WLM - Effective Setup and Usage of WLM Report Classes

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

- **WLM - Effective Setup and Usage of WLM Report Classes**

Abstract

– Rarely are WLM report classes used to their full potential. Most customers set them up so that they are used, but then never use them. Or do not even set them up effectively to begin with. Then when it comes time to debug a performance problem, or when management asks a question about a specific workload, the report class setup falls short of its objective.

– During this session WLM expert Peter Enrico will walk you through an effective report class setup. Report examples will be provided so you can see first-hand the power of a great report class structure.

EPS Sessions at Share

Peter Enrico

Day	Time	Location	Presentation
Wed	11:15	Asia 3	SMF 113 Processor Cache Counter Measurements – Overview, Update, and Usage
Wed	1:45	Asia 3	WLM – Effective Setup and Usage of WLM Report Classes
Thu	11:15	Asia 3	zProcessor Consumption Analysis (including z13), or What is Consuming All the CPU?

Scott Chapman

Day	Time	Location	Presentation
Tue	11:15	Asia 3	Memory Management in the TB Age
Tue	3:15	Southern Hemisphere 4	Lessons Learned from implementing an IDAA
Fri	11:15	Asia 3	WLM in One Page

Performance Workshops Available

During these workshops you will be analyzing your own data!

- WLM Performance and Re-evaluating of Goals
  - Instructor: Peter Enrico and Scott Chapman
  - September 28 – October 2, 2015 – Columbus, Ohio, USA

- Parallel Sysplex and z/OS Performance Tuning
  (Web / Internet Based!)
  - Instructor: Peter Enrico and Scott Chapman
  - November 17 – 19, 2015

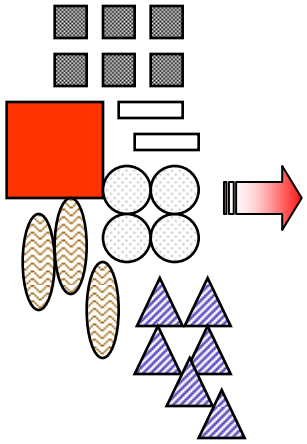
- Essential z/OS Performance Tuning Workshop
  - Instructors: Peter Enrico, Scott Chapman, Tom Beretvas
  - October 19 - 23, 2015 – Dallas, Texas, USA

- z/OS Capacity Planning and Performance Analysis
  - Instructor: Ray Wicks

Presentation Overview

- What are report classes?
  - Why use report classes
  - History of report classes
  - Overhead of report classes
- General Guidelines for report classes
  - General guidelines
  - Subsystem and workload guidelines
- Using report classes to evaluate resource usage
- Using report classes to evaluate goals

Overview of WLM Service Definition



Work coming Into the System

Service Definition

ASCH
CB
CICS
DB2
DDF
IMS
IWEB
JES
LSFM
MQ
NETV
OMVS
SOM
STC
TSO
SYSH

Classification Rules to Service Class

Service Classes have goals and importance

Workload **ONLINE**

SC=B	RC=RB
SC=C	RC=RC
SC=D	RC=RD
-	
SC=A	RC=RA

Workload **SYSTEM**

SC=O	RC=RO
SC=P	RC=RP
-	
SC=M	

Workload **BATCH**

SC=F	RC=RF
SC=F	RC=RG
SC=H	RC=RH
-	
SC=E	RC=RE

Workload **SYSPROG**

SC=N	RC=RN
SC=M	

Workload **WEB**

SC=J	RC=RJ
SC=K	RC=RK
-	
SC=L	

Workload **NEWWORK**

SC=NEWWORK
RC=NEWMQ
RC=NEWLSFM

Policy01

SC overrides

Policy02

SC overrides

Application Environments

AE1	AE4
AE2	AE5
AE3	

Scheduling Environments

SE1	Res01
SE2	Res02
	Res03

Miscellaneous Controls

- **Service Definition Coefficients**
- **I/O Management**

Resource Groups

RGRP01

Minimum=100
Maximum=1500

RGRP02

Minimum=0
Maximum=1000

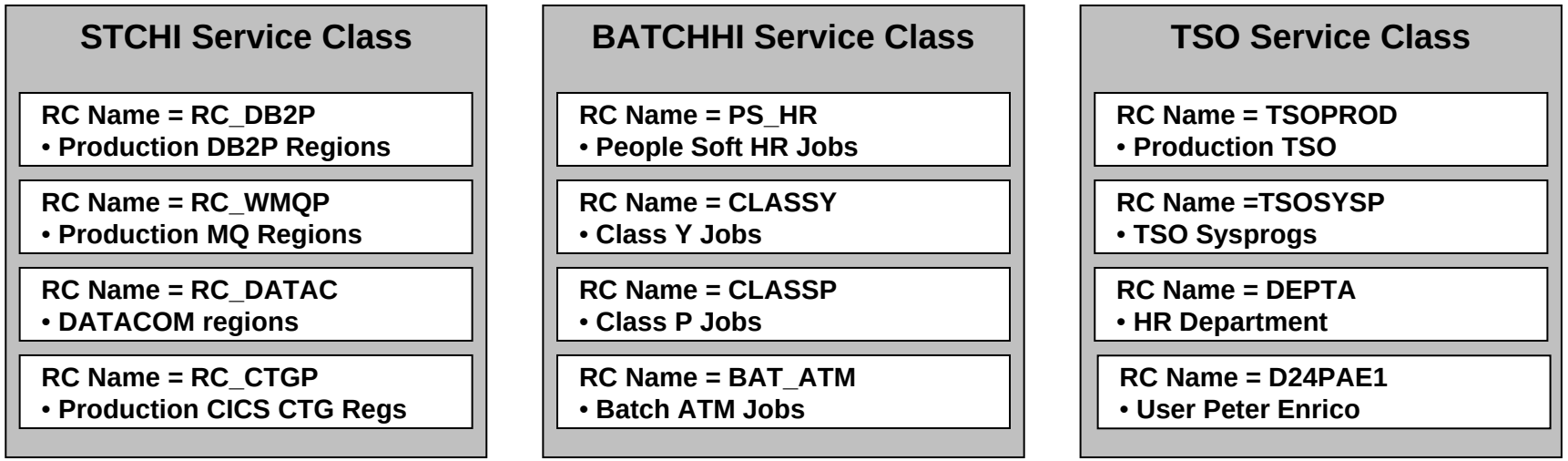
WLM Service Definition Review

- A WLM Service Definition consists of
  - *Service Policies* – named sets of overrides to defined goals in service policy
  - *Workloads* – aggregation of service classes for reporting purposes
  - *Service Classes* – subdivided into periods, groups of work with similar performance goals, business importance, and resource requirements for reporting and management purposes
  - *Report Classes* – group of work for 'more/less granular' reporting purposes
  - *Resource Groups* – define processor capacity boundaries across a sysplex
  - *Classification Rules* – determine how to assign incoming work to a service class and/or a report class
  - *Application Environments* – groups of application functions that execute in server address spaces and can be requested by a client
  - *Scheduling Environments* – lists of named resources along with their required state
  - *Global Settings* – miscellaneous settings for WLM controls

What are WLM report classes?

- In the world of WLM, all work is classified
  - To a service class for management and reporting purposes
  - and optionally assigned to a report class strictly for reporting

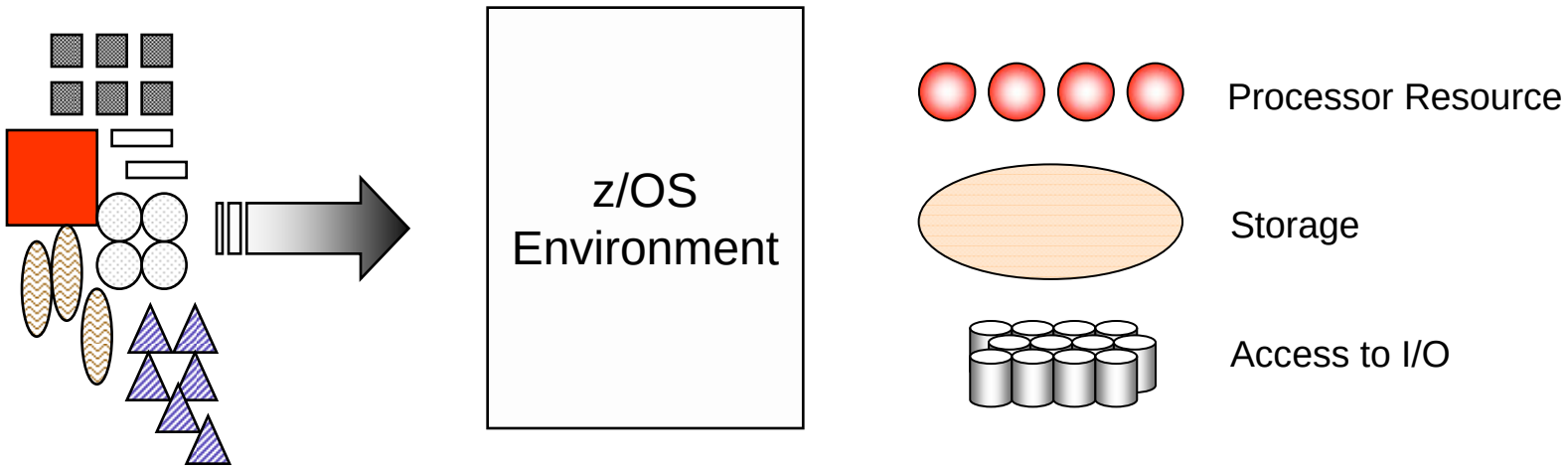
- A report classes are groupings of work used for installation monitoring
  - Division of work into separate WLM defined report classes has no effect WLM goal or resource management
  - Defined as an installation sees fit and is strictly for reporting purposes



Why use report classes?

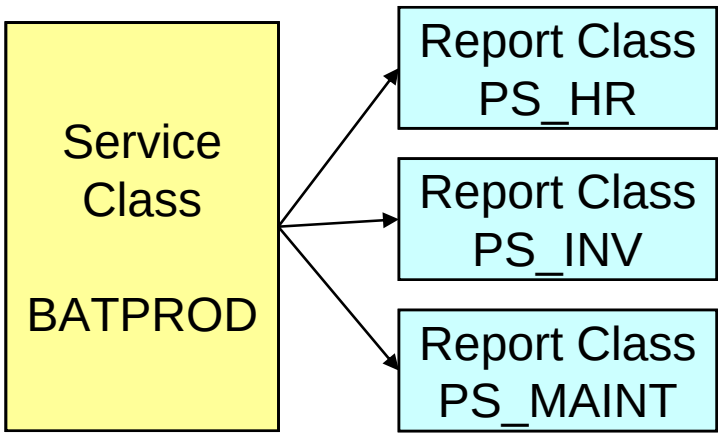
- As a reminder, the performance objectives of any installation are
  - Optimize the use of all system resources across all system in Sysplex
    - Make management happy
  - Enable all work in the system to achieve the installation's run time objectives
    - Make the customers happy

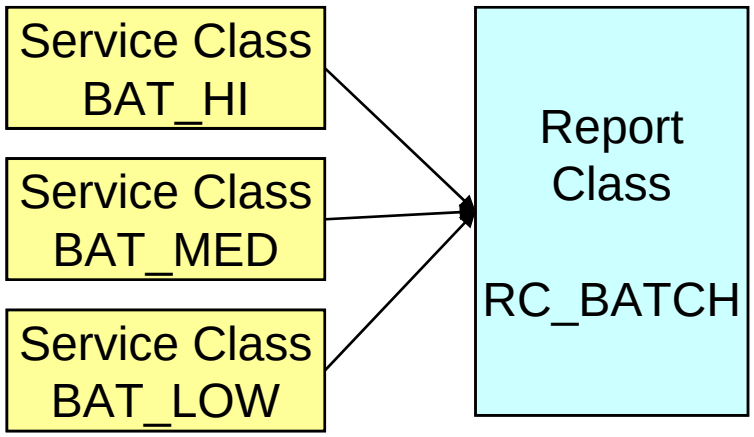
- Report classes provide more (or less) granular measurements to assist in
  - evaluation of workload resource usage
  - evaluation of goals, importance levels, multiple periods, etc



Why use report classes?

- Report classes are used to allow:
  - Reporting of workloads on a *more granular* level than service class
    - Example: Breakdown of service class measurements into smaller groupings
  - Reporting of workloads at a *less granular* level than service classes
    - Example: Combine the measurements of work in several service classes to a larger set





Overview of Report Classes

- Report class measurements are contained in the SMF 72 records
  - Very low overhead, but does result in SMF records

- SMF 72 report class measurements are mostly the same set of measurements available to service classes
  - Throughput data
  - Velocity achieved data
  - Response time and response time distribution data
  - Resource consumption data (CPU, Storage, I/O)
  - State samples
  - Work manager delay data

- A unit of work can
  - only be classified and managed into a single service class
  - optionally classified into a single report class
  - Thus, a unit of work can never be in more than service class and never be in more than one report class
    - So no double accounting possible

Simple Example of Using Report Classes

- Report class definition is made up of a name and a description
- When work is classified it is optionally also classified to a report class
  - In the below example all work classified to service class TSO
    - But work is broken up into several report classes so measurements are more granule

```
Subsystem-Type  Xref  Notes  Options  Help
------------------------------------------------------------------------------
                 Modify Rules for the Subsystem Type       Row 1 to 4 of 4
Command ===> _____________________________________________ SCROLL ===> PAGE

Subsystem Type . : TSO         Fold qualifier names?   Y  (Y or N)
Description  . . . TSO Workload Classification

Action codes:   A=After     C=Copy        M=Move     I=Insert rule
                B=Before    D=Delete row  R=Repeat   IS=Insert Sub-rule
                                                        More ===>
          --------Qualifier--------               -------Class--------
Action    Type       Name     Start                Service    Report
                                              DEFAULTS: TSO        RDEFTSO
 ____  1  UI         TSOS*    ___                  TSO        RTSOSYS
 ____  1  UI         TSOP*    ___                  TSO        RTSOPROD
 ____  1  UI         TSOD*    ___                  TSO        RTSODEV
****************************** BOTTOM OF DATA ********************************
```

Simple Example of Not Using Report Classes

- But report classes are totally optional
  - In the below example, none of the work assigned to service class TSO will be reported in any report class

```
Subsystem-Type  Xref  Notes  Options  Help
------------------------------------------------------------------------------
                 Modify Rules for the Subsystem Type       Row 1 to 4 of 4
Command ===> _____________________________________________ SCROLL ===> PAGE

Subsystem Type . : TSO         Fold qualifier names?   Y  (Y or N)
Description  . . . TSO Workload Classification

Action codes:   A=After     C=Copy        M=Move     I=Insert rule
                B=Before    D=Delete row  R=Repeat   IS=Insert Sub-rule
                                                        More ===>
          --------Qualifier--------               -------Class--------
Action    Type       Name     Start                Service    Report
                                              DEFAULTS: TSO      _______
 ____  1  UI         TSOS*    ___                  TSO        _______
 ____  1  UI         TSOP*    ___                  TSO        _______
 ____  1  UI         TSOD*    ___                  TSO        _______
****************************** BOTTOM OF DATA ********************************
```

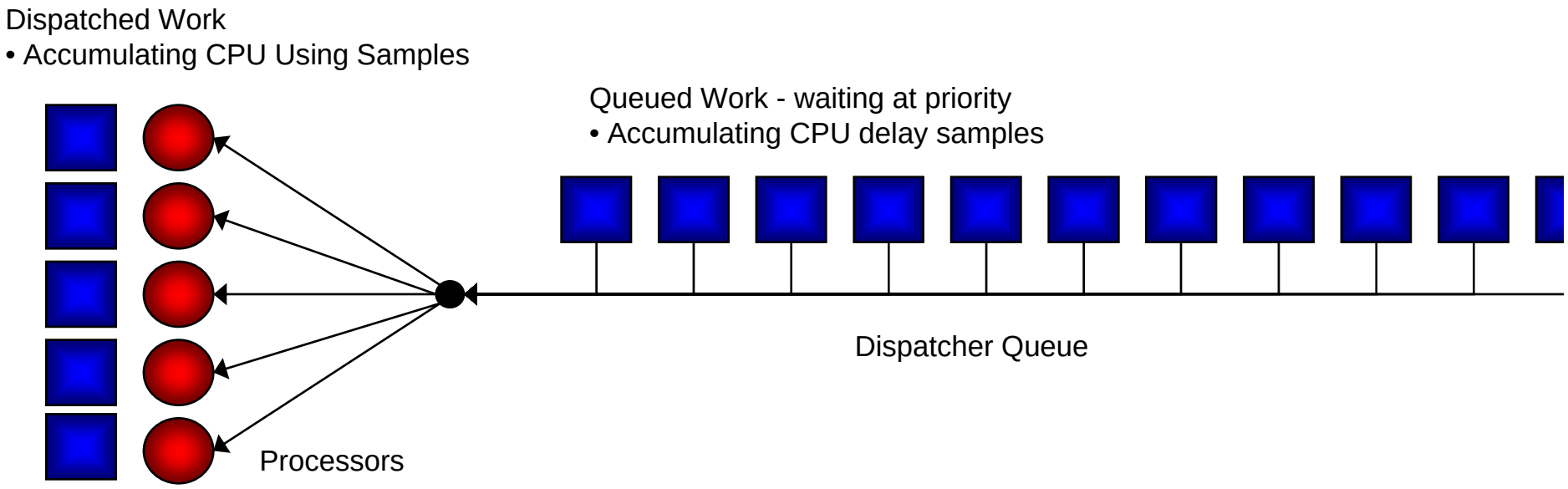
History of Report Classes

- Just as a fun background story…
  - Today, a WLM service definition
    - Can have up to 100 service classes
    - Can have up to 2047 report classes
  - However, the first release of WLM had limits that were very different
    - Could have up to 999 service classes
    - Could have up to only 100 report classes

    The thought was that units of work would be assigned their own service classes and WLM management was to be very granular
  - But with the introduction of fair share dispatching to improve WLM responsiveness, it was realized that fewer service class (periods) would result in better WLM management of workloads
    - So fewer service classes were expected, but more report classes was required
  - So the limits were switched
    - It was much later than the report class limit was raised from 999 to 2047

Fair Share Dispatching

- Fair share dispatching
  - At a given CPU dispatching priority
    - Each unit of work gets fair share access to the processors
  - Eliminates the need for work to be secluded to a period for micromanagement of access to the CPU
  - Lesson:
    - It is OK that like work is grouped and managed in the same service class period



General Report Class Setup Guidelines

General Guidelines

- A report class setup should be as sophisticated as the reporting intentions
  - First… Report classes are not required
    - From a technical point-of-view, if you have zero report classes defined, or 500 report classes defined, their existence or absence will have no affect on WLM management of the workloads or resources
    - However, at the very least it, is helpful to have report classes defined to assist in debugging and analysis of reporting issues
      - Examples given later in this presentation
  - Second… It is never too late to change your report class structure
    - The degree of difficulty to change an established report class structure is directly related to how much report classes are being used at your installation for reporting
      - Installations that make heavy use of report classes and have lots of daily / weekly / monthly / YTD reports will have a much more difficult time making report class changes
      - Installations that do not heavily depend on reports that use report classes will have a much easier time

Determine why report classes are needed

- Remember, all measurements are available at the service class level
  - So most performance analysis and debug can be done with service class measurements
  - When designing a report class setup, it is important to ask what the reporting needs are
  - Will they be used for evaluation of workload performance?
    - When a group of work is managed together, but insights into workload components is desired at a more granular level to better understand workload performance
    - Granular reporting of
      - Transaction throughput such as ended transactions
      - Velocities
      - Response time and response time distribution
      - Discretionary work performance
      - SYSTEM / SYSSTC work
      - Workload constraints
      - Etc.

Determine why report classes are needed

- Are the report classes to be used for evaluation of resource consumption?
  - For capacity planning purposes?
    - Example: Trending workload and resource growth for projection or verification purposes
  - For guidelines and insights tor MSU usage and pricing purposes?
    - Example: Breaking down work the R4HA was composed of
  - General debug and insights into what workloads are using the resources at a more granular level than at the service class level
    - Example: When work is to be managed together but more granular measurements are needed to evaluate what makes up that service class

```
--------Qualifier--------                -------Class--------
 Action    Type       Name     Start                Service     Report
                                          DEFAULTS: STCDEF      RDEFSTC
:
:
  _____  1  TN         DB2P*    ___                  STCPROD     RDB2PROD
  _____  1  TN         MQPD*    ___                  STCPROD     RWMQPROD
  _____  1  TN         DLIPROD  ___                  STCPROD     RDLIPROD
****************************** BOTTOM OF DATA ********************************
```

Report class naming guidelines

- Report classes can be assigned nearly any 8 character name
  - Name (required) - Eight character identifier of the report class.
  - Description (optional) - An area of 32 characters to describe the report class
- Report class descriptions – optional, but highly recommended
- Report class names - Installations use different naming conventions
  - Most installation have no naming conventions
    - > Example: XCFAS, RTSO, JOE
  - Others use name of the transaction or address space
    - > Example: Report class XCFAS for address space XCFAS
  - Others start report class names with the letter R
    - > Example: RTSOPROD, RDB2P
  - Some use cryptic name but meaningful descriptions
    - > Example: Report class RSTCD22, Description: DB2PMSTR address space
  - – So long as report class measurements can be mapped and used, then all is OK

Do not assign report class names already being used by service classes

- please, Please, PLEASE!
  - Use report class names that are different than service class names
  - Confusing when report class names are the same as service class names

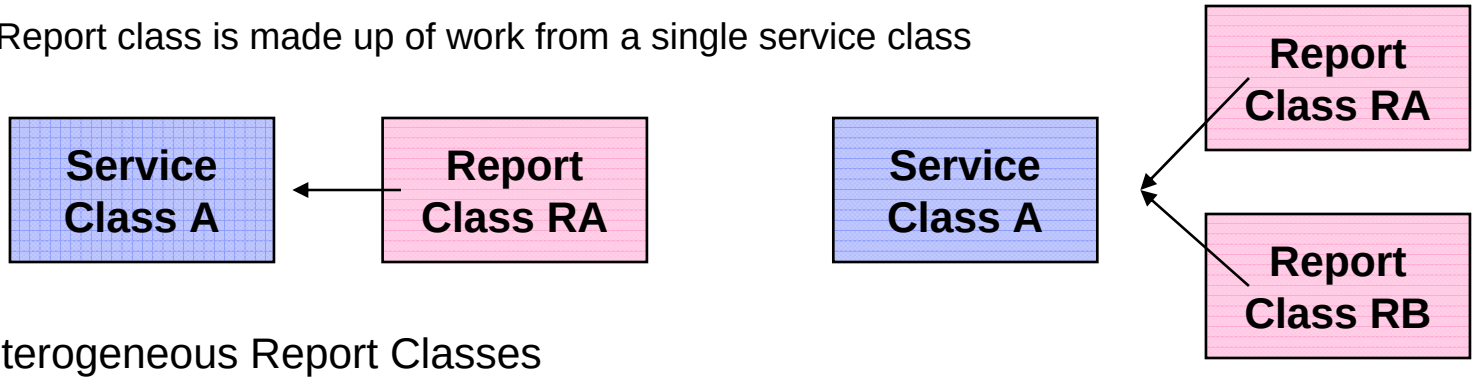
```
Subsystem-Type  Xref  Notes  Options  Help
------------------------------------------------------------------------------
                Modify Rules for the Subsystem Type      Row 1 to 8 of 47
Command ===> _____________________________________________ SCROLL ===> PAGE

Subsystem Type . : STC         Fold qualifier names?   Y  (Y or N)
Description  . . . Started Tasks

Action codes:   A=After     C=Copy        M=Move     I=Insert rule
                B=Before    D=Delete row  R=Repeat   IS=Insert Sub-rule
                                                        More ===>
          --------Qualifier--------               -------Class--------
Action    Type       Name     Start                Service     Report

(all other STC rules are here, and then last rule is as follows:

____  1  TNG        STCHI    ___                  STCHI       STCHI
____  1  SPM        SYSTEM   ___                  SYSTEM      SYSTEM
____  1  SPM        SYSSTC   ___                  SYSSTC      SYSTC
```

Consider renaming to
RSTCHI
DEFSYSTM
DFSYSSTC

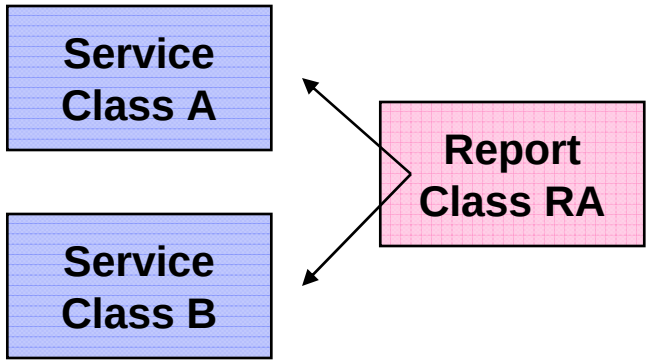
Heterogeneous versus Homogeneous report classes

- Report classes are either heterogeneous or homogenous relative to the service class where the work in the report class is classified

  - Homogenous Report Classes
    - Report class is made up of work from a single service class

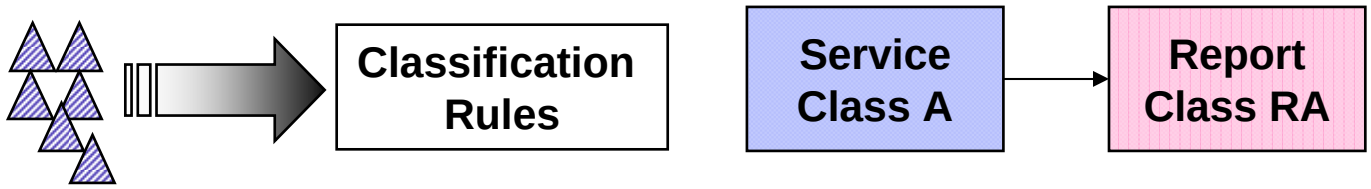


  - Heterogeneous Report Classes
    - Report class is made up of work from multiple service classes
    - Even if goal of the different service class is the same



Make sure all report classes are homogenous

- Classification rules can assign work to report classes
  - Work can only be assigned to one report class



- Problems
  - Difficult to monitor goals using heterogeneous report classes
    - Since work in report class could be managed to different goals and different importance levels
  - Report classes can be multiple period like their associated service class
    - But heterogeneous report classes associated with different multiple period service classes can result in misleading evaluations and measurements
  - Response time distribution data in heterogeneous report classes inaccurate if the response time goals of the associated service classes are different

Example
Heterogeneous versus Homogeneous

- Ensure that all work classified to a report class comes from a single service class
  - In the below example
    - PRODBAT is a heterogeneous report class - it contains work from both BATCHHI and BATCH
    - TESTBAT is a homogeneous report class - it contains work from just BATCH

```
Subsystem-Type  Xref  Notes  Options  Help
------------------------------------------------------------------------------
                Modify Rules for the Subsystem Type       Row 1 to 4 of 4
Command ===> _____________________________________________ SCROLL ===> PAGE

Subsystem Type . : JES         Fold qualifier names?   Y  (Y or N)
Description  . . . JES2 Batch

Action codes:   A=After     C=Copy        M=Move     I=Insert rule
                B=Before    D=Delete row  R=Repeat   IS=Insert Sub-rule
                                                        More ===>
          --------Qualifier--------                -------Class--------
Action    Type       Name     Start                 Service     Report
                                          DEFAULTS: BATCH       _________
 _____  1  TN         BOBA*    ___                   TPNS        _________
 _____  1  TN         BACK*    ___                   BATCHHI     PRODBAT
 _____  1  TN         COMP*    ___                   BATCH       PRODBAT
 _____  1  TN         TEST*    ___                   BATCH       TESTBAT
****************************** BOTTOM OF DATA ********************************
```

SMF data indicates if report class is heterogeneous or homogeneous

Class Type	Class Name	Heterogeneous?
RC	ASYNC	N
RC	BATRPT	N
RC	BATRPT	Y
RC	CPSM	N
RC	DCSDB2	Y
RC	DDFRPT	N
RC	DPAPP1	N
RC	DPAPP2	N
RC	DPAPP3	N
RC	DPCOL1	N
RC	EDI	N
RC	EXCI	N
RC	FOR	N
RC	MONITORS	N
RC	MONITORS	Y
RC	MQRPT	Y
RC	NEONRPTO	Y
RC	NEONRPTW	Y
RC	NEONRPTX	Y
RC	NEONRPTY	Y
RC	NEONRPTZ	N
RC	NETWK	Y
RC	OMVS	Y
RC	OMVSDFLT	N
RC	ONLRPT	Y
RC	P11ADM	N
RC	P11BTC	N
RC	P11DB2	Y
RC	P11DIA	N
RC	P11GEN	N
RC	P11ICL	N
RC	P11SMQ	N
RC	P11SPO	N
RC	P11STPR	N
RC	P11UP2	N
RC	P11UPD	N

Class Type	Class Name	Heterogeneous?
RC	PF1DDF	N
RC	PF1DIA	N
RC	PF1SPO	N
RC	PF1STPR	N
RC	PF1UP2	N
RC	PF1UPD	N
RC	PG1ADM	N
RC	PG1BTC	N
RC	PG1DB2	Y
RC	PG1DDF	N
RC	PG1DIA	N
RC	PG1DRDA	N
RC	PG1SPO	N
RC	PG1STPR	N
RC	PG1UP2	N
RC	PG1UPD	N
RC	PH1ADM	N
RC	PH1BTC	N
RC	PH1DB2	Y
RC	PH1DDF	N
RC	PH1DIA	N
RC	PH1SPO	N
RC	PH1STPR	N
RC	PH1UP2	N
RC	PH1UPD	N
RC	PM1ADM	N
RC	PM1BTC	N
RC	PM1DB2	Y
RC	PM1DDF	N
RC	PM1DIA	N
RC	PM1SPO	N
RC	PM1STPR	N
RC	PM1UP2	N
RC	PM1UPD	N
RC	POR	N
RC	PR1ADM	N

Class Type	Class Name	Heterogeneous?
RC	PS1ADM	N
RC	PS1BTC	N
RC	PS1DB2	Y
RC	PS1DDF	N
RC	PS1DIA	N
RC	PS1DRDA	N
RC	PS1SPO	N
RC	PS1STPR	N
RC	PS1UP2	N
RC	PS1UPD	N
RC	QOR	N
RC	SAPCICS	N
RC	SAPRPT	Y
RC	SRDS05M	N
RC	SRDS10S	N
RC	SRSD02M	N
RC	SRSD05M	N
RC	SRSERV	N
RC	STCRPT	Y
RC	SYNC	N
RC	TOR	N
RC	TSORPT	N
RC	TSORPT	Y
RC	ZOSRPT	Y
SC	BATCHHI	N
SC	BATCHLO	N
SC	DDF	N
SC	HOTBATCH	N
SC	KILLIT	N
SC	NEON	N
SC	NEWWORK	N
SC	ONLINEHI	N
SC	ONLINELO	N
SC	SAPBW	N
SC	SAPHI	N
SC	SAPLO	N

Use SMF 30 to figure out what address spaces in a report class

Job_Name	AS_Type	SC_Name	RC_Name	SYS1	SYS2
PQ0ACHIN	SYS	STCHI	MQRPT	380.19	
PQ0AMSTR	SYS	SYSSTC	MQRPT	404.52	
PQ1ACHIN	SYS	STCHI	MQRPT	448.57	
PQ1AMSTR	SYS	SYSSTC	MQRPT	1280.54	

Job_Name	AS_Type	SC_Name	RC_Name	SYS1	SYS2
CICPWUI1	SYS	SYSSTC	MONITORS	26.16	
MVCAS	SYS	SYSSTC	MONITORS	1.91	
SVOS	SYS	STCHI	MONITORS	29.13	
TMONDB2	SYS	SYSSTC	MONITORS		2.01
TMONDB2L	SYS	SYSSTC	MONITORS	170.15	73.96
TMONDLFL	SYS	SYSSTC	MONITORS	14.71	0.54
TMONDLFS	SYS	SYSSTC	MONITORS		0.55
TMONHUBP	SYS	SYSSTC	MONITORS	7.36	0.08
TMONHUBR	SYS	SYSSTC	MONITORS	0.28	0.32
TMONMLFS	SYS	SYSSTC	MONITORS	26.35	33.84
TMONMQS	SYS	SYSSTC	MONITORS	263.95	
TMONMSA	SYS	SYSSTC	MONITORS	2.28	
TMONMST	SYS	SYSSTC	MONITORS	313.31	342.1
TMONMVS	SYS	SYSSTC	MONITORS	24.82	280.43
TMONQLFS	SYS	SYSSTC	MONITORS	1.19	
TMONTCP	SYS	SYSSTC	MONITORS	11.75	4.88
TMONTLFS	SYS	SYSSTC	MONITORS	0.19	0.36
TMONULFS	SYS	SYSSTC	MONITORS	0.04	
TMONUSS	SYS	SYSSTC	MONITORS	0.01	
TMONVLFS	SYS	SYSSTC	MONITORS	3.24	
TMONVTM	SYS	SYSSTC	MONITORS	9.12	
XMANAGER	SYS	SYSSTC	MONITORS	4.79	7.4

These report classes are examples of report classes used to evaluate resource consumption.

In both these cases you may decided to live with the heterogeneous report classes

Just be careful when using report class measurements to evaluate goal

Example of RMF Report

```
                                        W O R K L O A D   A C T I V I T Y
                                                                                                          PAGE  47
   z/OS V1R12             SYSPLEX PRODPLEX          DATE 11/07/2014           INTERVAL 15.00.036    MODE = GOAL
                          RPT VERSION V1R12 RMF     TIME 12.45.00

                                    POLICY ACTIVATION DATE/TIME 05/14/2014 09.07.06

------------------------------------------------------------------------------------------------ REPORT CLASS PERIODS

REPORT BY: POLICY=DAILY02                    REPORT CLASS=ICMIGRAT                               PERIOD=1
                                             HOMOGENEOUS: GOAL DERIVED FROM SERVICE CLASS DDFLOW

-TRANSACTIONS-  TRANS-TIME HHH.MM.SS.TTT  --DASD I/O--  ---SERVICE---   SERVICE TIME  ---APPL %---  --PROMOTED--  ----STORAGE----
AVG      0.02   ACTUAL                21  SSCHRT   2.1  IOC          0  CPU    5.776  CP      0.51  BLK    0.000  AVG        0.00
MPL      0.02   EXECUTION             20  RESP     0.3  CPU     404829  SRB    0.000  AAPCP   0.00  ENQ    0.001  TOTAL      0.00
ENDED     428   QUEUED                 0  CONN     0.2  MSO          0  RCT    0.000  IIPCP   0.00  CRM    0.000  SHARED     0.00
END/S    0.48   R/S AFFIN              0  DISC     0.1  SRB          0  IIT    0.000                LCK    0.000
#SWAPS      0   INELIGIBLE             0  Q+PEND   0.0  TOT     404829  HST    0.000  AAP      N/A                -PAGE-IN RATES-
EXCTD       0   CONVERSION             0  IOSQ     0.0  /SEC       450  AAP      N/A  IIP     0.02                SINGLE      0.0
AVG ENC  0.02   STD DEV               20                                IIP    0.161                              BLOCK       0.0
REM ENC  0.00                                           ABSRPTN    29K                                            SHARED      0.0
MS ENC   0.00                                           TRX SERV   29K                                            HSP         0.0

GOAL: RESPONSE TIME 000.00.05.000 FOR  80%

         RESPONSE TIME EX    PERF  AVG   --EXEC USING%--  -------------- EXEC DELAYS % -----------  -USING%-  --- DELAY % ---    %
SYSTEM      ACTUAL%    VEL%  INDX ADRSP  CPU AAP IIP I/O  TOT CPU   Q                                CRY CNT  UNK IDL CRY CNT  QUI
                                                                   MPL
PRD2           100     46.7  0.5   0.0   33 N/A 0.0 0.0   37  35 2.3                                0.0 0.0   30 0.0 0.0 0.0  0.0

                                            ----------RESPONSE TIME DISTRIBUTION----------
   ----TIME----     --NUMBER OF TRANSACTIONS--    -------PERCENT-------  0    10   20   30   40   50   60   70   80   90   100
   HH.MM.SS.TTT     CUM TOTAL         IN BUCKET     CUM TOTAL   IN BUCKET  |....|....|....|....|....|....|....|....|....|....|
<  00.00.02.500          428               428           100         100  >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
<= 00.00.03.000          428                 0           100         0.0  >
<= 00.00.03.500          428                 0           100         0.0  >
<= 00.00.04.000          428                 0           100         0.0  >
<= 00.00.04.500          428                 0           100         0.0  >
<= 00.00.05.000          428                 0           100         0.0  >
<= 00.00.05.500          428                 0           100         0.0  >
<= 00.00.06.000          428                 0           100         0.0  >
<= 00.00.06.500          428                 0           100         0.0  >
```

Keep report classes homogeneous across subsystems as well

- Try to keep report classes are homogeneous to subsystem as well
  - If the same service class is used in two different subsystem classification rules, then make sure the report classes do not cross subsystems as well.

- Example Part A – Below we see that TSO work goes to report classes

```
Subsystem-Type  Xref  Notes  Options  Help
------------------------------------------------------------------------------
                Modify Rules for the Subsystem Type       Row 1 to 4 of 4
Command ===> _____________________________________________ SCROLL ===> PAGE

Subsystem Type . : TSO         Fold qualifier names?   Y  (Y or N)
Description  . . . TSO Workload Classification

Action codes:   A=After     C=Copy        M=Move     I=Insert rule
                B=Before    D=Delete row  R=Repeat   IS=Insert Sub-rule
                                                        More ===>
          --------Qualifier--------               -------Class--------
Action    Type       Name     Start                Service    Report
                                          DEFAULTS: TSO        RDEFTSO
 ____  1  UI         TSOS*    ___                  TSO        RTSOSYS
 ____  1  UI         TSOP*    ___                  TSO        RTSOPROD
 ____  1  UI         TSOD*    ___                  TSO        RTSODEV
****************************** BOTTOM OF DATA ********************************
```

Keep report classes homogeneous across subsystems as well cont…

- Rarely is it necessary to have a service class for OMVS interactive users
  - OMVS interactive users should be classified to the TSO service classes
  - However, make sure the report classes of these OMVS users are different so there is insight into what TSO work ran TSO environment and what ran OMVS environment
- Example Part B – Below we see that OMVS interactive user are classified to different report classes

```
Subsystem-Type  Xref  Notes  Options  Help
--------------------------------------------------------------------------
                 Modify Rules for the Subsystem Type       Row 1 to 4 of 4
Command ===> ____________________________________________ SCROLL ===> PAGE

Subsystem Type . : OMVS         Fold qualifier names?   Y  (Y or N)
Description  . . . USS Workload Classification

Action codes:   A=After     C=Copy        M=Move     I=Insert rule
                B=Before    D=Delete row  R=Repeat   IS=Insert Sub-rule
                                                             More ===>
          --------Qualifier--------                -------Class--------
Action    Type       Name     Start                 Service     Report
                                         DEFAULTS: TSO          RDEFOMVS
 ____  1  UI         TSOS*    ___                   TSO          ROMVSSYS
 ____  1  UI         TSOP*    ___                   TSO          ROMVSRRD
 ____  1  UI         TSOD*    ___                   TSO          ROMVSDEV
****************************** BOTTOM OF DATA ******************************
```

Make all default report classes are unique

- Ensure every defined set of subsystem classification rules have a report class
  - And make sure each report class for each subsystem type is unique
  - This helps identify any work that is not explicitly classified via a rule
  - Helps to gain insight into work that may not be well understood
- Go one step further and make sure the measurements for these default report classes are always zero
  - If RDEF* measurements are zero then all work is explicitly classified

```
Subsystem-Type  Xref  Notes  Options  Help
------------------------------------------------------------------------------
                 Modify Rules for the Subsystem Type       Row 1 to 1 of 1
Command ===> _____________________________________________ SCROLL ===> PAGE

Subsystem Type . : CB          Fold qualifier names?   Y  (Y or N)
Description  . . . WAS EE and Component Broker

Action codes:   A=After     C=Copy        M=Move     I=Insert rule
                B=Before    D=Delete row  R=Repeat   IS=Insert Sub-rule
                                                        More ===>
          --------Qualifier--------               -------Class--------
Action    Type       Name     Start                Service     Report
                                          DEFAULTS: DFLTWAS     RDEFWAS
 ____   1  TC        WASAPPL1 ___                  WAS_HI      RWASAPP1
****************************** BOTTOM OF DATA ********************************
```

Example of a subsystem type's classification rules

Make all default report classes are unique cont…

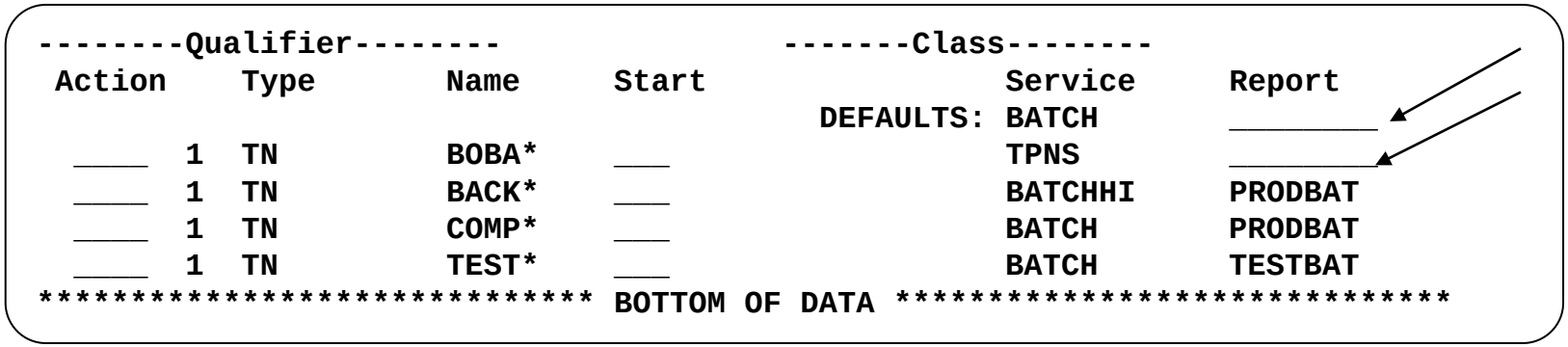
- It is recommended that only active subsystem types are defined
  - It makes for a simpler service definition
  - Work that exists for a subsystem type not used goes to SYSOTHER
    - So monitor SYSOTHER to make sure it is always void of activity
    - If not, then figure out the new type of work and add the necessary subsystem type and classification
    - But also use default report classes to determine what work is being classified by the default rules

```
Subsystem-Type  View  Notes  Options  Help
------------------------------------------------------------------------------
                Subsystem Type Selection List for Rules     Row 1 to 11 of 11
Command ===> ____________________________________________________________

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
              /=Menu Bar

                                                      ------Class-------
Action  Type      Description                         Service   Report
  __    CICS      CICS Transactions                   CICSHI    RDEFCICS
  __    DDF       All data_server requests            DDFDEF    RDEFDDF
  __    JES       JES2 Batch                          BATCHLO   RDEFBAT
  __    OMVS      Unix Services                       OMVS      RDEFOMVS
  __    STC       started Tasks                       STCLO     RDEFSTC
  __    TSO       Single service class                TSO       RDEFTSO
******************************* Bottom of data ********************************
```

Make sure all work is classified to a report class

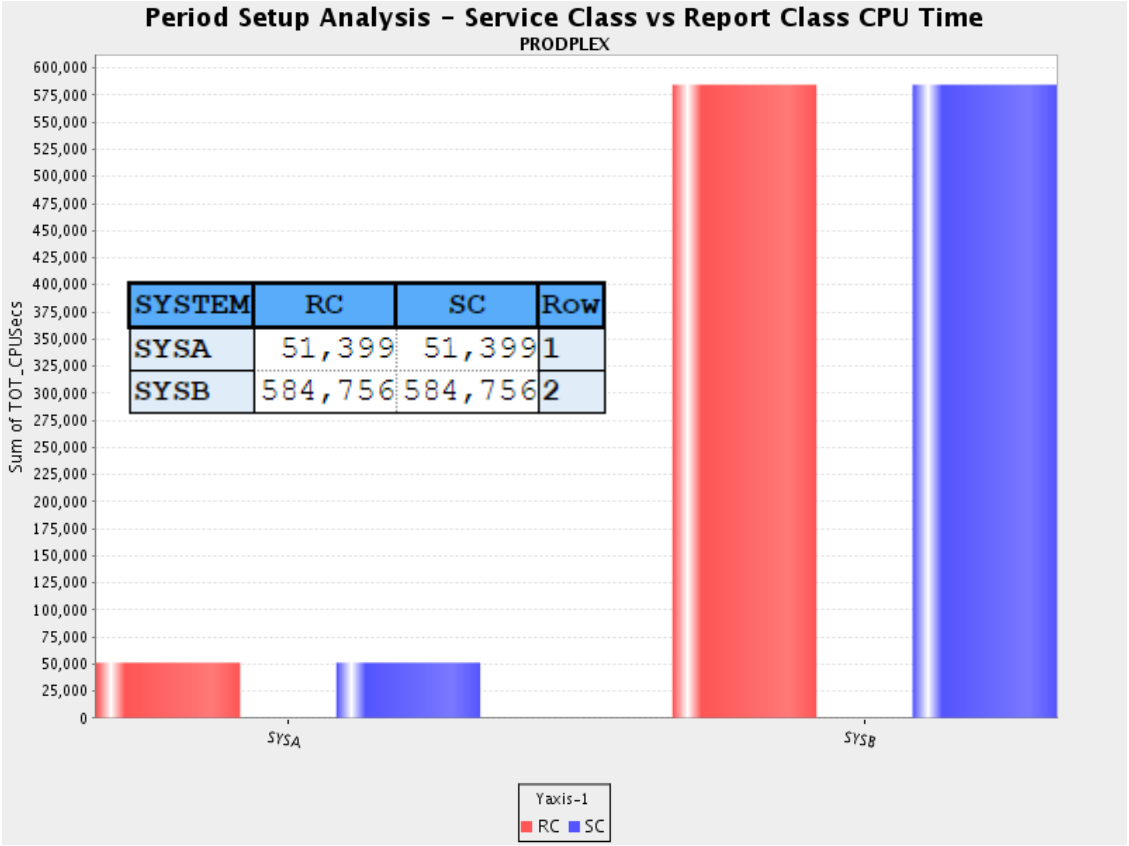
- Ensure that all work is classified to report classes
  - In theory, the total CPU time accumulated to service classes should equal the CPU time accumulated to report classes

- Benefits of all work classified to report classes
  - Service classes are setup for management purposes
    - But report classes allow a finer granularity of reporting
  - Helps with goal management of debugging
    - Work may be meeting is service class period goal, but finer granularity of reporting could show one part of the work doing well, but another part doing poorly



Comparing RC CPU seconds to SC CPU seconds

- Exercise: For a period of 24 hours, compare the summed CPU time of all report classes against summed CPU time of all service classes
  - If both numbers equal that good indicator that all work is assigned to a report class
  - Should do same exercise for number of transactions



SYSTEM	RC	SC	Row
SYSA	51,399	51,399	1
SYSB	584,756	584,756	2

Report Class Setup Guidelines

The foundation of a good report class setup is a good service class setup

- A good report class setup is based on a good service class setup
  - Reminder: Service classes are used to manage and report like work together
    - Work types
    - Performance goals
    - Resource requirements
    - Business importance to installation
  - Remember that not all work on the system is the same:
    - Interactive workloads versus background workloads
    - System workloads versus customer workloads
    - High importance workloads versus low importance workloads
    - High regular volume workloads versus low sporadic volume workloads
    - Short running work versus long running work
    - Distributed workloads versus single system workloads
    - e-business workload versus legacy workloads
    - And much more
  - If this separation is already achieved, and report classes are kept homo s a good foundation to built a report class structure upon

COMPBAT Service Class

Period 1
Goal = Velocity 15
Importance 4
RGRP = FENCED

PRODTSO Service Class

Period 1 – 500 Service
Goal = RT 0.5 sec, 95%
Importance 2
RGRP =

Period 2 – 1500 Service
Goal = RT 1.5 sec, 90%
Importance 3
RGRP =

Period 3
Goal = RT 3.0 sec, 80%
Importance 4
RGRP =

So, as a reminder…
Keep service classes homogeneous

- If you keep service classes homogeneous
  - Keep work in each service class relatively homogeneous
  - WLM takes different actions for different types of work to meet goals
  - Reports may not always reflect reality
  - Example: Don't mix CICS transactions in same service class as TSO

- Separate unlike work
  - Don't mix enclave work with non-enclave work
  - Don't mix interactive work with non-interactive work
  - Don't mix participants with non-participants
  - Don't mix server with non-servers
  - Don't mix regions managed towards region goal with regions managed towards transaction goals
  - Don’t mix Batch in WLM inits with batch in JES inits
  - Don't assign goals to spaces that should truly be in SYSTEM and SYSSTC
  - Don't put stuff into SYSTEM and SYSSTC that should not be there

Make sure report classes do not contain a mixture of different work unit types

- Continuing on the theme of ensuring report classes are homogenous
  - Do not mix into the same report class (or service class):
    - Address spaces and Enclaves and CICS / IMS transactions
  - Examples:
    - Do not put WAS transactions into same report class as WAS regions
    - Do not put CICS transactions into same report class as CICS regions
  - Exceptions:
    - OK for TCP and NETV work to be with STC work

Subsystems that use enclaves:
- **Component Broker**
- **DDF**
- **IBM HTTP Server**
- **MQSeries Workflow**
- **NetView**
- **LSFM**
- **TCP**
- **WebSphere**

Subsystems that have address space oriented transactions:
- **APPC**
- **JES2**
- **JES3**
- **UNIX System Services**
- **TSO/E**
- **Started Tasks**

Note: CICS and IMS subsystems do not use enclaves, but use a different set of services to support transactions to WLM.

STC report class setup

- Report class setup for STC work is especially important since the STC rules is where so many different workloads are classifies

- Separate into different groups of homogeneous report classes
  - Monitors
  - Production versus test
  - Various products
    - Example: DB2 regions, away from WMQ regions, away from WAS regions, etc..
  - Various subsystem instances of a product
    - Example: DB2 regions (MSTR, DBM1, DIST) for DB2P, away from DB2D, etc.
  - Certain SYSTEM and SYSSTC address spaces
    - Consider setting up report classes for the following address spaces since each has performance sensitivities
      - IXGLOGR, ZFS, GRS, WLM, XCFAS, etc.

STC report class setup…

- One or more report classes should be defined fore each of the possible groups of STC work
  - SYSTEM address spaces
  - SYSSTC address spaces
  - Monitors
  - DB2 regions
    - Separate by subsystem instance
    - Separate IRLM from other DB2 regions
    - Separate WLM stored procedure address spaces away from DB2 regions
    - Separate WLM stored procedure address space by application environment
  - CICS regions
    - Separate by subsystem instance
    - Separate TORs from AORs from FORs
    - Separate regions managed by velocity goals from those managed by transaction goals
  - IMS regions
    - Separate by subsystem instance
    - Separate DLISAS, DLI, DBRC, etc. from MPRs
    - Separate regions managed by velocity goals from those managed by transaction goals

STC report class setup…

- One or more report classes should be defined fore each of the possible groups of STC work
  - WMQ
    - Separate by subsystem instance
  - WAS
    - Separate by subsystem instance
    - Separate controllers from servants
    - Separate into their own report classes: NODE agents, Deployment managers, DAEMONs
  - Non-Participants each into their on report classes
    - ADABAS, IDMS, CTG, XCOM, VSAM RLS, etc

STC report class setup…

- Setup unique report classes for STC work being classified via SPM rules
  - Subsystem Parameter (SPM) rule for STC subsystem
    - Indicates that the system provided service class of SYSTEM or SYSSTC will be assigned if a started task with high dispatching priority, privileged, or system task attribute but is not assigned to a regular service class with goal

```
Subsystem-Type  Xref  Notes  Options  Help
------------------------------------------------------------------------------
                 Modify Rules for the Subsystem Type      Row 1 to 8 of 47
Command ===> _____________________________________________ SCROLL ===> PAGE

Subsystem Type . : STC         Fold qualifier names?   Y  (Y or N)
Description  . . . Started Tasks

Action codes:   A=After     C=Copy        M=Move     I=Insert rule
                B=Before    D=Delete row  R=Repeat   IS=Insert Sub-rule
                                                        More ===>
          --------Qualifier--------               -------Class--------
Action    Type       Name     Start                Service     Report

(all other STC rules are here, and then last rule is as follows:

____  1  SPM        SYSTEM   ___                  SYSTEM      DEFSYSTM
____  1  SPM        SYSSTC   ___                  SYSSTC      DEFSYSTC
```

JES (Batch) report class setup

- There are so many different types of batch
  - Normal Production Jobs submitted by a Job Scheduler
  - Critical Path Jobs submitted through a Job Scheduler
  - Ad-hoc Jobs (possibly submitted by a Job Scheduler)
  - Development Jobs
  - Normal System Support Jobs
  - High-Priority System Support Jobs
  - Logs, Archival, Backup, and D/R Jobs (possibly submitted by a Job Scheduler or an appropriate Subsystem)
  - Quick Utility Jobs
  - Emergency or Hot Jobs
  - Those requiring setup

- Report class recommendation
  - Separate jobs running in WLM managed inits away from JES managed inits
  - Figure out what batch reporting you need
  - Try to put applications into their own report classes
  - Try to separate using the above for batch types

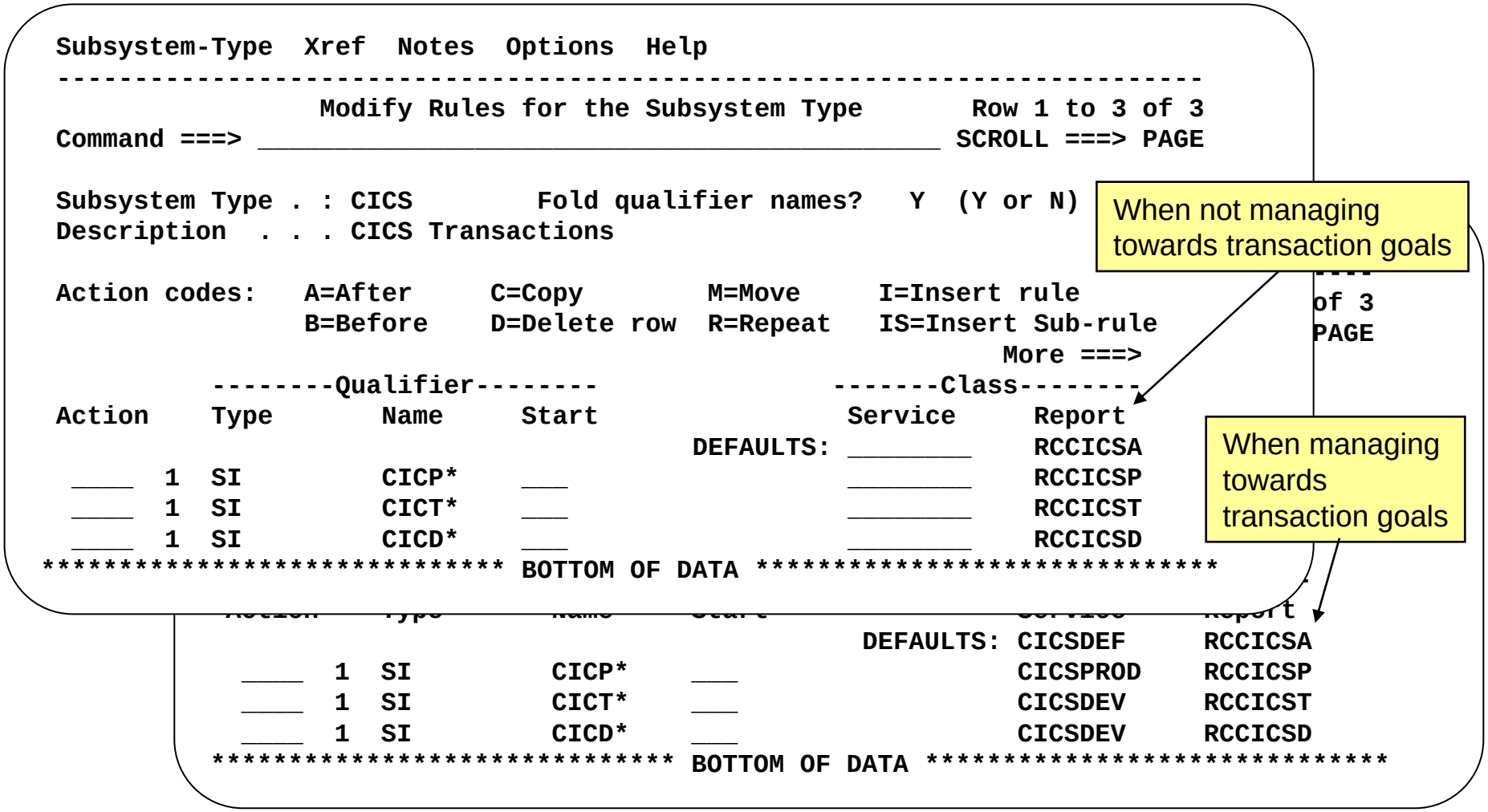
TSO, OMVS report class setup

- TSO
  - If helpful, separate users into different report classes as needed
  - Most common division is system programmers and all other users
  - Very useful to classify ‘problem’ users into their own report class
  - Example:
    - RTSOSYS, RTSOPRD, PETER, etc..

- OMVS
  - We need to remember there are distinct types of UNIX System Service work
    - Long running daemons
    - Interactive users
      - Production users or just system programmers doing their thing?
    - Batch Workloads
    - Things like JDBC and FTP users
  - Example:
    - ROMVSSYS, ROMVSPRD, ROMVSSTC, etc…

CICS and IMS Transactions

- Regardless if you are managing regions towards velocity goals or transaction goals, still classify all CICS and IMS transactions



Enclave transaction report class setup
DDF, CB (WAS),

- The world is your oyster!
  - Have fun!

- DDF – Distributed Data Facility transactions
  - Report classes are extremely useful to gain insights into transaction response time and resource consumption
  - A good DDF report class structure may help to avoid the need for some SMF 101 processing

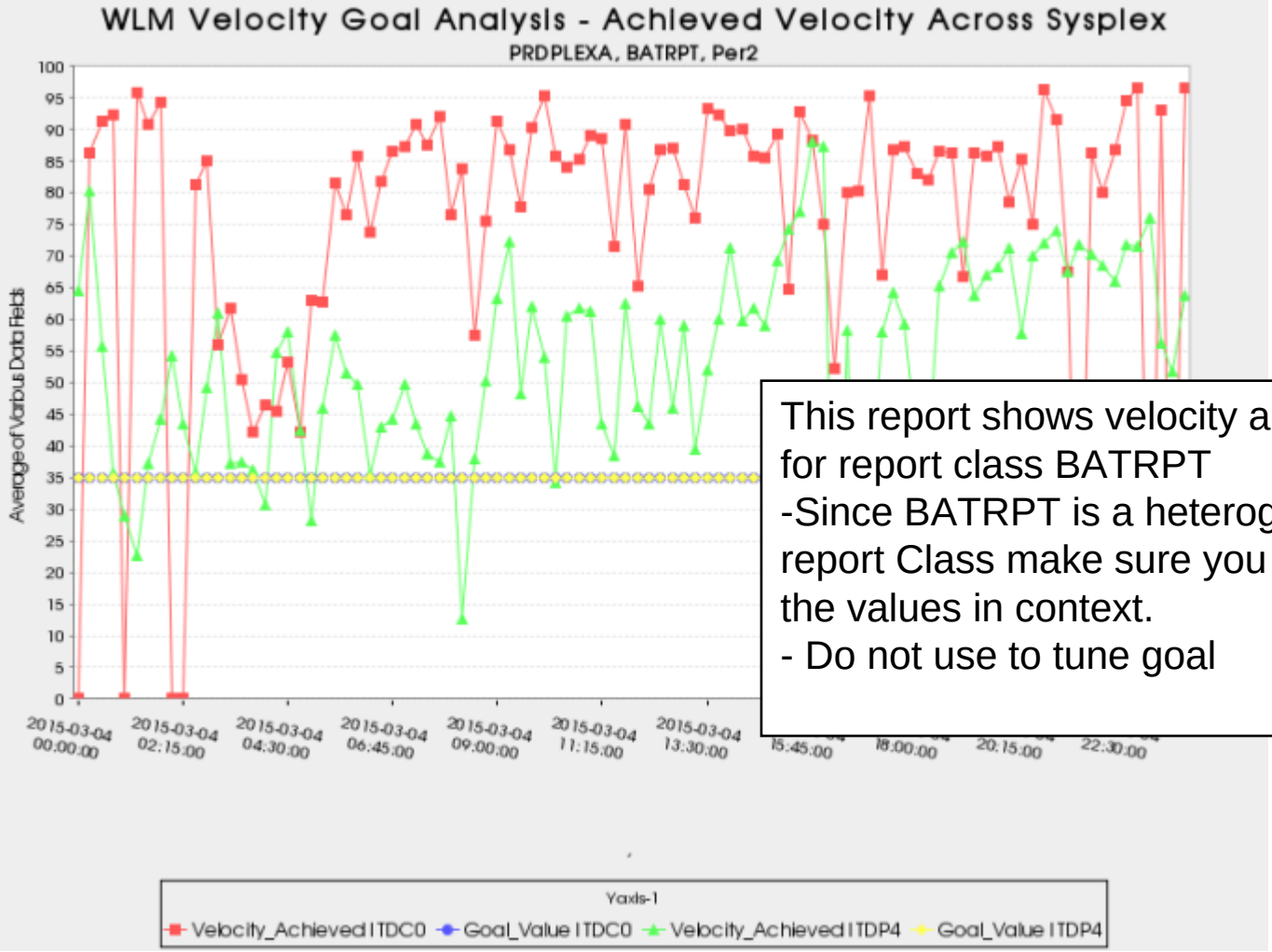
- CB – WebSphere Application Server transactions
  - Report classes are extremely useful to gain insights into transaction response time and resource consumption
  - A good CB report class structure may help to avoid the need for some SMF 120.9 processing

Enclave transaction report class setup
TCP, NETV, IWEB

- TCP and NETV
  - NETV
    - Includes all NetView network management subtasks and system automation subtasks created by Tivoli NetView for z/OS
  - TCP
    - Enclave work requests processed by the z/OS Communication Server
  - Neither is worthy of their own service class, so classify this work to a service class such as STCHI
    - One of the few cases when it is OK to mix enclave and address spaces into the same service class
    - However, still put each into their own report class to gain insights into their resource consumption
- IWEB
  - Most installations do not enable IWEB classification rules
    - Instead they disable enclaves in the httpd.conf file
  - If enclave transactions are enabled, then put this work into its own service class and report classes

Using Report Classes

Example of Velocity Goals
- BATRPT is a Heterogeneous report class



This report shows velocity achieved for report class BATRPT
-Since BATRPT is a heterogeneous report Class make sure you look at the values in context.
- Do not use to tune goal

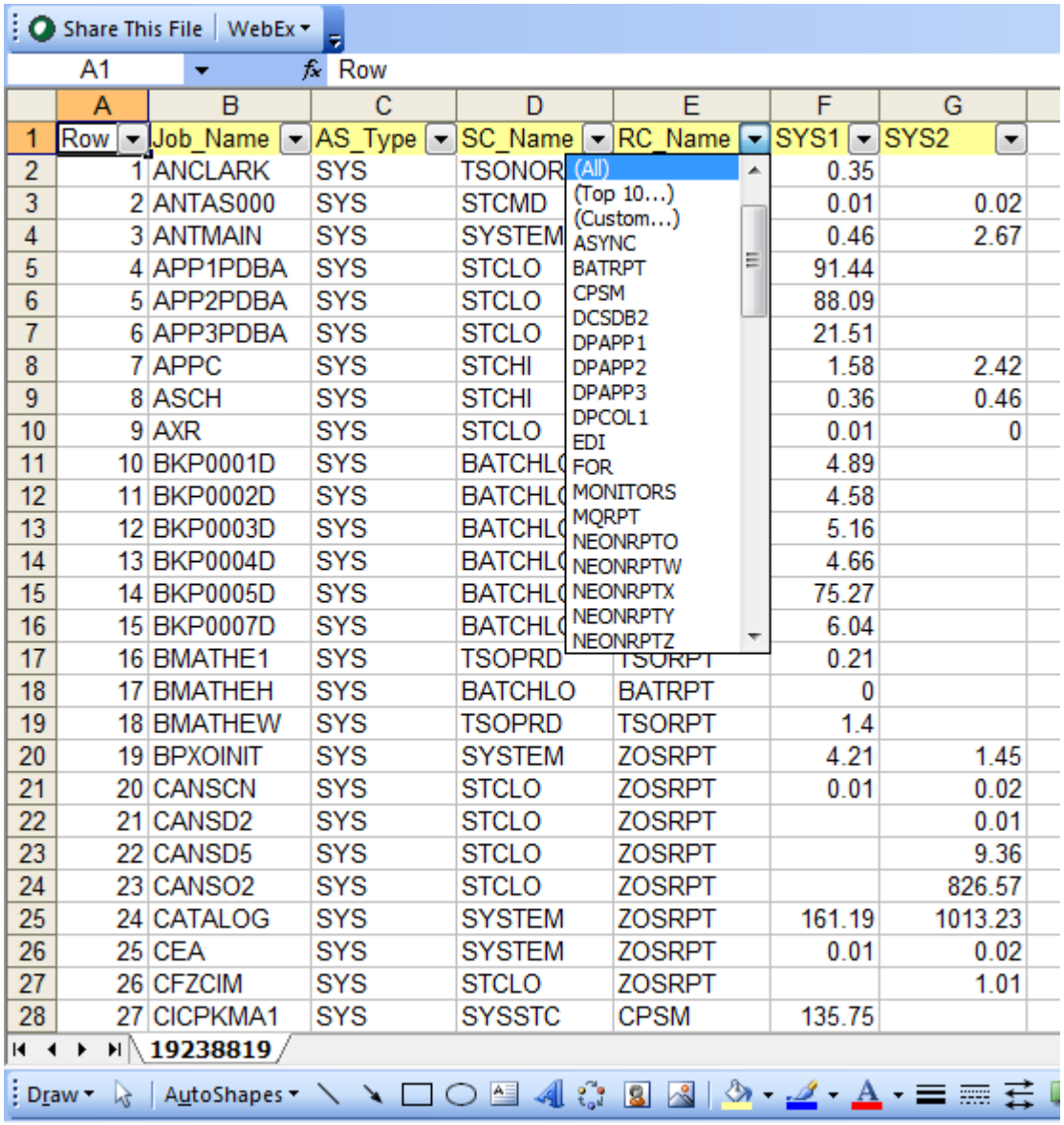
SMF 30 Measurements to Correlate to SMF 72.3

- 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.

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

Create a SMF30.2 to WLM Mapping



Share This File | WebEx

A1 | Row

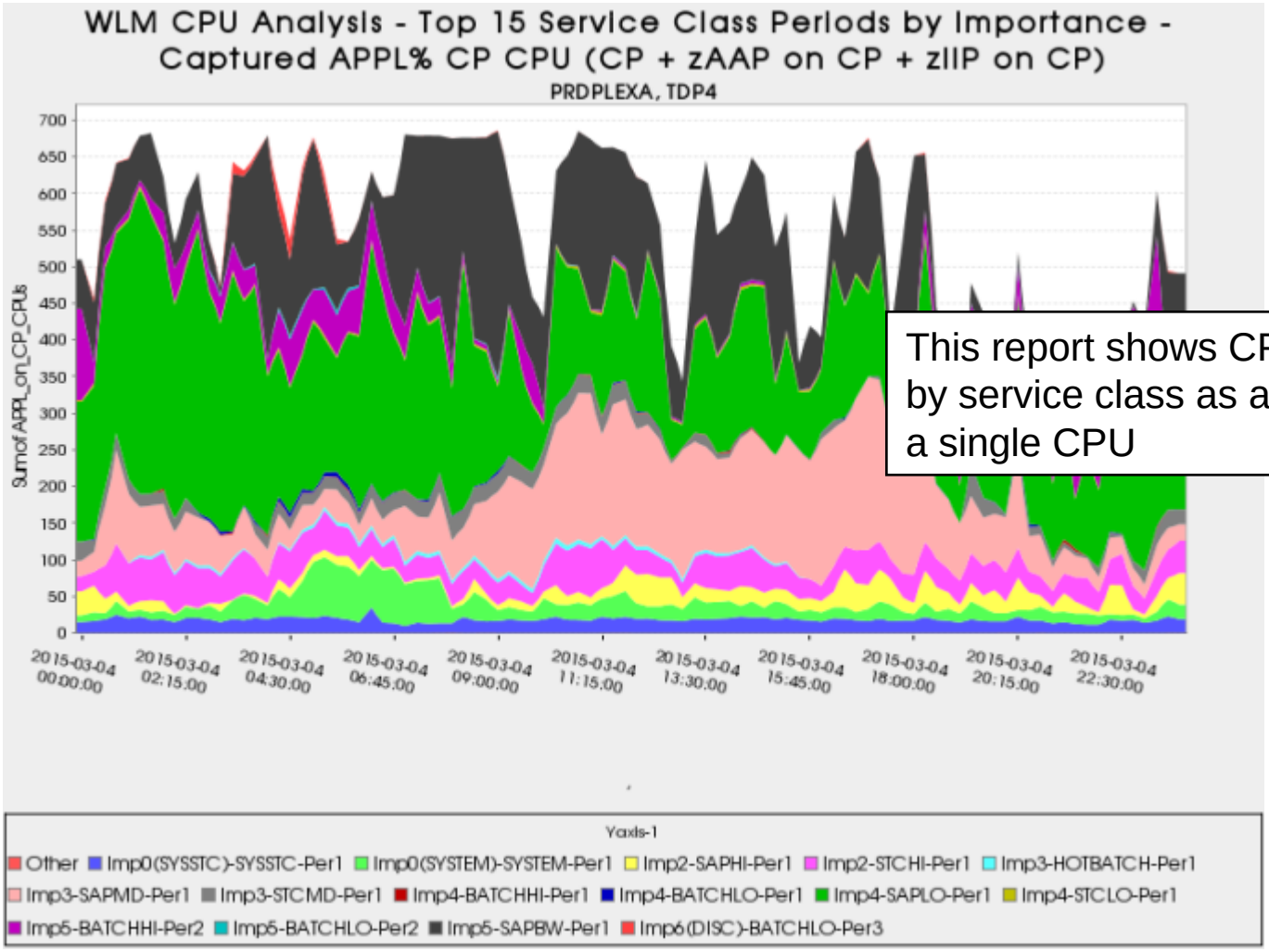
	A	B	C	D	E	F	G
1	Row	Job_Name	AS_Type	SC_Name	RC_Name	SYS1	SYS2
2	1	ANCLARK	SYS	TSONOR		0.35	
3	2	ANTAS000	SYS	STCMD		0.01	0.02
4	3	ANTMAIN	SYS	SYSTEM		0.46	2.67
5	4	APP1PDBA	SYS	STCLO		91.44	
6	5	APP2PDBA	SYS	STCLO		88.09	
7	6	APP3PDBA	SYS	STCLO		21.51	
8	7	APPC	SYS	STCHI		1.58	2.42
9	8	ASCH	SYS	STCHI		0.36	0.46
10	9	AXR	SYS	STCLO		0.01	0
11	10	BKP0001D	SYS	BATCHL		4.89	
12	11	BKP0002D	SYS	BATCHL		4.58	
13	12	BKP0003D	SYS	BATCHL		5.16	
14	13	BKP0004D	SYS	BATCHL		4.66	
15	14	BKP0005D	SYS	BATCHL		75.27	
16	15	BKP0007D	SYS	BATCHL		6.04	
17	16	BMATHE1	SYS	TSOPRD		0.21	
18	17	BMATHEH	SYS	BATCHLO	BATRPT	0	
19	18	BMATHEW	SYS	TSOPRD	TSORPT	1.4	
20	19	BPXOINIT	SYS	SYSTEM	ZOSRPT	4.21	1.45
21	20	CANSCN	SYS	STCLO	ZOSRPT	0.01	0.02
22	21	CANSD2	SYS	STCLO	ZOSRPT		0.01
23	22	CANSD5	SYS	STCLO	ZOSRPT		9.36
24	23	CANSO2	SYS	STCLO	ZOSRPT		826.57
25	24	CATALOG	SYS	SYSTEM	ZOSRPT	161.19	1013.23
26	25	CEA	SYS	SYSTEM	ZOSRPT	0.01	0.02
27	26	CFZCIM	SYS	STCLO	ZOSRPT		1.01
28	27	CICPKMA1	SYS	SYSSTC	CPSM	135.75	

(All)
(Top 10...)
(Custom...)
ASYNC
BATRPT
CPSM
DCSDB2
DPAPP1
DPAPP2
DPAPP3
DPCOL1
EDI
FOR
MONITORS
MQRPT
NEONRPTO
NEONRPTW
NEONRPTX
NEONRPTY
NEONRPTZ

19238819

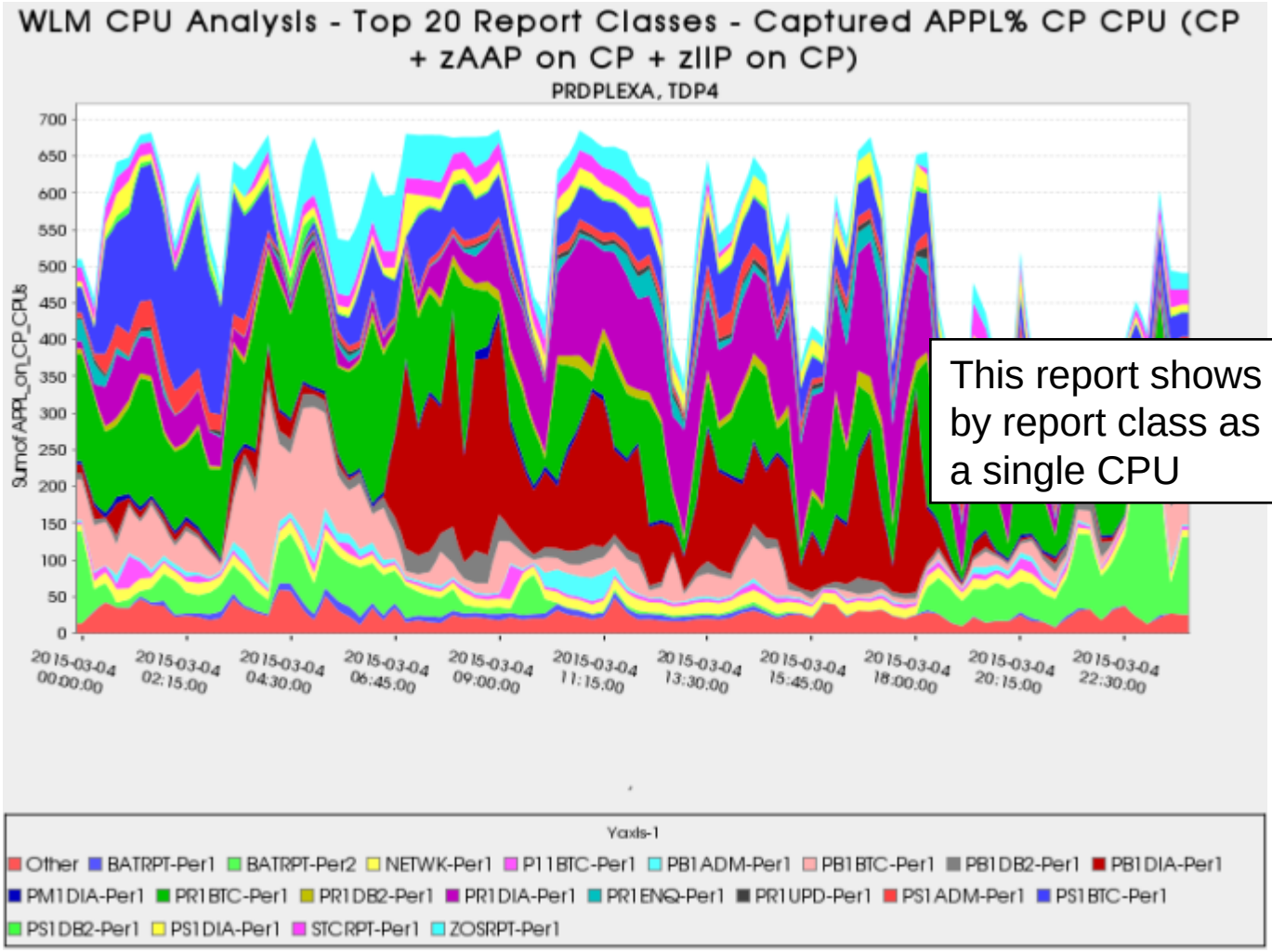
Draw | AutoShapes

Look at CPU consumption by service class



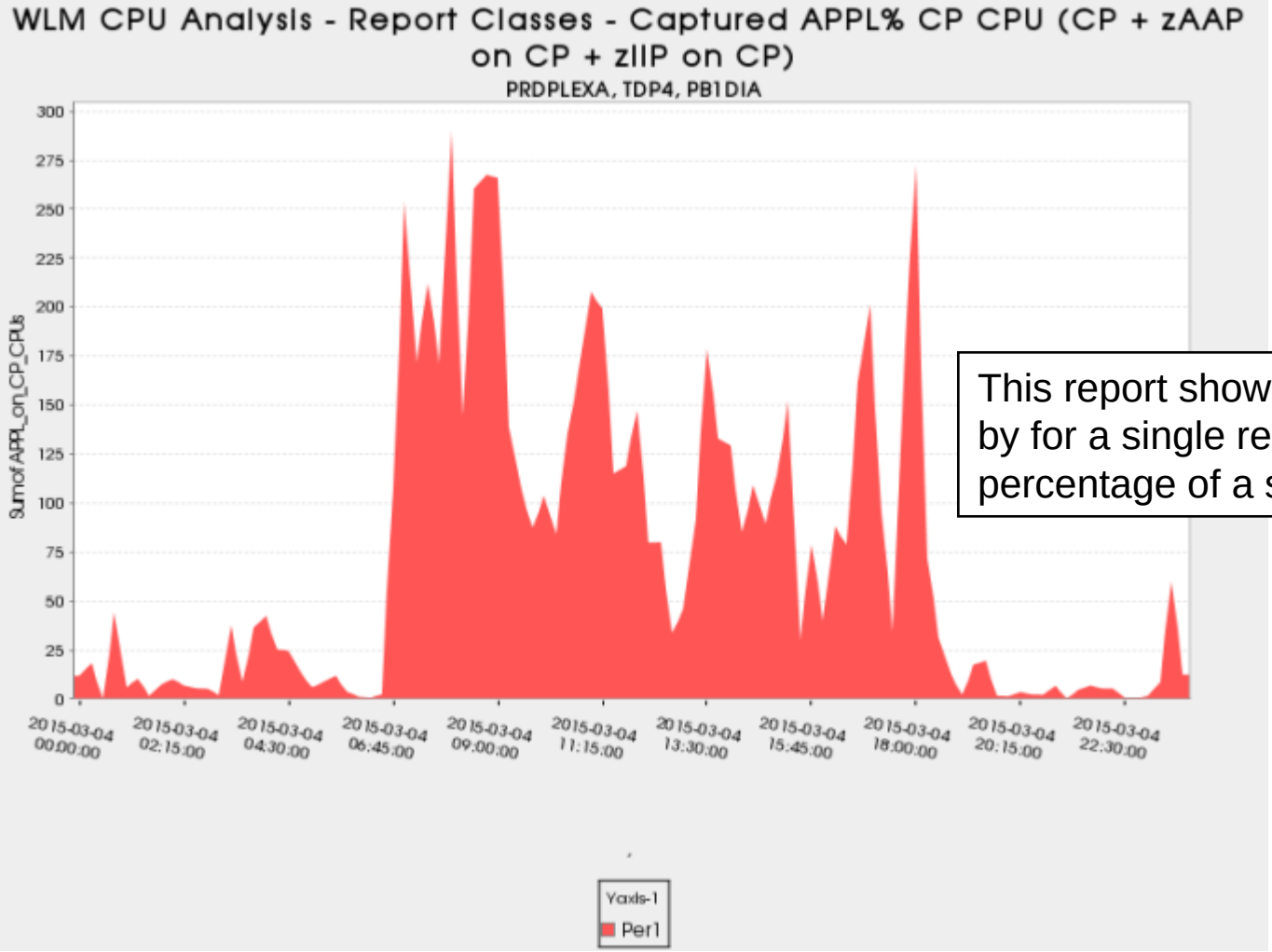
This report shows CPU consumption by service class as a percentage of a single CPU

CPU consumption by top report classes



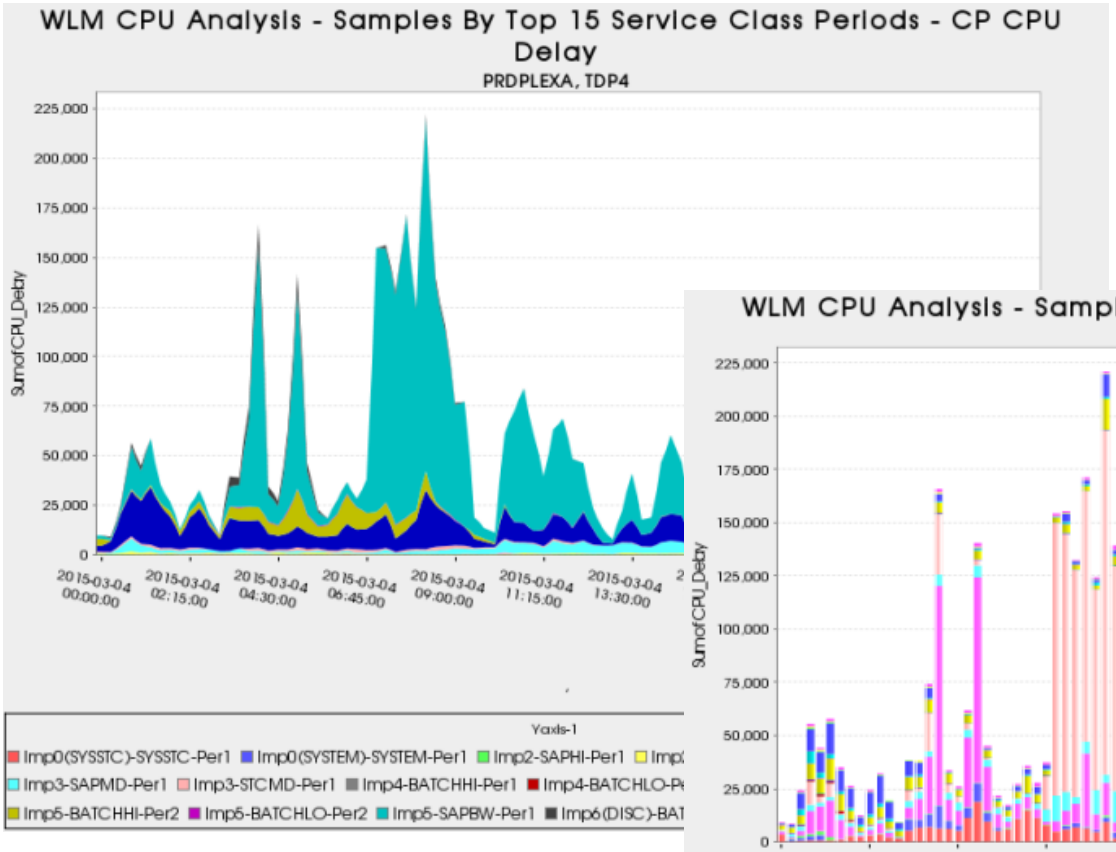
This report shows CPU consumption by report class as a percentage of a single CPU

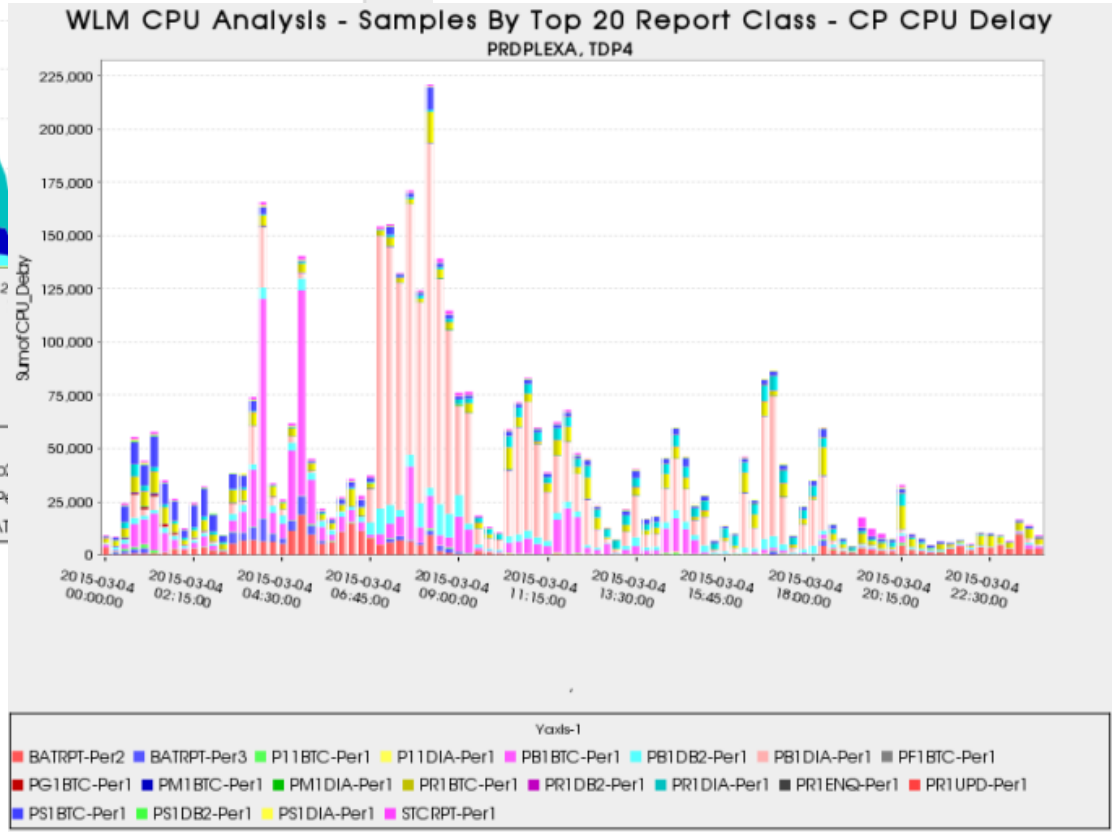
CPU Consumption for selected report class



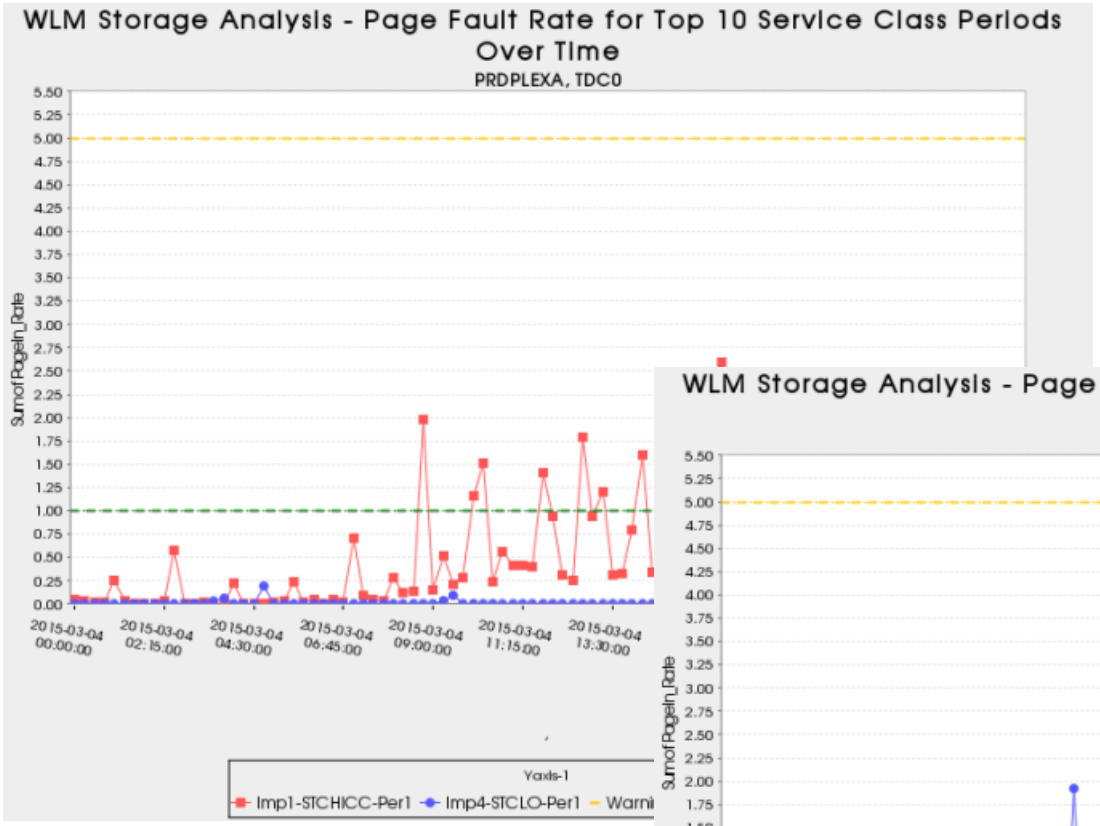
This report shows CPU consumption by for a single report class as a percentage of a single CPU

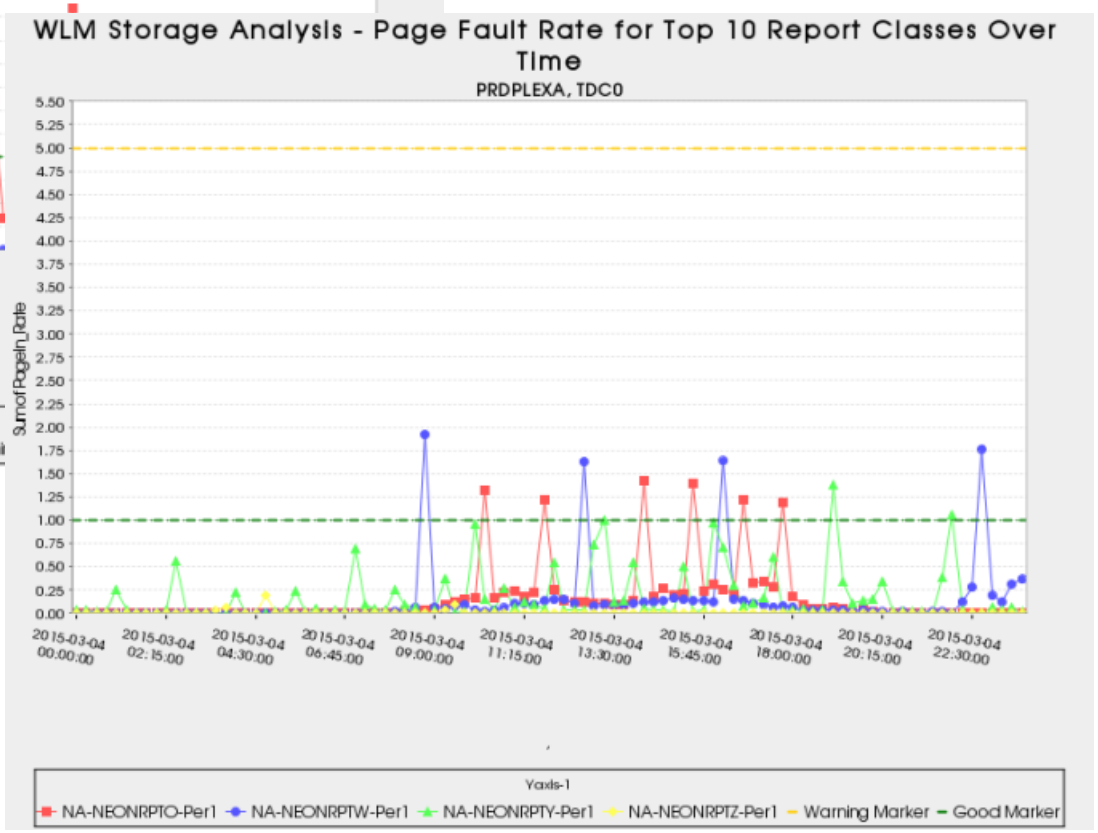
Example of CPU Delay Samples
- By Top Service Classes and Top Report Classes



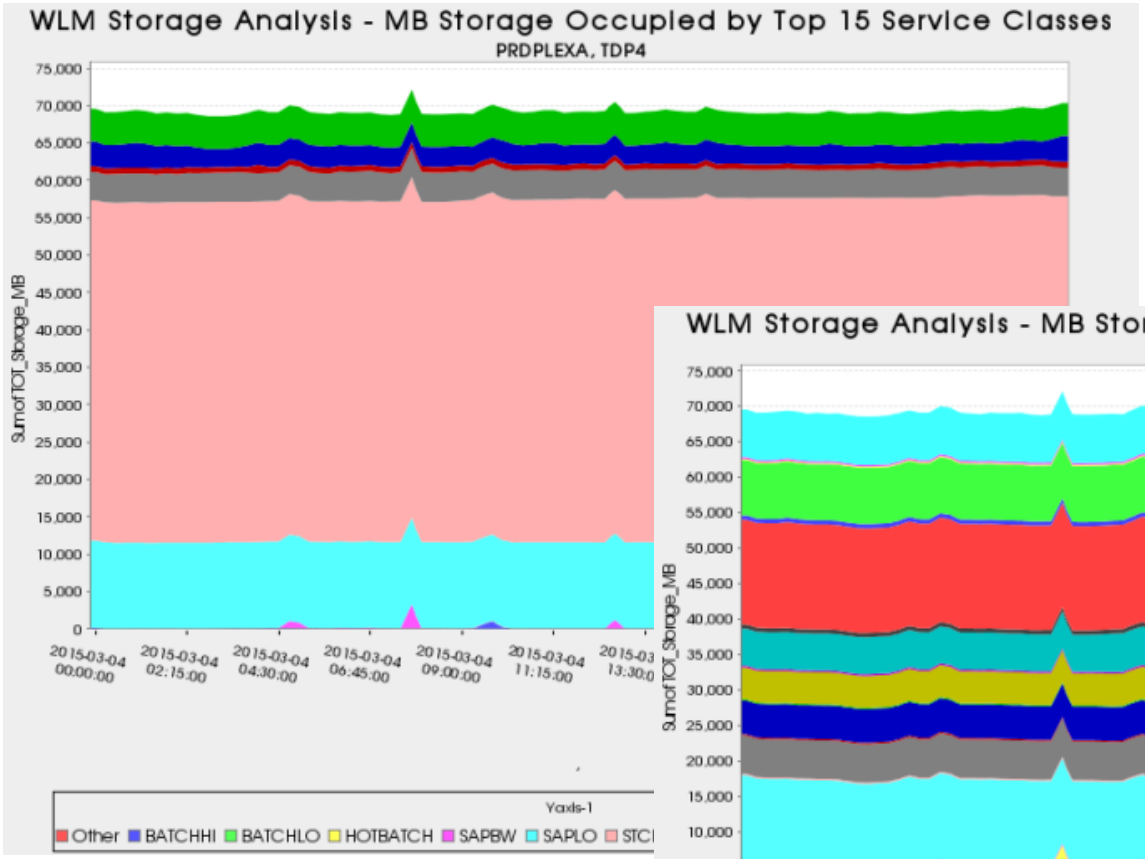


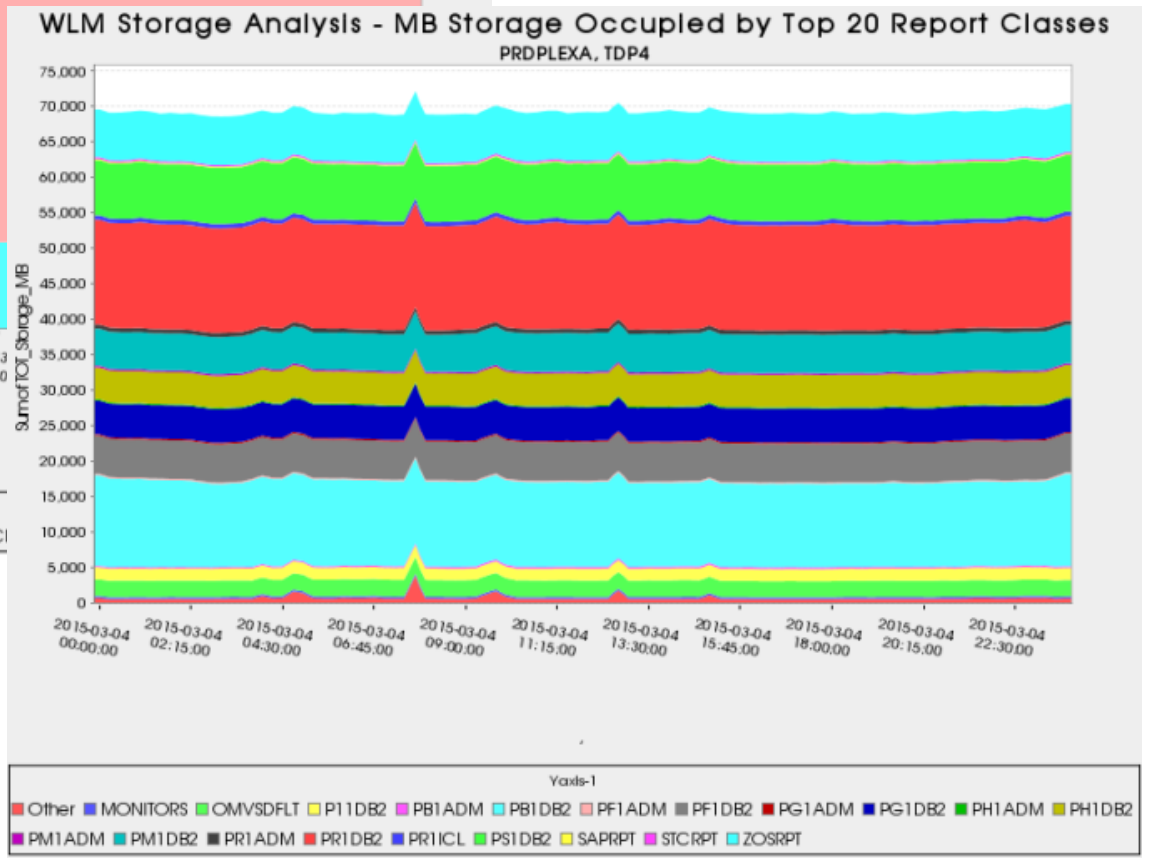
Example of Page Fault Rate
- By Top Service Classes and Top Report Classes





Example of Storage Megabytes
- By Top Service Classes and Top Report Classes





Performance Workshops Available

During these workshops you will be analyzing your own data!

- WLM Performance and Re-evaluating of Goals
  - Instructor: Peter Enrico and Scott Chapman
  - September 28 – October 2, 2015 – Columbus, Ohio, USA

- Parallel Sysplex and z/OS Performance Tuning
  (Web / Internet Based!)
  - Instructor: Peter Enrico and Scott Chapman
  - November 17 – 19, 2015

- Essential z/OS Performance Tuning Workshop
  - Instructors: Peter Enrico, Scott Chapman, Tom Beretvas
  - October 19 - 23, 2015 – Dallas, Texas, USA

- z/OS Capacity Planning and Performance Analysis
  - Instructor: Ray Wicks