Standard Measurements When Monitoring Transactions

(Back to Basics)

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 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:

Abstract

Standard Measurements when Monitoring Transactions

This is a back-to-basics presentation to discuss some of the most popular transaction measurements and formulas to use when you want to gain an understanding and further insights into your transactions' performance. Whether your transactions are batch, CICS, IMS, IDMS, or DB2, the basic measurements and formulas in this presentation will help ensure that you understand the services delivered to your customers.

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*
    - *Overseeing z/OS Performance Management with Your Outsourcer*
    - *z/OS Performance Management in an AI World*
    - *Understanding z/Architecture Processor Topologies*
    - *Processor Comparison Discussion*
    - *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

What is a Transaction?

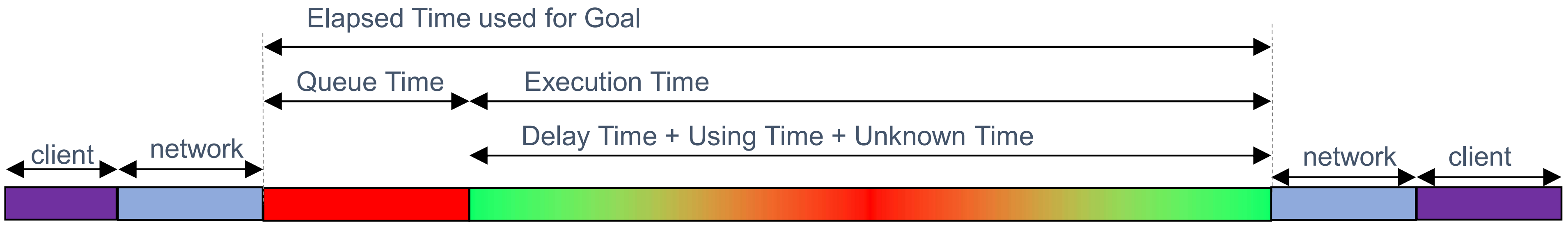
- When z/OS associates resources and performance characteristics to a unit-of-work, it does so to manage the unit-of-work called a transaction

- Transaction
  - A way of delineating a unit of work that is consuming service to complete a useful function
  - Has the performance characteristics and requirements
    - And is not necessarily the same as the address spaces processing the transaction

- Examples of transactions
  - APPC = Corresponds to work scheduled by ASCH
  - Batch = Corresponds to work scheduled by JES
  - Started Task = Generally the life of the address space
  - Web Server (IHS) = Web request (i.e. file serving, CGI, plug-in, etc)
  - TSO/E = Usually corresponds to a terminal interaction
  - DDF = Distributed SQL request
  - etc…

Basic Transaction Response Time Components

- Transaction timeline (i.e. response time) includes
  - Client time
    - Client-side code to either build request or evaluate / display response can be noticeable
    - If using browser interface, choice of browser can be significant
  - Network time
    - Faster networks offset by larger payloads for XML responses
  - Queue time
    - Time waiting to transaction to start execution
  - Execution time
    - Using Time: such as CP and zIIP CPU using time, I/O connect time
    - Delay Time: such as delayed for CP and zIIP CPUs, I/O delays, lock delays, other delays, etc.
    - Unknown Time: Portion of the response time, but we do not know if this is time using or delay.

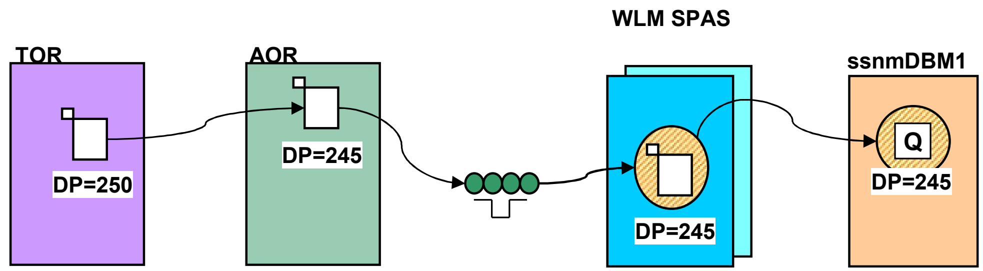


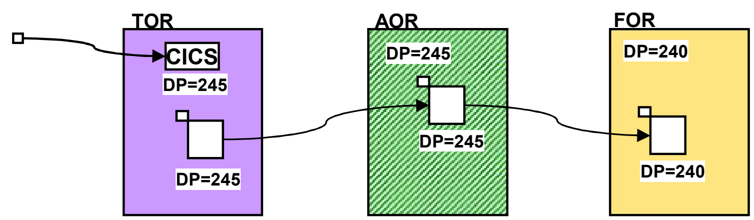
Importance of End to End

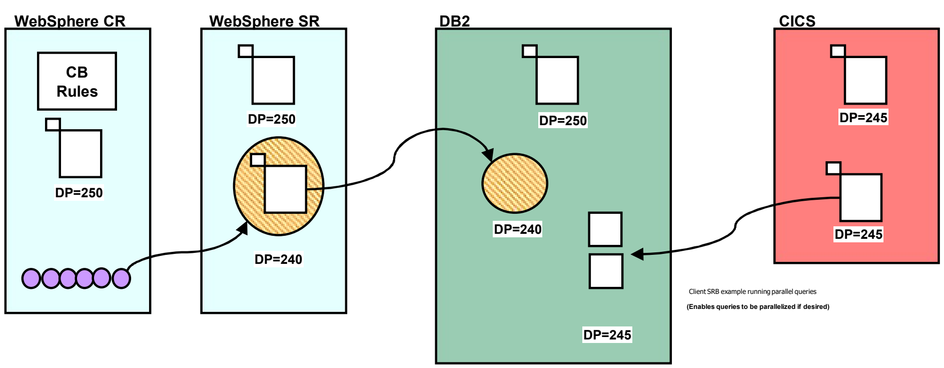
- Don’t ignore the components outside of the mainframe
  - Even though WLM can’t measure, much less manage that time, you must be concerned about it since this is what your users see
  - Network time
    - Faster networks offset by larger payloads for XML responses
  - Client time
    - Client-side code to either build request or evaluate / display response can be noticeable
    - If using browser interface, choice of browser can be significant
- Multiple transactions
  - Also remember that a single user interaction sometimes composed of multiple host transactions
  - “Screen scraper” type of applications often do this: 50 CICS transactions for a single user interaction
    - Reporting on these aggregate transactions possible with custom code, but can be challenging

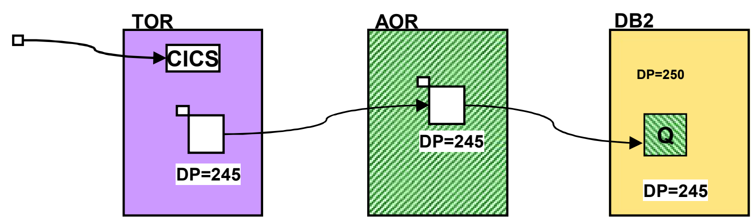
Learn about the application and transaction flows

- The applications and the transaction flows are at the heart of most performance concerns
  - Home grown or vendor product?
  - How are they deployed?
  - How do they work?
  - How are they measured?
  - Anything you can learn!



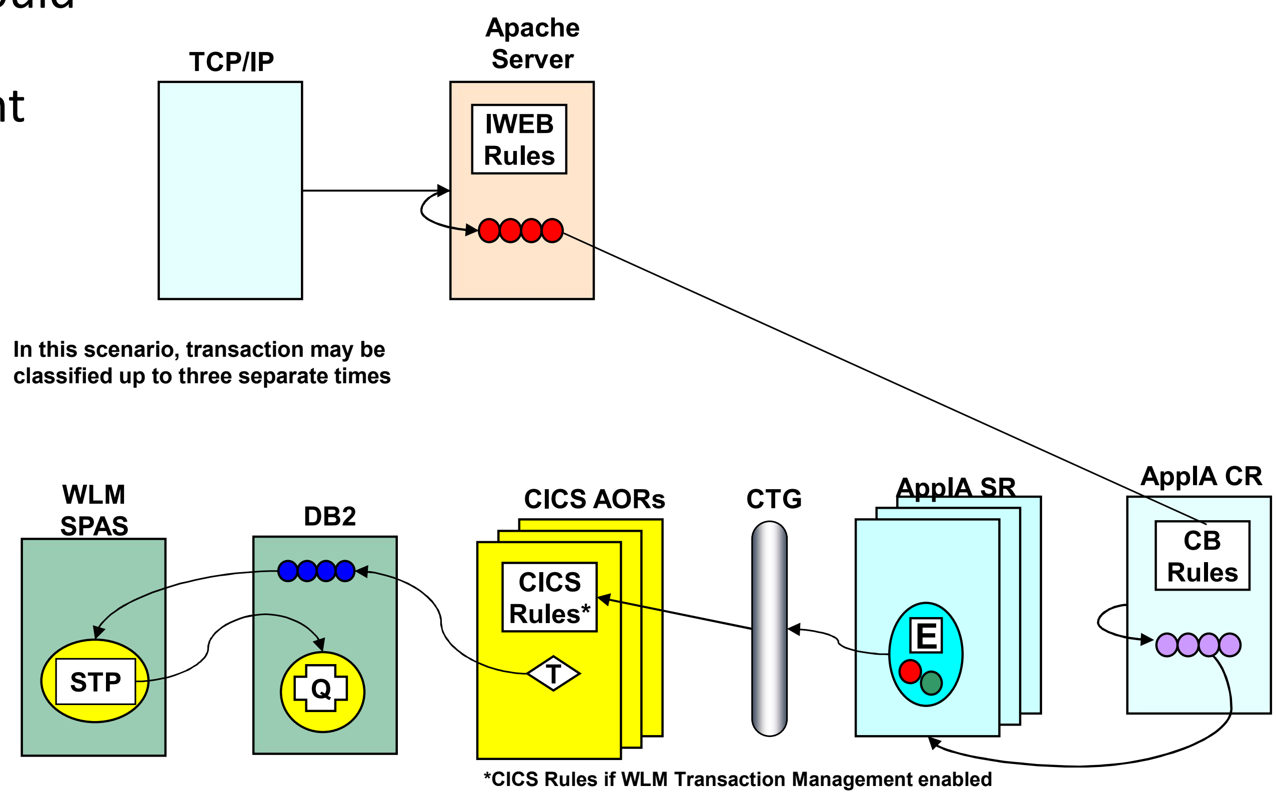






Learn about the application and transaction flows

- Remember the flows can be complex
  - Measurement and tuning could be dependent upon the architecture and deployment
  - What products are involved in the flow of the application?
  - Who suffers and benefits?
  - When does the application run and matter?



Basic Formulas for Any Transaction Analysis

- When doing transaction performance analysis:
  - There are ***many*** standard metrics and formulas that apply to all transaction types
  - Also, each transaction type (i.e. CICS, Batch, IMS, etc..) have their own unique measurements

- So whenever evaluating transaction performance, it is best to start with the basic standard measurements and metrics that apply to all transaction types

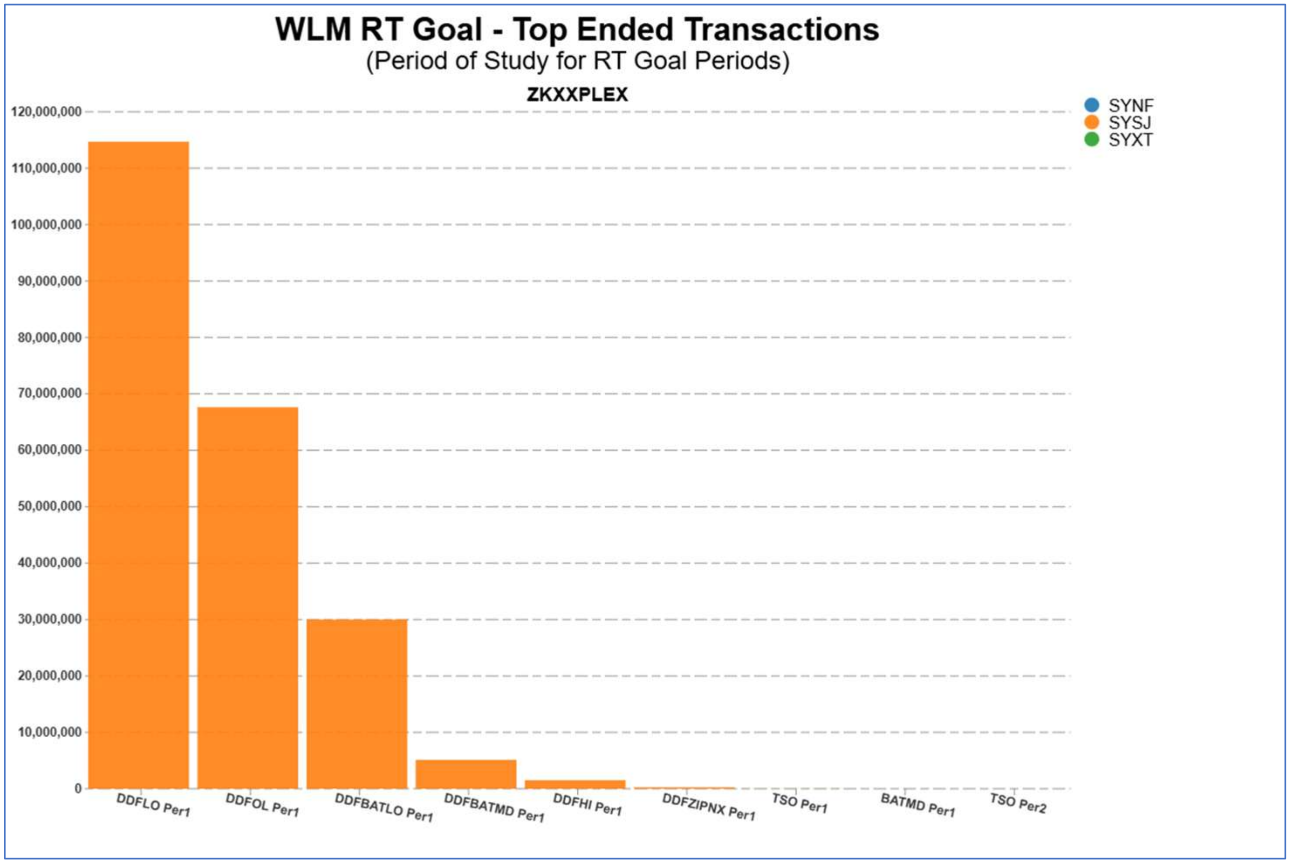
- As a starting point, some of the most basic transaction measurements include:
  - Ended transaction counts
  - Transaction response times and distributions
  - Response time components
    - CPU, I/O, Other
  - Transaction rates
    - External Throughput Rates (ETRs)
    - Internal Throughput Rates (ITRs)

Transaction Counts

Transactions Ended

- Ended: The number of transactions that completed execution within a specific time period
- Basic Formula: *Ended = (Number of Completions)*
- Common use is when examining the relative quantity of one transaction type, or ended relative to another transaction type
  - But need to know the definition of the transaction
- Examples of use:
  - Showing transaction load in terms of ended transactions
  - Provides indications of
    - High volume interactive transactions
    - Medium volume for workloads such as batch
    - Low volume for sporadic workloads, or workloads with very long transactions ending
- Drawbacks
  - Could be difficult to draw conclusions without knowledge of what is normal

Example Total Ended Transactions

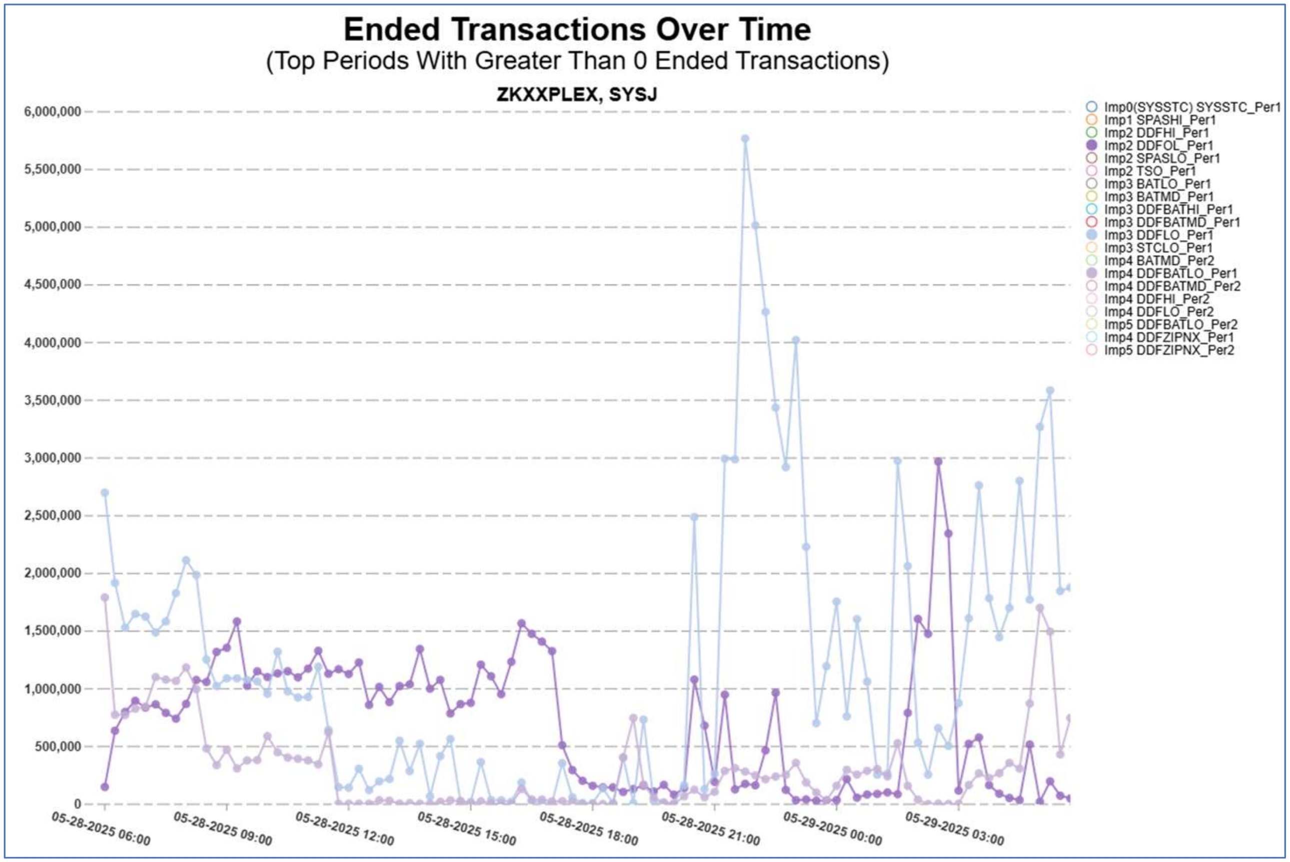


This chart shows the absolute number of ended transaction for a period of study for service classes that are assigned response time goals.

Such a chart is useful to help understand which workloads have the most ended transactions for a set window of time.

When doing an analysis and you do not know which workloads to look at first, sometime the workloads with the most transactions is a good starting point.

Example Ended Transactions Over Time



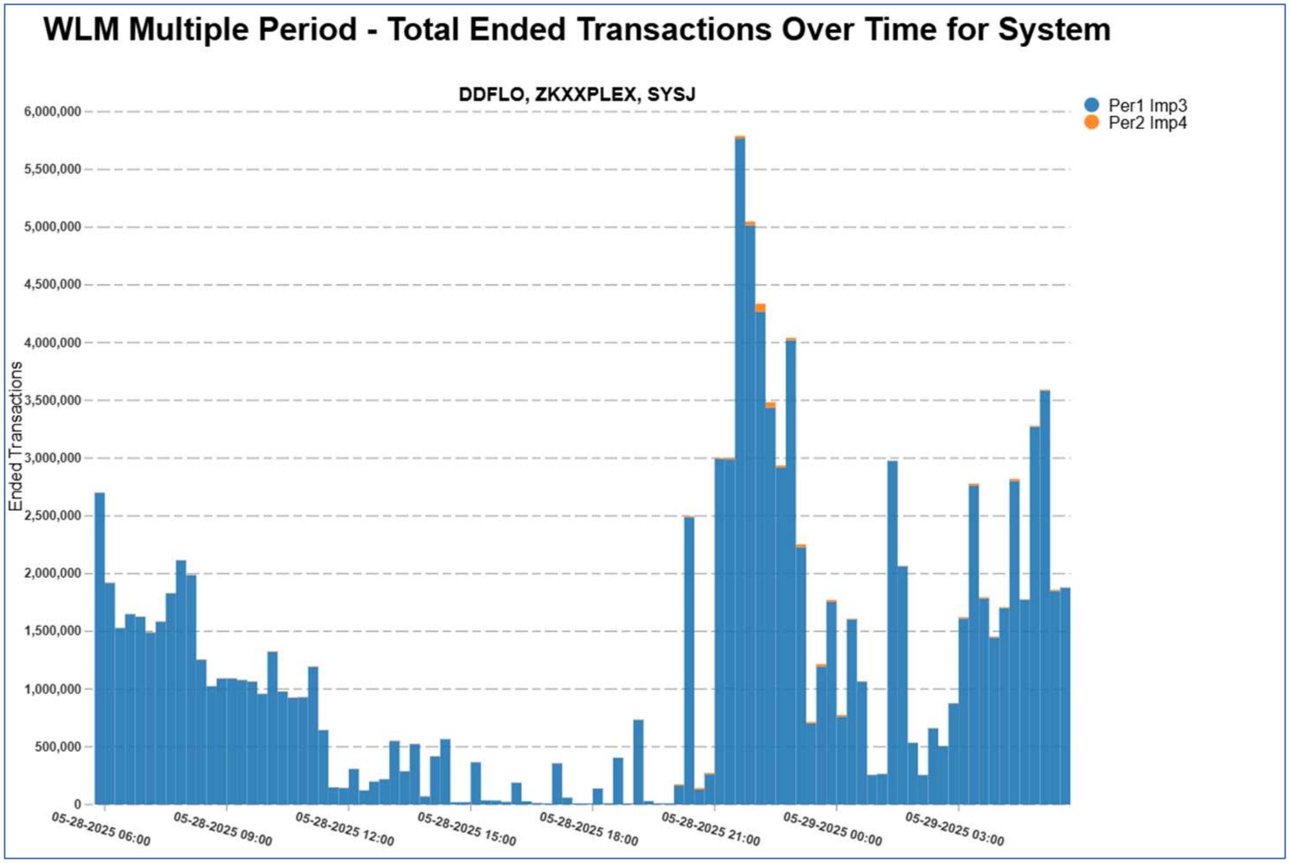
This chart shows the absolute number of ended transaction over time.

This chart is useful to help pin-point peak periods of time for when transactions are ending.

It is also useful to understand the number of ended transactions of one transaction type versus another.

When doing an analysis remember that looking at transaction peak periods is not the same as periods of time of peak CPU usage. It is always helpful to understand when users are executing transactions.

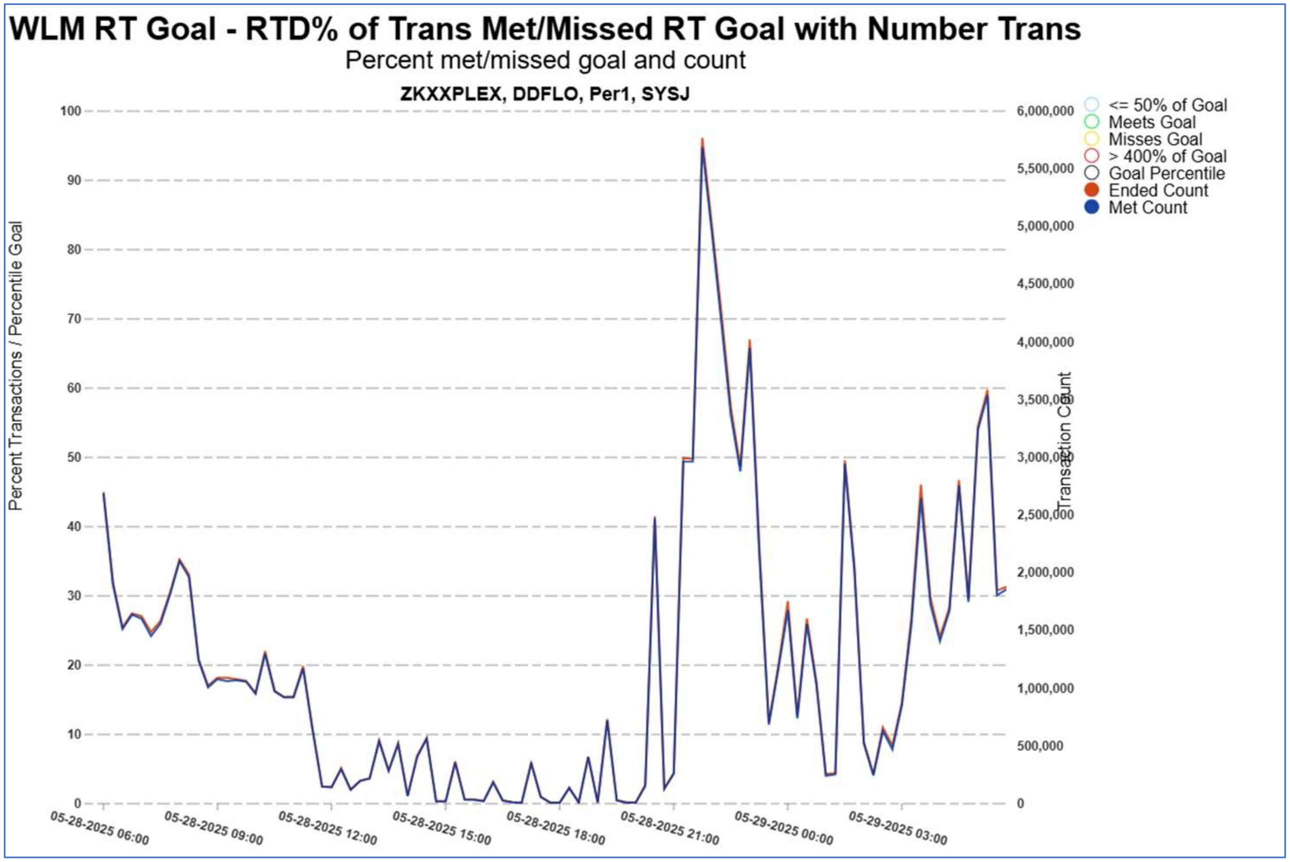
Example Ended Transactions Over Time by Service Class Period



This chart shows the absolute number of ended transaction over time, but for a specific service class, and the number of transactions that completed within each period of this multiple period service class

This chart is useful gain insight into when, and the quantity, of transactions ending in each period of the service class.

Example Ended Trans and Ended Meeting Goal



This chart is another variation of the number of ended transactions for a specific service class periods.

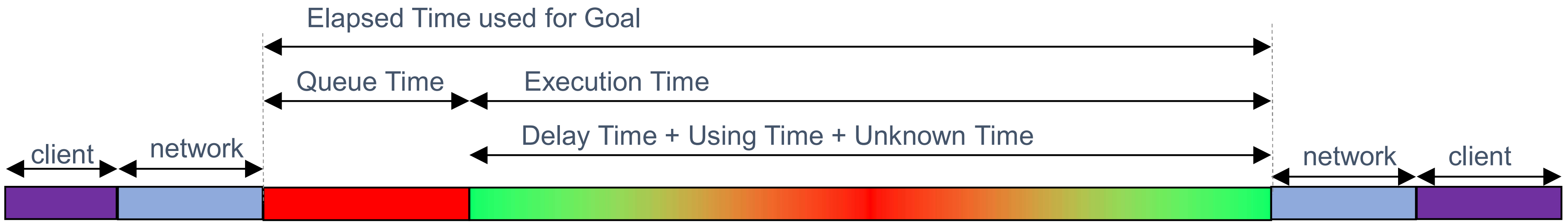
In this example, we are looking at

- The number of transactions that ended
- The number of transactions that met the goal value

Transaction Response Time Components

Basic Transaction Response Time Components

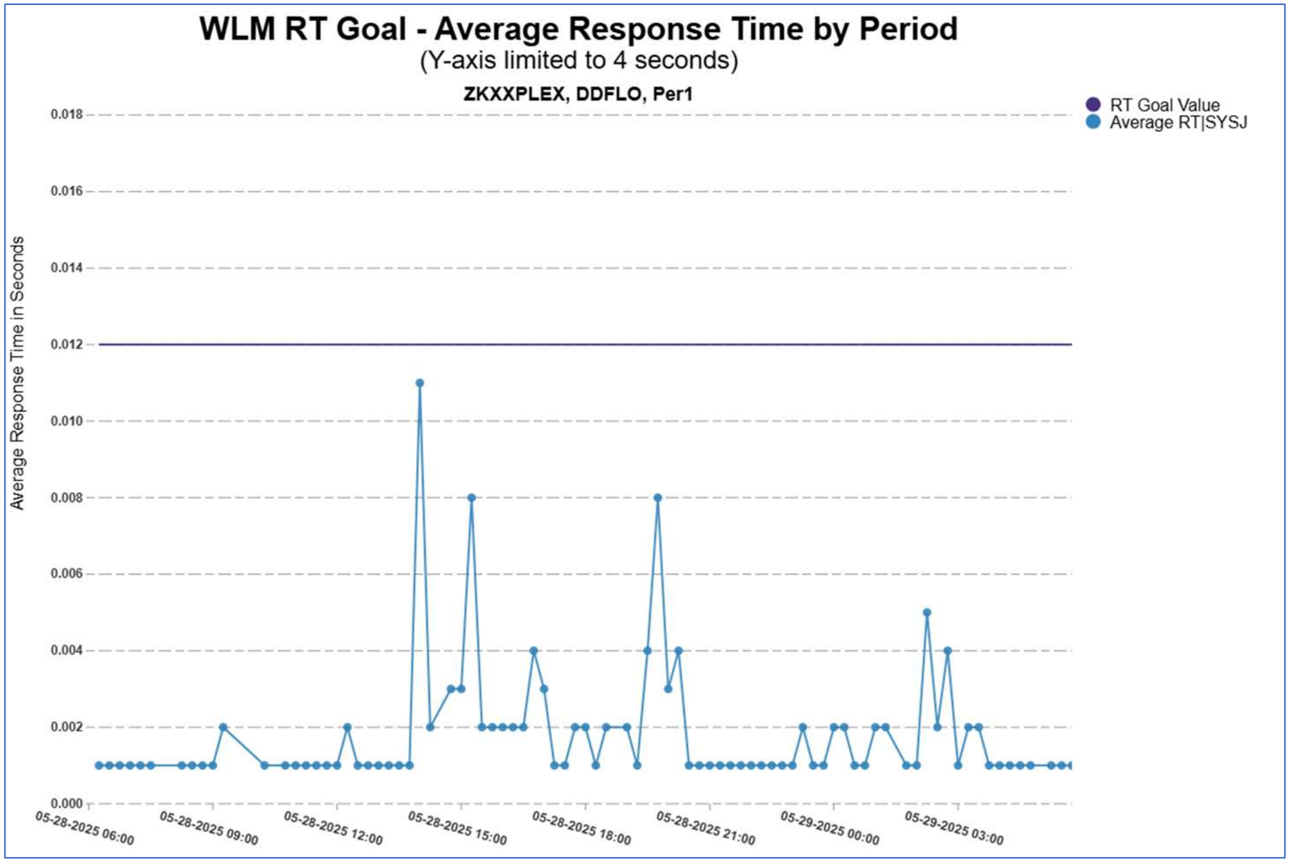
- z/OS measurements allow us to focus on the following timeline components
  - Queue time
    - Time waiting to transaction to start execution
  - Execution time
    - Using Time: such as CP and zIIP CPU using time, I/O connect time
    - Delay Time: such as delayed for CP and zIIP CPUs, I/O delays, lock delays, other delays, etc.
    - Unknown Time: Portion of the response time, but we do not know if this is time using or delay.



Transaction Response Times

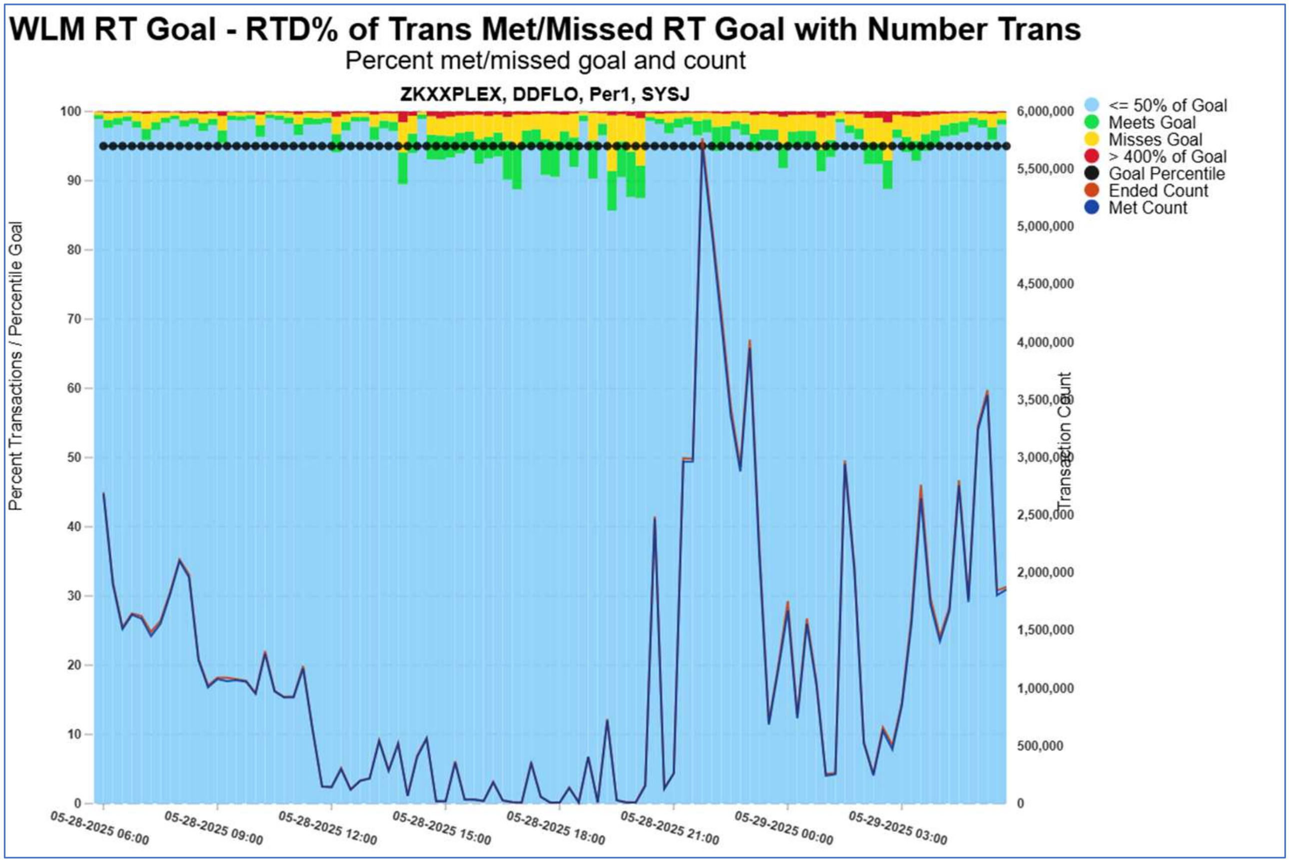
- Average Transaction Response Times :
  - The measure of central tendency of ended transactions
  - That is, given a specific transaction, or group of transactions, the average response time of that transaction
  - Basic Formula: *Average = (Sum All Transaction RTs) / (Number of Ended Trans)*
- Transaction Response Time Distributions
  - Percentage of transactions that completed within defined response time buckets
- Breakdown of transaction response times
  - Many workloads have measurements to understand the time spent within the transaction

Example : DDFLO Average Response Times for Transactions in a Service Class Period



This chart shows the average response time for the transactions of a specific service class period relative to the response time goal for this workload.

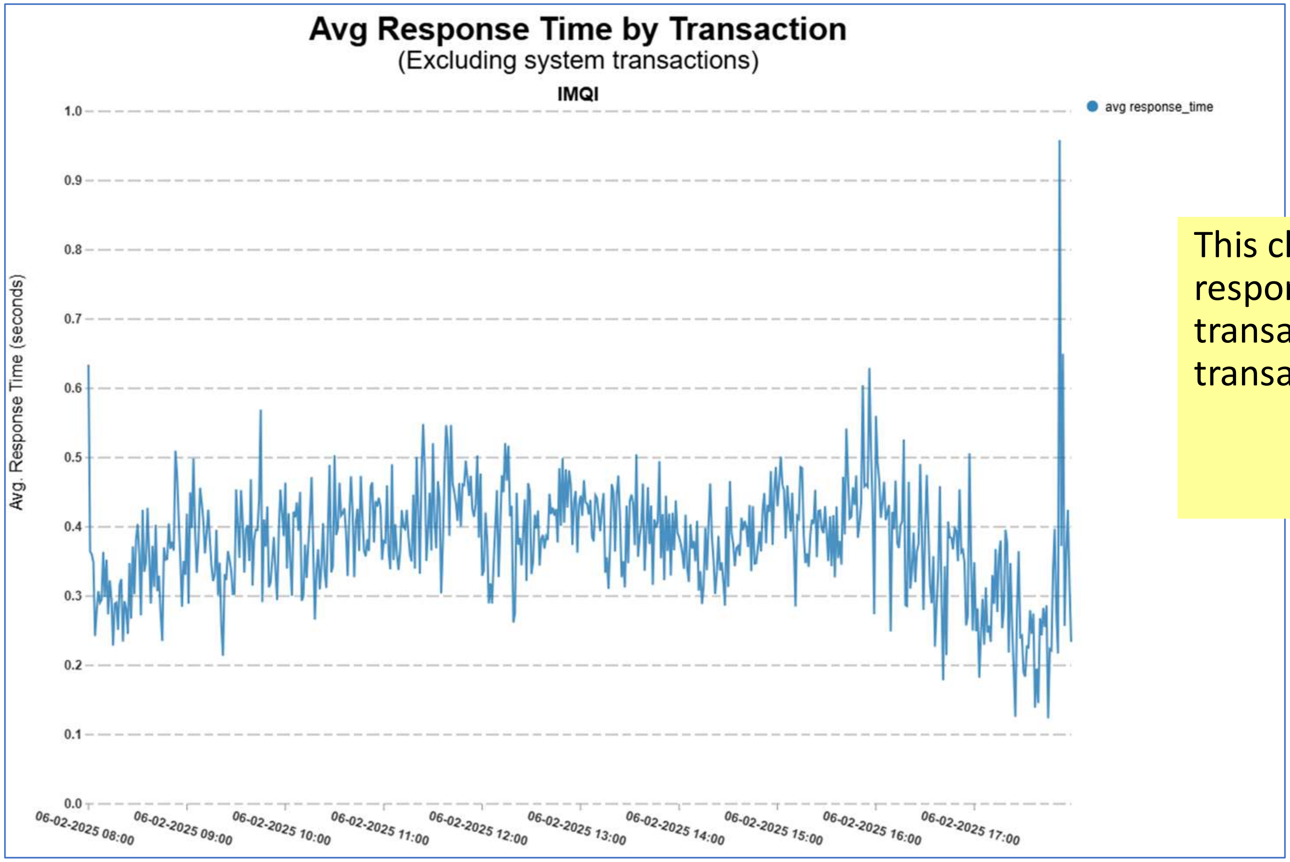
Example : DDFLO Average Response Time Distribution for Transactions in a Service Class Period



This chart shows the response time distribution for the ended transactions for a specific service class period.

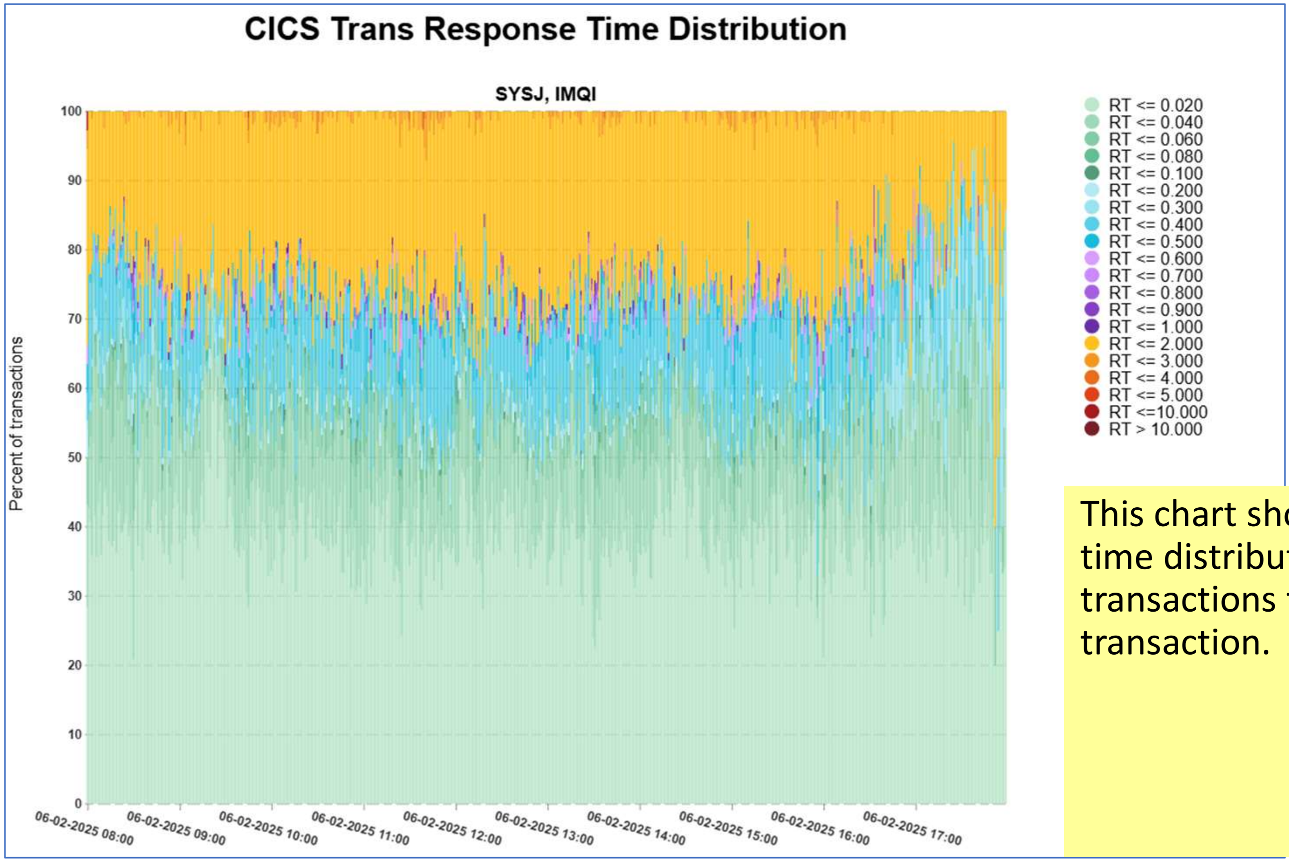
It is always helpful to show the distribution relative to the number of ended transactions so we can gain insights as to how transaction response times may be affected by the load as indicated by the number of ended transactions.

Example of Average Response Times for Transactions in a Service Class Period



This chart shows the average response time for the transactions of a specific CICS transaction

Example of Average Response Time Distribution for Transactions in a Service Class Period



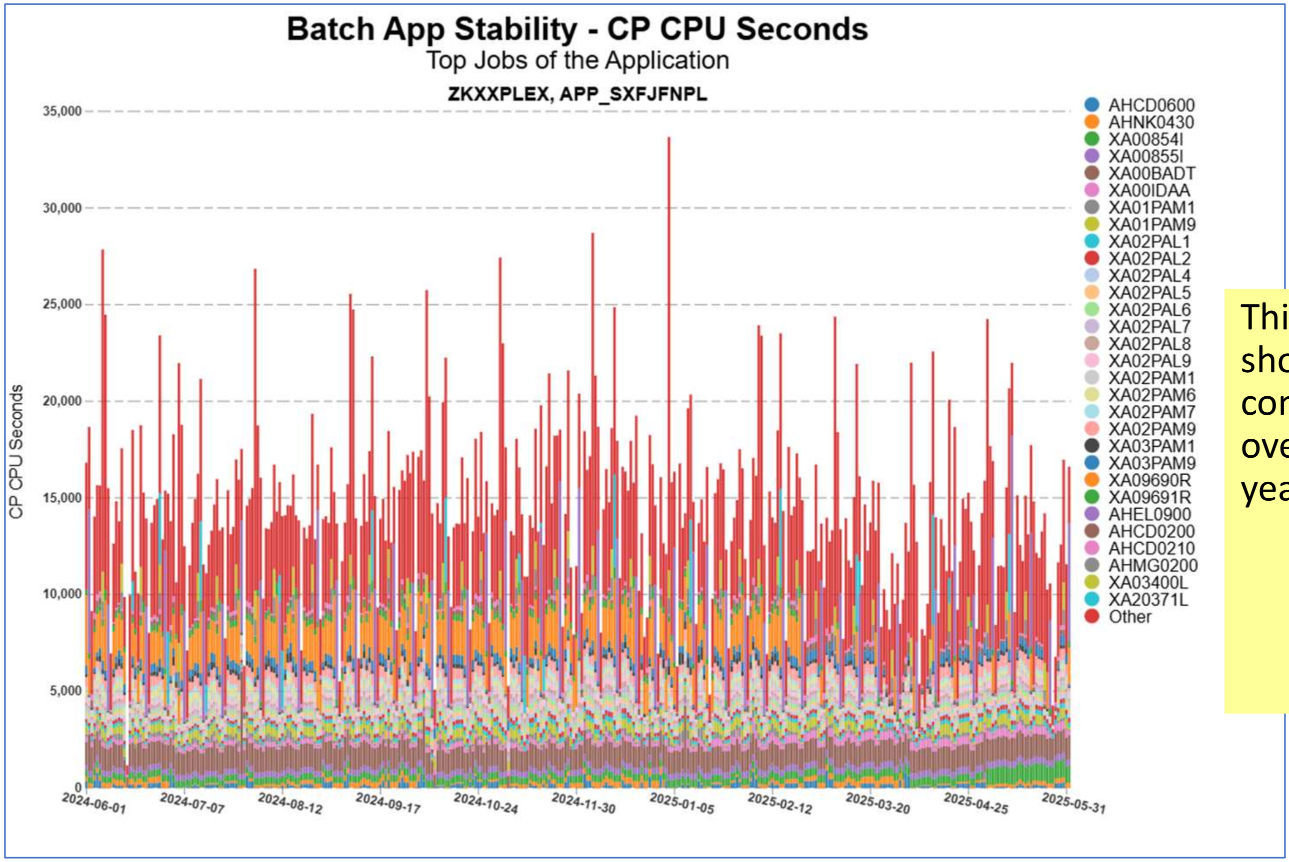
This chart shows the response time distribution for the ended transactions for a specific CICS transaction.

Example: Key Batch Job Time Components

init select_date	init select_time	Job_Name	Job_Number	System	Job Execution_Time	CPU_Time	TIME_ON_zIIP	TIME_zIIP_ON_CP	DASD_IO_Time	CONN_Sec	DISC_Sec	PEND_Sec	SSCH_Count	Initiator Wait	Ineligible wait	conversion wait	RS_Affinity Wait
5/28/2025	4:49:59	XA00CDMA	2065	SYSJ	10028.96	3009.58	0	0	14.24	13.88	0.04	0.32	13883	0.6892	0	0	0
5/28/2025	5:50:06	XA02PAM9	2309	SYSJ	785.81	202.85	0.06	0	1581.42	1575.84	0.05	5.52	1575844	1.0537	0	0	0
5/28/2025	6:00:00	VVDSBKUP	2327	SYSJ	96.56	4.03	0	0	48.08	44.6	2.13	1.35	44599	0.8038	0	0	0
5/28/2025	6:00:00	OPCOLTPM	2326	SYSJ	138.28	18.98	0.22	0.01	384.67	378.6	0.54	5.53	378597	1.622	0	0	0
5/28/2025	6:00:01	PVDP2025	2325	SYSJ	6.07	0.48	0	0	1.75	1.68	0.03	0.04	1679	1.9405	0	0	0
5/28/2025	6:00:01	PV002085	2330	SYSJ	10.24	0.83	0	0	1.42	1.37	0.01	0.03	1373	6.3887	0	0	0
5/28/2025	6:00:01	USR0BKUP	2352	SYSJ	16.39	0.4	0	0	13.13	13.04	0.07	0.02	13042	0.4936	0	0	0
5/28/2025	6:00:01	MCATBKUP	2350	SYSJ	18.97	0.72	0	0	15.95	15.87	0	0.07	15874	3.4417	0	0	0
5/28/2025	6:00:01	SYSSBKUP	2338	SYSJ	20.85	0.55	0	0	25.06	25.05	0	0.01	25046	0.2417	0	0	0
5/28/2025	6:00:01	IMSBKUPC	2335	SYSJ	26.3	0.76	0	0	36.65	36.63	0	0.02	36633	0.1372	0	0	0
5/28/2025	6:00:01	TST0BKUP	2333	SYSJ	27.68	1.64	0	0	37.39	37.26	0.08	0.05	37260	3.54	0	0	0
5/28/2025	6:00:01	PRD0BKUP	2332	SYSJ	174.04	4.5	0	0	227.14	226.8	0.15	0.19	226799	1.0342	0	0	0
5/28/2025	6:00:01	DB2PBKUP	2331	SYSJ	813.3	24.12	0	0	599.03	598.17	0.47	0.4	598168	1.0312	0	0	0
5/28/2025	6:00:01	DB2TBKUP	2329	SYSJ	940.86	17.52	0	0	868.78	868.3	0.11	0.37	868302	0.8561	0	0	0
5/28/2025	6:00:02	OPCOSR08	2355	SYSJ	0.52	0.03	0	0	0.05	0.05	0	0	48	0.6789	0	0	0
5/28/2025	6:00:02	OPCOSR22	2357	SYSJ	0.94	0.02	0	0	0.05	0.05	0	0	46	0.683	0	0	0
5/28/2025	6:00:02	NE050350	2356	SYSJ	1.79	0.04	0	0	0.14	0.13	0.01	0	126	2.1473	0	0	0
5/28/2025	6:00:03	SWMNMAIL	2372	SYSJ	1.03	0.02	0	0	0.03	0.02	0	0	25	0.2519	0	0	0
5/28/2025	6:00:03	XP00AEDX	2364	SYSJ	1.41	0.03	0	0	0.02	0.02	0	0	18	2.1432	0	0	0
5/28/2025	6:00:03	XP00AMFX	2362	SYSJ	1.5	0.03	0	0	0.02	0.02	0	0	21	2.1617	0	0	0
5/28/2025	6:00:03	FI100210	2361	SYSJ	2.9	0.09	0	0	0.4	0.39	0	0.01	389	2.1627	0	0	0
5/28/2025	6:00:03	NZ030202	2376	SYSJ	14.61	[illegible]	[illegible]	[illegible]	[illegible]	5.6	0.04	0.13	5605	0.5898	0	0	0
5/28/2025	6:00:03	NZ010202	2375	SYSJ	14.96	[illegible]	[illegible]	[illegible]	[illegible]	16.65	0.08	0.2	16653	0.5427	0	0	0
5/28/2025	6:00:03	NZ020801	2371	SYSJ	131.08	[illegible]	[illegible]	[illegible]	[illegible]	140.76	0.12	1.15	140761	0.1362	0	0	0
5/28/2025	6:00:04	NZ020202	2377	SYSJ	13.45	[illegible]	[illegible]	[illegible]	[illegible]	5.27	0.05	0.12	5270	3.5727	0	0	0

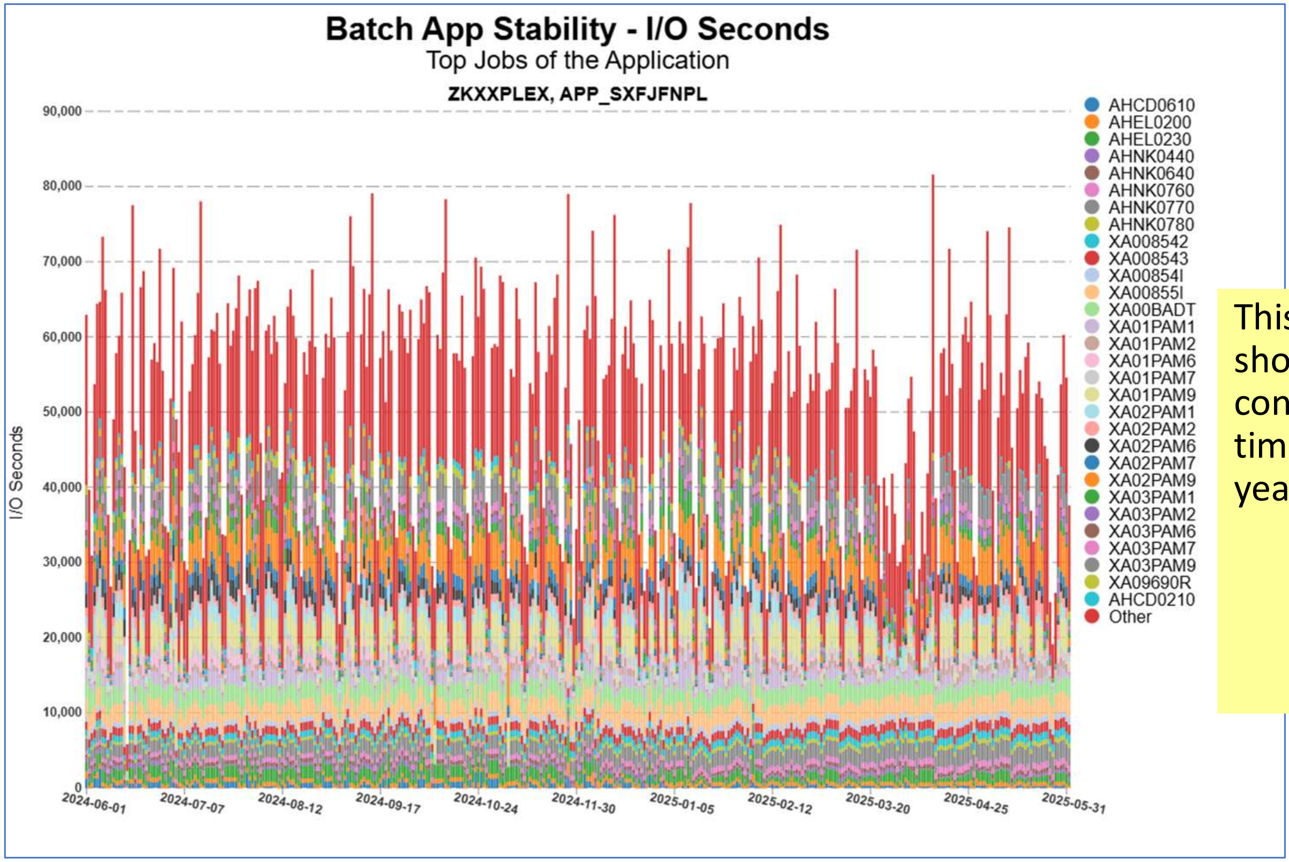
This table reports some key measurements for the batch workload. The same measurements are available at the job step level

Example: Batch CPU Time for One Year



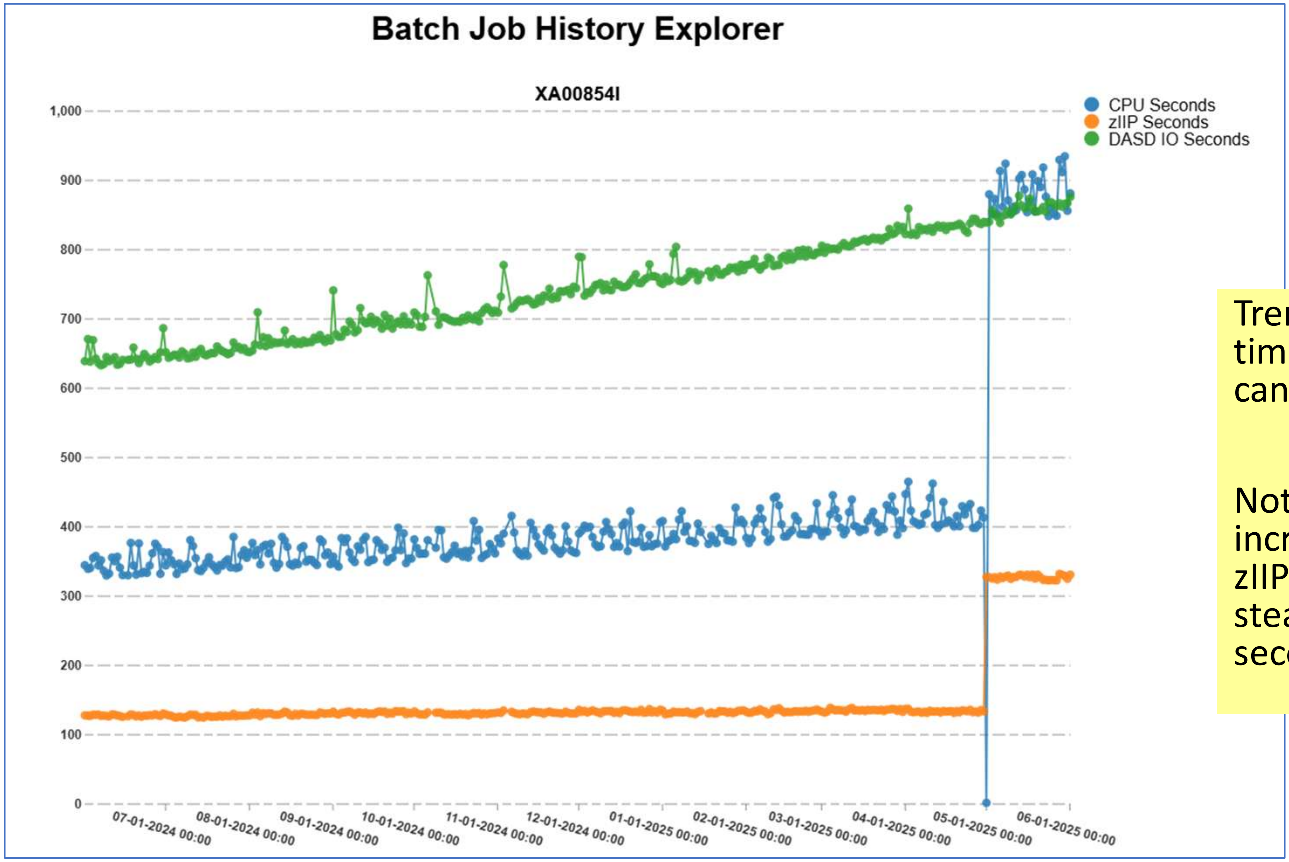
This stability report shows the top batch jobs consuming the most CPU over a period of one year.

Batch: I/O Time for One Year



This stability report shows the top batch jobs consuming the most I/O time over a period of one year.

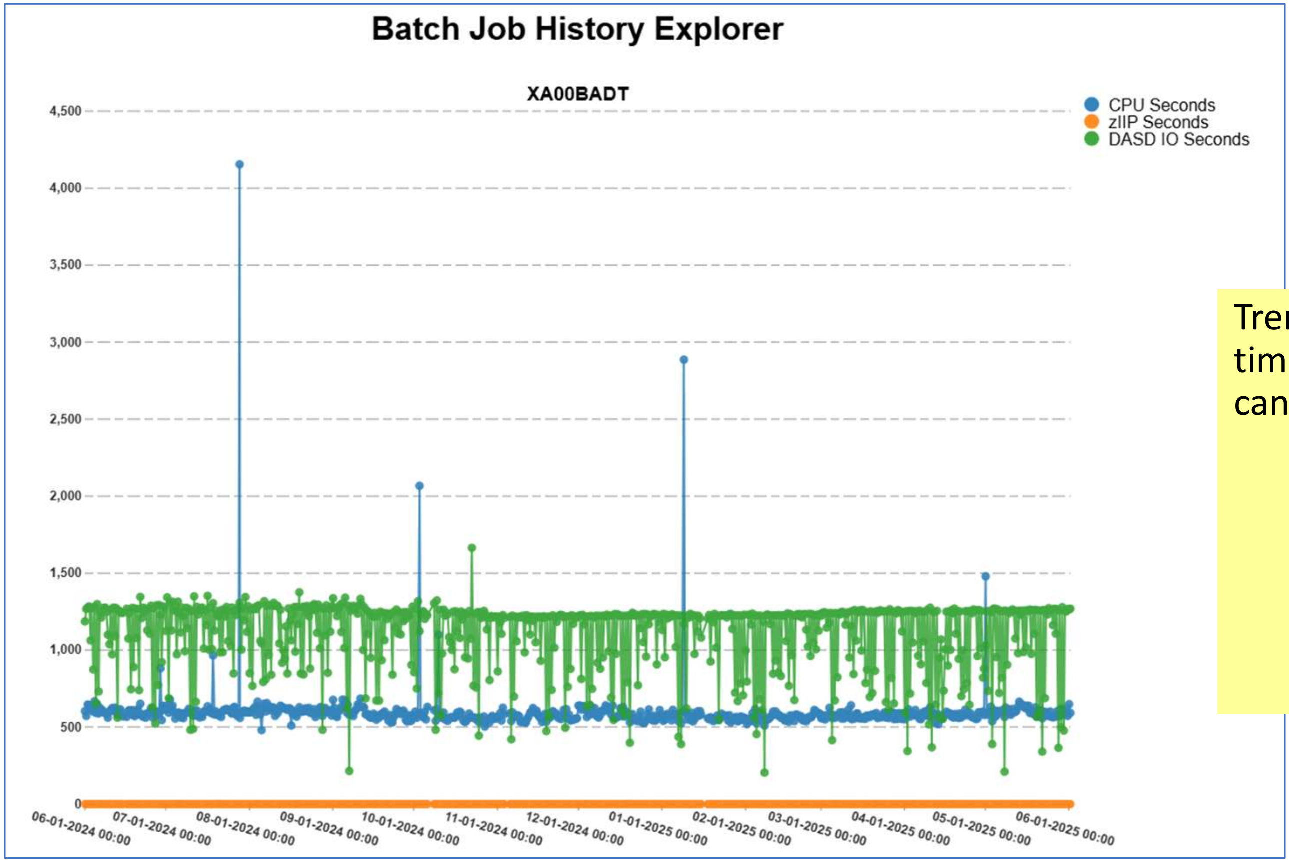
Example: Batch Job XA00854I for One Year



Trending batch response time components over can be insightful.

Notice the sudden increase in the CPU and zIIP time values, and the steady increase in the I/O seconds

Example: Batch Job XA00BADT for One Year



Trending batch response time components over can be insightful

Transaction Throughput Rates

Throughput Rates

- Rate: A measure of frequency against some other measure
  - Many times, the other measure is time (for per second rates)

- Basic Formula Throughput *Rate = (Ended) / (Time)*

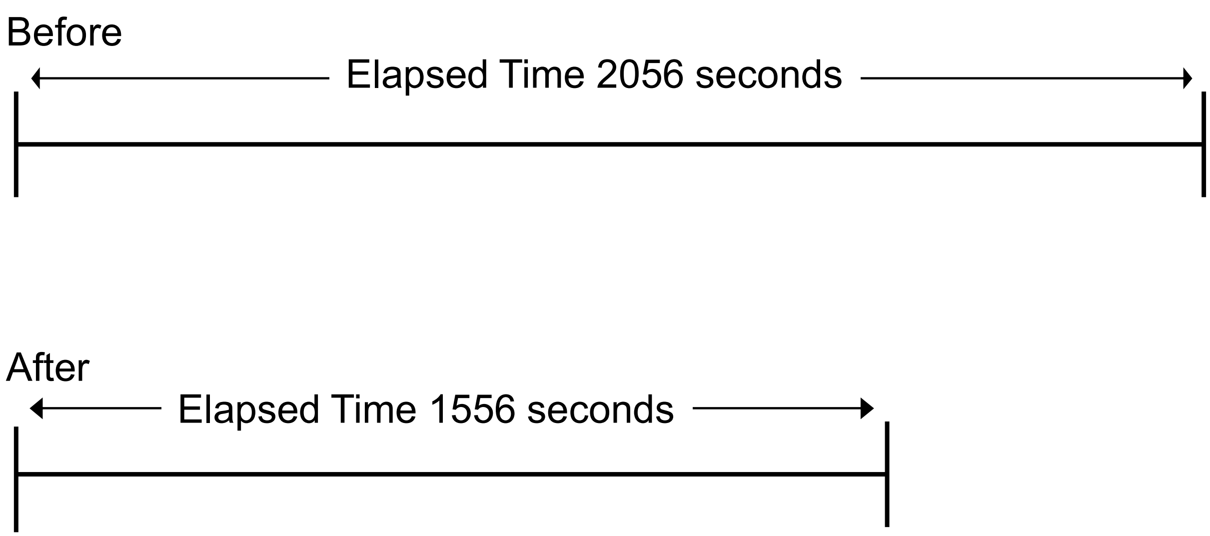
- Common use to understand the frequency of a unit of work

- Examples of use:
  - Showing the rate of work
    - Example: transactions per second, jobs per shift, etc.

  - Showing the rate of the usage of a resource
    - Example: I/Os per second, service units per second, etc.

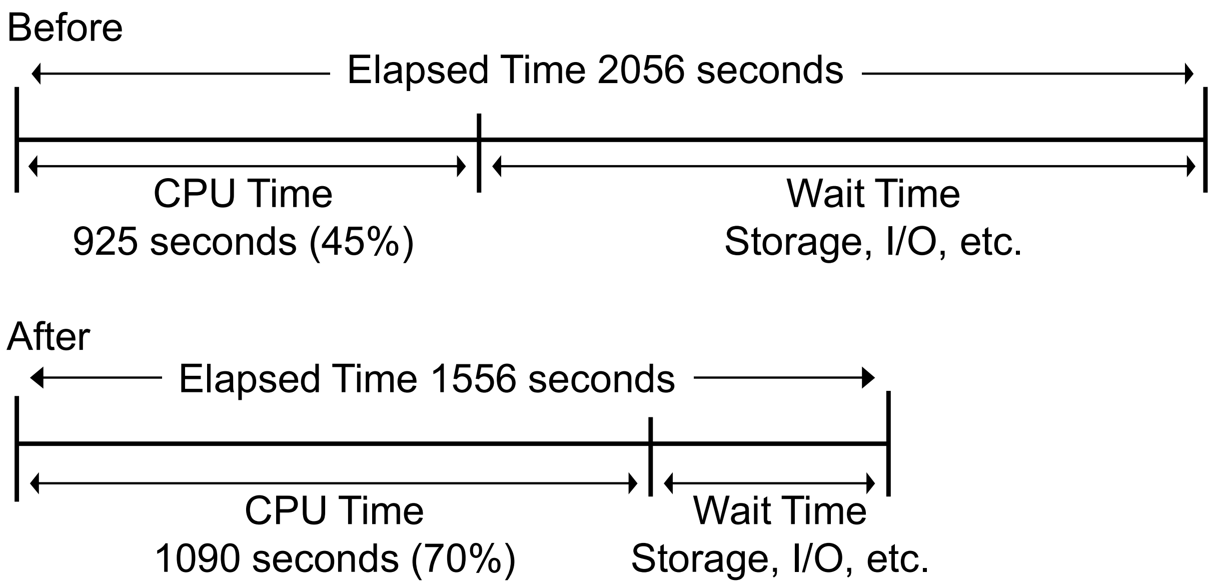
Why Are We Interested In Throughput Rates?

- Using the below figure, answer the following two questions:
  - Which is the better system to run the workload?
  - Which system has the better processor to run the workload?
  - Note: Assume that 1,000 transactions are run in the elapsed time



Understanding Workload Throughput

- Which is the better system to run the workload?
  - *For improved transaction throughput – The After System*
- Which system has the better processor to run the workload?
  - *For less usage of the CPU = The Before System*
- Need to consider ETRs and ITRs



External Throughput Rate - ETR

- A measure that focuses on system capacity
  - A measure of throughput as defined as by the number of transactions per wall clock second
  - Helps to answer the question"
    - 'Which system better processes a workload' for transaction throughput

$$\text{ETR} = \frac{\text{Units of Work}}{\text{Elapsed Time}} = \frac{\text{Units of Work}}{\text{Second}}$$

- Inverse of ETR formula is 'Average transaction response time'

$$\text{Avg Trans Response Time} = \frac{\text{Elapsed Time}}{\text{Units of Work}}$$

- Previous Example:
  - Before ETR = 1000 trans /2056 elapsed sec = 0.486 transactions / second
  - After ETR = 1000 trans /1556 elapsed sec = 0.643 transactions / second

Internal Throughput Rate - ITR

- A measure that focuses on processor capacity
  - A measure of throughput as defined as by the number of transactions per CPU second
  - Since this is a busy time measurement, it helps to answer the question:
    - 'Which processor better processes a workload?'
    - Useful when comparing processors

$$\text{ITR} = \frac{\text{Units of Work}}{\text{Processor Busy Time}} = \frac{\text{Units of Work}}{\text{CPU Second}} = \frac{\text{ETR}}{\text{Utilization}}$$

  - Processor Time should include system overhead
    - On n-way machine, should include busy time of all processors
    - Attempts to factor in only processor as the performance factor

- Previous Example:
  - Before ITR = 1000 trans / 925 CPU sec = 1.081 transactions / CPU second
  - After ITR = 1000 trans / 1090 CPU sec = 0.917 transactions / CPU second

ETR vs ITR

- ETR - used to characterize **system** capacity
  - Since it is an elapsed time measure
    - It encompasses the performance of the processor, the operating system, and all the external resources
    - e.g. disk, cache, storage, network, operations, etc
    - All resources are potential inhibitors
  - The highest ETR achieved is the processing capability of the system
- ITR - used to characterize **processor** capacity
  - Since only based on CPU time
    - It encompasses the performance of just the processor
    - When measured, all external resources must be adequate
    - Thus, whenever two processors are compared, they must be measured at the same utilization
  - Could be used to evaluate the efficiency of a workloads use of CPU

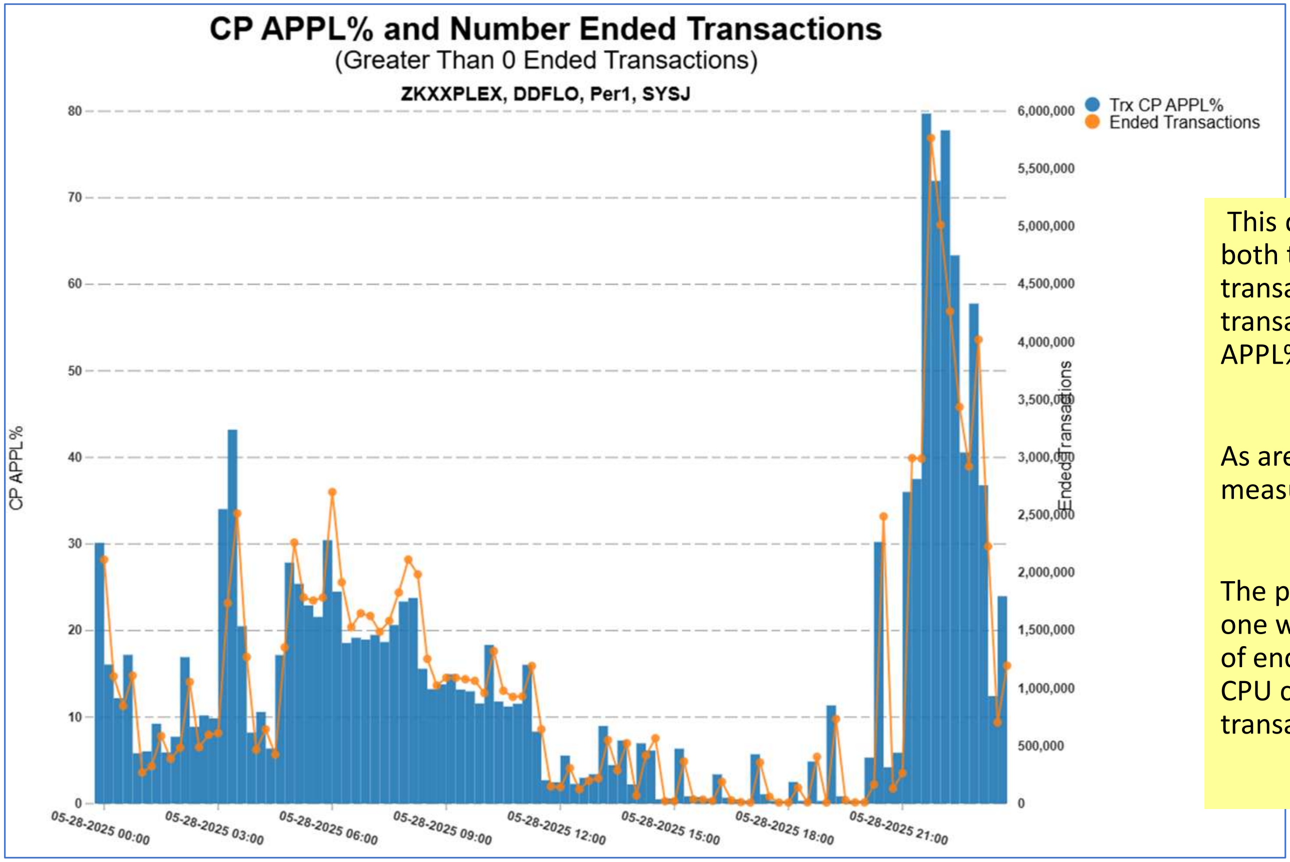
Example of Using ITR/ETR Relationship

- Since $ITR = \dfrac{ETR}{CPUBusy\,\%}$
  - It is simple to derive an ITR from a standard ETR without having to run a special test where the processor is the only limiting factor
    - Useful when evaluating an application, or system change
    - Below example: Was a 50% increase in CPU% and a 22% decrease in efficiency of the CPU by the workload worth an 18% improvement in throughput?

	Before Change	After Change	% Change
Elapsed Seconds	900	900 ↔	
Processor Seconds	540	810 ↑	
Transaction Count	1100	1300 ↑	18%
CPU Utilization (%)	60%	90% ↑	50%
ETR	1.22	1.44 ↑	18%
ITR	2.04	1.60 ↓	-22%

Installation must decide the value of the change.

Example 1: DDFLO Transactions

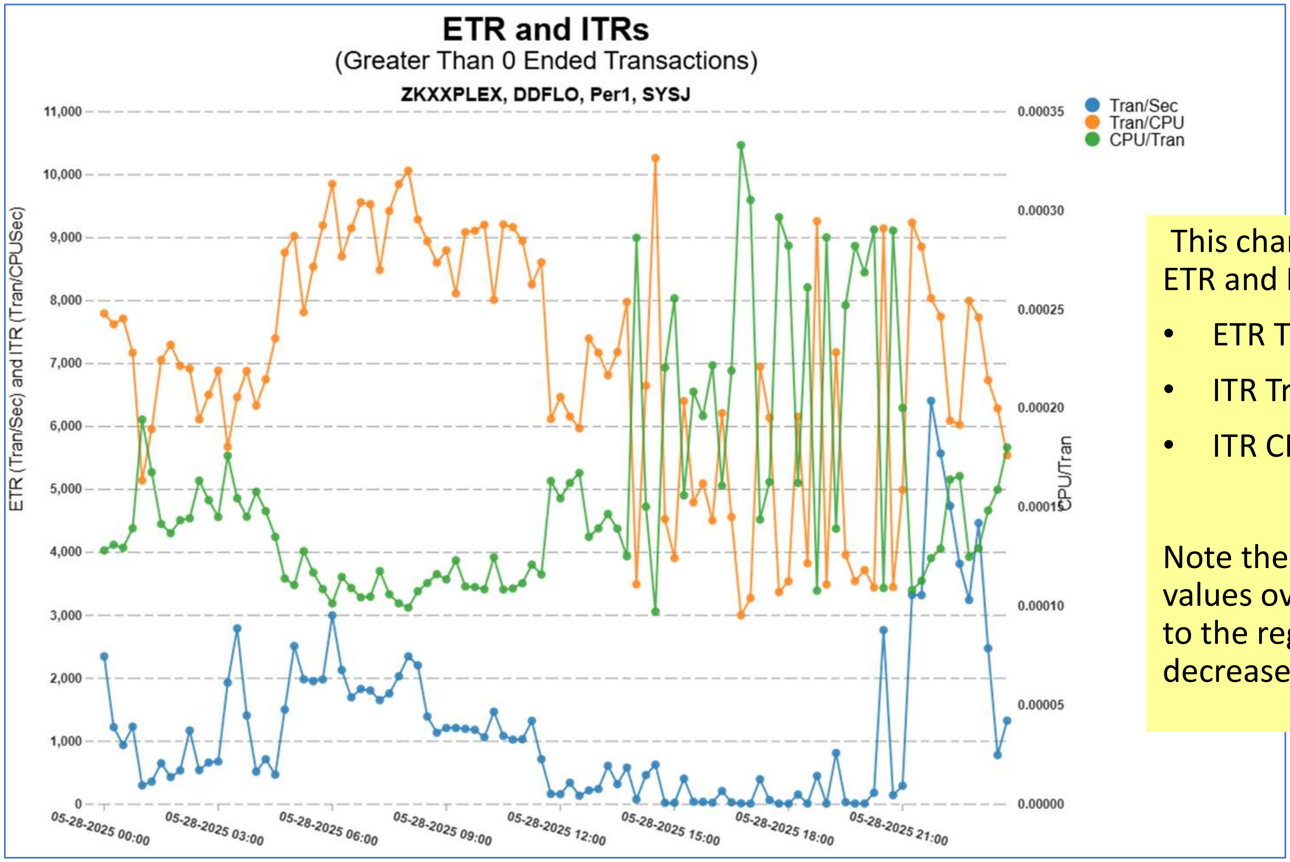


This double y-axis chart shows both the number of ended transactions for DDFLO transactions, as well as the CP APPL% of these transactions.

As are reminder, APPL% is a measure as a percentage of 1 CPU.

The point is, this chart shows for one week compares the number of ended transactions against the CPU consumption for those transactions

Example 1: DDFLO Transactions

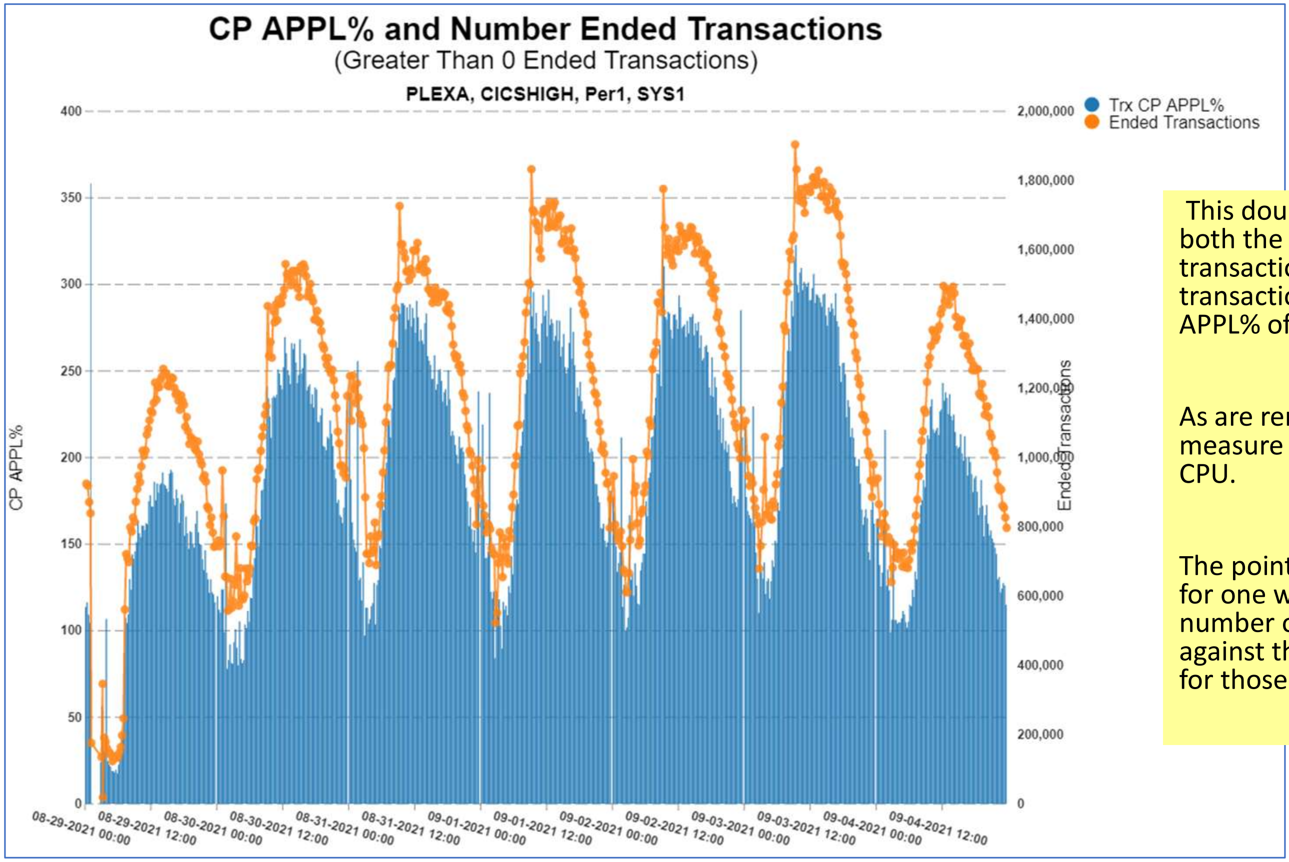


This chart shows the three ETR and ITR values:

- ETR Transactions/Sec
- ITR Transactions/CPU Sec
- ITR CPU / Transaction

Note the stability of the ITR values over the week relative to the regular increases and decreases in the ETR.

Example 2: CICSHIGH Transactions

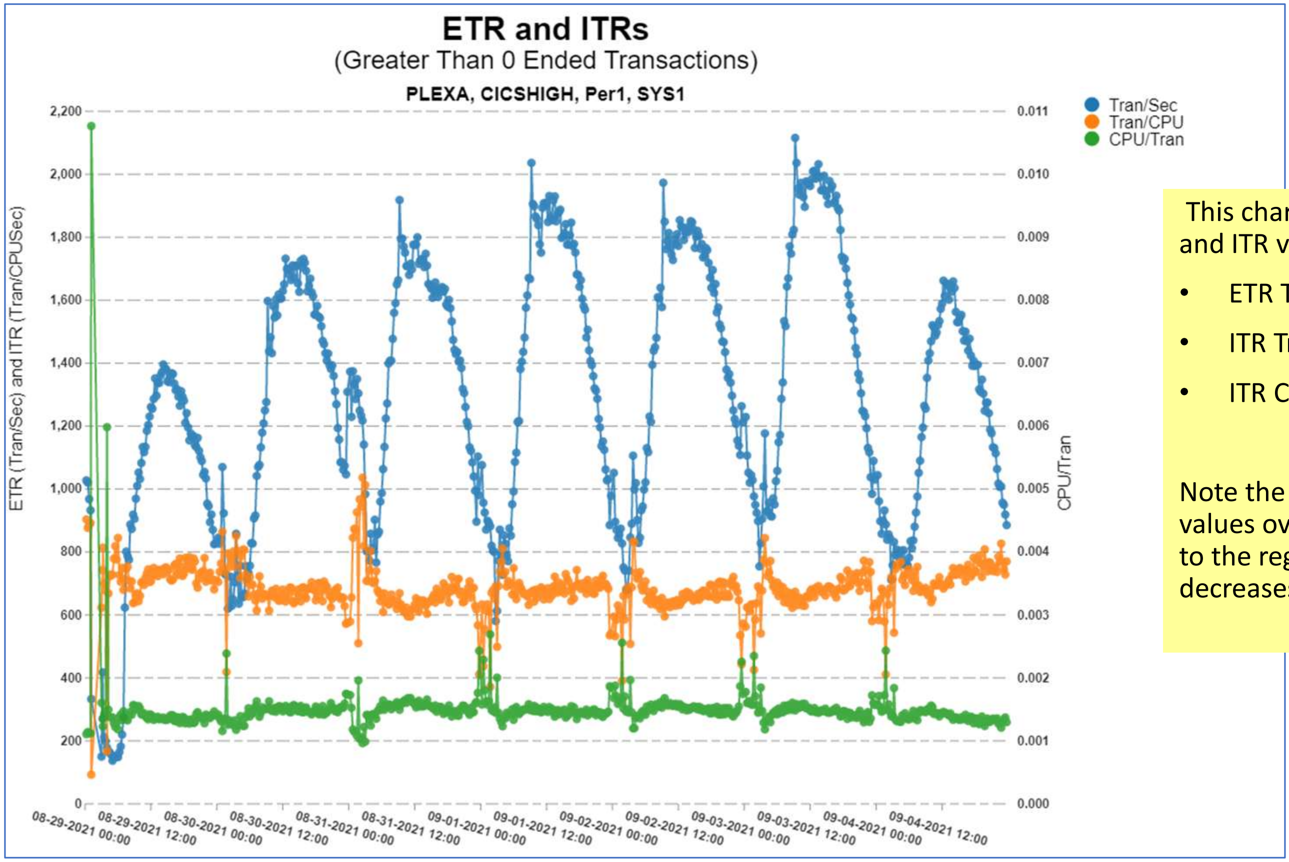


This double y-axis chart shows both the number of ended transactions for CICSHIGH transactions, as well as the CP APPL% of these transactions.

As are reminder, APPL% is a measure as a percentage of 1 CPU.

The point is, this chart shows for one week compares the number of ended transactions against the CPU consumption for those transactions

Example 2: CICSHIGH Transactions

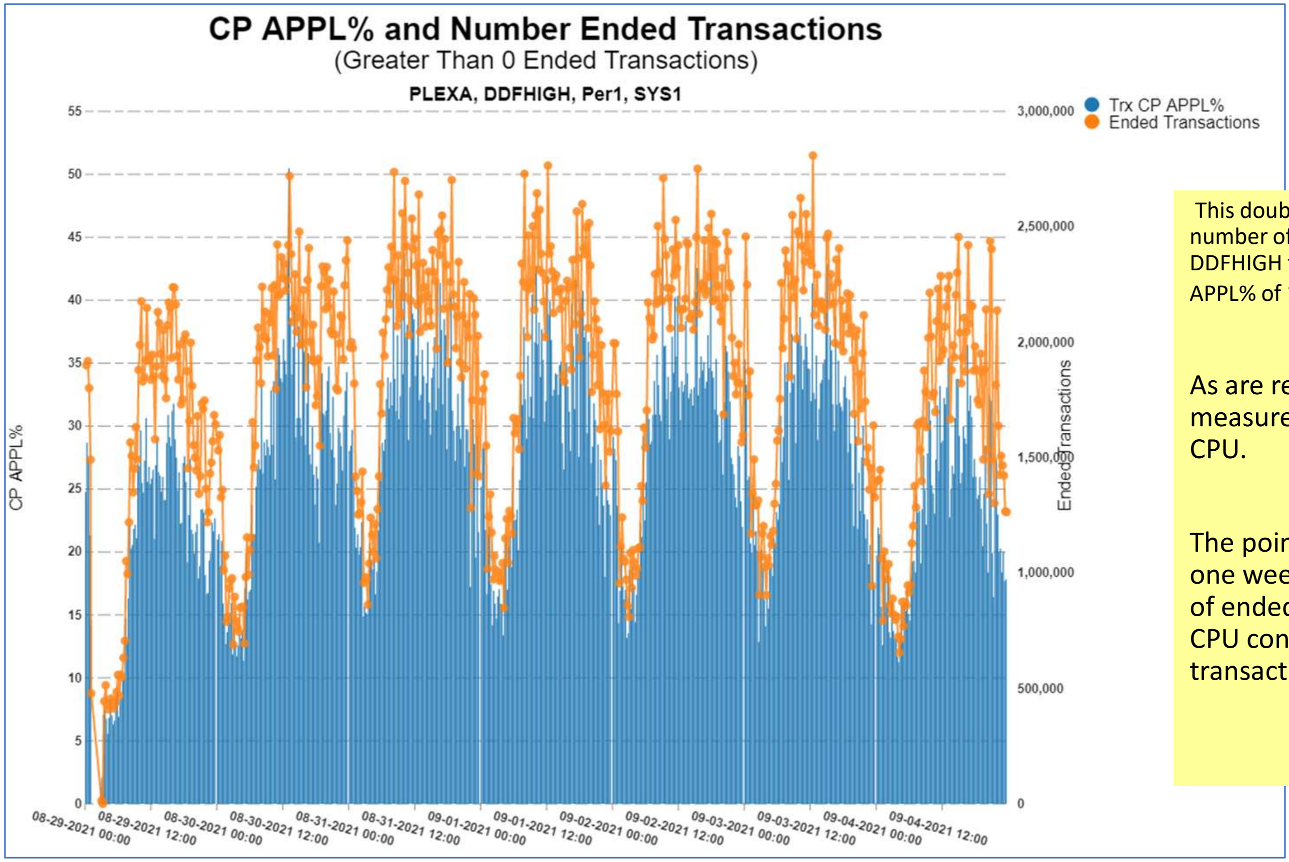


This chart shows the three ETR and ITR values:

- ETR Transactions/Sec
- ITR Transactions/CPU Sec
- ITR CPU / Transaction

Note the stability of the ITR values over the week relative to the regular increases and decreases in the ETR.

Example 3: DDFHIGH Transactions

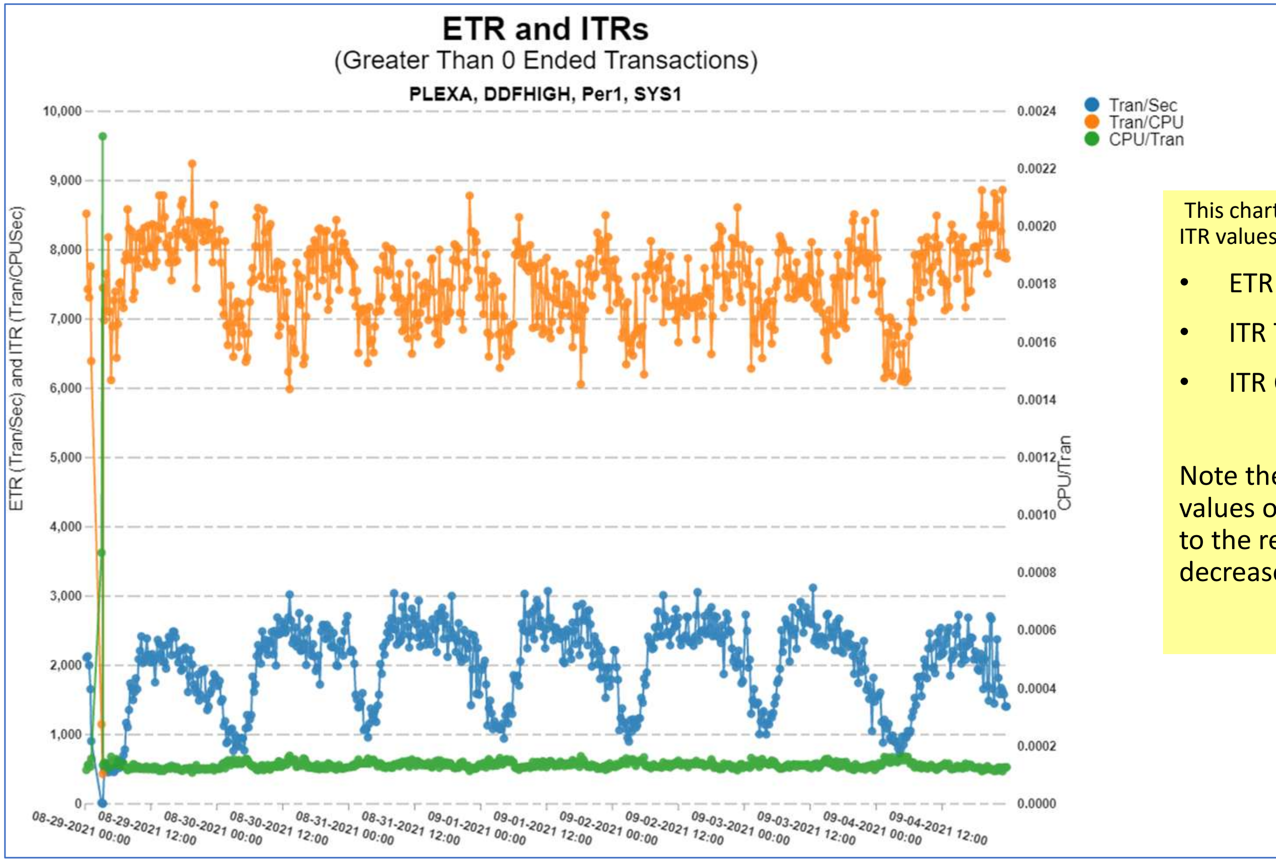


This double y-axis chart shows both the number of ended transactions for DDFHIGH transactions, as well as the CP APPL% of these transactions.

As are reminder, APPL% is a measure as a percentage of 1 CPU.

The point is, this chart shows for one week compares the number of ended transactions against the CPU consumption for those transactions

Example 3: DDFHIGH Transactions



This chart shows the three ETR and ITR values:

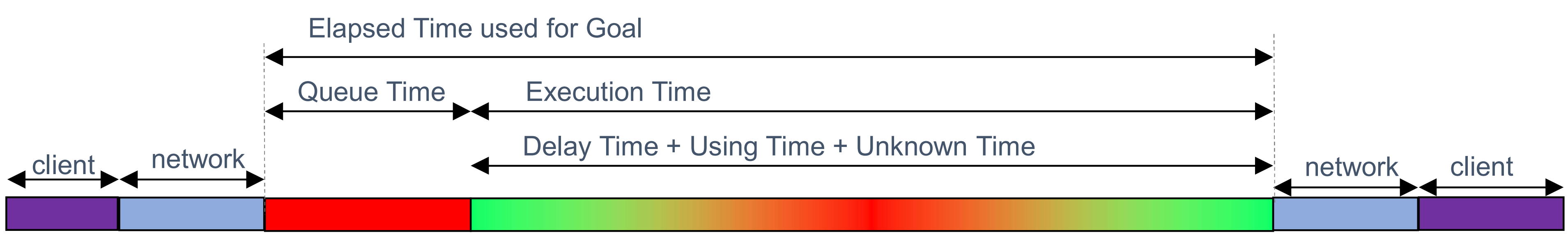
- ETR Transactions/Sec
- ITR Transactions/CPU Sec
- ITR CPU / Transaction

Note the stability of the ITR values over the week relative to the regular increases and decreases in the ETR.

Summary

- Transaction timeline (i.e. response time) includes
  - The z/OS environment has lots and lots and lots of measurements to help us understand a workload's transaction performance and timeline

- This presentation just scratched the service by showing some of the basic measurements all analysts should look at on a regular basis



Thank You!