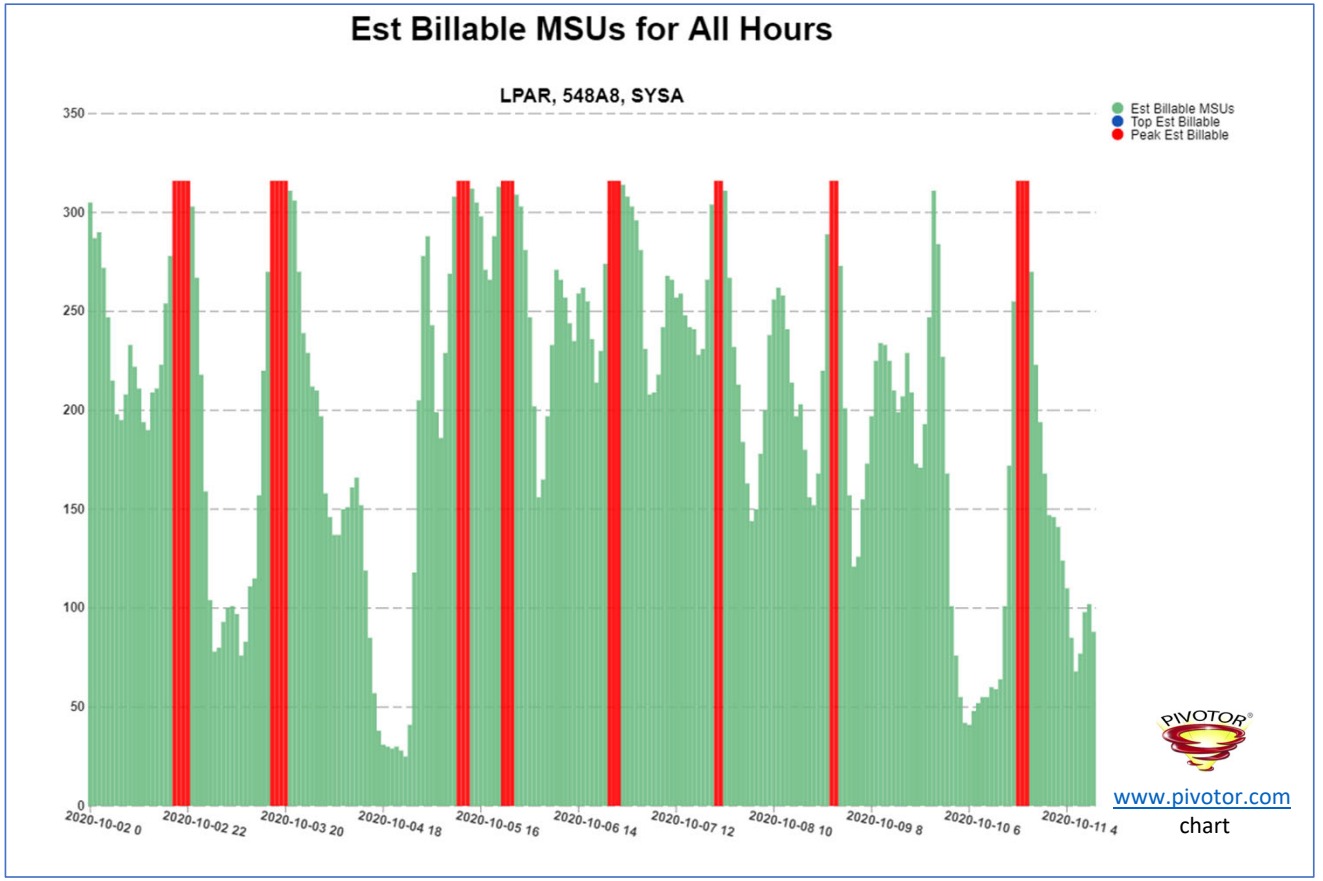




www.pivotor.com
charts

Looking for capping due to defined capacity limits

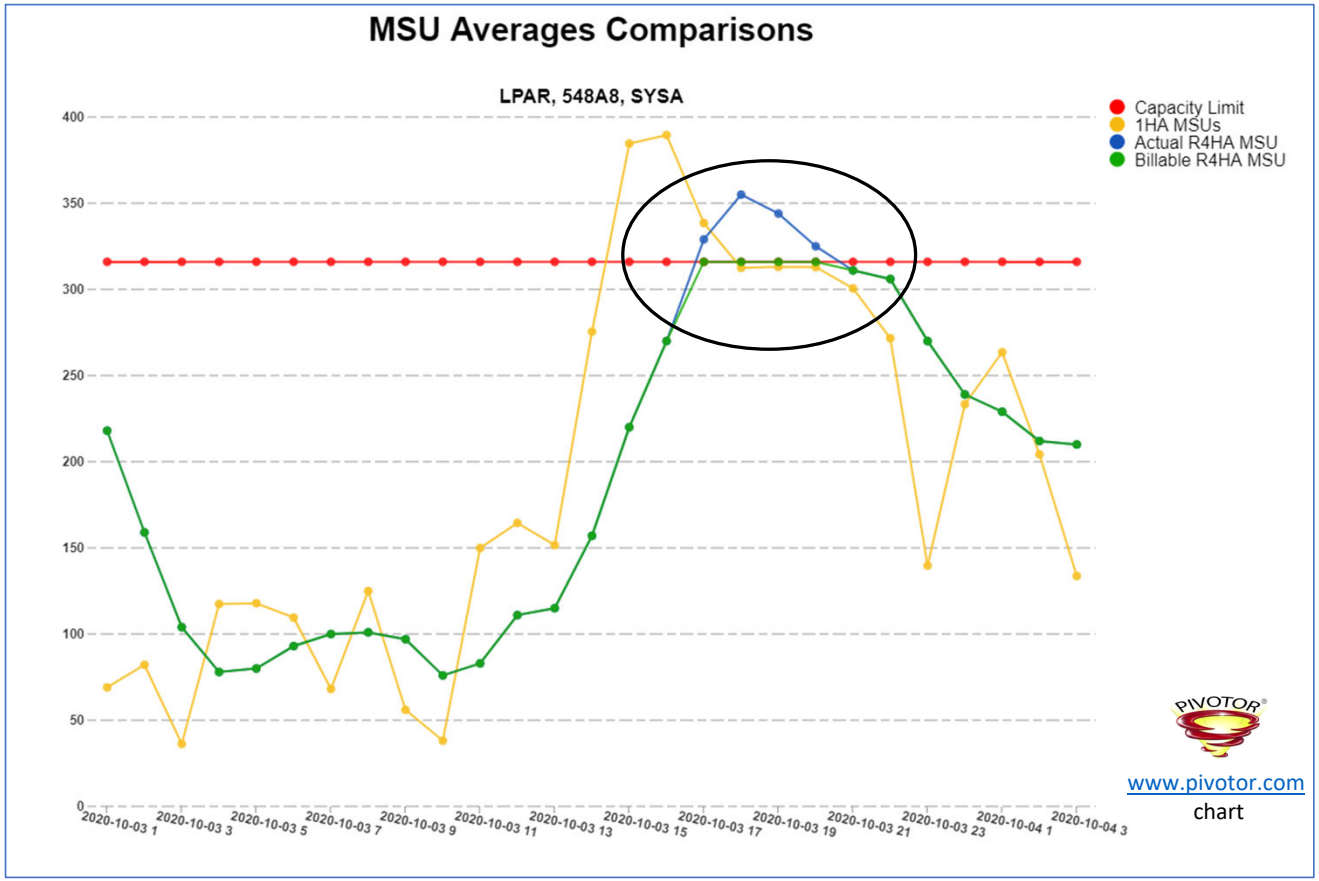
- Another area to examine is peak billable periods of the month.
- This report is for the first 11 billable days of October.
- Note that this customer hits their peaks quite often
- So probably working off a capacity limit
- It is worth examining one of these peak periods



Looking at MSU R4HA, Actual, and Image Capacity

- This chart shows the CEC image capacity, actual MSUs consumed, and the R4HA

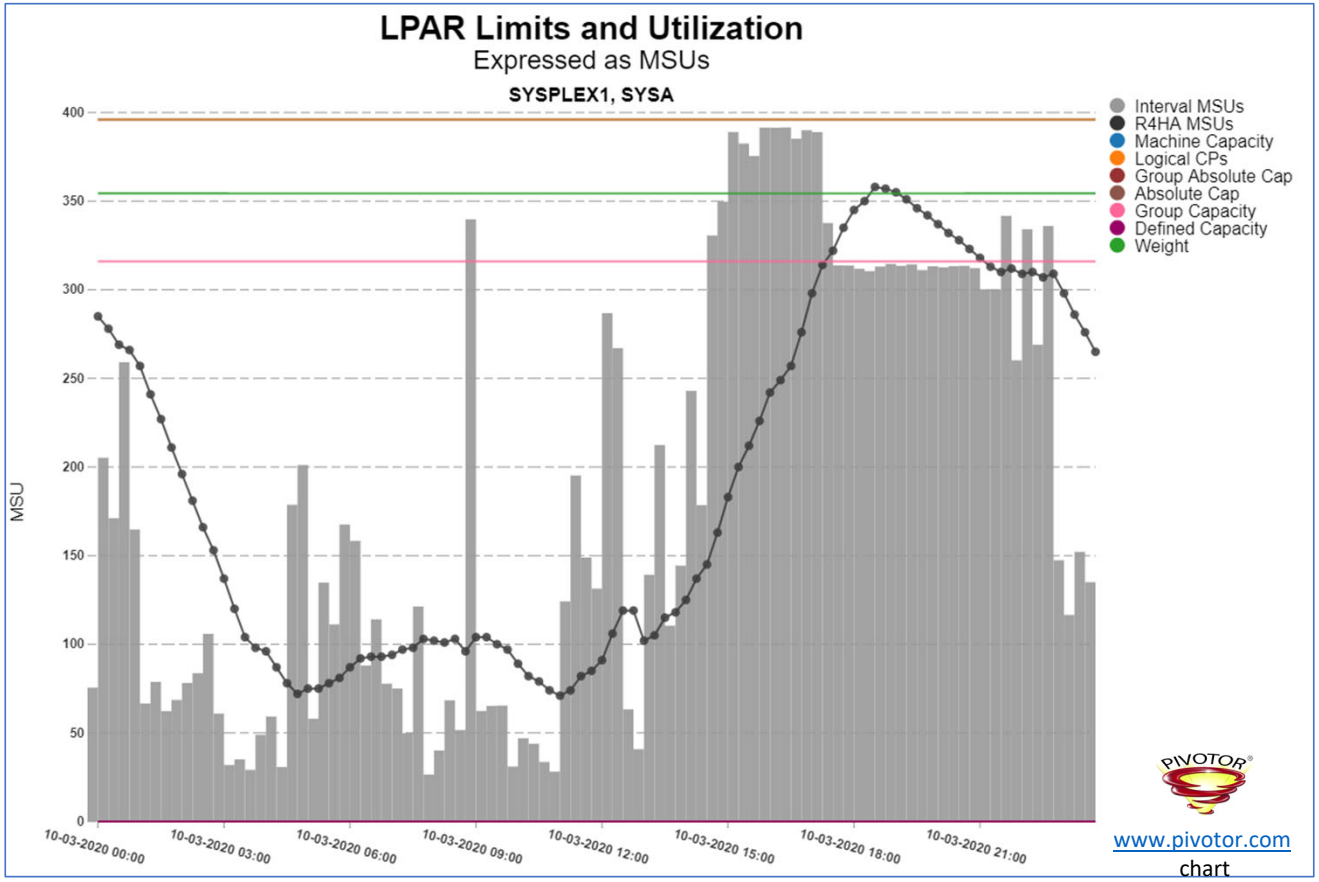
- Note on October 3 there are some interesting periods of time when it appears capacity limits are reached
  - Capping is occurring



Looking for capping due to defined capacity limits

- But capping is not the only resource control that limits an LPAR and its workloads

- For example, this chart shows 7 different limits that, if met, would limit an LPARs workloads from consuming CPU
  - And there are other limits, as well

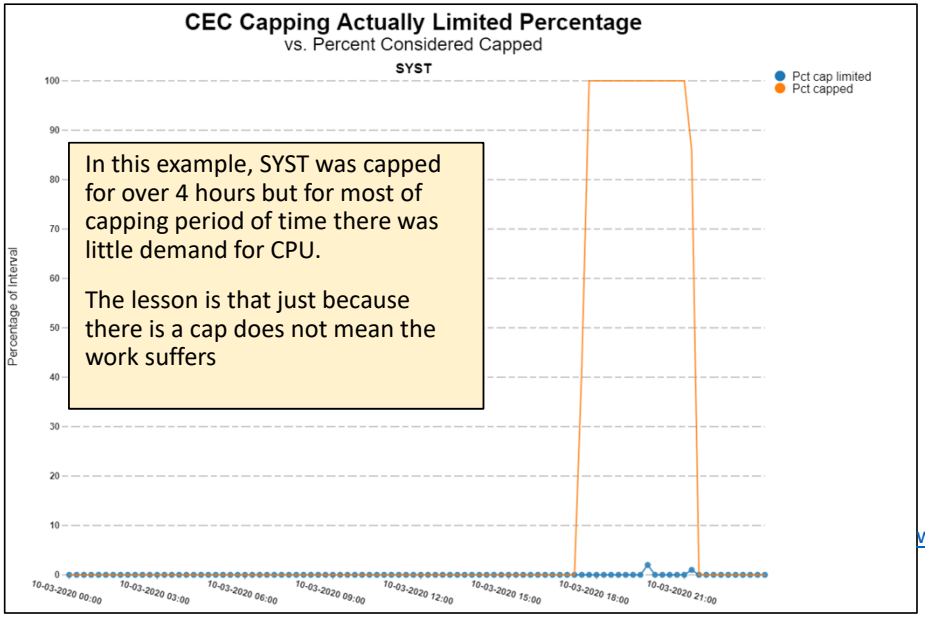


Did Capping Actually Limit the LPAR?

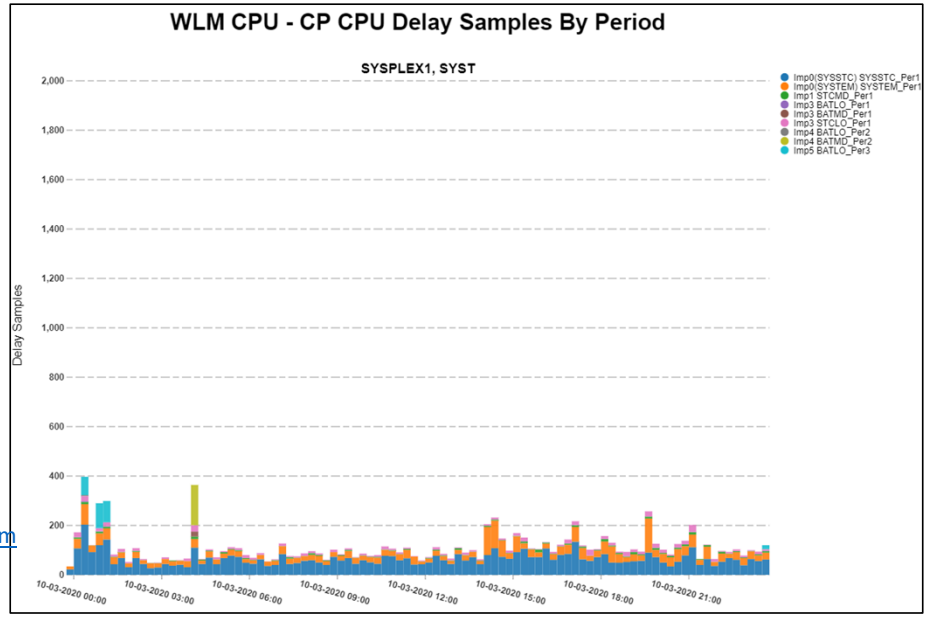
- Remember that capping does not always affect the workloads
- If demand for CPU is less than the cap, the cap isn't really limiting the LPAR
- RMF records:
  - Samples when the LPAR is considered capped
  - Samples where the cap limited the usage of processor resources
- "Considered capped" will usually work out to 100%, except for the first and last intervals when the cap is coming on or off
- "Actually limited" may vary throughout the capping period
  - Lower "actually limited" vs. "considered capped" means capping is causing less latent demand – i.e. capping is causing less delays for work
    - Likely because there's not demand for the full cap amount

Sometimes capping has no effect

- A cap could be in place, but if the workloads have no demand during the cap, then the cap is probably not causing much latent demand

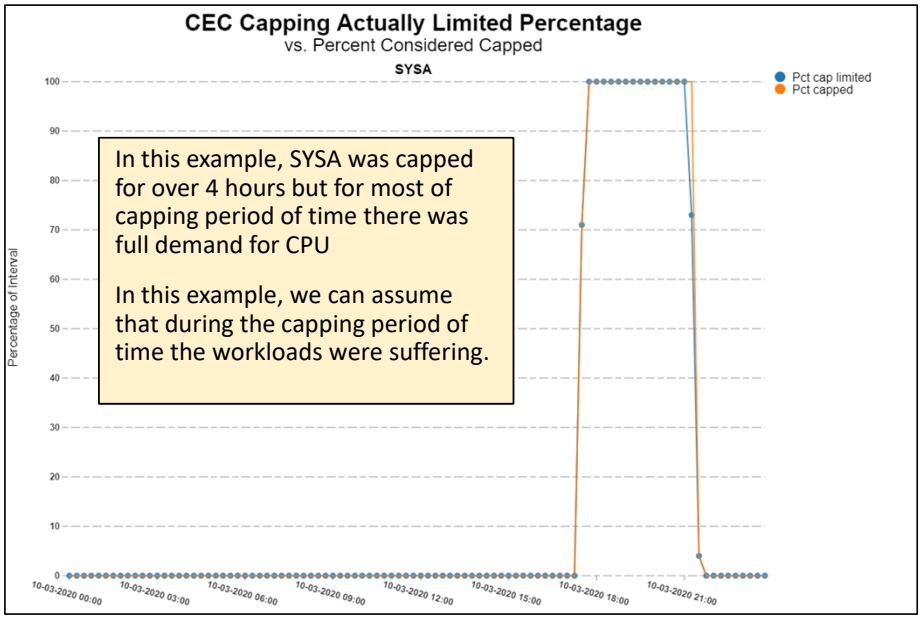


www.pivotor.com
charts

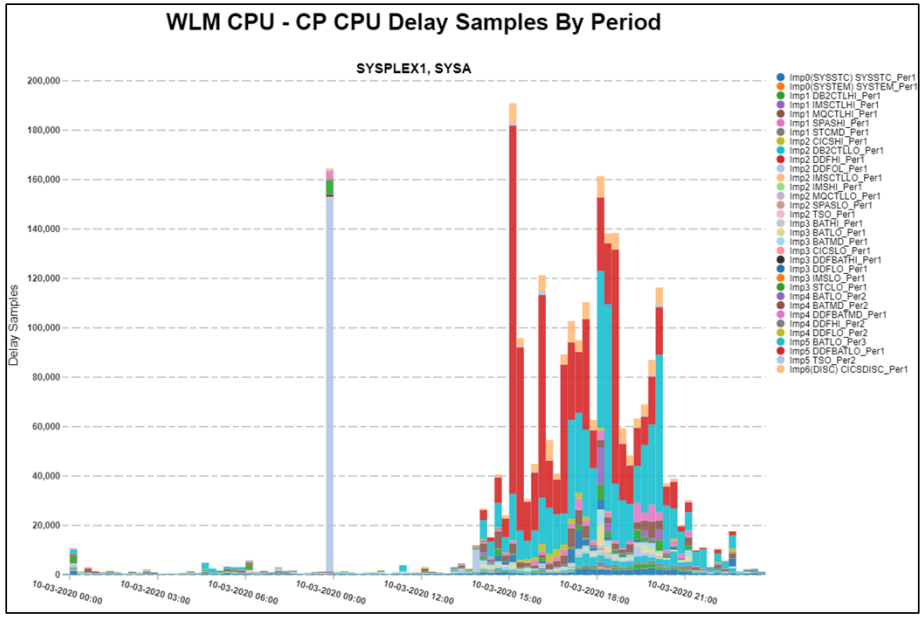


Sometimes capping has a big effect

- A cap could be in place, and if the workloads have lots of demand, then there would be a build up of latent demand
  - "Actually limited" may vary throughout the capping period
  - Lower "actually limited" vs. "considered capped" means capping is causing less latent demand – i.e. capping is causing less delays for work
  - Likely because there's not demand for the full cap amount



www.pivotor.com
charts



Looking at latent demand

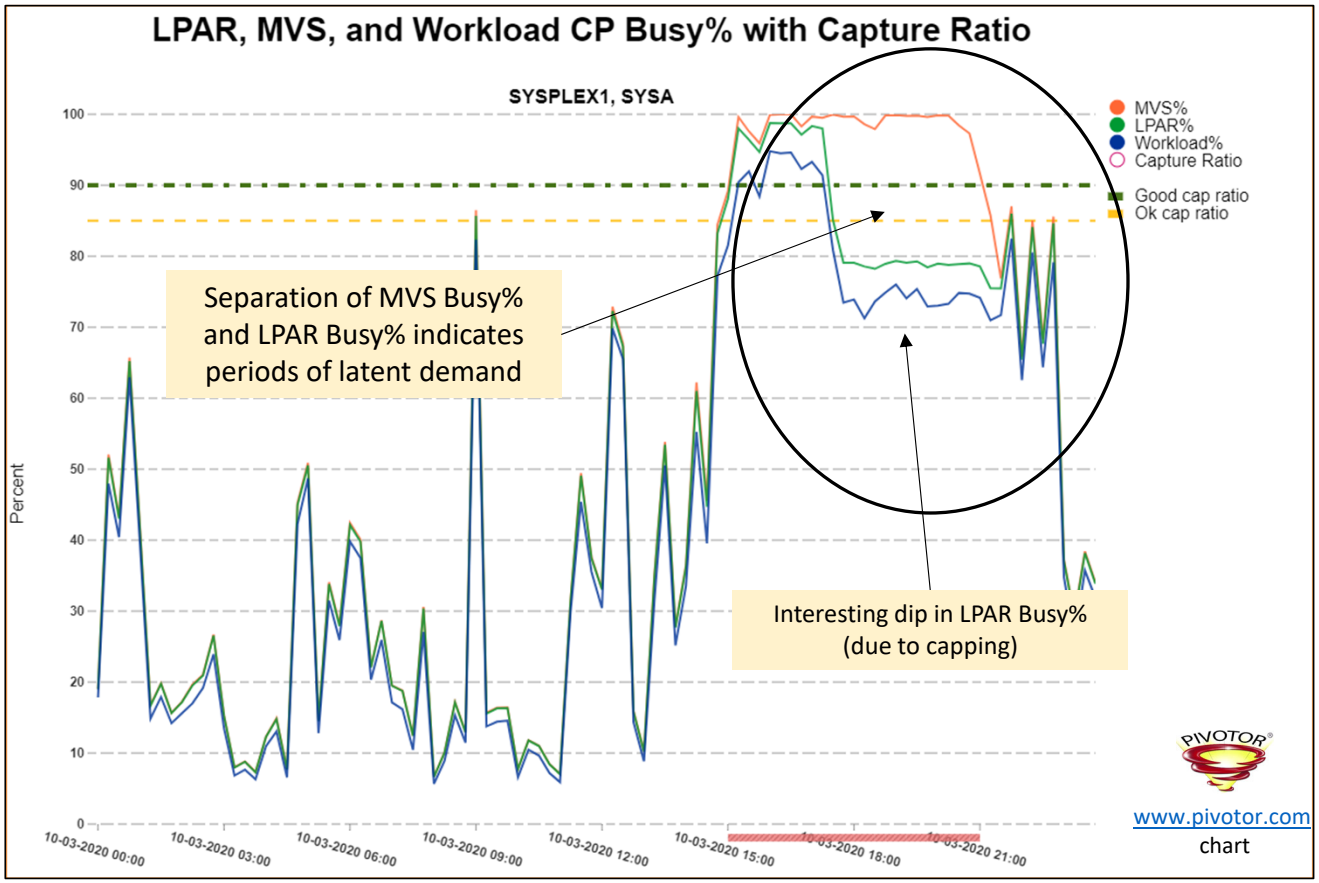
How does capping manifest itself?

What does it do to the workloads?

What can we see in the measurements?

LPAR Busy % vs MVS Busy %

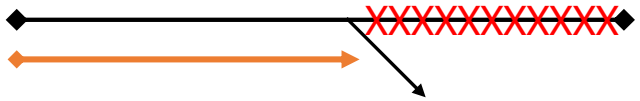
- LPAR Busy % measures how busy the LPAR kept its logical processors
- MVS Busy % measures how much of the logical resource the LPAR wanted
- Differences indicate latent demand
  - Flat lines usually indicate capping or weight enforcement



Understanding Dispatching to Gain Insights to MVS Busy %

- Dispatch Time
  - Time logical processor is associated with a physical processor

- MVS Time
  - Time z/OS was busy before voluntarily giving up a processor

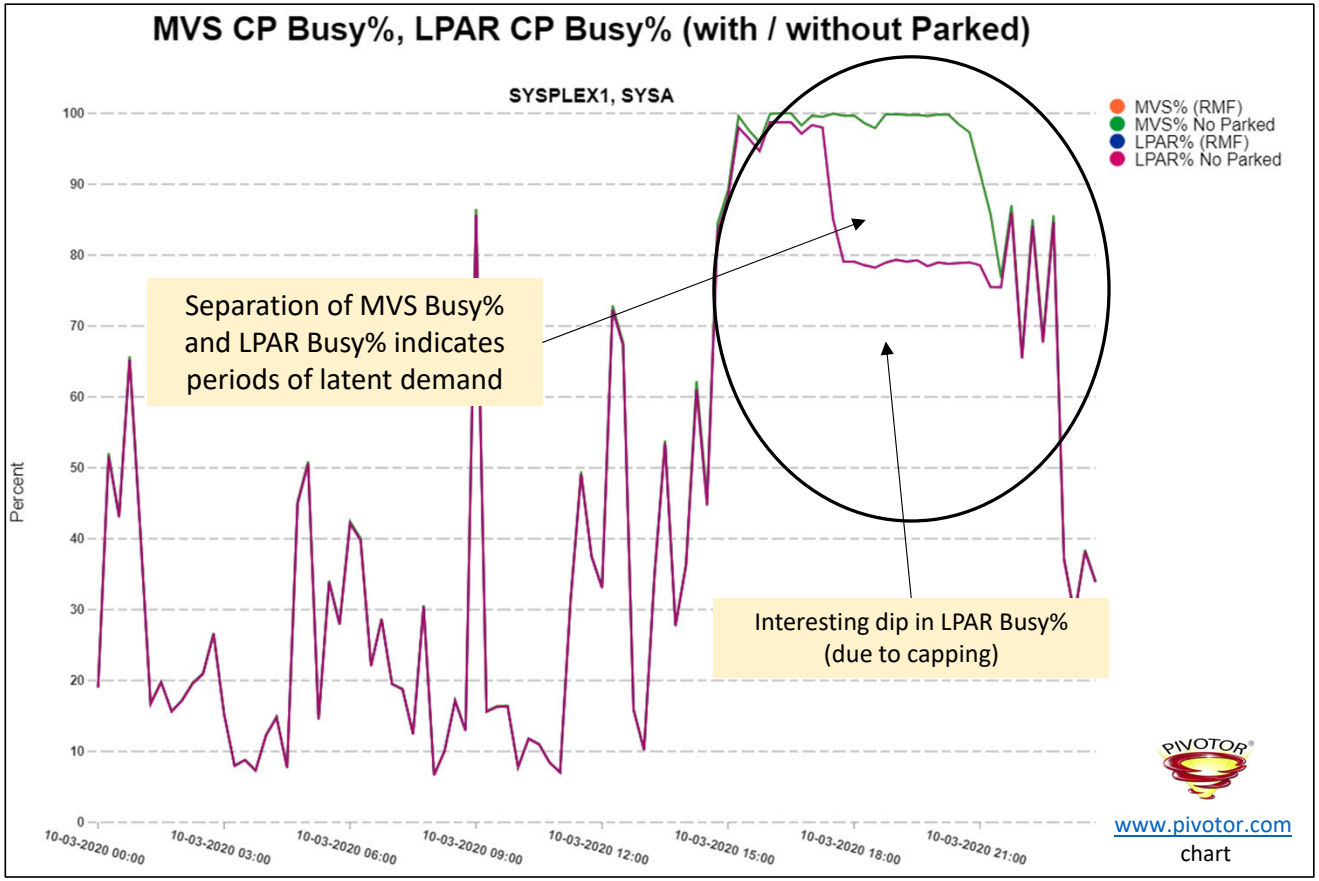


Voluntary Wait
- z/OS voluntarily gives up the processor
- MVS time equals dispatch time

Involuntary Wait (mostly on vertical mediums)
- z/OS does not give up the processor voluntarily
- Instead PR/SM un-dispatches the partition
- MVS time will be greater than dispatch time

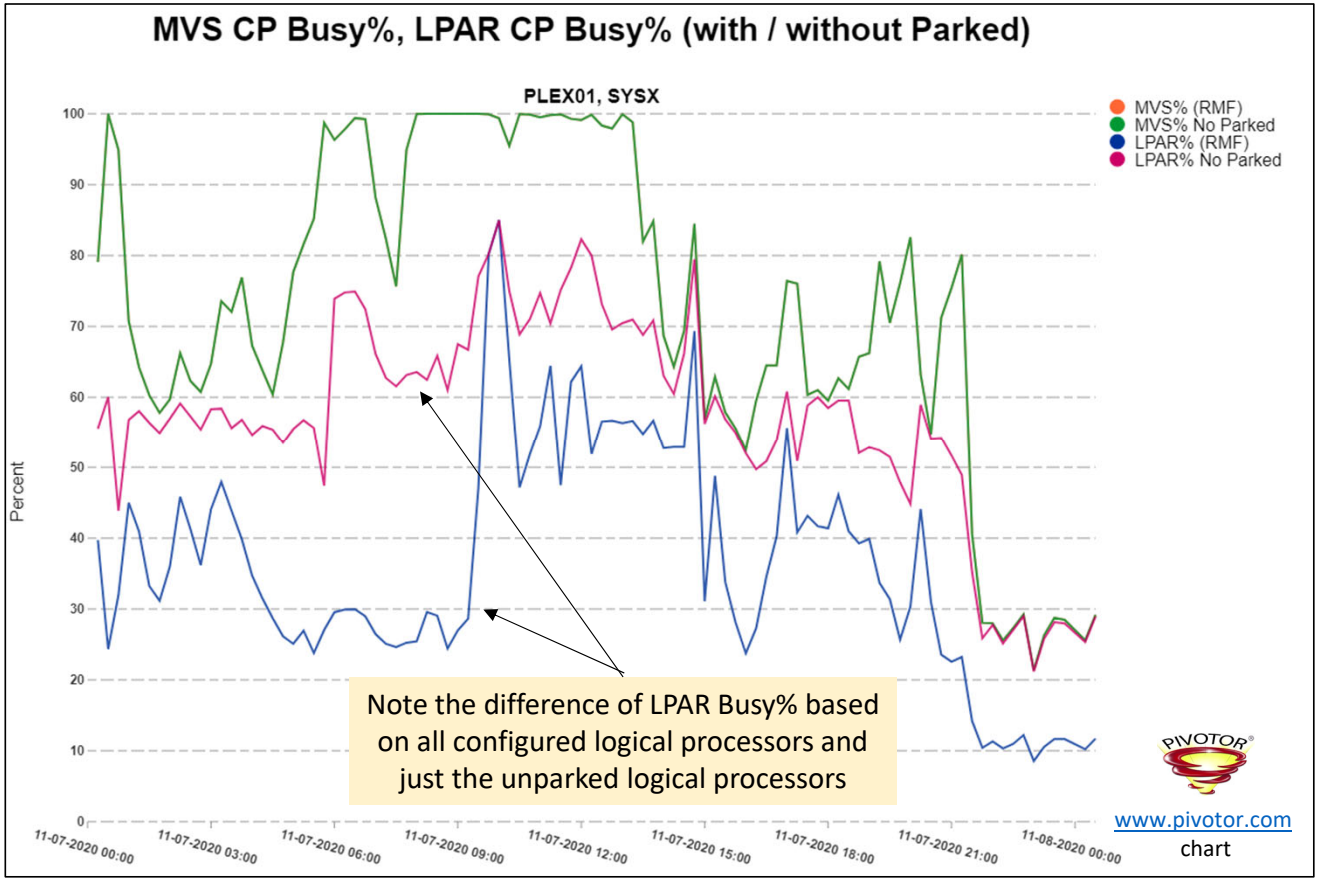
LPAR Busy % with Config CPs and only Unparked CPs

- LPAR Busy % based on configured number of logical processors
  - Reports logical constraint of the LPAR
- LPAR Busy % based on unparked number of logical processors
  - Reports the HiperDispatch constraint



LPAR Busy % with Config CPs and only Unparked CPs

- LPAR Busy % based on configured number of logical processors
  - Reports logical constraint of the LPAR
- LPAR Busy % based on unparked number of logical processors
  - Reports the HiperDispatch constraint



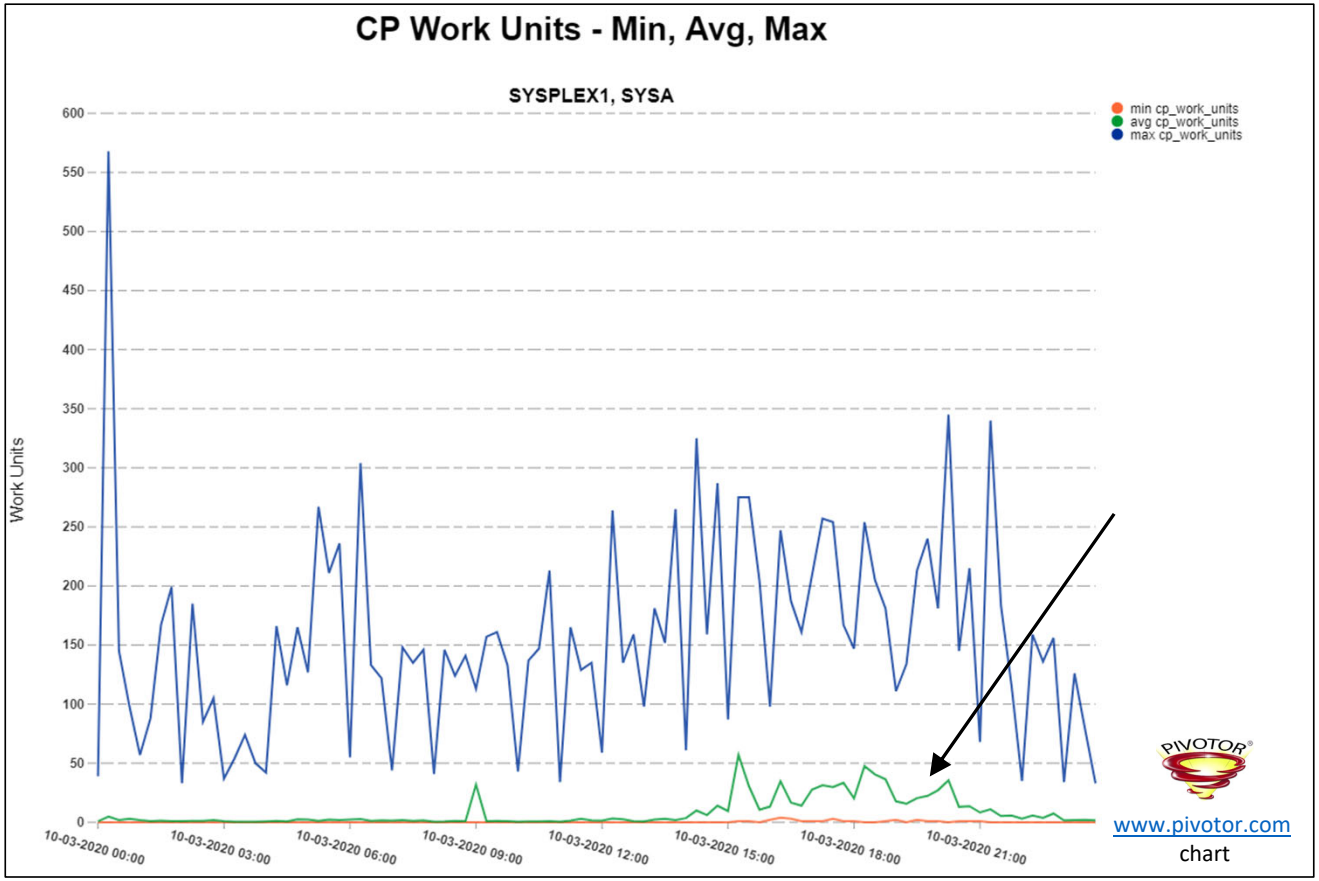
Work unit Queuing

- Work units is a dispatchable unit of work
  - Also known as: TCBs, SRBs, PGM=program, threads, etc.
  - Not the same as address spaces
  - Since an address space could be multiple threaded and have multiple work units
    - Thus, Work Units are a more accurate representation of work because we have an ever increasing number of multi-threaded address spaces

- z/OS measures the running or waiting work units
  - Values by processor type (GP, zIIP)
  - Plot min, average, max over time
    - Max is often far larger than average
  - Distribution of observations
    - Based on the number of online and not parked processors (N)
    - Counts in buckets: N, N+1, N+2, N+3, N+5, N+10, N+15, N+20, N+30…

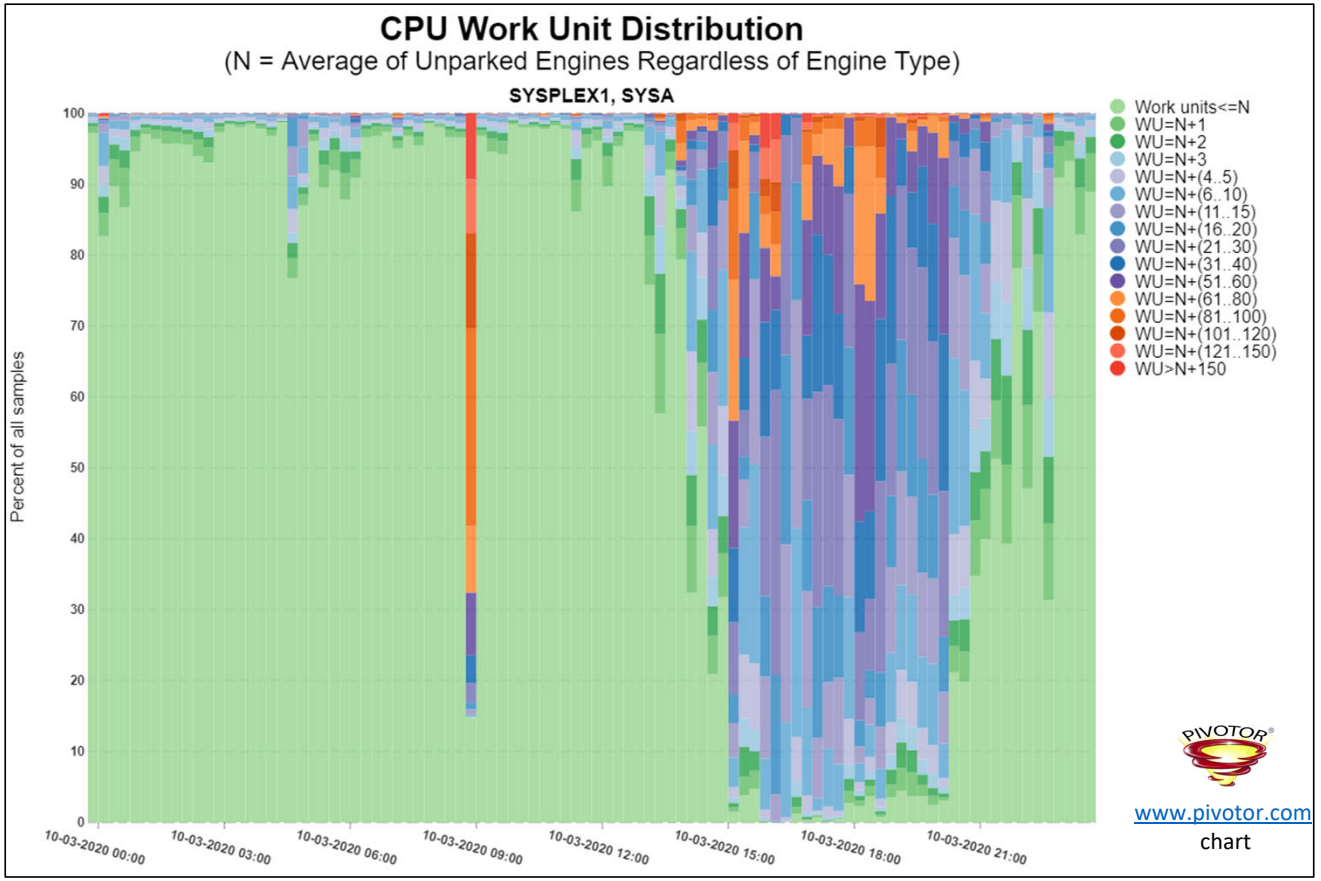
Minimum / Maximum / Average work unit queue length

- The following system has 3 CP GCPs
- At 18:15
  - Min WU queue is 0
  - Avg WU queue is 47
  - Max WU queue is 254
- An indicator of latent demand
- But to be fair, notice the minimum spike at about 18:00
  - Probably an influx of nighttime workload



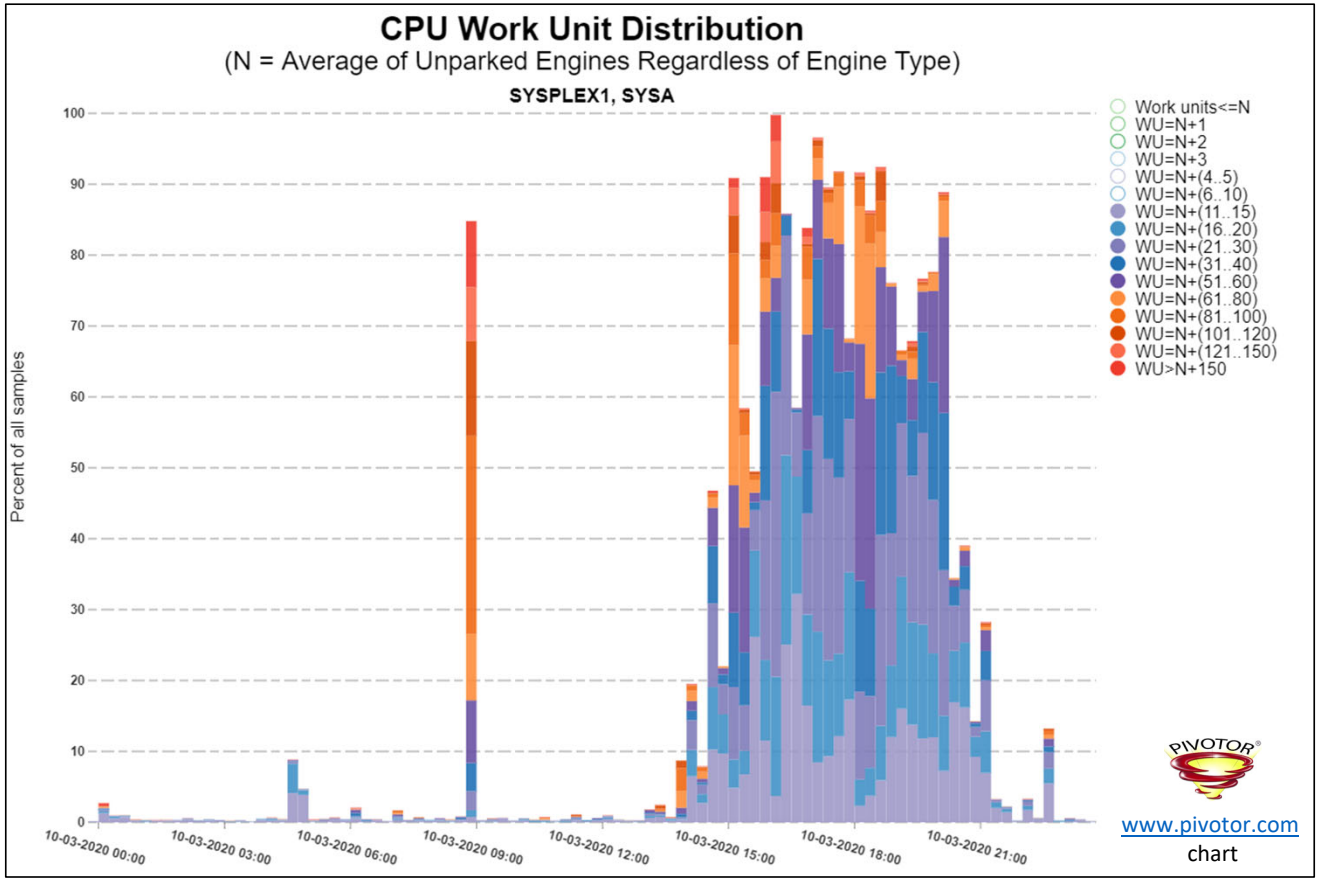
Distribution of work unit queue lengths

- Each bucket of the distribution represents the percentage of the measurement interval the queue of work waiting to use the CPUs is a certain length:
  - N = number of unparked CP + zIIP engines



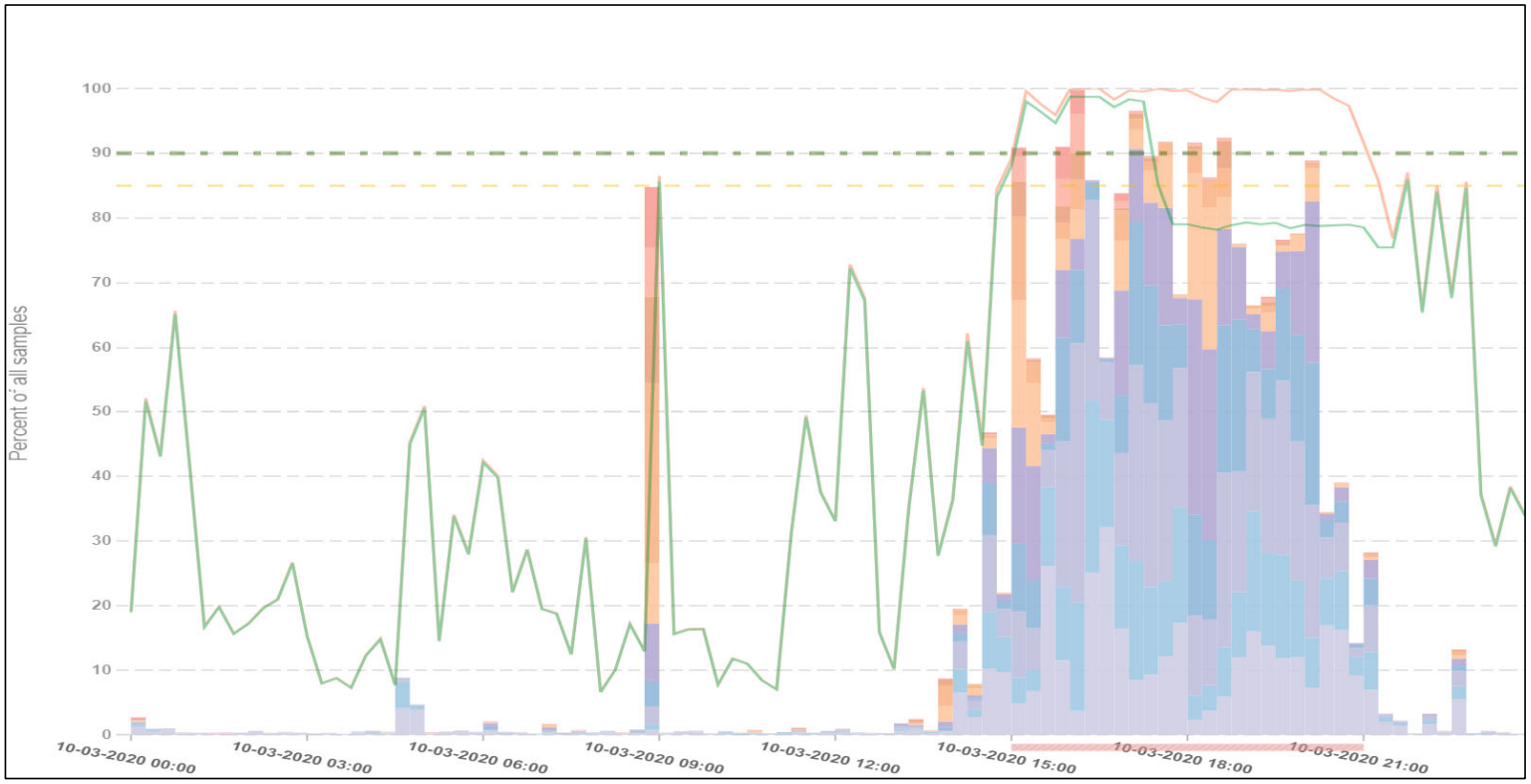
Distribution of work unit queue lengths

- How much latent demand is too much, too unhealthy?
- Assuming a rule of thumb that CP queues lengths of > 3 times the number of CP CPUs is unhealthy latent demand
  - We see here that during the evening hours we have continuous unhealthy latent demand
  - With large percentages of the measurement intervals of more than 100 Work Units queued up



Relationship of LPAR% delta to MVS%, and Work Unit Queuing

- When we overlay the two charts, we see a correlation



www.pivotor.com
chart

So, what might be at risk when capping?

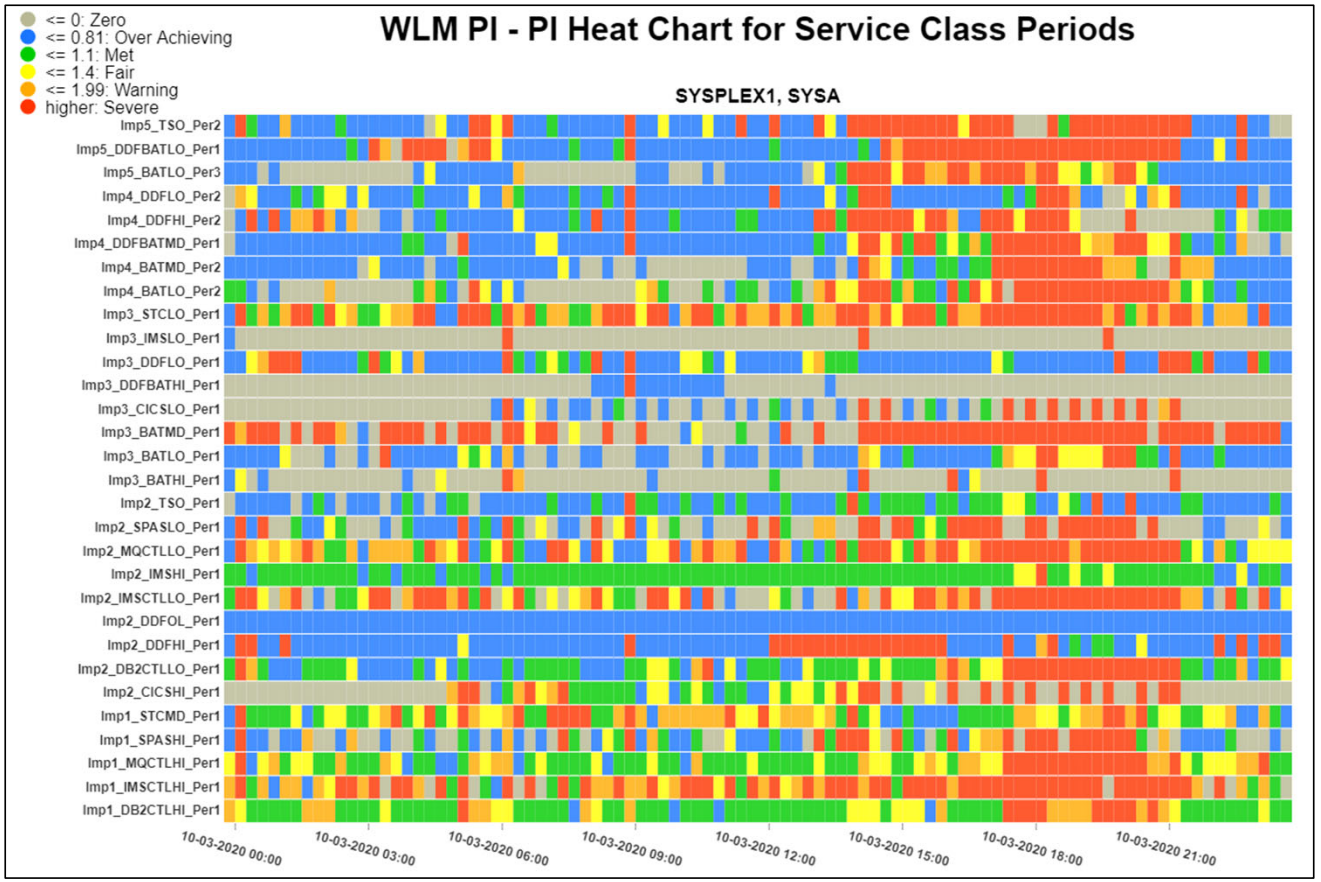
- Before you start capping you should consider what might be at risk

- Workloads that are doing better than their goal (PI < 1) may be degraded to their goal
  - Potentially, even to help lower importance workloads
- If high importance workloads are missing their goal, hopefully you have lower importance workloads that WLM can borrow from
  - If everything is importance 1, nothing is important

- You should revisit your policy before capping
  - And periodically of course!

WLM Performance Indexes can indicate latent demand

- This WLM PI chart shows that when capping is enforced goals are affected
- The question is, are the lower importance workloads being hurt more
- Never assume goals and importance level are correct

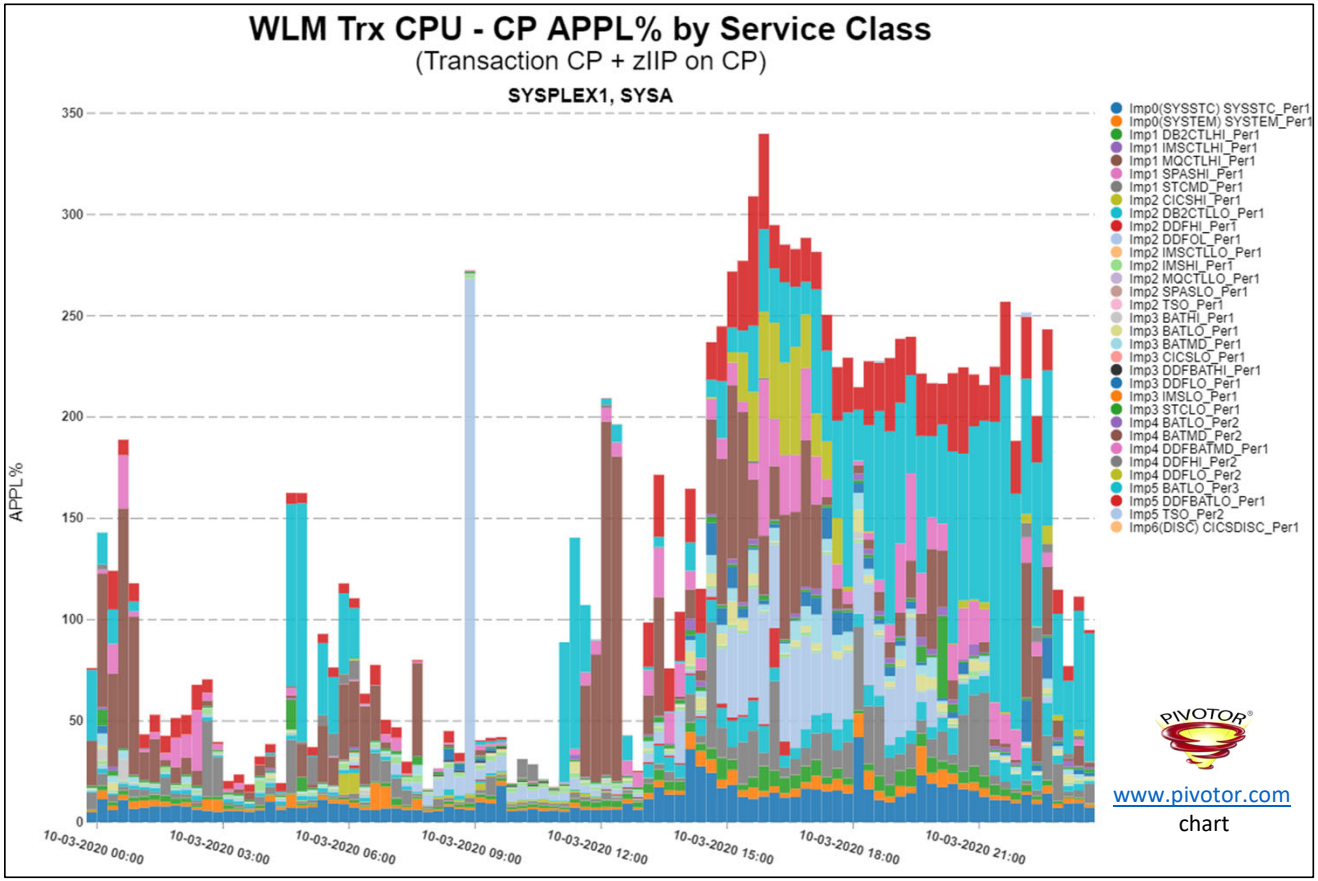


www.pivotor.com
chart

CPU APPL% Consumption – By Importance, By Service Class Period

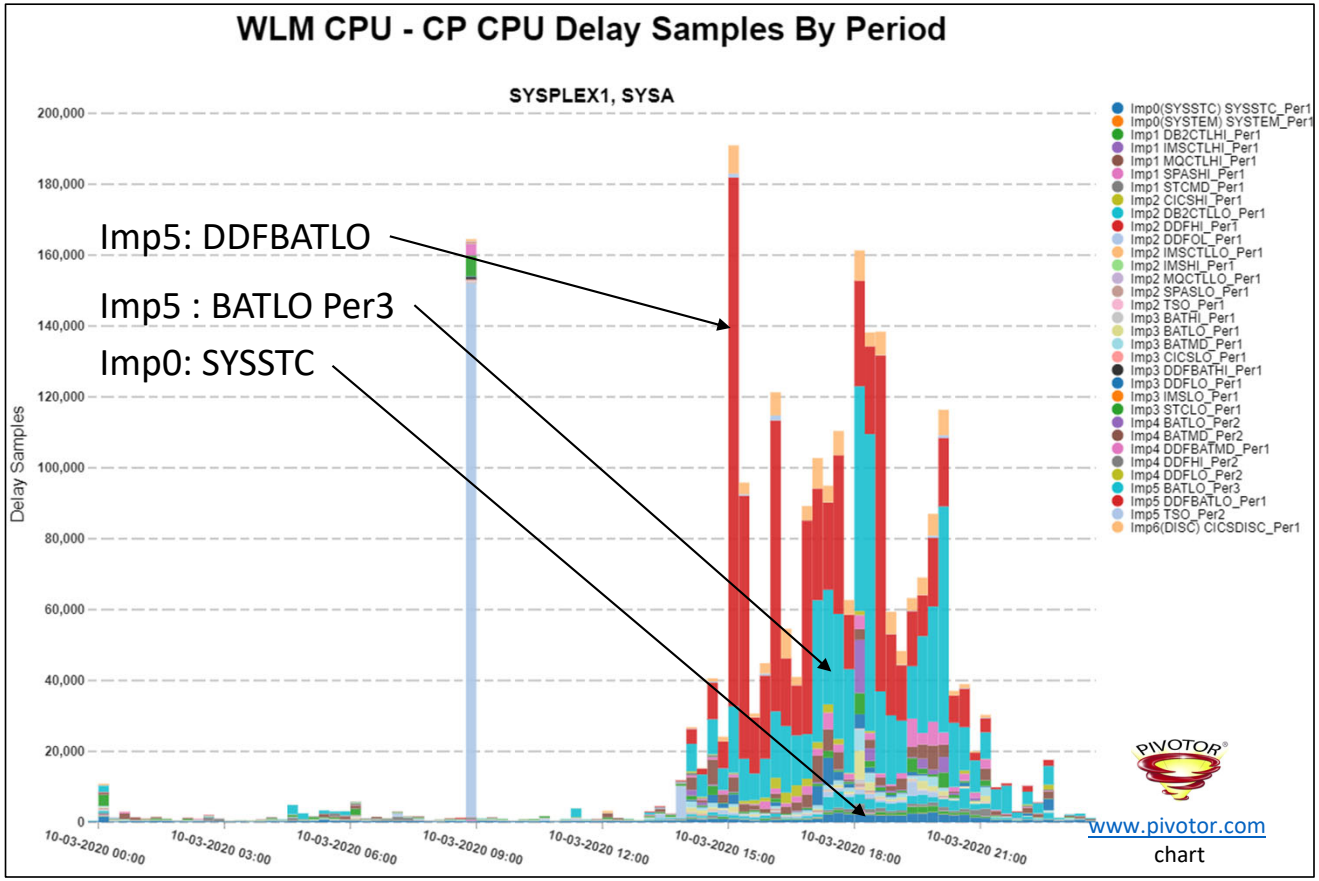
- Examine CPU consumption during capping periods

- We see here that the largest workloads are
  - BATLO per 3 at Imp 5
  - DDFBATLO at Imp 5
  - Is this work consuming CPU before higher importance work?
  - Is it ok that work running at lower importance suffer CPU delays



CPU delay samples – By Importance, By Service Class Period

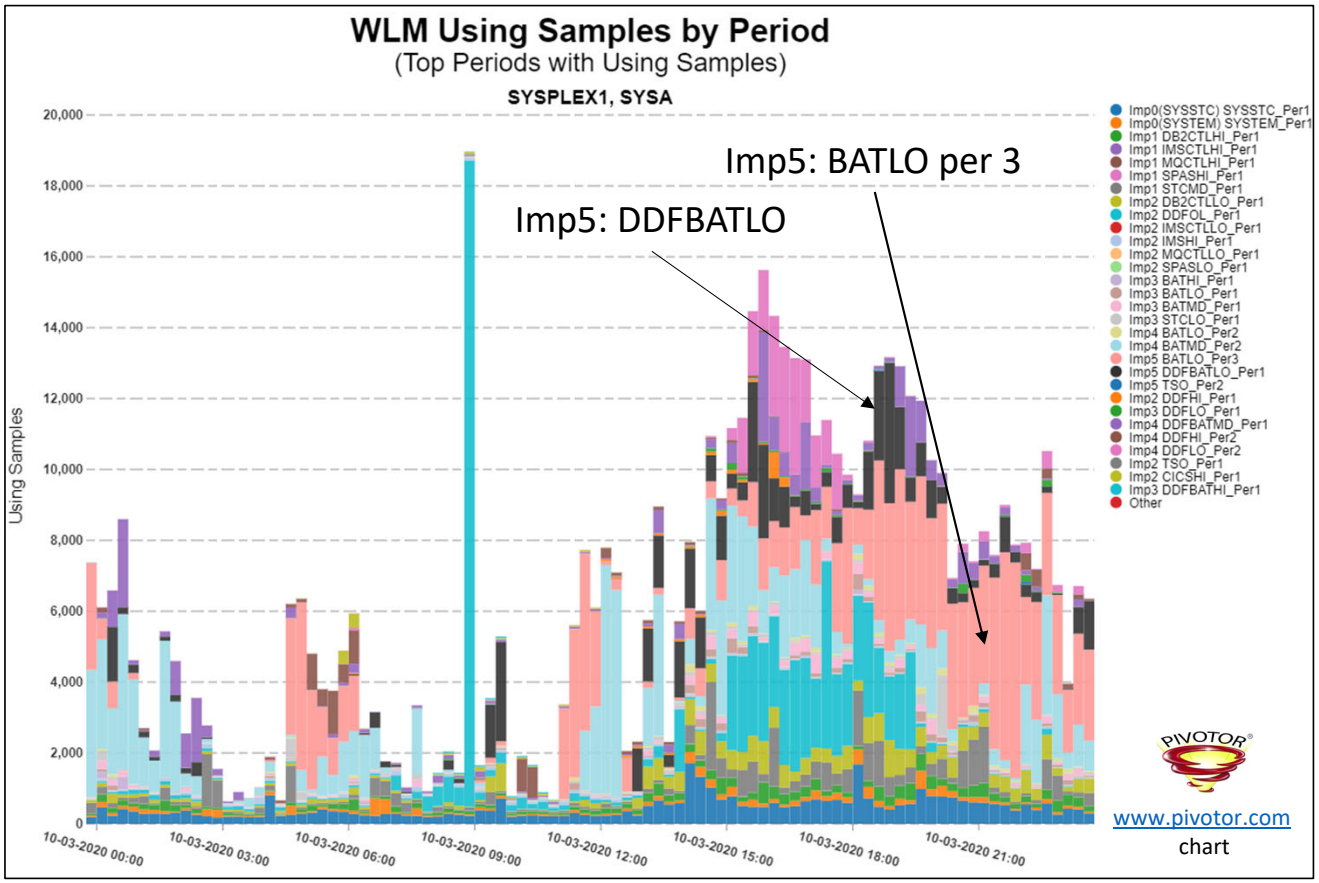
- So look at CPU delays
  - Other delay types will be of interest, but for capping, CPU delay will be the most interesting
- What work is delayed?
- Is the right work delayed?
- Is delay proportional to the work?



CPU Using samples – By Importance, By Service Class Period

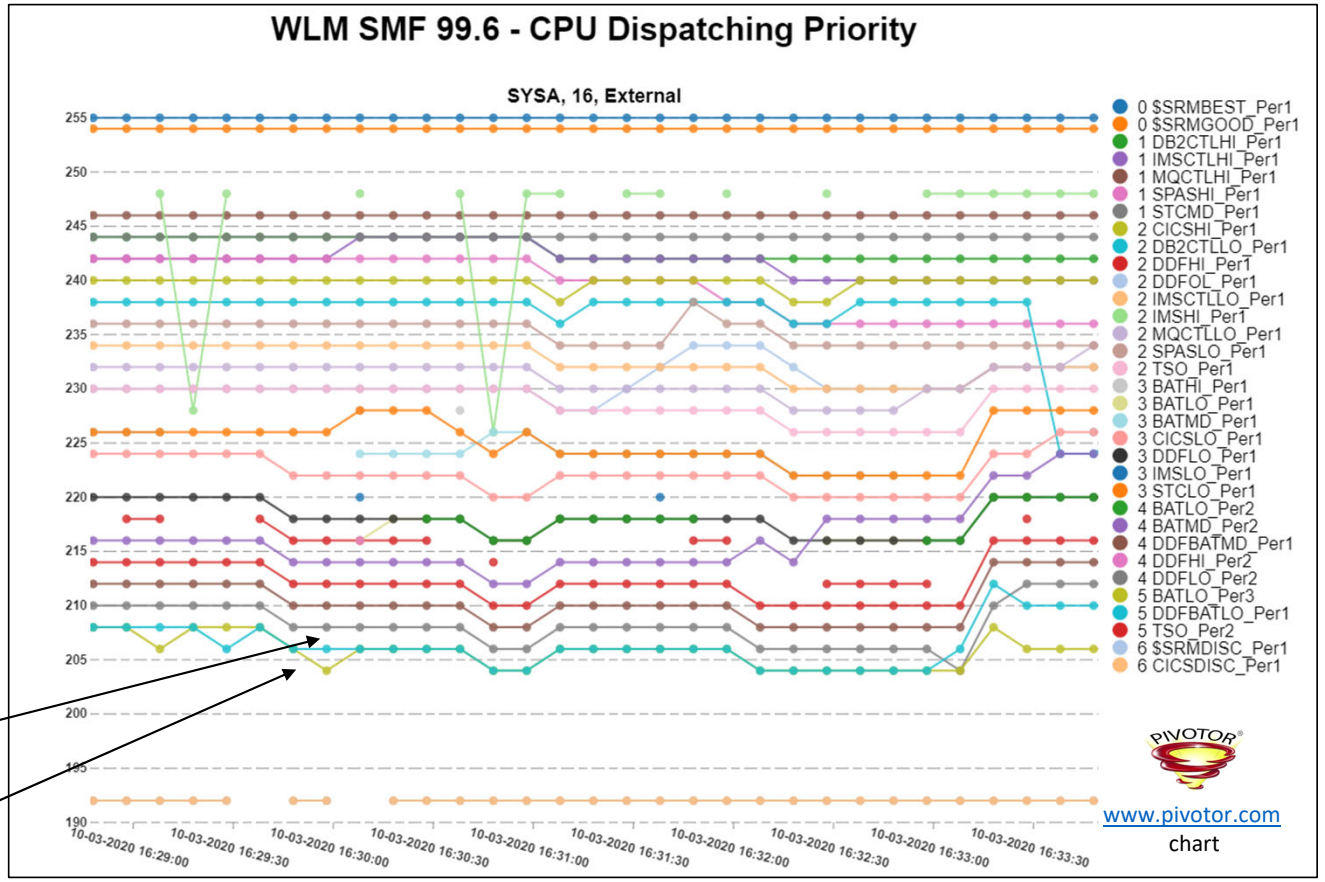
- Also look at CPU using samples
- In this case we also see that BATLO also has lots of CPU using samples

- BATLO Per 3
  - Consumes lots of CPU
  - Has lots of CPU delay
  - Has lots of CPU using



Looking at CPU Dispatching Priorities

- Do not assume goals and importance levels are correct
- Verify CPU dispatching priorities
- When a cap is enforced
  - Do the right workloads have first access to the CPU?



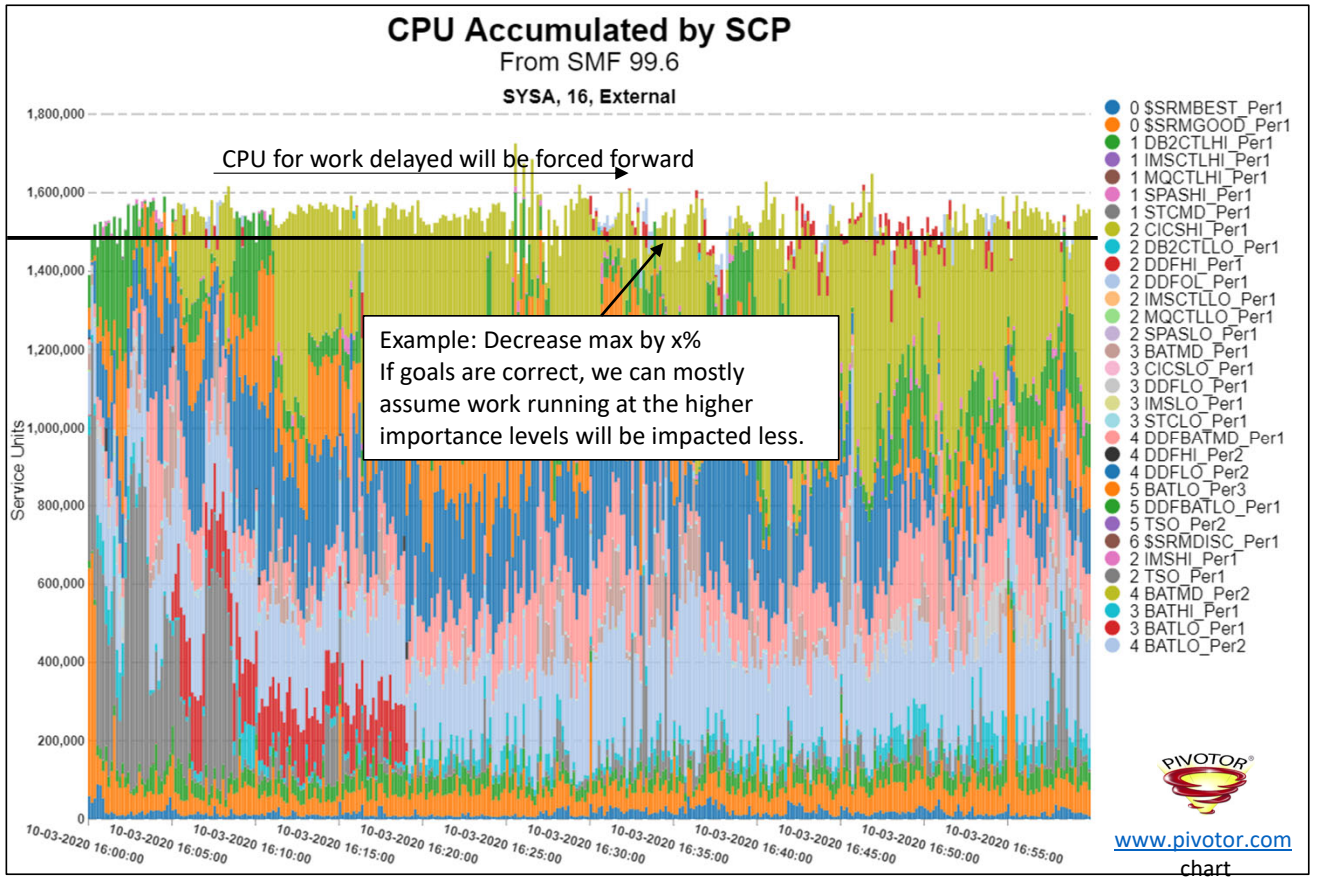
Imp4: DDFLO per 2

Imp5: BATLO per 3

Imp5: DDFBATLO per 1

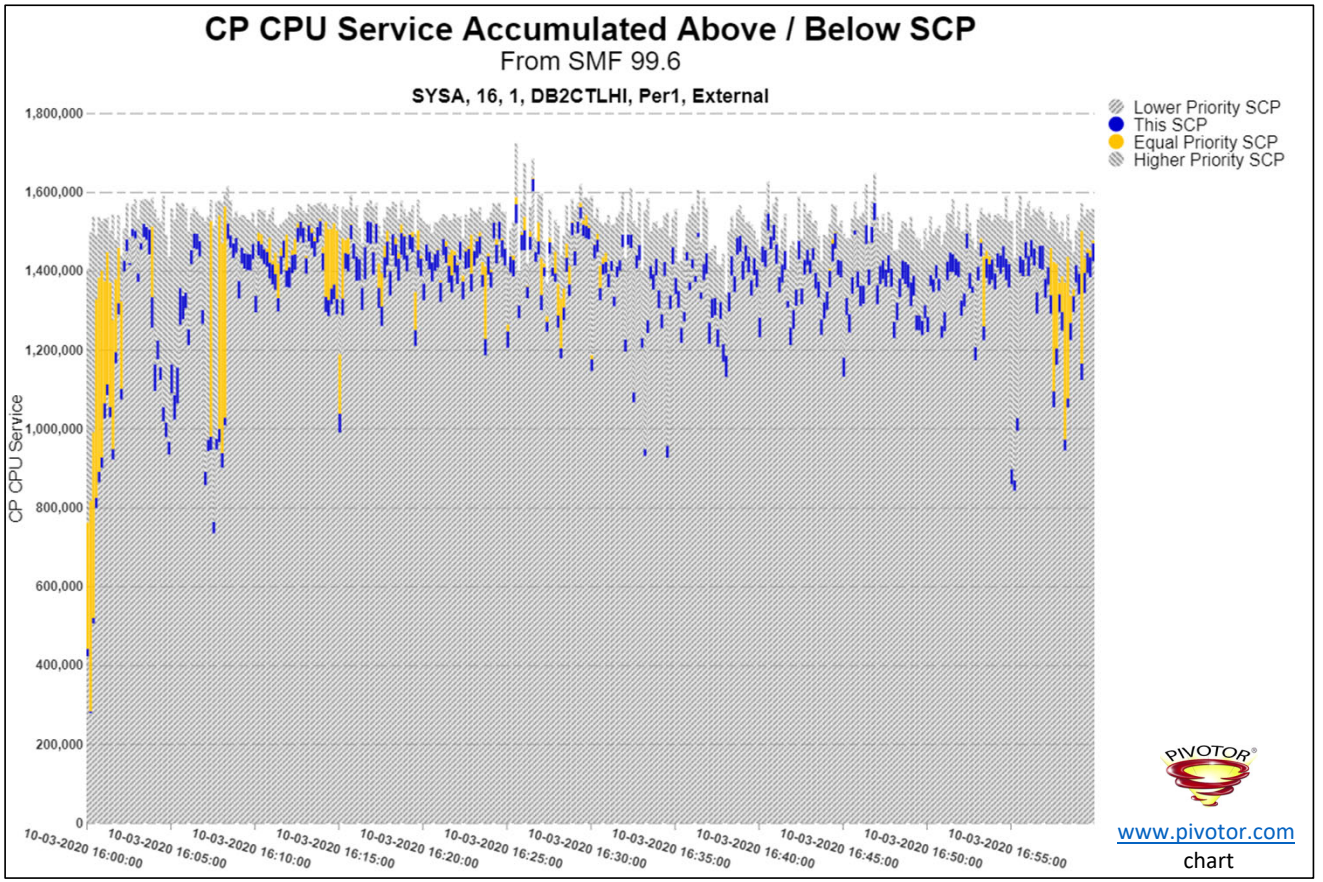
Typical Exercise – If considering capping (or lowering an existing cap)

- Here is a chart of consumed by each WLM service class period and ordered by WLM importance level
  - Chart just shows 16:00 hour
- There are challenges in doing this, but assuming goals are correct, and CPU dispatching priorities are as hoped, then a somewhat straight forward exercise:
  - If you are thinking of imposing or lowering a maximum, one can roughly project which workloads will suffer
  - Must still look at how response times and velocities will change



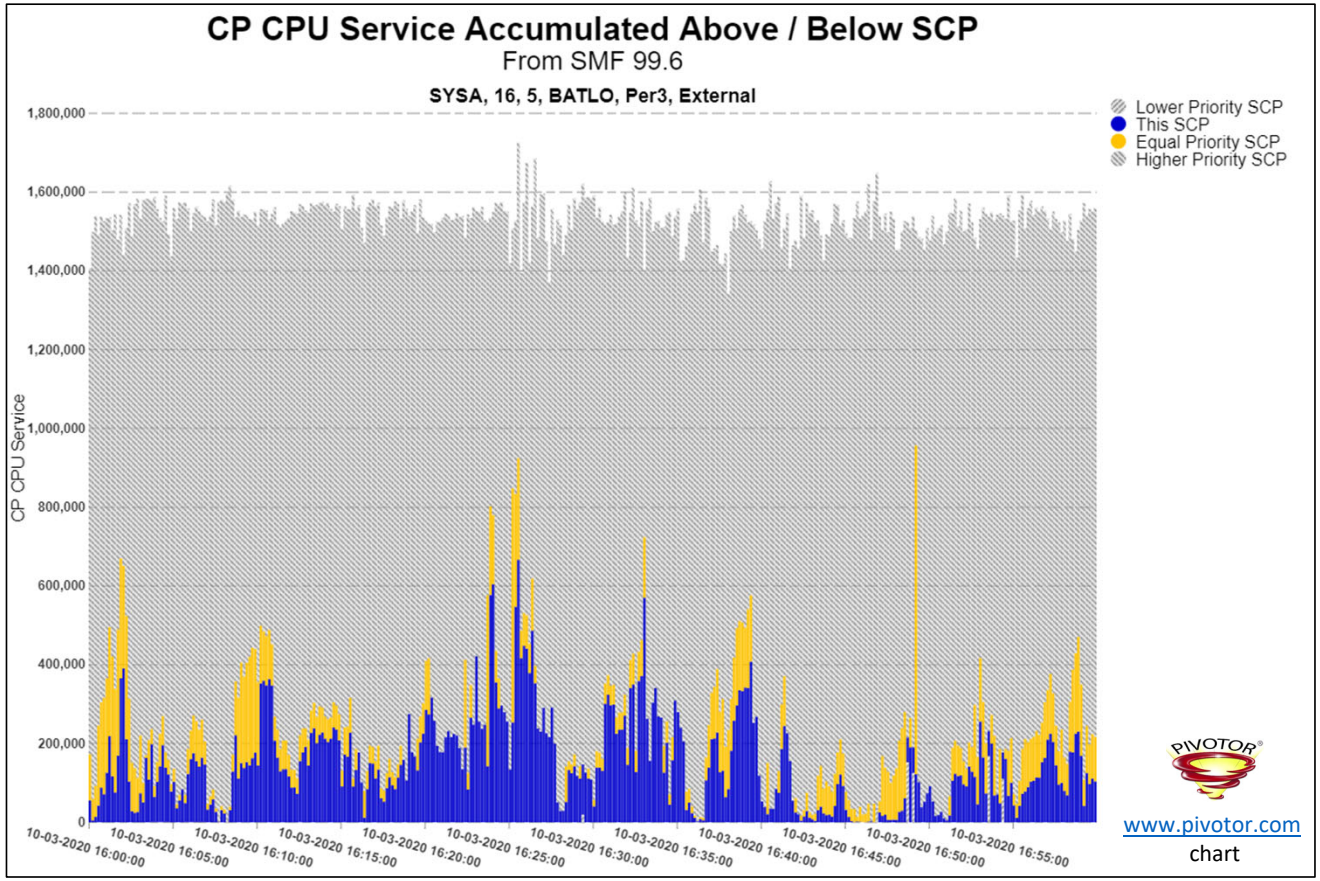
Service consumed at CPU dispatching priorities

- Also look to see how much service is being made available above and below the large consuming workloads
- In this example, during capping DB2CTLHI does not use much CPU (dark blue)
  - But not much CPU used at higher priorities
  - And lots of CPU available to lower priority work



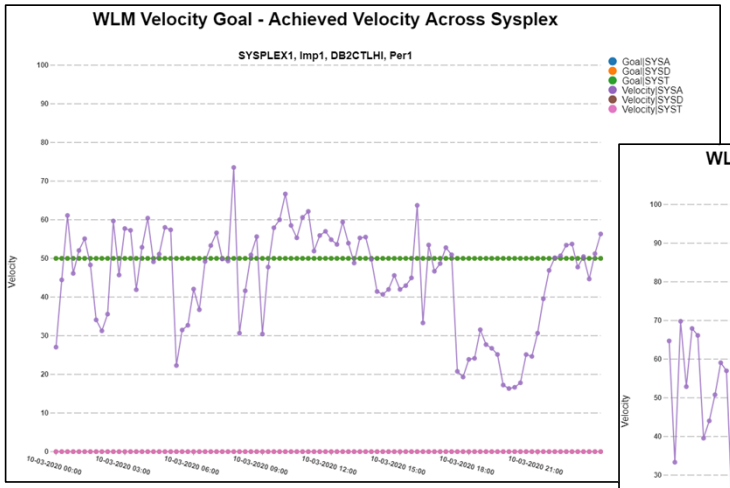
Service consumed at CPU dispatching priorities

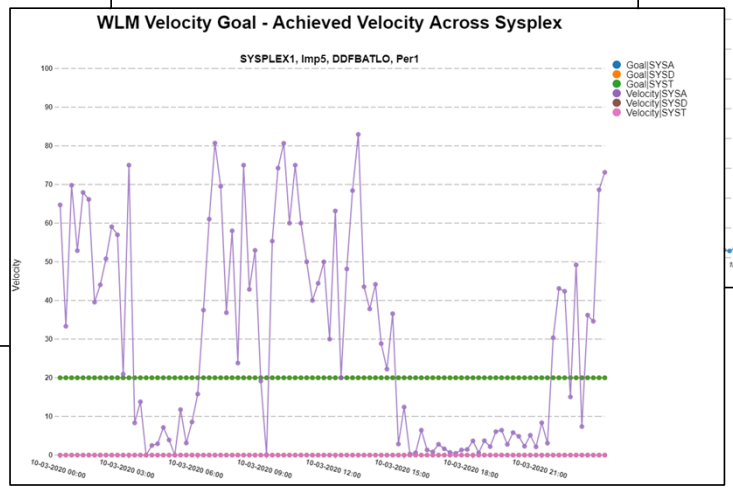
- In this example, we see that, relatively speaking that for BATMDSCH, Per1:
  - Not much CPU used at higher priorities
  - Very little left to the lower priority work
  - Is it enough
    - This is why CPU delay samples are important

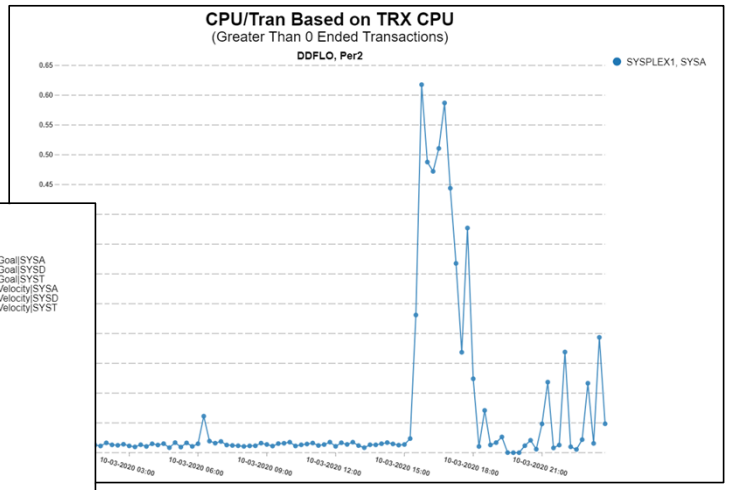


Look at goals, response times, CPU/Tran

- Of course, make sure you look at all the typical performance indicators to evaluate the impact of when CPU is limited to a workload (regardless of capping type)



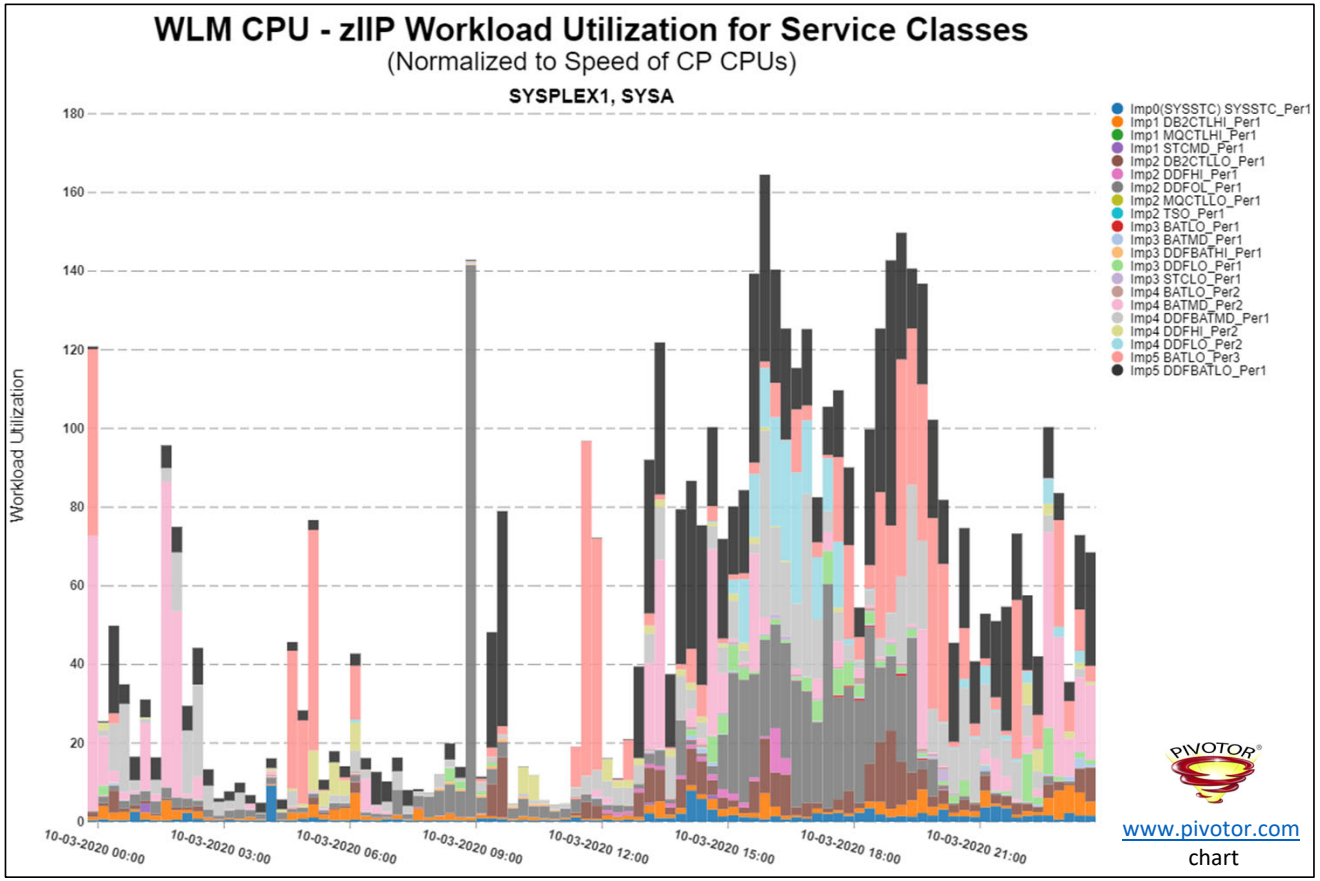




www.pivotor.com
charts

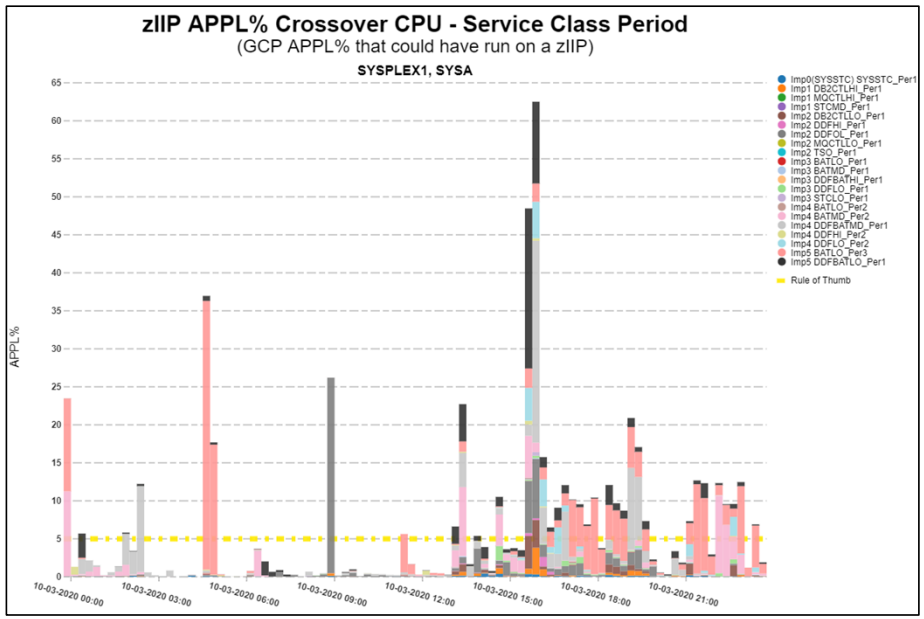
Same exercise for zIIP engines

- It is also possible that zIIP engines have latent demand.
- So same analysis exercise applies
- But zIIPs can also cross over and affect CPU usage on the CP engine.

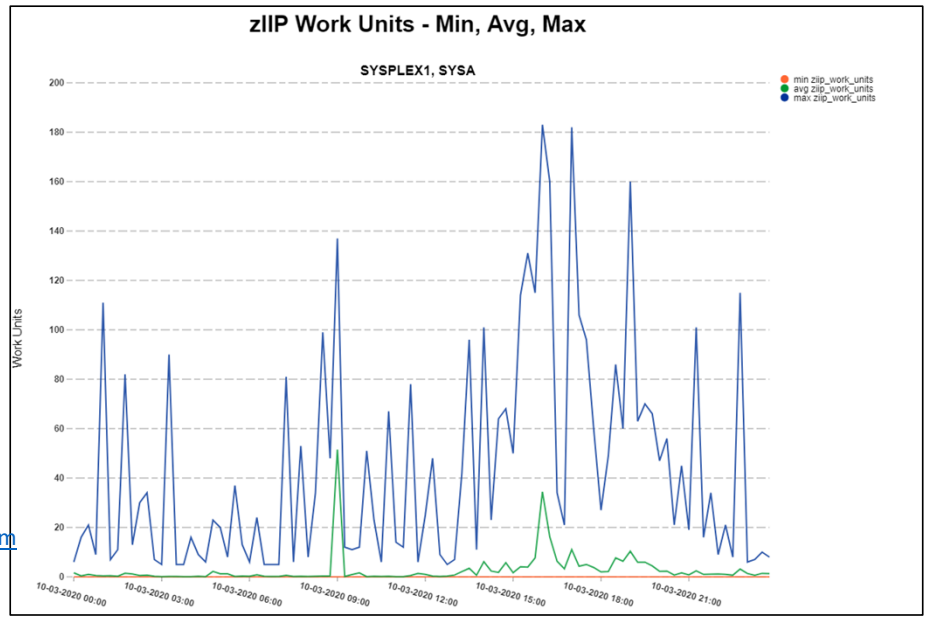


zIIP crossover

- Remember to keep an eye on zIIP cross over
- If excessive cross over during capping periods of time, then re-examine zIIP capacities, zIIP weights, zIIP latent demands, and zIIP controls
- In many cases it does not make sense to allow crossover when capping CP processor and there is enough zIIP capacity



www.pivotor.com
charts



Thank You!