More Details
Exploring z/OS SMF 30 Address Space CPU Measurements

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:

© 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

Exploring z/OS SMF 30 Address Space CPU Measurements

Address space activity measurements are recorded in the SMF 30 record. There are so many measurements in the SMF 30 records that we could probably do an entire webinar series on just the SMF 30 record.

During this webinar, Peter Enrico will both introduce the SMF 30 record and then explore the key processor measurements you may be interested in. Also discussed will be some ways you may want to examine and use these address space CPU measurements.

EPS: We do z/OS performance…

- We are z/OS performance!

- Pivotor
  - Performance reporting and analysis of your z/OS measurements
  - Example: SMF, DCOLLECT, other, etc.
  - Not just reporting, but cost-effective analysis-based reporting based on our expertise

- Performance Educational Workshops (while analyzing your own data)
  - Essential z/OS Performance Tuning
  - Parallel Sysplex and z/OS Performance Tuning
  - WLM Performance and Re-evaluating Goals

- Performance War Rooms
  - Concentrated, highly productive group discussions and analysis

- MSU reductions
  - Application and MSU reduction

Enterprise Performance Strategies, Inc.
©

Like what you see?

- Free z/OS Performance Educational webinars!
  - The titles for our Winter 2022 webinars are as follows:
    - ✓ *SMF Recording Options to Improve Your Performance Analysis*
    - ✓ *SMF 98 and 99: Pinpointing Transient Performance Problems*
    - ✓ *Exploring z/OS Processor Storage Measurements*
    - ✓ *Exploring PR/SM Physical and Logical CPU Utilization Measurements*
    - ✓ *Exploring Locking and Locking Measurements on z/OS (with Bob Rogers)*
    - ✓ *Exploring z/OS SMF 30 Address Space CPU Measurements*
    - ~~*Exploring z/OS XCF Message Traffic Measurements*~~
    - *More Details - Exploring z/OS SMF 30 Address Space CPU Measurements*
    - *Exploring z/OS SMF 14 / 15 Records for Tape and DASD File Activity*
    - *Exploring z/OS WLM CPU Measurements: SUs vs CPU Secs vs APPL% vs Workload%*
    - *Exploring the Coupling Facility Lock Structure Measurements*
  - Dozens of past webinars are available at our website.

- 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

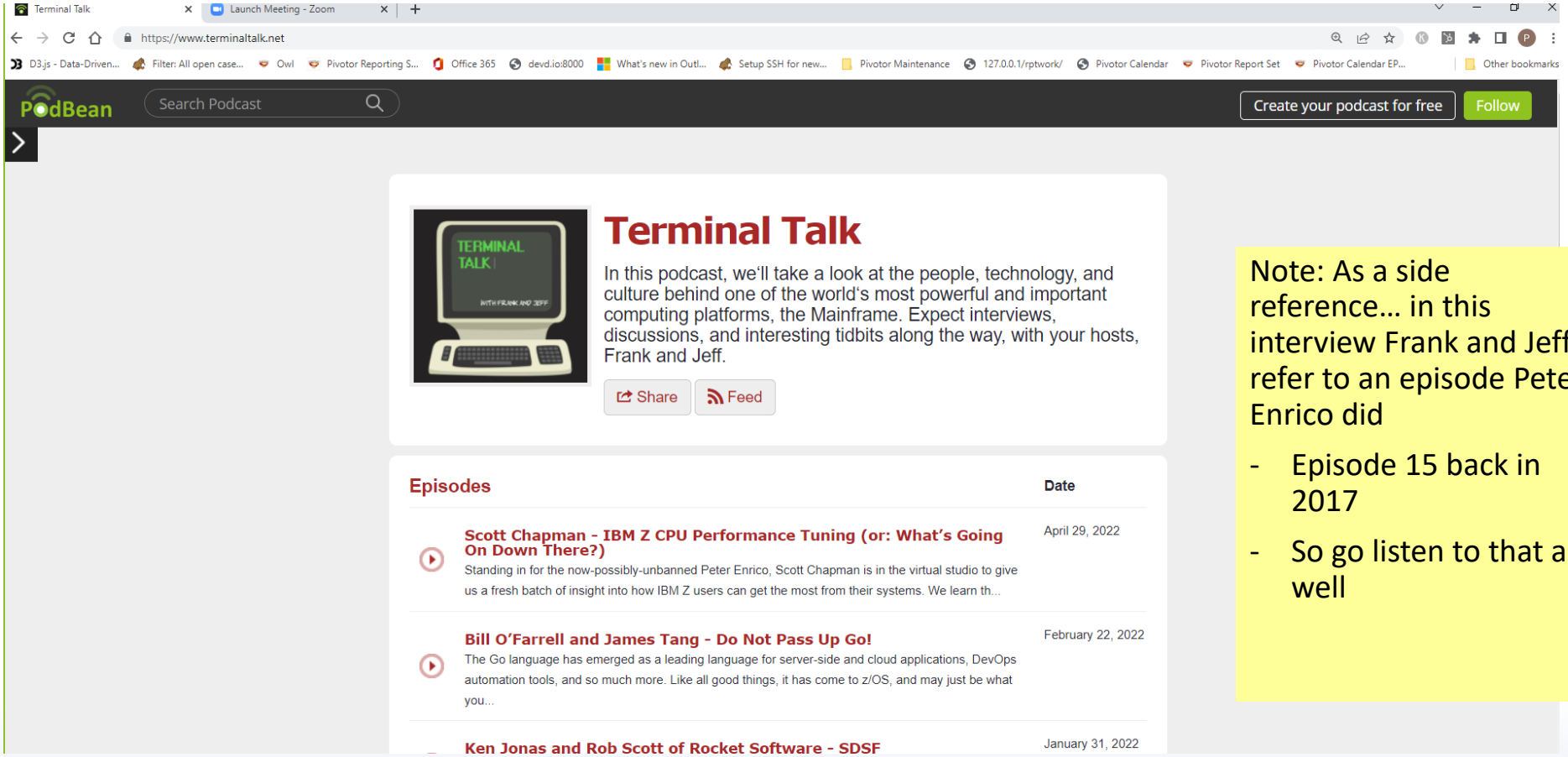
Enterprise Performance Strategies, Inc.
©

z/OS Performance workshops available

During these workshops you will be analyzing your own data!

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

Scott's interview on Terminal Talk

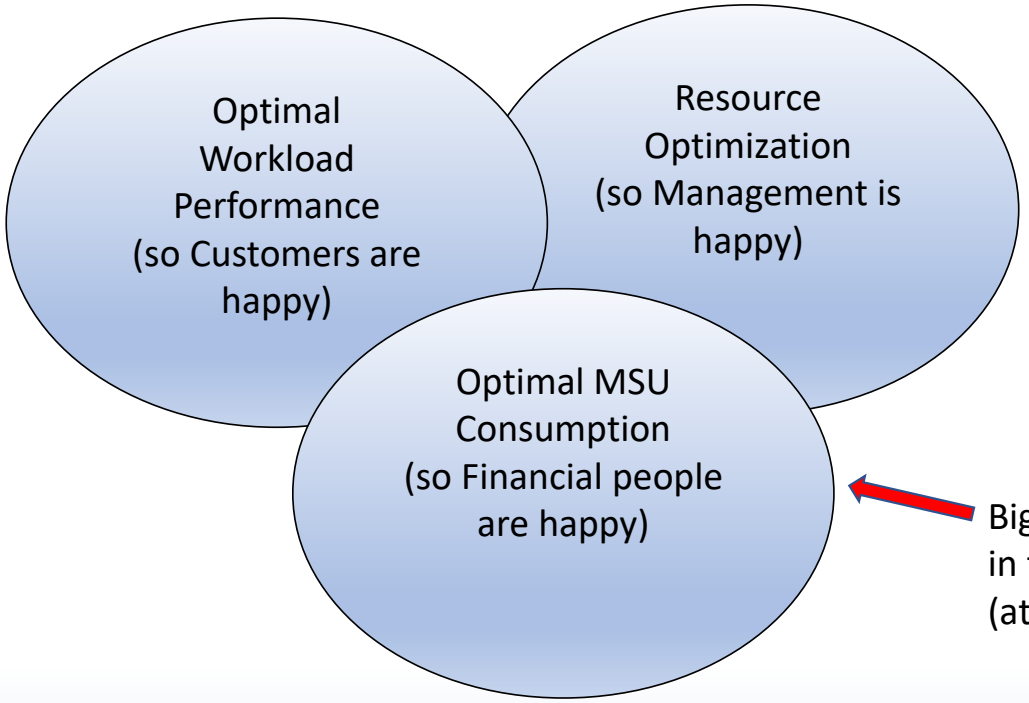


Note: As a side reference… in this interview Frank and Jeff refer to an episode Peter Enrico did

- Episode 15 back in 2017
- So go listen to that as well

The Performance Balancing Act

- Performance on z/OS is about finding an optimal balance of the following:

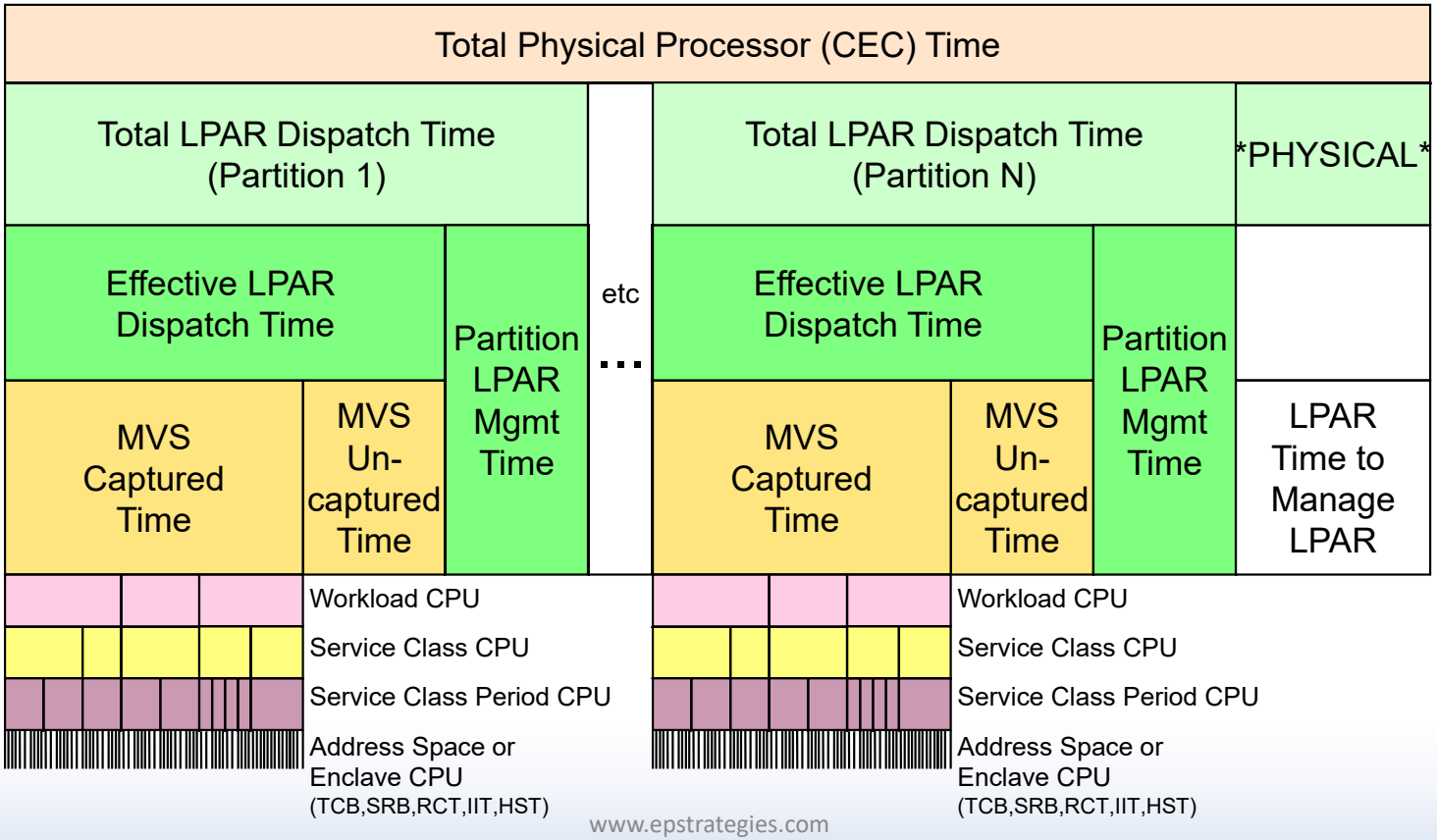


Biggest change to z/OS performance in the last 25 years of WLM (at least in my opinion)

© Enterprise Performance Strategies, Inc.

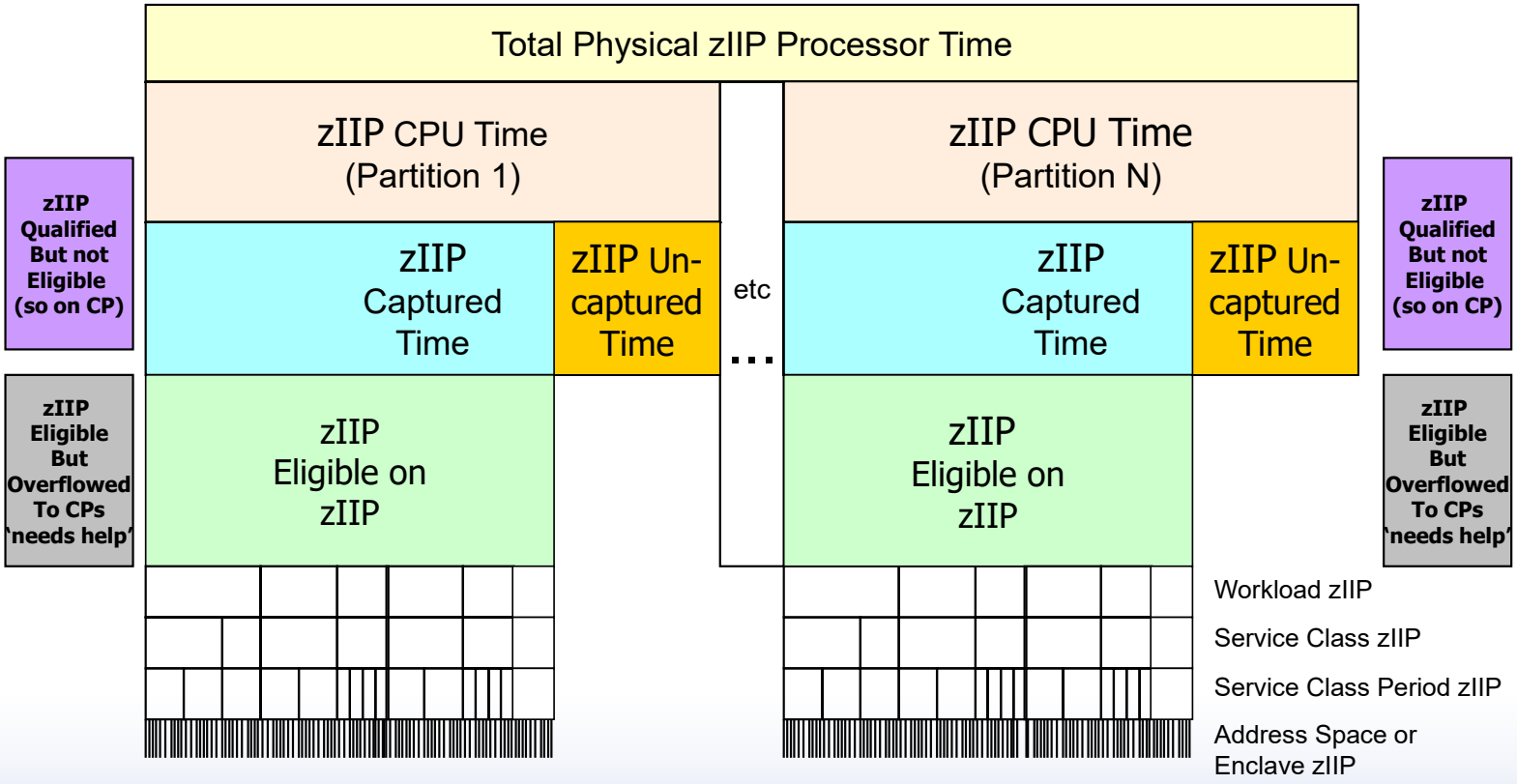
Breakdown of General-Purpose Processor

- We always needed to understand the break down of CP CPU consumption



Breakdown of zIIP Engine Time

- We need to understand how PR/SM allocates the zIIP processor resource
  - In all measurements zIIPs



SMF 30 Record Summary

- SMF 30
  - Measurements at the address space level
  - Important Note: There is no equivalent record for enclaves, but the enclave CPU time still needs to be accumulated to some associated address space
    - To be discussed…
- SMF 30.2 (and SMF 30.3) for interval recording (to match up to SMF 72.3)
  - Useful when need to understand CPU during specific periods of time
  - Also useful for looking at measurements over a period of time
- SMF 30.4 for step end for understanding resources consumed by job step
  - Useful for understanding CPU measurement on job step basis
    - By Step / Program
  - If summed, then useful for looking at CPU measurement for the entire job
- SMF 30.5 for job end for understanding resources consumed by completed jobs
  - Useful for understanding CPU measurement for a completed job

Identification Measurements

Identification Measurements

- SMF 30 job / address space identification information

Name	Description
SMF30JBN	Job or session name.
SMF30PGM	Program name (taken from PGM= parameter on EXEC card).
SMF30STM	Step name (taken from name on EXEC card).
SMF30UIF	User-defined identification field
SMF30JNM	JES job identifier.
SMF30STN	Step number (first step = 1, etc.).
SMF30CLS	Job class (blank for TSO/E session or started tasks)
SMF30SSN	Substep number. This field is set to zero for non-z/OS UNIX System Services steps. When the z/OS UNIX System Services exec function is requested, a new substep is begun and this value is incremented.
SMF30EXN	Program name. For a z/OS UNIX program, this contains the UNIX program that was run or the 8 character name of an MVS program that was run.
SMF30ASI	Address Space identifier

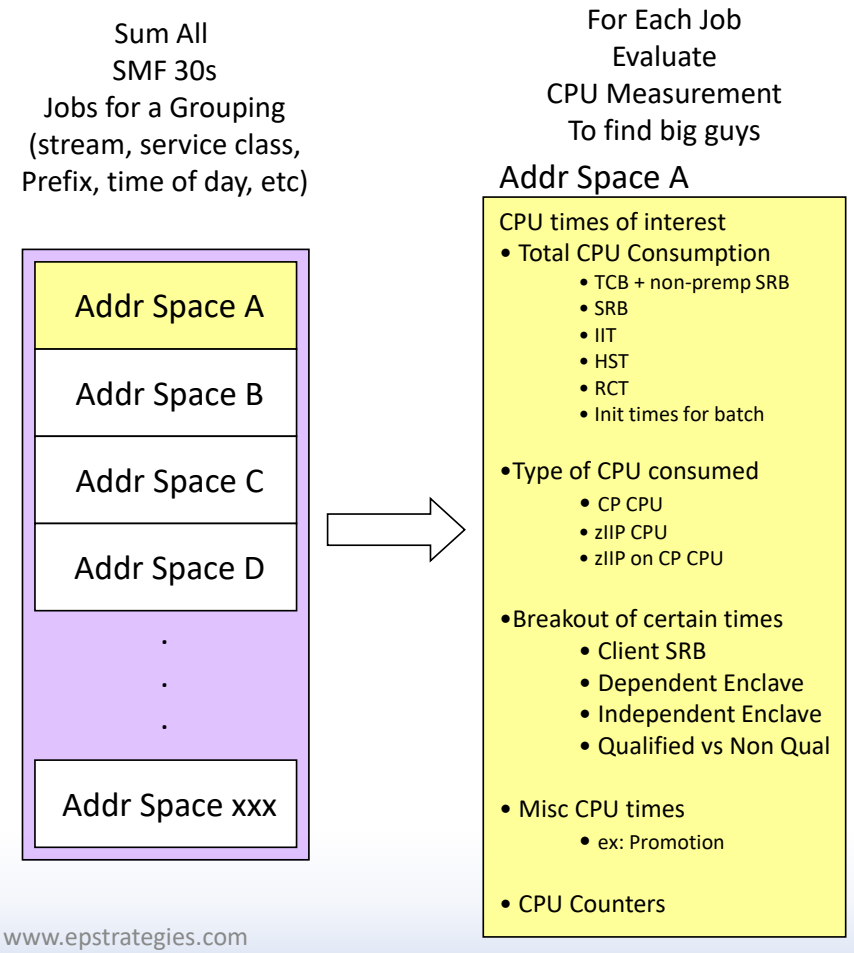
SMF 30 Measurements to Correlate to SMF 72.3

- Can also use SMF 30 WLM information to correlate measurements to the SMF 72.3 records
  - Use Service Class name and Report Class name to correlate measurements to the SMF 72.3 records

Name	Description
SMF30TRS	Number of system resources manager (SRM) transactions.
SMF30WLM	Workload name.
SMF30SCN	Service class name.
SMF30GRN	Resource group name.
SMF30RCN	Report class name.
SMF30ETC	Independent enclave transaction count.

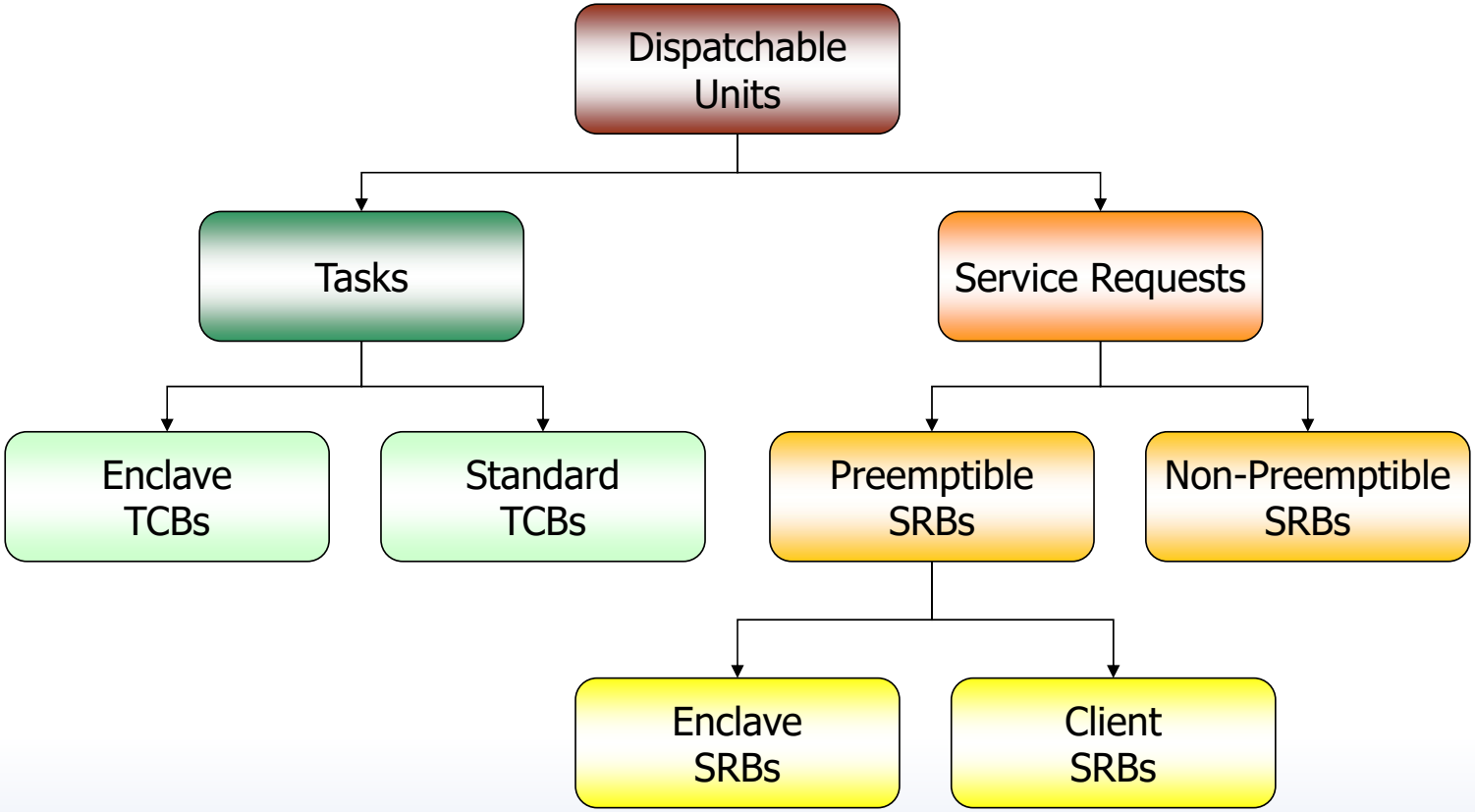
Ways of Looking at SMF 30 Data

- SMF 30 records can be cut for each address space
  - In theory, if you add up all the SMF 30 CPU time for all the address spaces in a service class, it should equal the CPU time in the SMF 72.3 record for that service class
- There is lots of possible double accounting that you need to be careful of
- And there are many cases of CPU time accumulated on one address space that is accounted for in a different address space
  - Or at least broken out to show this



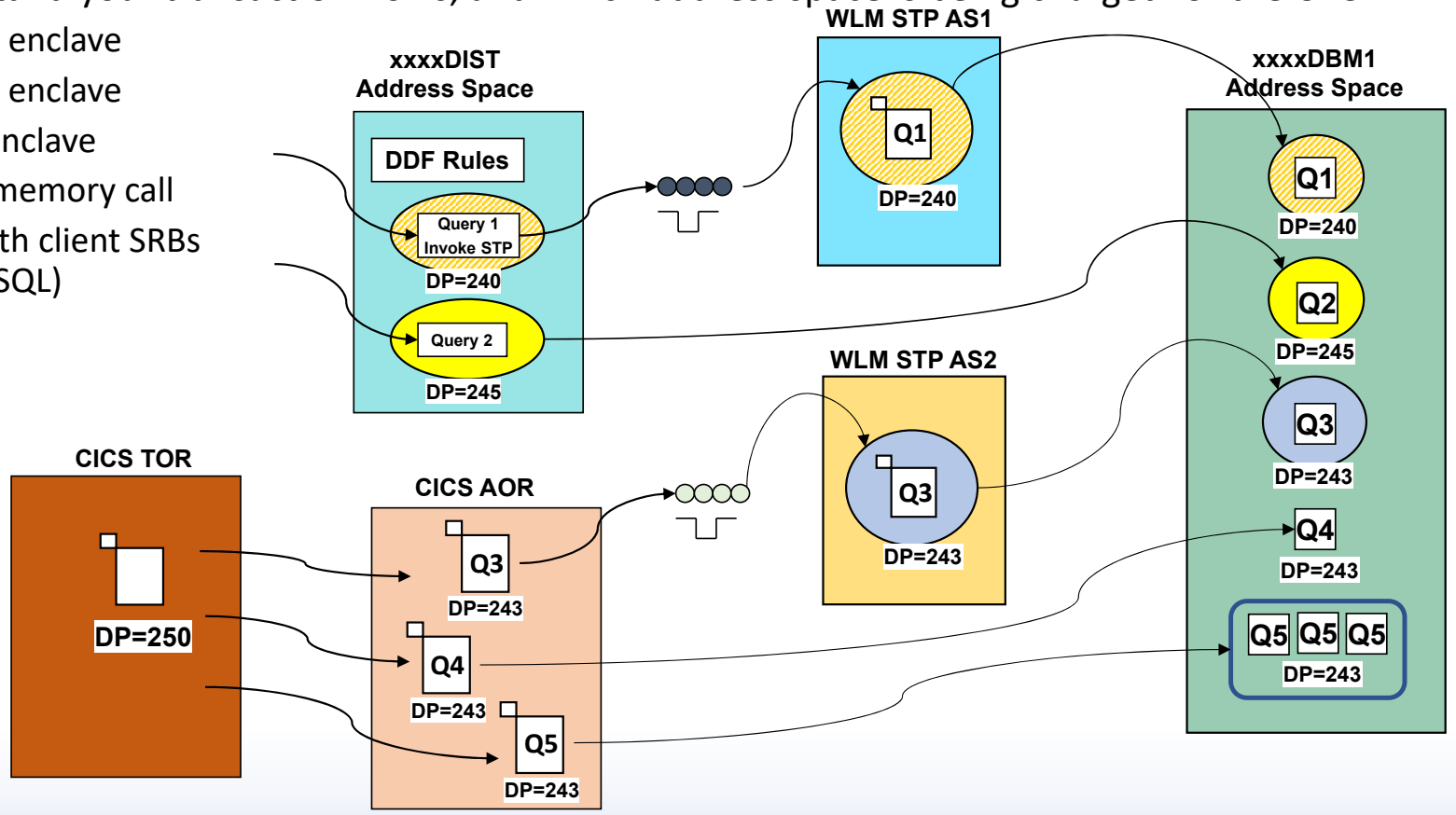
Summary of Dispatchable Unit Types

- Dispatchable units represent the programs that run on the CPUs



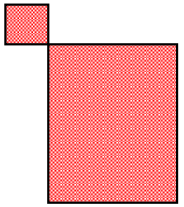
For server address spaces with/without enclaves… it does get a bit complicated

- Key Lesson – Understand your transaction flows, and which address space is being charged for the CPU
  - Q1 – Independent enclave
  - Q2 – Independent enclave
  - Q3 – Dependent enclave
  - Q4 – Direct cross memory call
  - Q5 – Direct call with client SRBs (for parallel SQL)



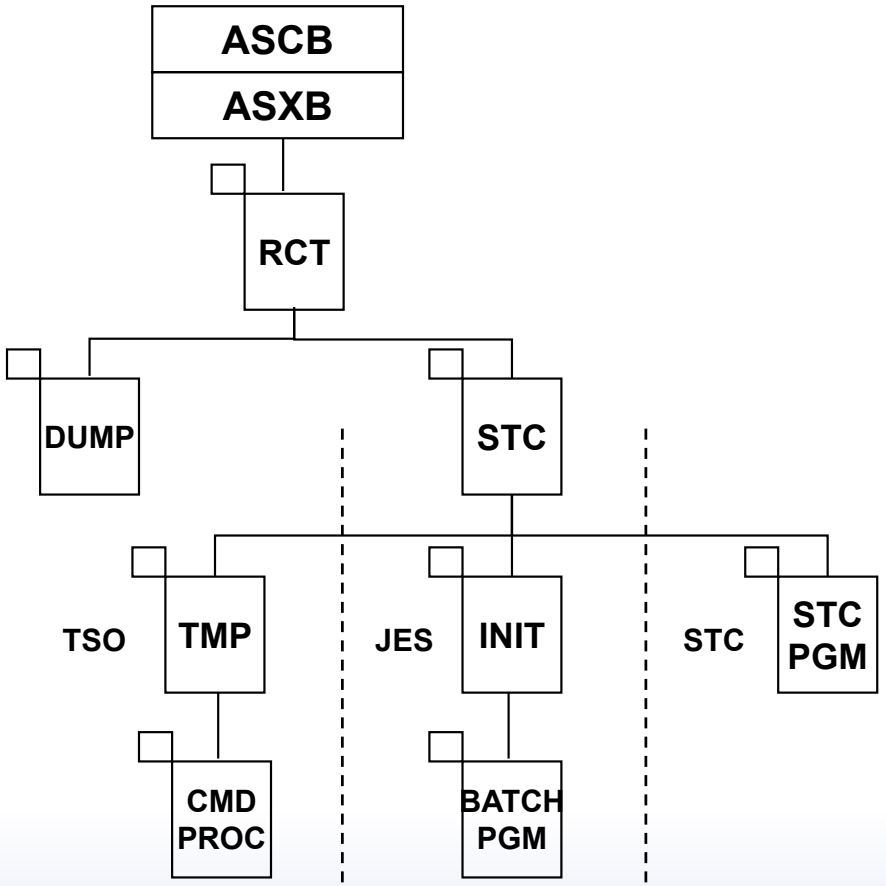
Overview of Tasks (TCBs)

- Tasks
  - Represented internally on z/OS as TCBs (Task Control Blocks)
  - Represents a single dispatchable unit of work
  - Can only be dispatched on one, individual, CP at a time
  - Represents a single program
  - Characteristics of TCBs include:
    - created by the ATTACH macro
    - can own storage
    - can issue supervisor calls
    - can issue I/O requests
    - CPU time measurements typically thought of as problem program time
    - are usually long running
    - run as preemptible (i.e. can be interrupted)
    - Dispatching Priority
      - Standard TCBs typically run at dispatch priority of home address space
      - Enclave TCBs run at the dispatching priority of the enclave



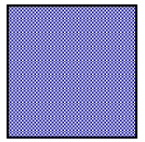
Typical Address Space's TCB Structure

- The diagram on this slide is a crude representation of the TCB structure of TSO, JES, and STC type address spaces



Overview of Non-preemptible SRBs

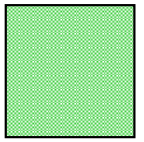
- Non-preemptible Service Request Blocks
  - Known as SRBs - Service Request Blocks
  - This foil only discusses non-preemptible SRBs
  - Represents a single dispatchable unit of work (i.e. program), so can only be dispatched on a single CP
  - Characteristics of non-preemptible SRBs include:
    - created by authorized programs, initiated via the SCHEDULE macro
    - scheduled globally or locally
    - very inexpensive to create and run
    - cannot own storage
    - cannot issue supervisor calls (including I/O)
    - meant for short duration work
    - 'SRB time' in SMF records
    - traditionally run at dispatch priority of home address space



  - Once dispatched, non-preemptible SRBs will continue to run until they incur a voluntary interrupt such as a suspend/page fault, of when they complete
    - Even if higher priority work is ready to use the CPU

Overview of Preemptible SRBs

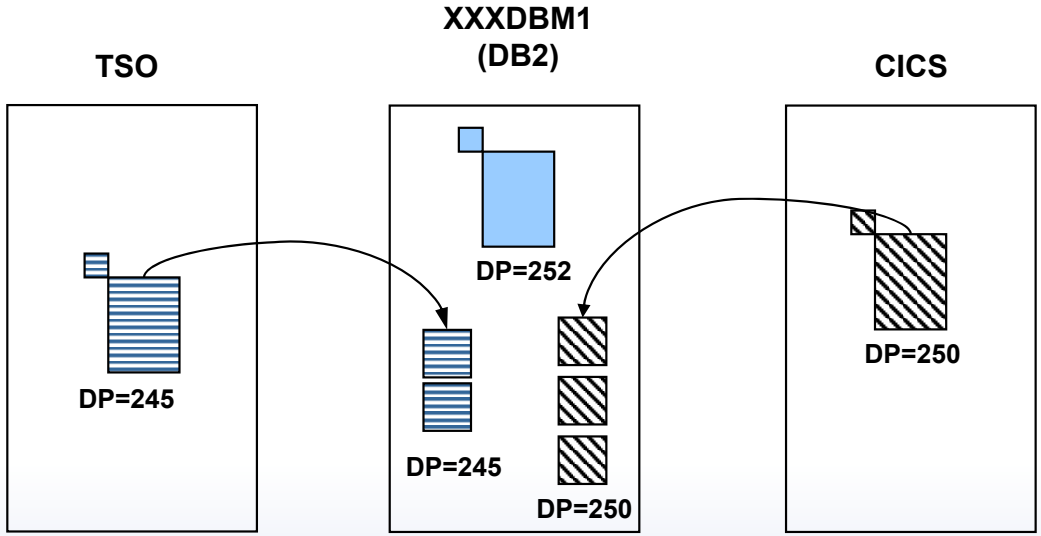
- Preemptible Service Request Blocks
  - Known as SRBs - Service Request Blocks
  - This foil only discusses preemptible SRBs
  - Represents a single dispatchable unit of work (i.e. program), so can only be dispatched on a single CP
  - Considered an alternative to task mode
  - Characteristics of preemptible SRBs include:
    - created by authorized programs, initiated via the SCHEDULE macro
    - scheduled globally or locally
    - very inexpensive to create and run
    - cannot own storage
    - cannot issue supervisor calls (including I/O)
    - do not have to be for short duration work
    - Can be preempted off a processor
    - CPU time measurements for these SRBs are usually included with the CPU time measurements of TCBs



Preemptible SRB

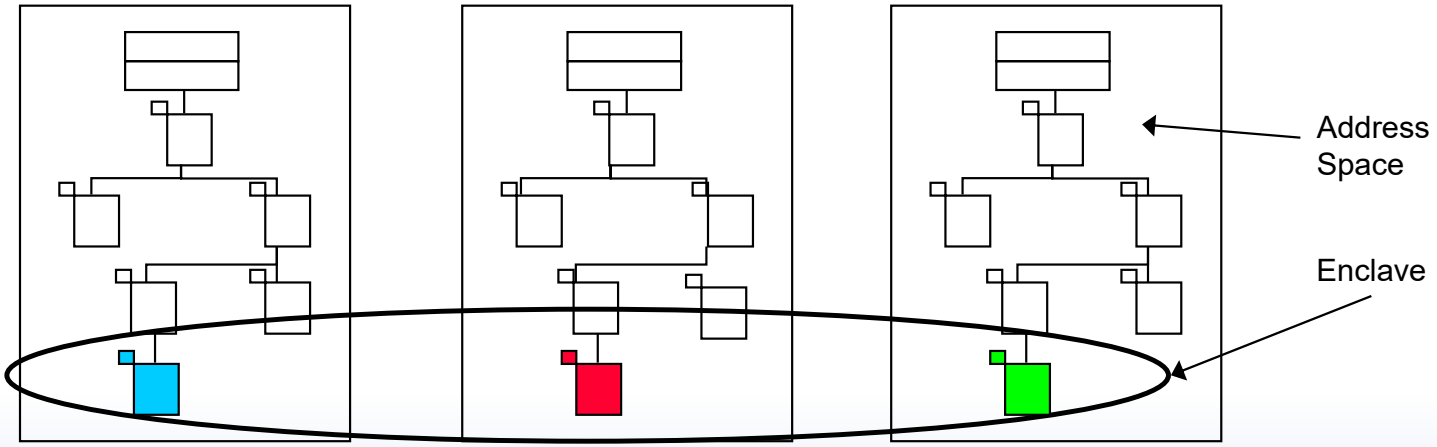
Introduction to Preemptible Client SRBs

- Client SRBs are preemptible SRBs
  - Originally created to allow for DB2 CPU parallelism of complex queries
  - Client SRBs run in an address space (such as DB2), but execute work on behalf of some other 'client' address space (such as TSO or CICS)
  - All CPU controls are derived from the client address space
  - CPU time consumed is accumulated by the client SRB is accumulated back to client address space and considered part of client's transaction



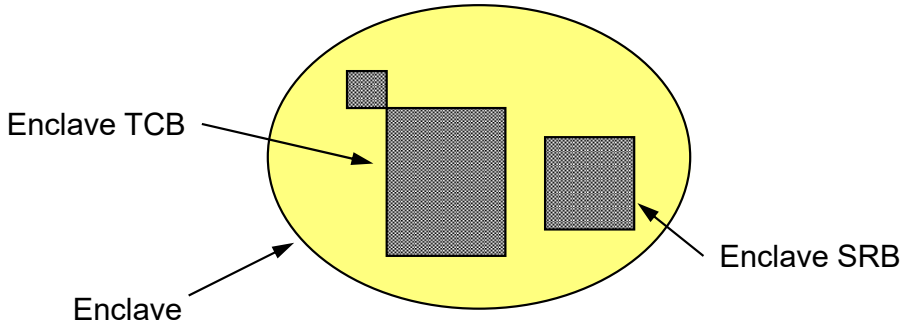
Definition of Enclave

- Very different than address spaces, but similar in concept
- An enclave represents a transaction that can span multiple dispatchable units of work (SRBs and TCBs) in one or more address spaces, and reported and managed as a single unit



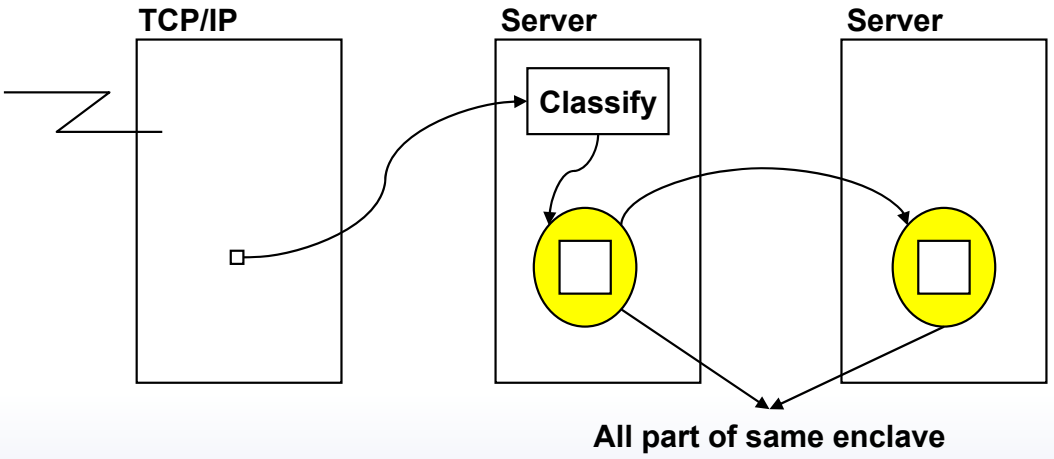
Enclave SRBs and TCBs

- Enclave SRBs and Enclave TCBs
  - Preemptible SRBs or TCBs that run in a target address space, but executes work on behalf of some enclave to which it is associated
  - Dispatch priorities are derived from the enclave
  - CPU time consumed accumulated back to address space that created the enclave, and included as CPU service in the server address space
  - When the enclave transitions to a new period, so does the enclave SRB and TCB



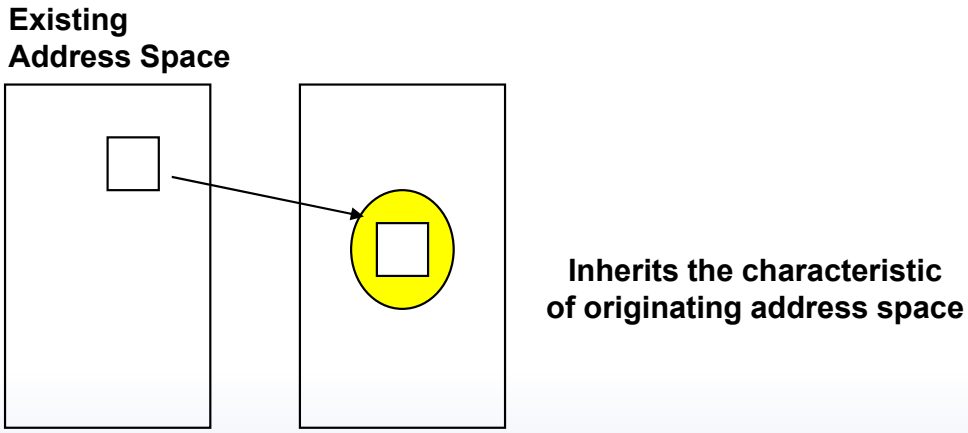
Types of Enclaves - Independent

- Independent Enclave
  - Represents a complete independent transaction (ie. a true SRM transaction)
    - Usually, a transaction that is arriving from the network
  - Separately classified and managed in a service class
  - Transaction starts, enclave created, classified, runs



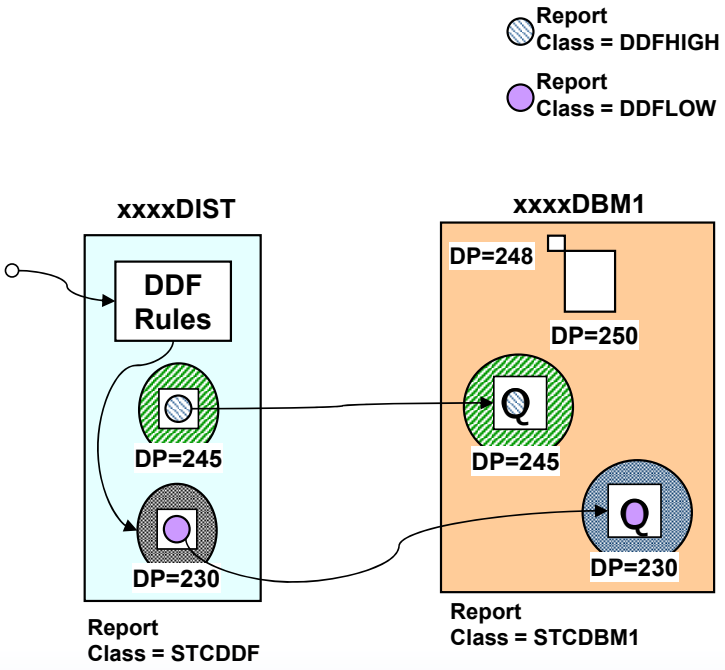
Types of Enclaves – Dependent

- Dependent Enclave
  - Represents a transaction that is a continuation of an already existing transaction
  - Starts in an address space and 'spawns' into an enclave
  - Inherits the service class of the original transaction
  - Accumulated service treated as though it were accumulated by the requesting address space



DDF and Independent Enclaves

- Chargeback to the address space that created enclave
  - In this case, the xxxxDIST
- When zIIP engines are configured SMF30ENC is the enclave CPU time that was qualified for zIIP but was prevented from running on zIIP



SMF 72

STCDDF

CPU=1%

SMF 72

STCDBM1
(for these transactions)

CPU=1%

SMF 72

DDFLOW

CPU=26.25%
zIIP=33.75

SMF 72

DDFHIGH

CPU=8.75%
zIIP=11.25%

SMF 30

DDF

CPU=36%
ENC=35%
zIIP=45%

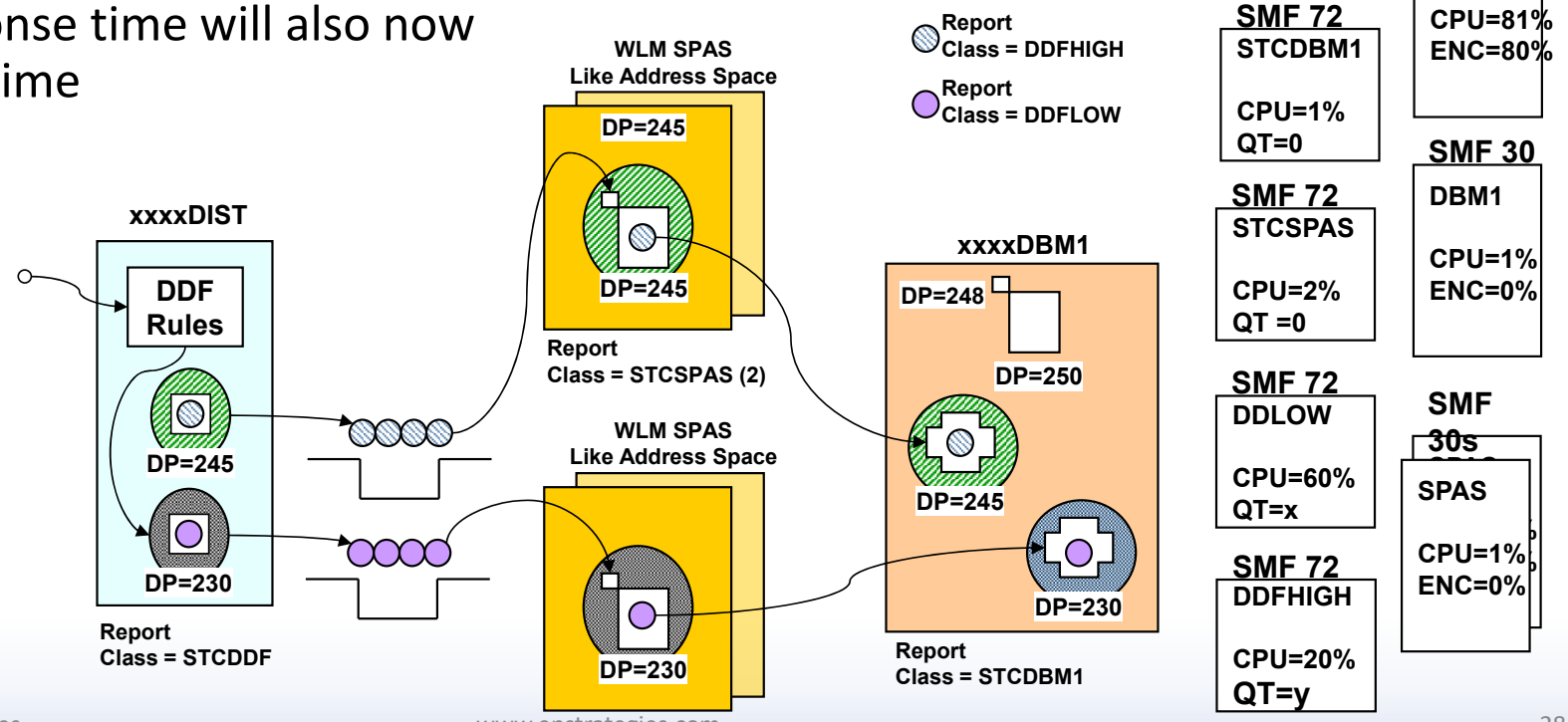
SMF 30

DBM1

CPU=1%
ENC=0%
zIIP=0%

DDF and Independent Enclaves and Application Environments

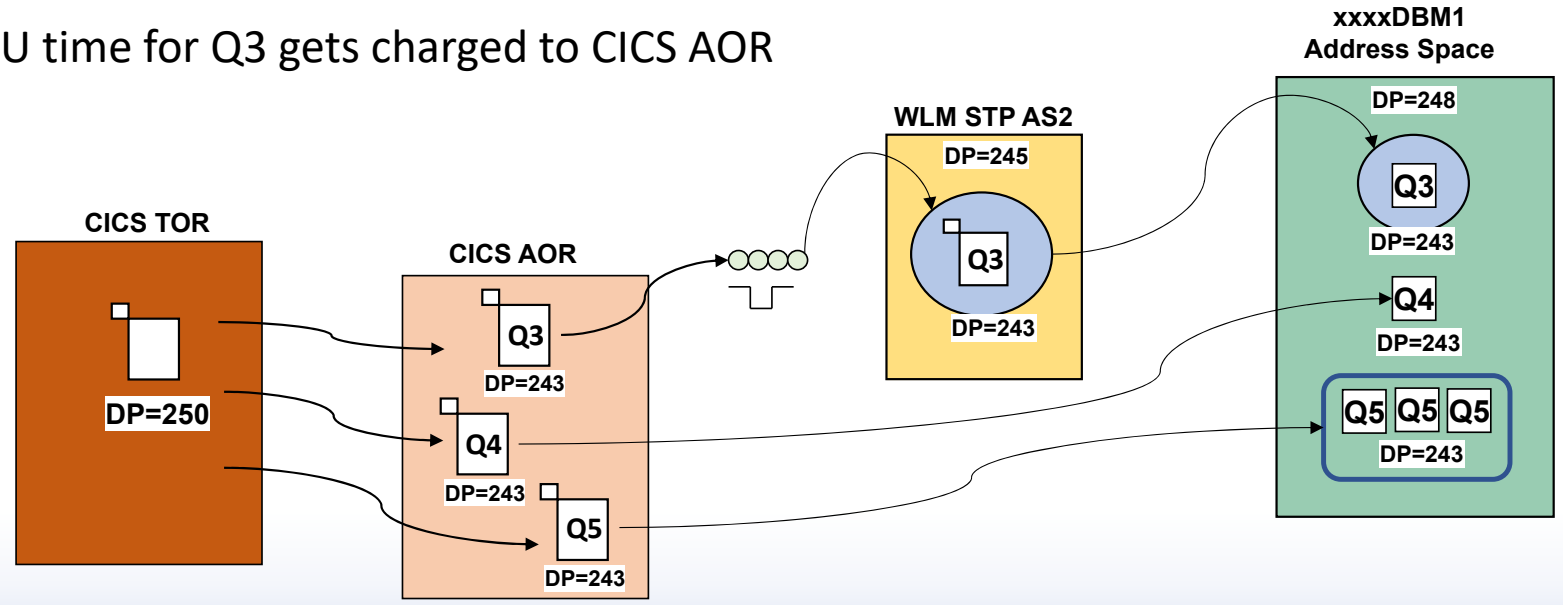
- Once again, chargeback to the address space that created enclave
  - In this case, the xxxxDIST
- Note that response time will also now include queue time



CICS and Dependent Enclaves and Application Environments

- Key Lesson – Understand your transaction flows, and which address space is being charged for the CPU
  - Q3 – Dependent enclave
  - Q4 – Direct cross memory call
  - Q5 – Direct call with client SRBs (for parallel SQL)

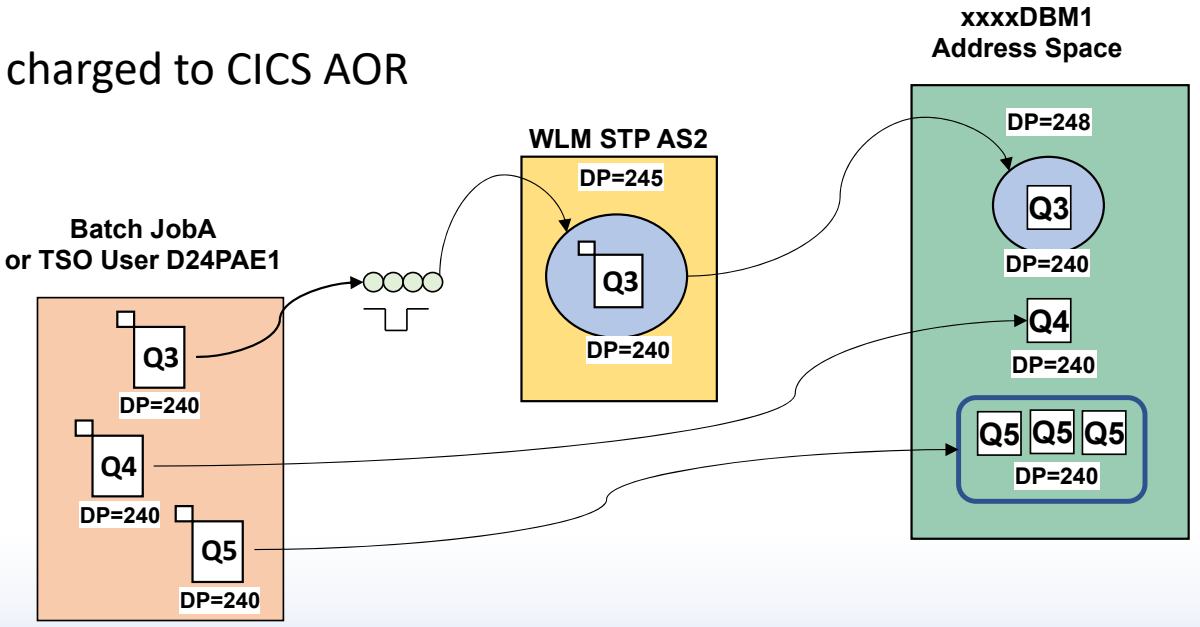
- In this case, the CPU time for Q3 gets charged to CICS AOR



Batch or TSO and Dependent Enclaves and Application Environments

- Key Lesson – Understand your transaction flows, and which address space is being charged for the CPU
  - Q3 – Dependent enclave
  - Q4 – Direct cross memory call
  - Q5 – Direct call with client SRBs (for parallel SQL)

- In this case, the CPU time for Q3 gets charged to CICS AOR

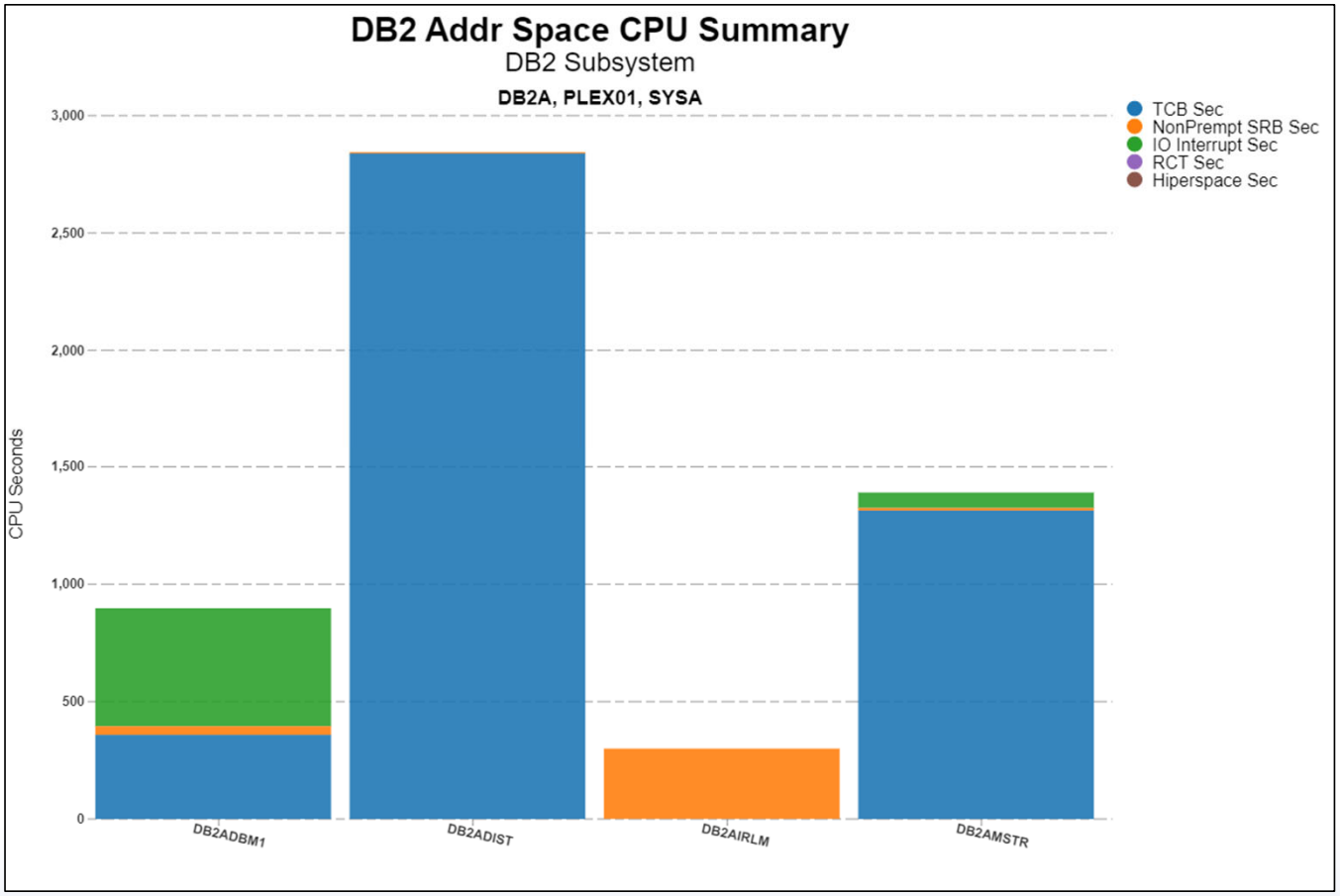


SMF 30 General Purpose CPU Time Values

- SMF 30 Processor Accounting Section of SMF 30 contains key CPU time values of interest

Name	Description
SMF30CPT	CPU time under the task control block (TCB). Includes: • Enclave time • Preemptible class SRB time (such as client SRB time) • Includes cross over zIIP work running on a CP
SMF30CPS	CPU time under non-preemptible SRBs
SMF30IIP	Amount of CPU time used to process I/O interrupts.
SMF30RCT	Amount of CPU time used by the region control task (RCT).
SMF30HPT	Amount of CPU time used for hiperspace transfers (HST).
SMF30ICU	Initiator CPU time under the task control block (TCB) (step init and step term)
SMF30ISB	Initiator CPU time under the service request block (SRB) (step init and step term)
SMF30ASR	Additional CPU time accumulated by the pre-emptable SRBs and client SRBs for this job. Value is included in SMF30CPT.
SMF30_TIME_ON_zIIP	Time spent on zIIP. (includes enclave time).
SMF30_TIME_zIIP_ON_CP	CPU time spent running zIIP eligible work
SMF30ENC	CPU time used by the enclaves created by this address space. This value is also included in the value in SMF30CPT.
SMF30_ENCLAVE_TIME_ON_zIIP	Enclave time spent on zIIP.
SMF30_ENCLAVE_TIME_zIIP_ON _CP	zIIP enclave time spent on a standard CP.
SMF30DET	Dependent enclave CPU time. This value is also contained in SMF30CPT.
SMF30_DEPENC_TIME_ON_zIIP	Dependent enclave time spent on zIIP.
SMF30_DEPENC_TIME_zIIP_ON_CP	zIIP dependent enclave time spent on a standard CP.
SMF30_ENCLAVE_TIME_zIIP_QUAL	Normalized enclave time qualified to be on zIIP in hundredths of a second. This is the SRB time for an enclave that a program (Db2, for example) has identified to WLM for zIIP eligibility.
SMF30_DEPENC_TIME_zIIP_QUAL	Normalized dependent enclave time qualified to be on zIIP in hundredths of a second. This is the enclave SRB time that a program (Db2, for example) has identified to WLM for zIIP eligibility.

Total CP CPU Time

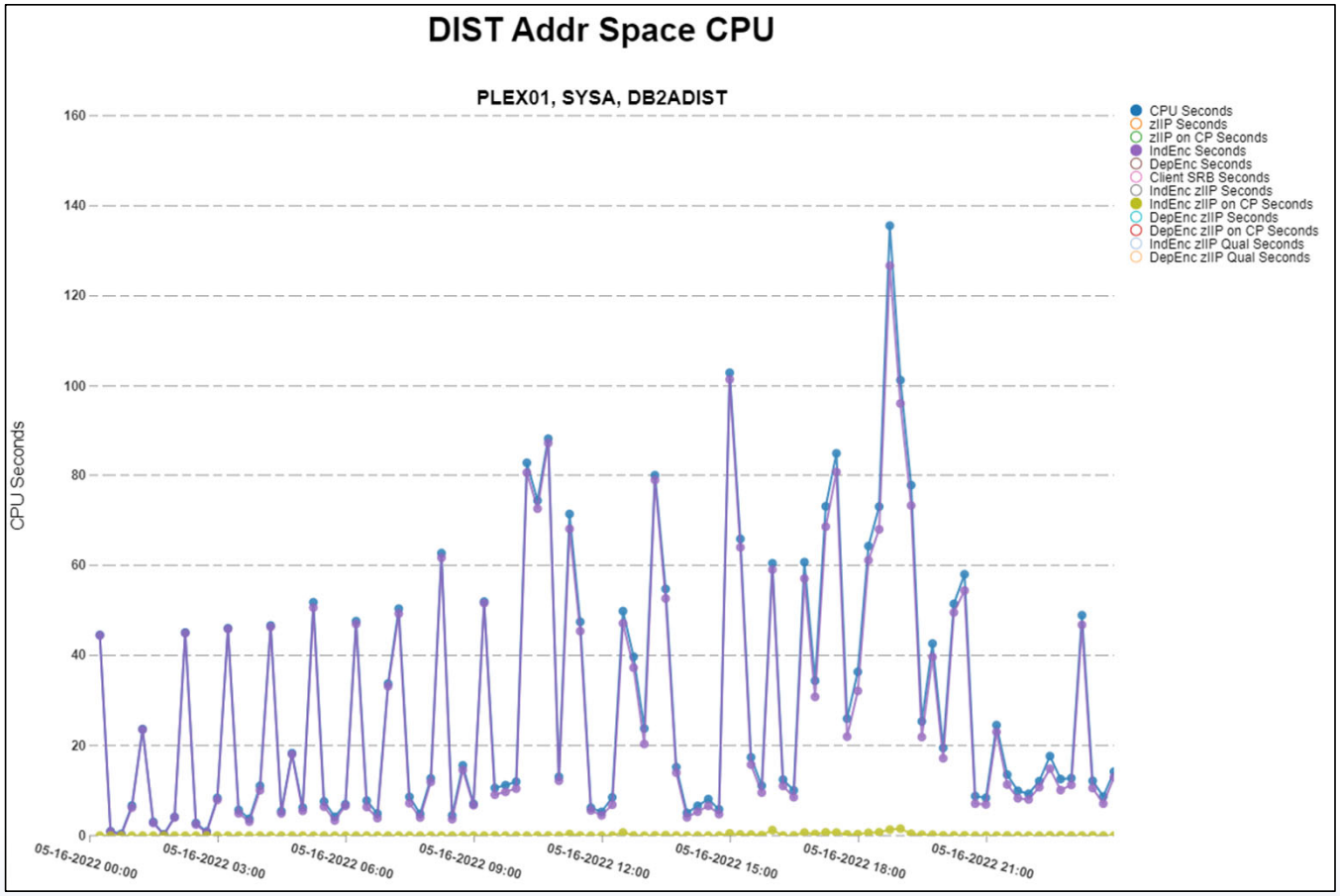


Total CPU CP Time for an address space:

- TCB + non-premp SRB
- SRB
- IIT
- HST
- RCT
- Init times for batch

Typically, this is the most interesting number when starting an address space CPU analysis

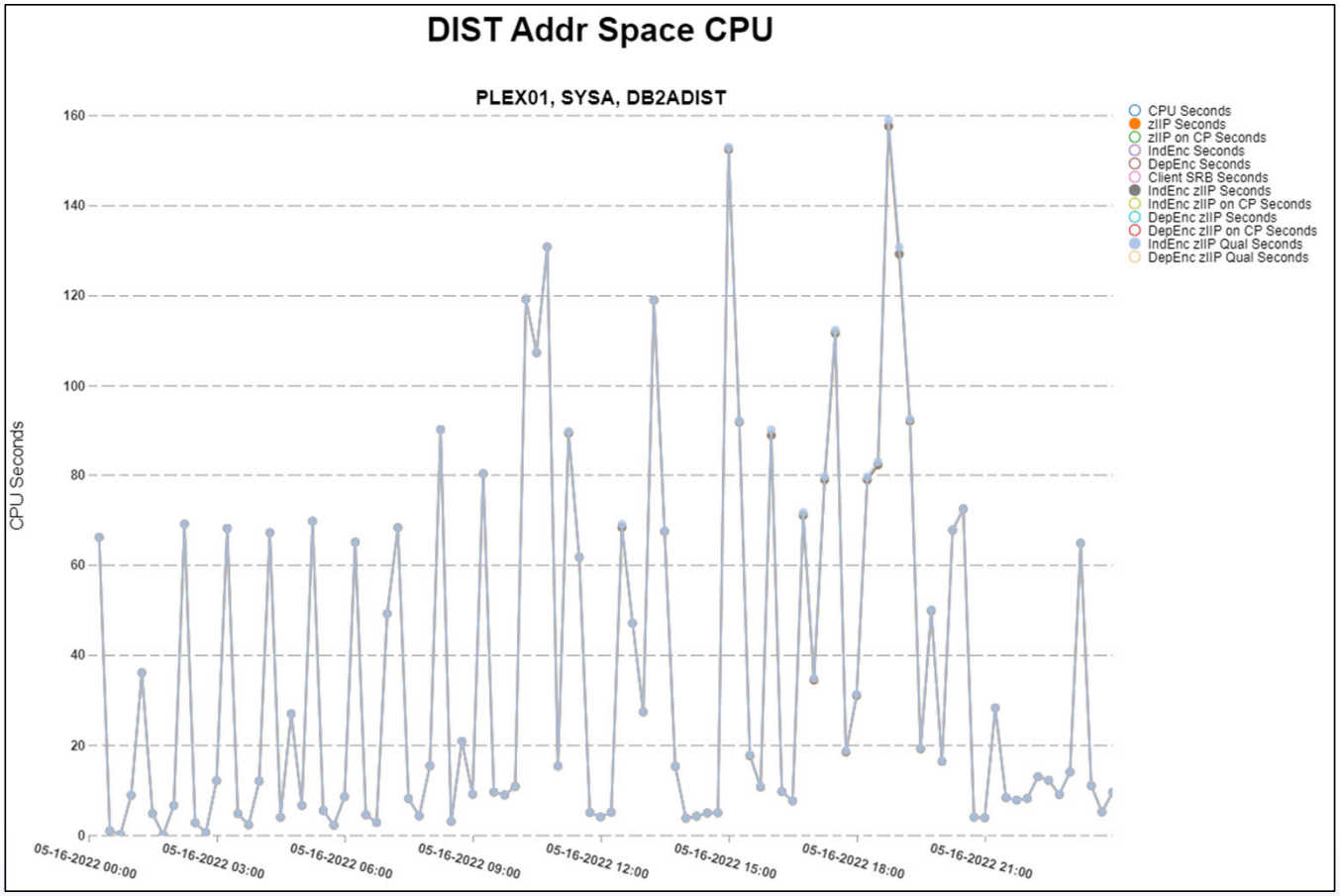
xxxxDIST CP CPU Time



Total CPU CP Time for a DIST address space:

- Note Total CPU seconds
- Very close to Independent Enclave seconds
- zIIP on CP small here

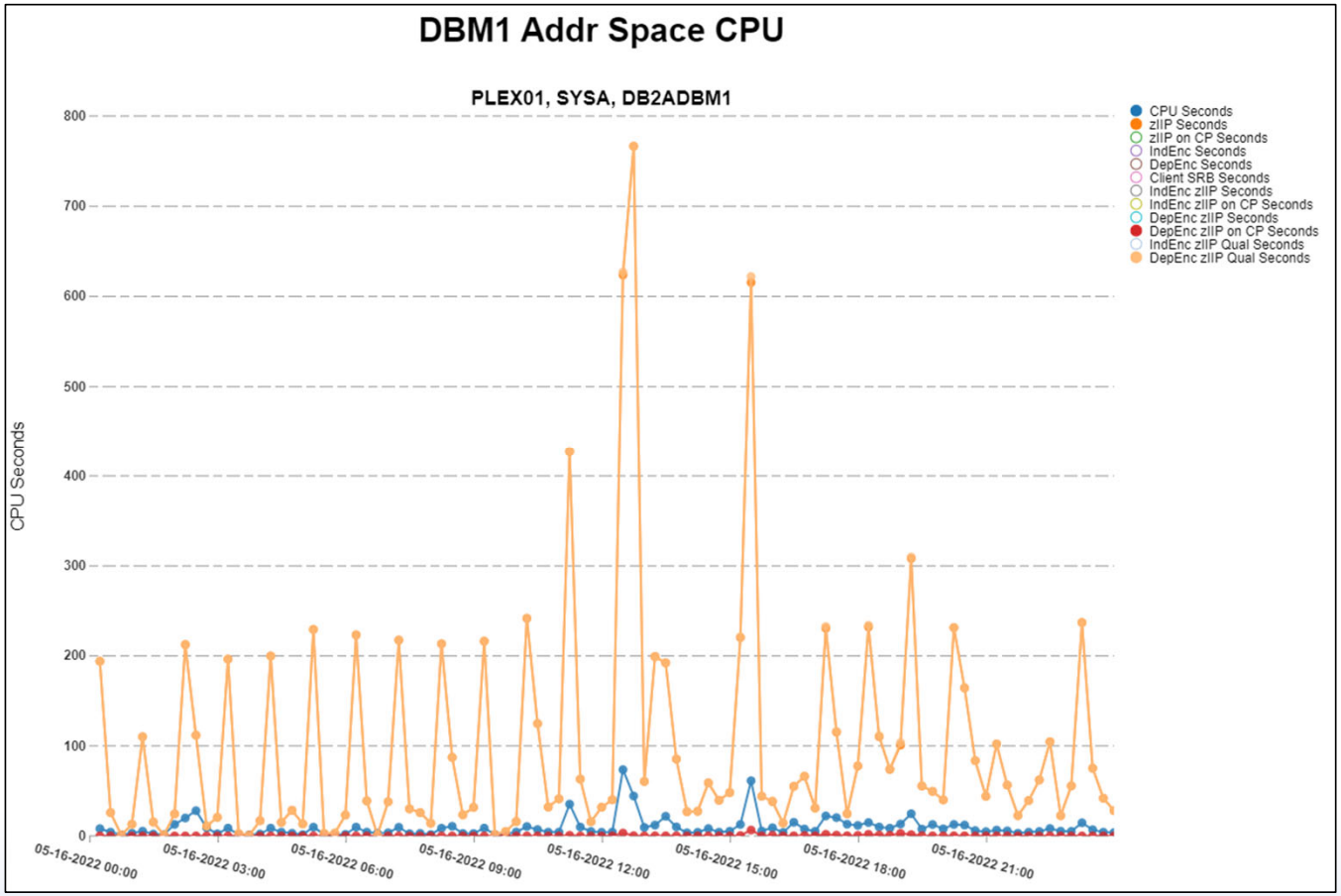
xxxxDIST zIIP CPU Time



Total CPU zIIP Time for a DIST address space:

- Note total zIIP seconds is nearly the same as independent enclave zIIP seconds.
- These are nearly identical to zIIP qualified time

xxxxDBM1 zIIP CPU Time

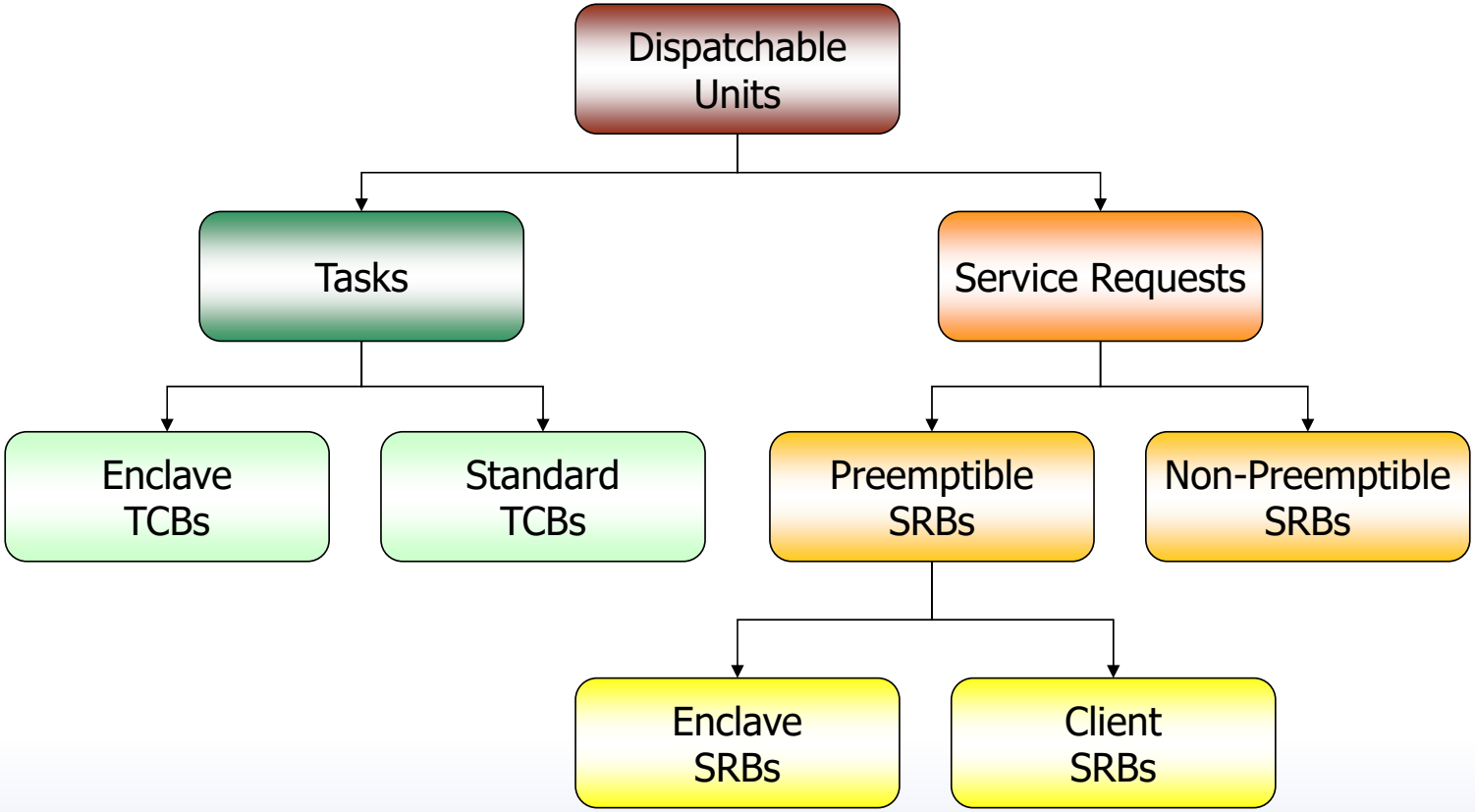


Total CPU zIIP Time for a DIST address space:

- Note total zIIP seconds is nearly the same as independent enclave zIIP seconds.
- These are nearly identical to zIIP qualified time
- Notice CPU time small since most of that time is charged to the caller

Summary of Dispatchable Unit Types

- Dispatchable units represent the programs that run on the CPUs



SMF 30 General Purpose CPU Time Values

- SMF 30 Processor Accounting Section of SMF 30 contains key CPU time values of interest

Name	Description
SMF30CPT	CPU time under the task control block (TCB). Includes: • Enclave time • Preemptible class SRB time (such as client SRB time) • Includes cross over zIIP work running on a CP
SMF30CPS	CPU time under non-preemptible SRBs
SMF30IIP	Amount of CPU time used to process I/O interrupts.
SMF30RCT	Amount of CPU time used by the region control task (RCT).
SMF30HPT	Amount of CPU time used for hiperspace transfers (HST).
SMF30ICU	Initiator CPU time under the task control block (TCB) (step init and step term)
SMF30ISB	Initiator CPU time under the service request block (SRB) (step init and step term)
SMF30ASR	Additional CPU time accumulated by the pre-emptable SRBs and client SRBs for this job. Value is included in SMF30CPT.
SMF30_TIME_ON_zIIP	Time spent on zIIP. (includes enclave time).
SMF30_TIME_zIIP_ON_CP	CPU time spent running zIIP eligible work
SMF30ENC	CPU time used by the enclaves created by this address space. This value is also included in the value in SMF30CPT.
SMF30_ENCLAVE_TIME_ON_zIIP	Enclave time spent on zIIP.
SMF30_ENCLAVE_TIME_zIIP_ON _CP	zIIP enclave time spent on a standard CP.
SMF30DET	Dependent enclave CPU time. This value is also contained in SMF30CPT.
SMF30_DEPENC_TIME_ON_zIIP	Dependent enclave time spent on zIIP.
SMF30_DEPENC_TIME_zIIP_ON_CP	zIIP dependent enclave time spent on a standard CP.
SMF30_ENCLAVE_TIME_zIIP_QUAL	Normalized enclave time qualified to be on zIIP in hundredths of a second. This is the SRB time for an enclave that a program (Db2, for example) has identified to WLM for zIIP eligibility.
SMF30_DEPENC_TIME_zIIP_QUAL	Normalized dependent enclave time qualified to be on zIIP in hundredths of a second. This is the enclave SRB time that a program (Db2, for example) has identified to WLM for zIIP eligibility.

Remember!!!!!
Always ask…
To which address space are various CPU measurements accumulated to?

Questions?