Optimizing your Batch Window

Scott Drummond (spd@us.ibm.com)
Horst Sinram (sinram@de.ibm.com)
IBM Corporation

Wednesday, August 9, 2011
Session 9968

Agenda

- What differentiates “batch” workloads from online workloads?
- 7 Key Strategies for Batch Window Reduction
- z/OS Storage Enhancements for Batch Optimization
  - Optimizing I/O
  - Improving Application Efficiency
  - Increasing Parallelism
  - Reducing the Impact of Failures
- Tuning Knobs in the Workload / Performance Management area
  - Service Policy Overrides
  - Intelligent Resource Director
  - WLM-managed Batch Initiators
  - Scheduling Environments
  - Batch & online concurrently
    - Resource Groups, CPU Critical

What differentiates “batch” workloads from online workloads?

Batch processing *system for processing data with little or no operator intervention. This allows efficient use of the computer and is well suited to applications of a repetitive nature, such as file format conversion, payroll, or the production of utility bills.*

*In **interactive computing**, by contrast, data and instructions are entered while the processing program is running.*

Hutchinson Encyclopedia

Batch vs. Online

- Differences are highly dependent on workload specifics
- Sequential access pattern for batch vs. direct access for online
- Workload may move to different systems
- In batch window different devices may be used than during online (DASDs, tape)
- Business importance inversion: night shift vs. prime shift

7 Key Strategies for Batch Window Reduction

- Ensuring the system is properly configured
- Implementing data in memory (DIM) (not covered in this presentation)
- Optimizing I/O
- Optimizing parallelism
- Reducing the impact of failures
- Increasing operational effectiveness
- Improving application efficiency

Thanks to Martin Packer, IBM Mainframe Performance Consultant for use of his 7 strategies

z/OS Storage enhancements for Batch Optimization

This section is based on strategies from the IBM Redbook “Batch Modernization on z/OS” (sg24-7779), which in turn is based on ideas from “SG24-2557 Parallel Sysplex Batch Performance”

Optimizing I/O - I/O optimization is most important when the batch workload is I/O-intensive. Where there is little spare CPU capacity I/O optimization could cause an I/O bottleneck to be replaced by a CPU bottleneck.

Optimizing I/O

- Optimizing I/O means ensuring the I/O processing done by batch jobs is performed as quickly as possible, such as:
  - The correct use of DS8000 Cache functions – this can be monitored with OMEGAMON XE for Storage and RMF – For multi-operating system DS8000's use TPC for Disk
  - Exploiting modern Channel Programming functions such as zHPF and MIDAWs
  - Using the fastest FICON channels – 8 Gb/s
  - Using HyperPAV to minimize IOSQ time and take advantage of 3390 mod 9, "27", "54" and EAV ("A") format volumes

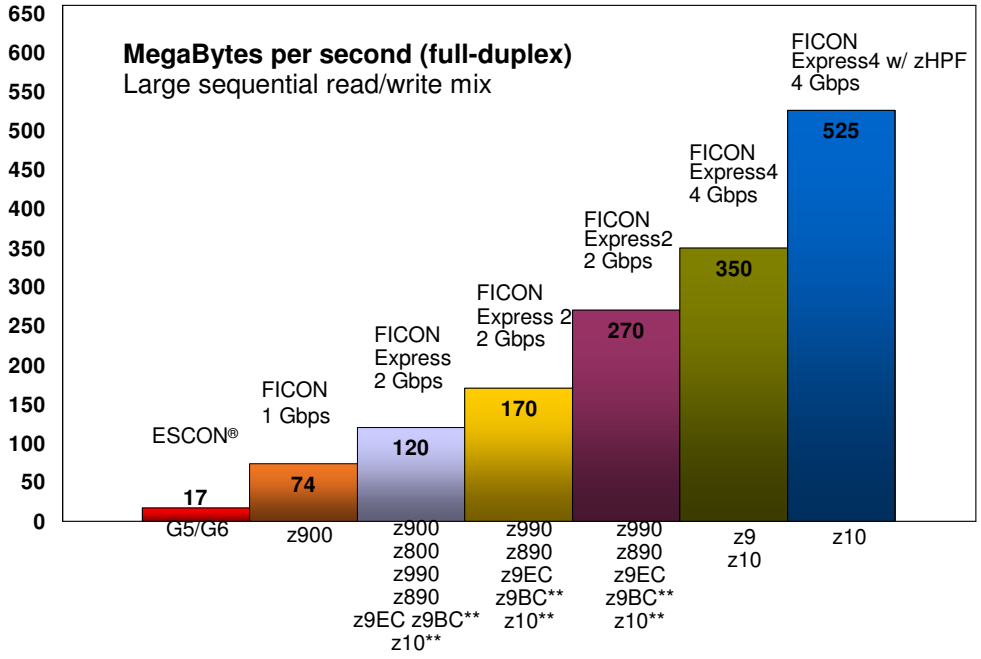
Optimizing I/O - continued

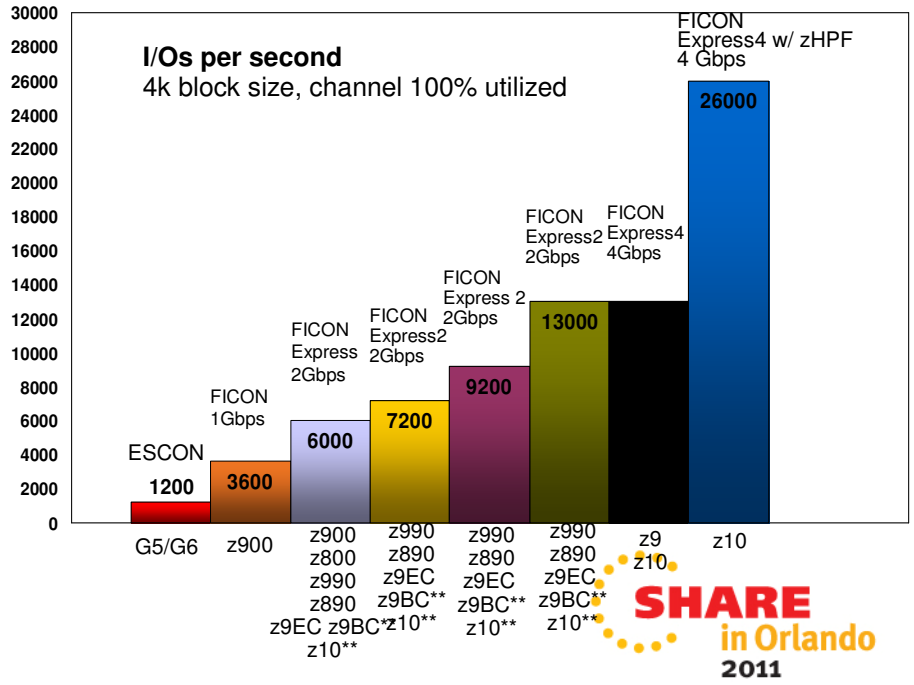
- Optimizing I/O means ensuring the I/O processing done by batch jobs is performed as quickly as possible, such as:
  - Using DS8000 DSN level FlashCopy for interim backup steps
  - Dataset separation by volume (DB2)
  - DFSMShsm and DFSMSrmm Fast Tape Positioning
  - DFSMShsm usage of VTL's (TS7720 and TS7740) and new - TS7680 De-Duplication support
  - VSAM buffering improvements and SMS construct usage
  - Avoid outages by using z/OS HyperSwap

Optimizing I/O:
High Performance FICON for System z (zHPF)

VALUE

- Performance/Growth - Up to 2x higher I/O throughput per channel
  - Realistic workloads with a mix of data set transfer sizes can see 30 – 70% of FICON I/Os utilizing zHPF, resulting in up to a 10-30% channel utilization savings.
- Management - Can reduce manual tuning
- TCO – May require fewer channels on the z server and on the disk system



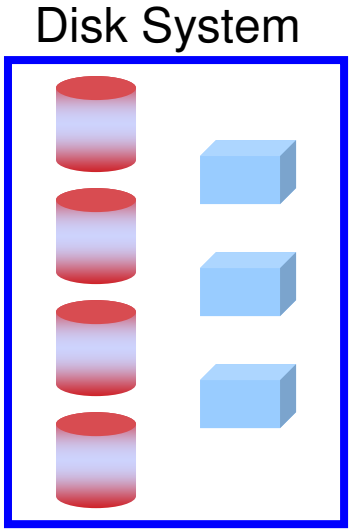


SHARE in Orlando 2011

Optimizing I/O: DFSMS and SSD

disk drive

SSD

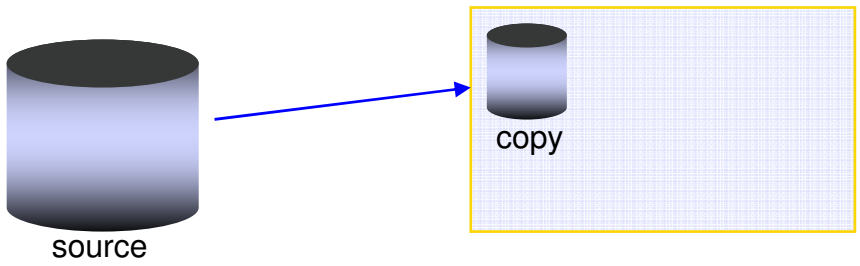


Value

- Management - Helps users allocate new data sets to the desired type of disk
- Management - Supports tools and commands that report on types of disk installed

Optimizing I/O:
IBM FlashCopy SE (Space-Efficient)

A technology optimization for short-term point-in-time copies



- Shared pool of capacity
- Holds unchanged source data only

Value

- Growth/TCO - Reduces physical capacity needed to hold volume copies
  - Target volume capacity savings can be up to 80%
- Growth/TCO - Helps reduce energy and floor space requirements

Optimizing I/O: IBM HyperSwap

Higher, cost-effective information availability for System z

VALUE

- Availability - “Hides” disk system outages from applications
- Management - Managed by Tivoli Storage Productivity Center for Replication (TPC-R) for System z

Improving application efficiency - Looking at ways to make the applications process more efficiently

Improving application efficiency

- Replacing self-written processing with DFSORT processing as this is optimized for speed.

- Older DFSORT enhancements that people might have missed
  - OUTFIL writes multiple outputs in a single pass over the input data
    - Note: 1 or zero sorts in a single invocation
  - Automatic BatchPipes/MVS detection for input and output data sets
  - ICETOOL - combines new features with previously available DFSORT features to perform complex sorting, copying, merging, reporting and analytical tasks using multiple data sets in a single job step.
  - Substring search, Bitwise operations and Arithmetic
  - Timestamps and Y2K conversions
  - And relative date arithmetic

- New DFSORT ICETOOL enhancements
  - DATASORT
    - An easy way to sort records between header and trailer records
    - Header and trailer records are left in place
  - SUBSET
    - An easy way to create a subset of the records, based on record number
    - And easy to send the complement of the subset to another data set

Increasing Parallelism - running more things alongside each other

Note: To be able to run more work in parallel, you need adequate resources, most importantly spare CPU capacity and memory, and adequate I/O bandwidth.

Increasing Parallelism

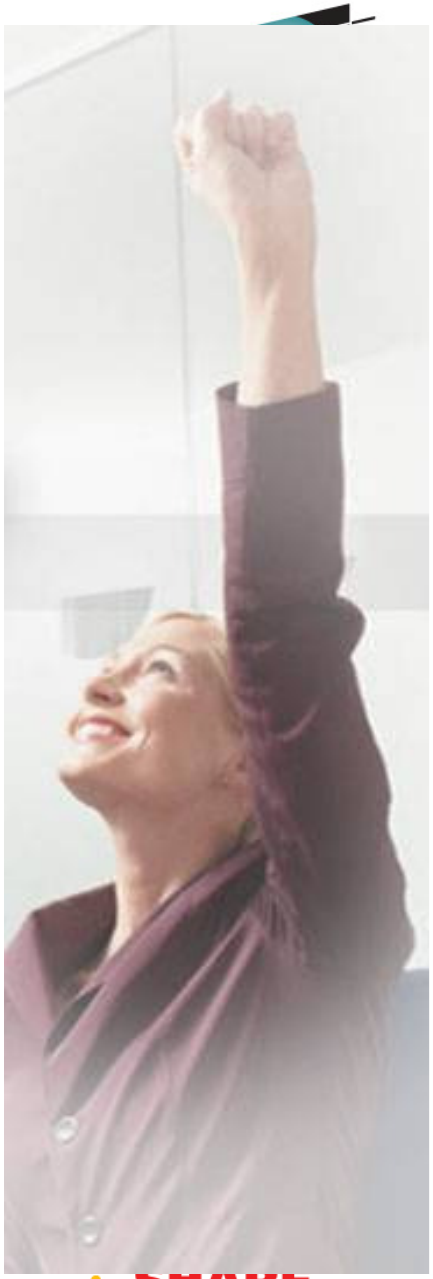
- Reducing the elapsed time of a set of jobs by running more of them alongside each other.
  - Performing I/O in parallel, using techniques such as DFSMS striping
  - Cloning batch jobs to work against subsets of the data
  - Usage of Batchpipes
  - Use of VSAM Record Level Sharing
  - Use of DFSMStvs – Transactional VSAM Services

Reducing the impact of failures - Reducing the impact of failures means ensuring that any prolongation of the run time of the batch workload by failures is minimized.

Reducing the impact of failures: TPC for Replication Family

- TPC for Replication is software running on a server that manages IBM disk based replication functions, i.e.:
  - FlashCopy
  - Metro Mirror
  - Global Mirror
  - Metro Global Mirror

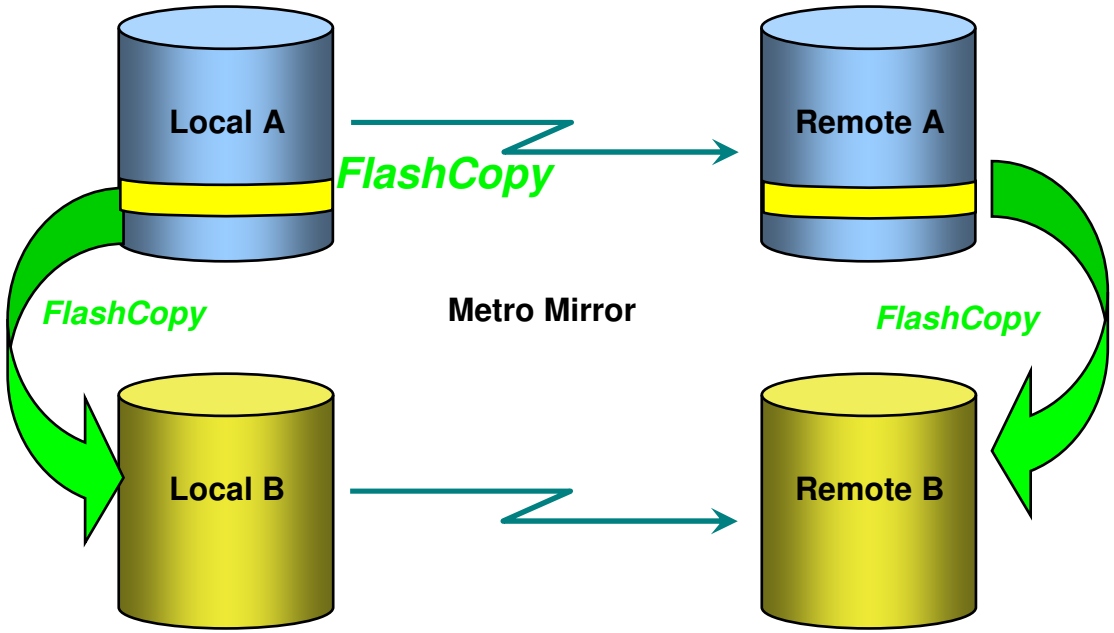
- TPC for Replication
  - Provides central control of your replication environment
  - Helps simplify and automate complex replication tasks without scripts
  - Allows testing of D/R at any time using Practice Volumes
  - Provides end to end management and tracking of the copy services
  - Provides management of planned and unplanned disaster recovery procedures
  - Manage your replication from the server of your choice, z/OS or open systems servers, z/OS and open data



Reducing the impact of failures : Remote Pair FlashCopy

Synchronous Remote Mirroring Optimization

- For z/OS data sets
- For volumes of all servers

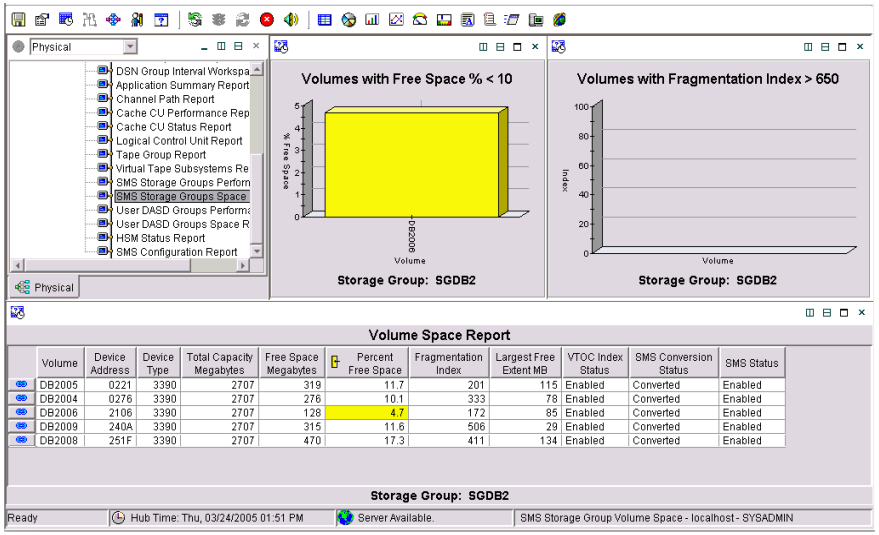


Value

- Availability – Remote site quickly reflects same data as the local site
- Management - Promotes local-remote data consistency for ease-of-recovery
- Performance/TCO - Reduces link bandwidth use, helps improve the performance of other work

Reducing the impact of failures : IBM Tivoli OMEGAMON XE

- **A mainframe STORAGE monitor, real-time and historical**
- **XE user interface, comes with the CUA UI component**
- **A wide breadth of mainframe storage information:**
  - Space (storage groups or user groups … define your own)
  - Performance (storage groups or user groups … define your own)
  - Tape / VTS
  - DFSMSrmm support
  - CACHE
  - Channels (FICON)
  - Control Units
  - DFSMShsm (View your HSM queues, control Datasets, etc.)
  - DFSMShsm/DFSMSdss/DFSMSrmm/ICKDSF online toolkit
  - SMS constructs
  - DS8000 support
  - Ability to see all logical volumes on a physical disk
  - Powerful applications view
  - Powerful dataset view and action capability
  - Integration capabilities from TEP interface
  - Submit JCL in response to monitored situations

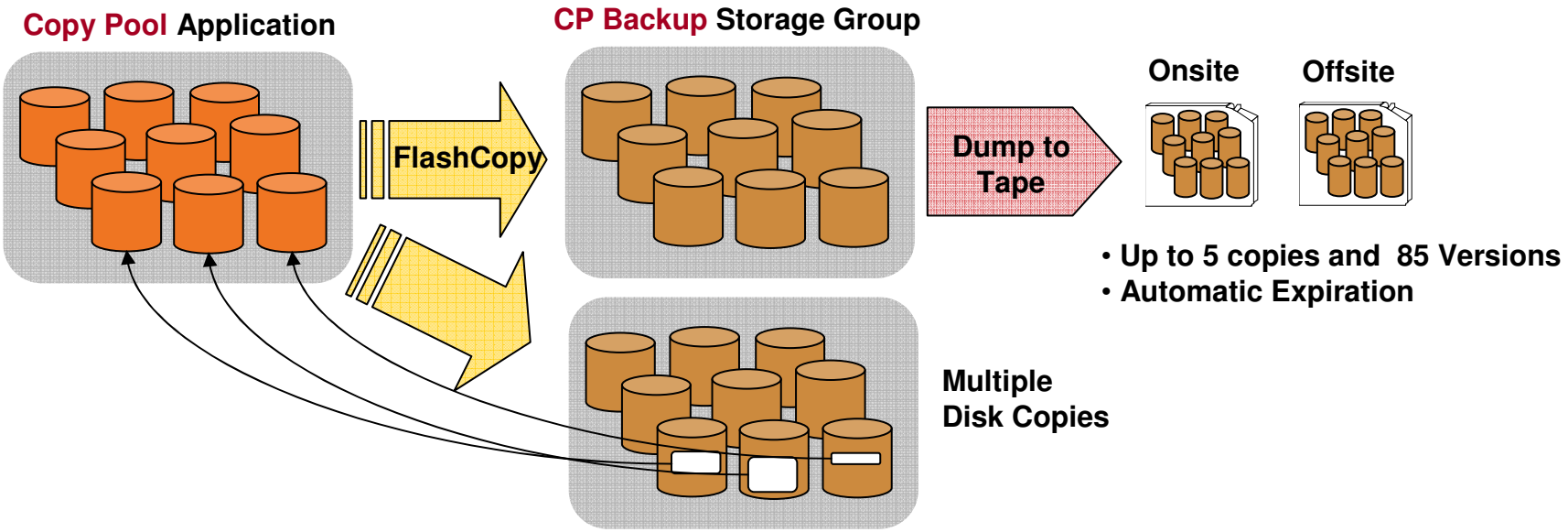


Tivoli Enterprise Portal (TEP) Enabled

Reducing the impact of failures : Fast Replication Overview

HSM function that manages Point-in-Time copies

- Combined with DB2 BACKUP SYSTEM, provides non-disruptive backup and recovery to any point in time for DB2 databases and subsystems (SAP)



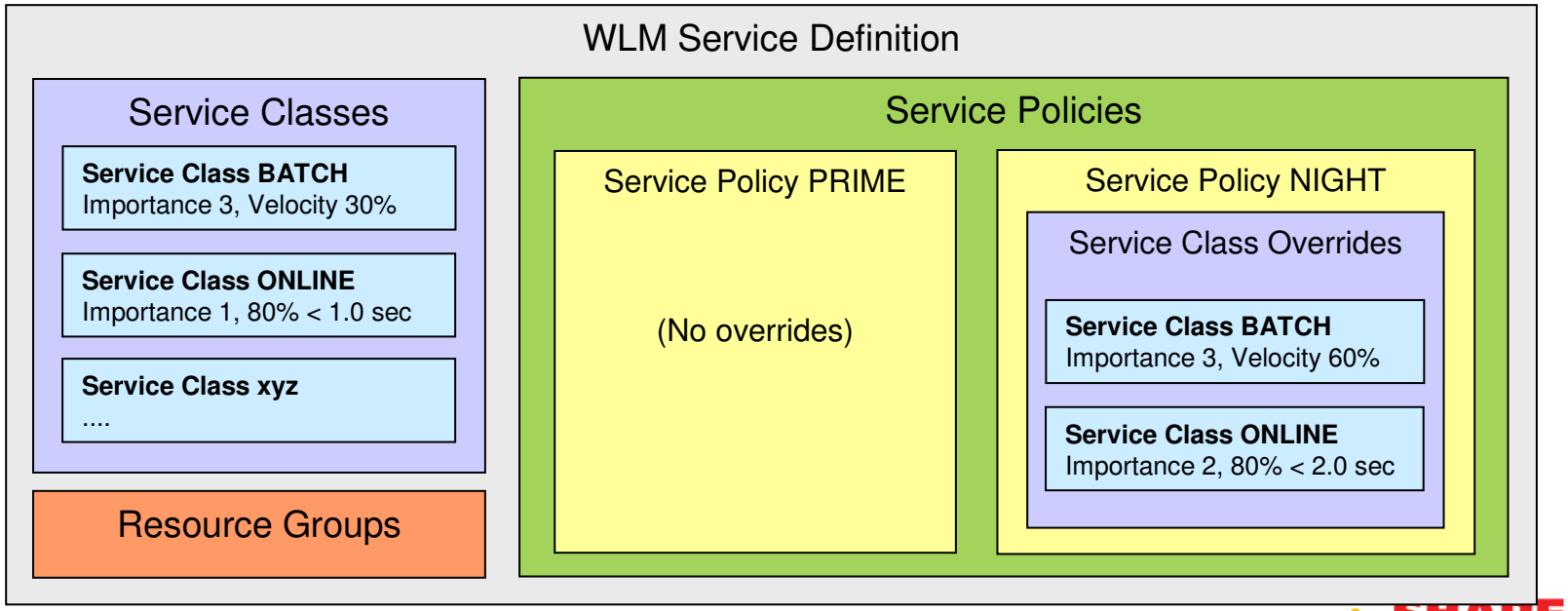
★Recovery at all levels from either disk or tape!

- Entire copy pool, individual volumes and …
- Individual data sets

Tuning Knobs in the Workload / Performance Management area

Ensuring proper system configuration – Service Policy Overrides

- By specifying several Service Policies in a WLM Service Definition you can define different goals for prime shifts and night shifts
- In a Service Policy you can override specific goals or resource groups in the base service definition
- A Service Policy can be activated manually or via automation

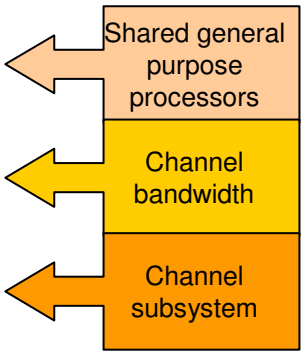


Ensuring proper system configuration – Intelligent Resource Director (IRD)

What is IRD?

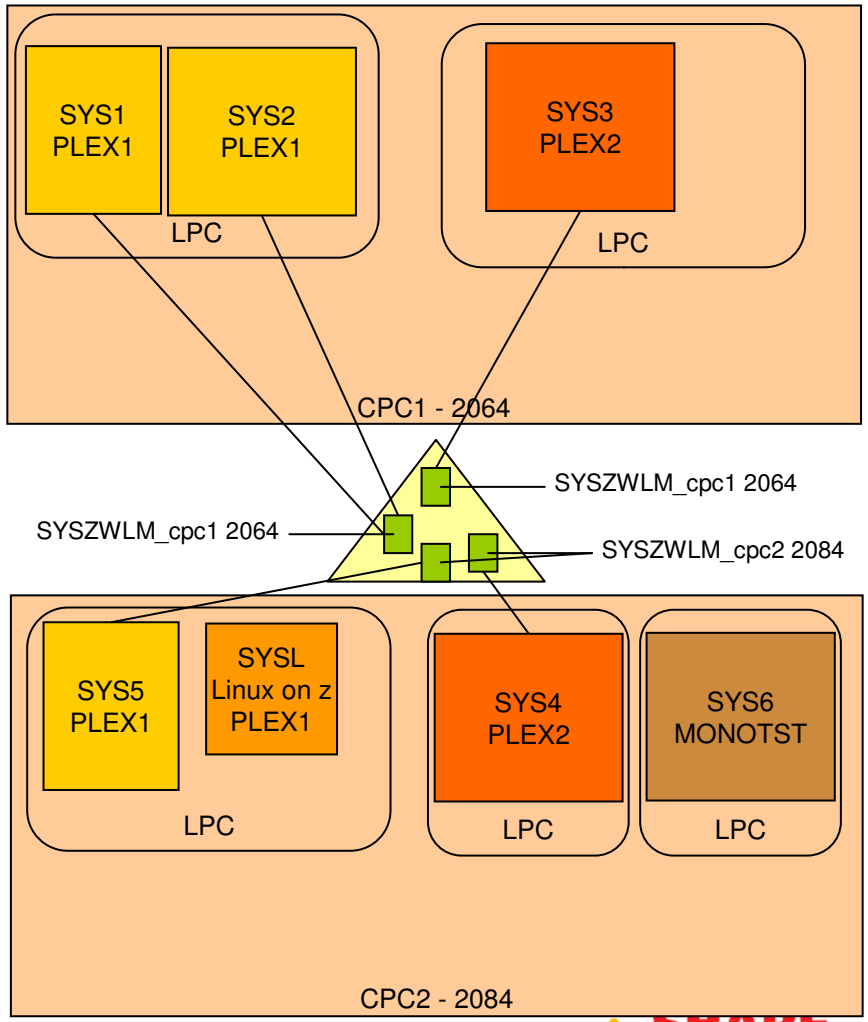
- Set of functions that distribute Central Processor Complex (CPC) resources based on business importance
- Problem areas being addressed:
  - Workloads may change over the course of a day, week, month, year…
  - Business priorities may change over the course of a day, week, month, year…
  - E.g., online vs. batch, production vs. test, workload fluctuations, or periodic work
  - Single static configuration may be sub-optimal to handle different workload mixes
    - Distribute resources based on workload and service level agreements (WLM goals)
  - Reliability problems, e.g. caused by single points of failure

- Consists of
  1. LPAR CPU Management
     - *LPAR Weight Management*
     - ~~*LPAR Vary CPU Management*~~ →*HiperDispatch*
  2. Dynamic Channel Path Management (DCM)
  3. Channel Subsystem Priority Queuing (CSSPQ)
     - *BTW… Not part of IRD: I/O Priority (at control unit level)*



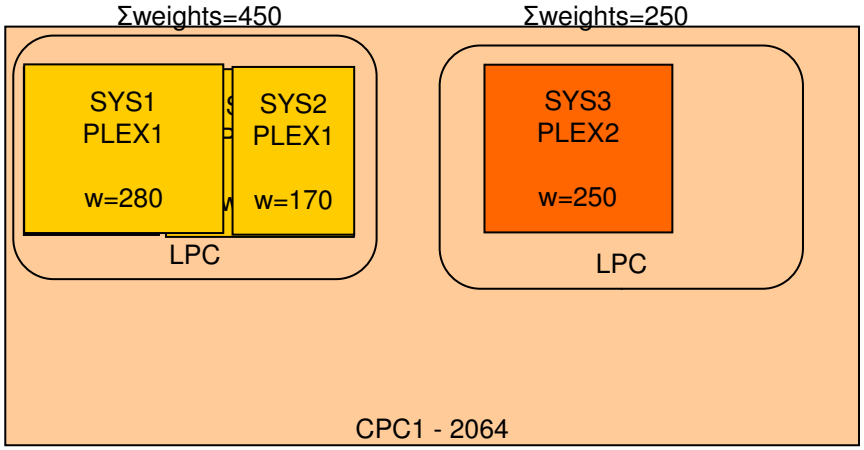
LPAR Clusters

- **Scope of IRD management is the LPAR Cluster (LPC)**
  - **= Set of LPARs on same CPC, which are part of same Sysplex**
- LPARs with dedicated General Purpose Processors (CPs) *can* join a cluster
  - But will not be enabled for WLM LPAR Weight and Vary CPU Management.
- Multi-image/Sysplex LPCs require a CF structure (except for CSSPQ)



LPAR Weight Management

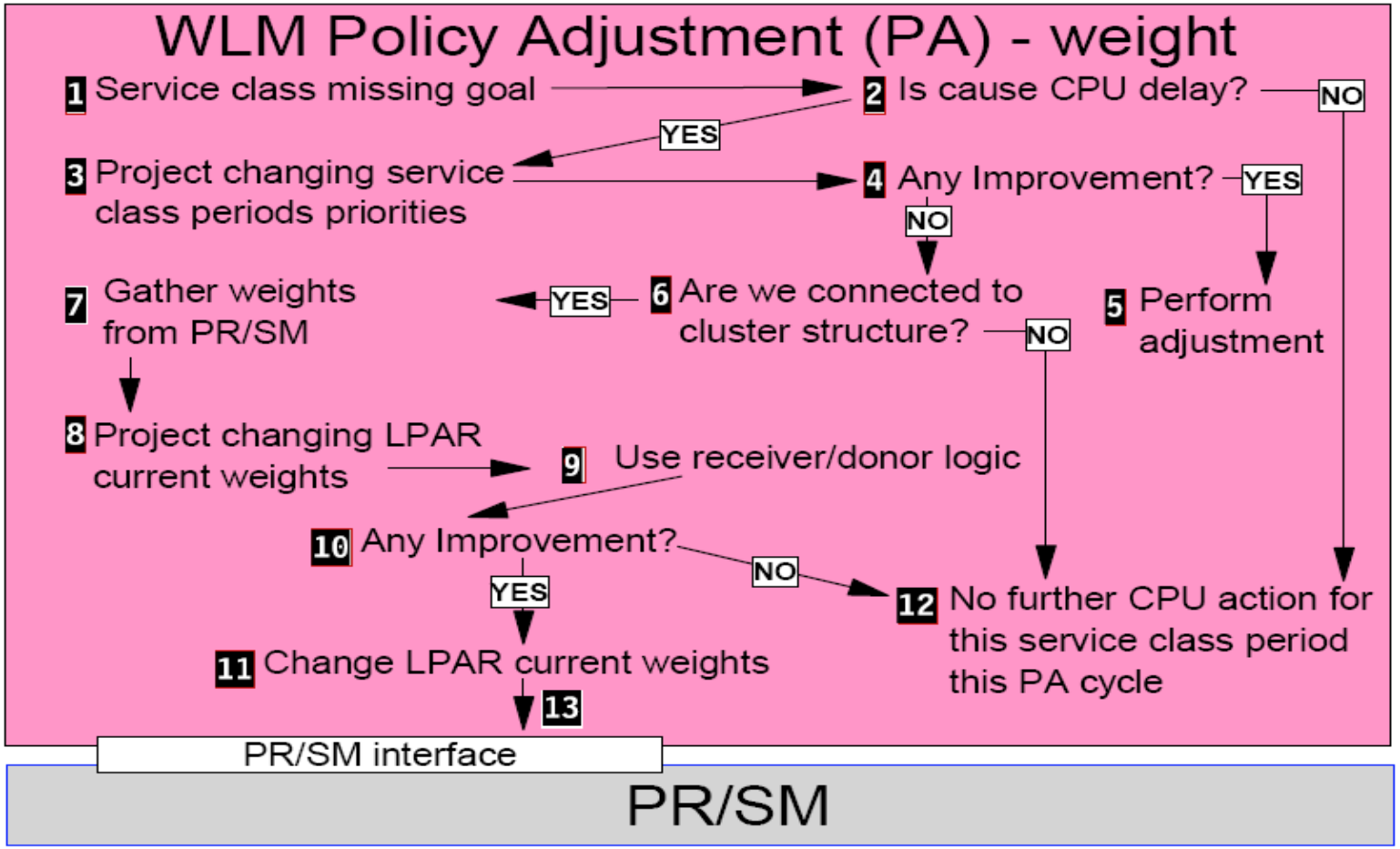
- Weights are shifted within an LPAR cluster (LPC)
  - The weight of an LPC does *not* change.
  - Single image LPCs cannot perform LPAR weight management



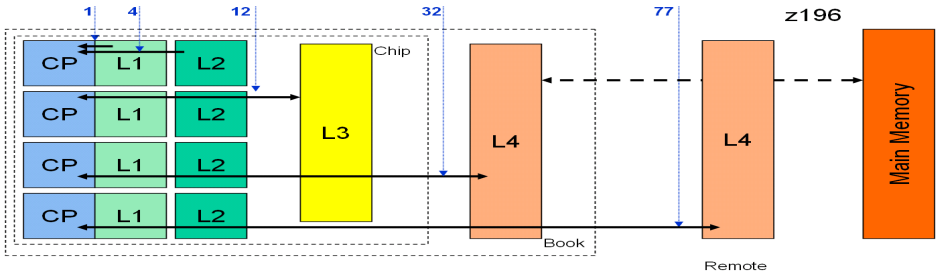
- The weight of an LPAR may take any value in the range defined by minimum and maximum processing weight values.
  - LPARs may consume more capacity when other LPARs have no demand and no capping is in effect
  - Weight will not be increased while LPAR is being capped

- Note: Tivoli System Automation ProcOps or a homegrown solution can be used to change weights beyond the LPAR cluster scope

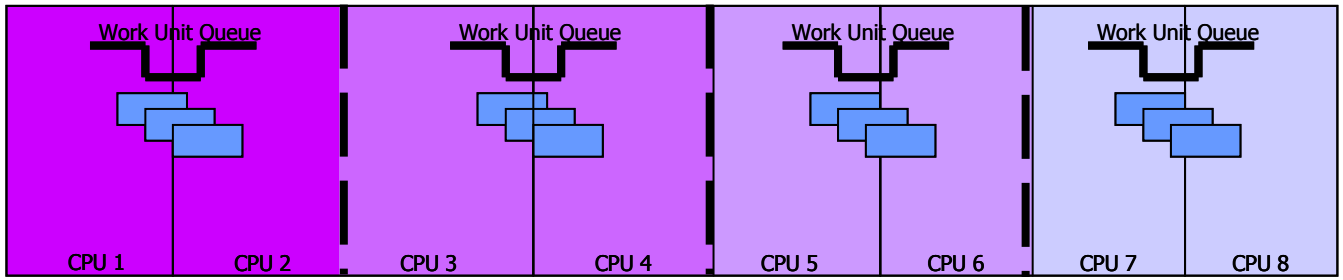
Weight Management Algorithm



HiperDispatch



- Optimizes throughput
  - Manages to cache topology: Attempts to redispatch work in the same processor context
  - Can „park“ logical processors when they are not needed to consume LPAR weight
- Allows to configure the logical configuration for flexibility
  - Reduces need to configue for optimum logical:physical CP ratio



CPU Weight Management: Controls

- LPAR Controls at HMC or Support Element
  - "WLM Managed" check box
  - Initial, Minimum and Maximum processing weights
    - Configure for flexibility but use safety net
  - Number of non-dedicated Central Processors

Change LPAR Controls

CPC Name: IP3TVM90
Last reset profile attempted:
Input/Output configuration data set (IOCDS): A0 328AF09

Logical Partition	Active	Defined Capacity	Current Weight	WLM Managed	Initial Processing Weight	Minimum Processing Weight	Maximum Processing Weight	Initial Capping	Current Capping	Number of Dedicated Central Processors	Number of Non-dedicated Central Processors	Logical Partition
SCLM1	Yes	0	400	☐	400	0	0	☐	No	0	7	SCLM1
SCLM2	Yes	0	202	☐	202	0	0	☐	No	0	3	SCLM2
SCLM3	Yes	0	202	☐	202	0	0	☐	No	0	3	SCLM3
VM9	Yes	0	0	-	-	-	-	-	No	8	0	VM9
COM1	Yes	0	100	☐	100	0	0	☐	No	0	3	COM1
COM2	Yes	0	80	☐	80	0	0	☐	No	0	5	COM2
COM4	Yes	0	80	☐	80	0	0	☐	No	0	5	COM4
IRL1	No	0	0	☐	1	0	0	☐	No	0	1	IRL1
IRD1	Yes	0	251	☑	200	50	500	☐	No	0	7	IRD1
IRD2	Yes	0	151	☑	200	50	500	☐	No	0	4	IRD2
IRD3	Yes	0	100	☑	102	15	500	☐	No	0	3	IRD3
CFIRD	Yes	0	49	☐	49	0	0	☑	No	0	1	CFIRD

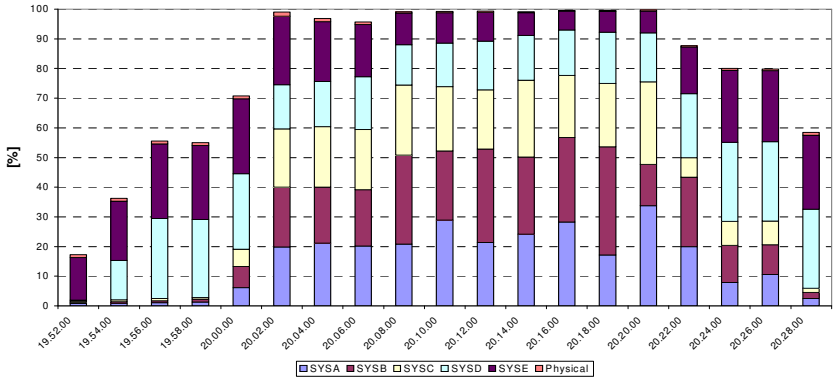
Processor running time

Warning: It is recommended that you select 'Dynamically determined by the system.' Selecting 'Determined by the user' risks suboptimal use of processor resources.

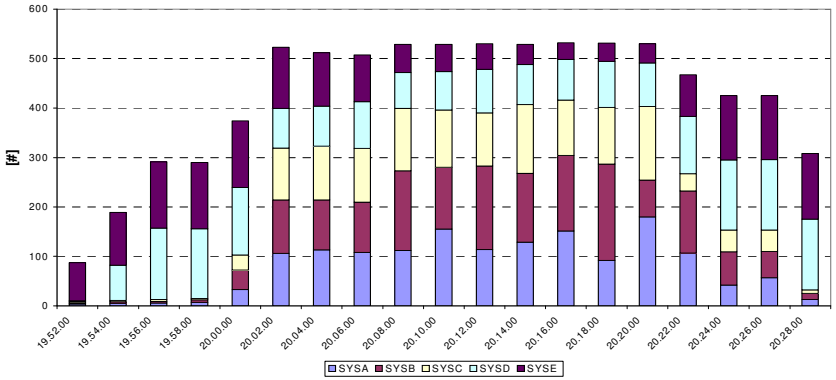
◉ Dynamically determined by the system

IRD CPU Management in Action

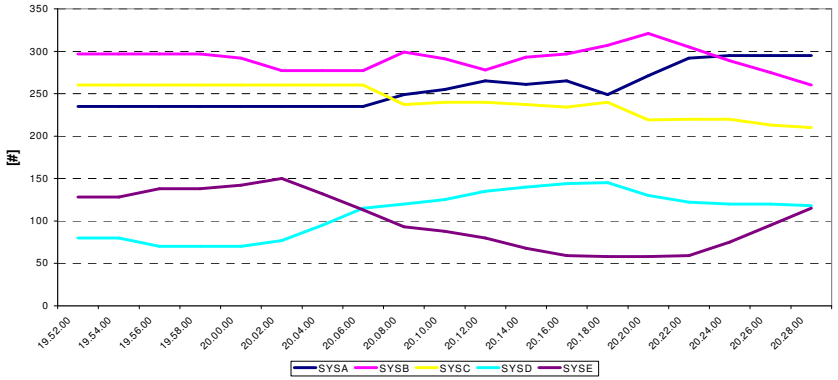
LPAR Utilization
(physical)



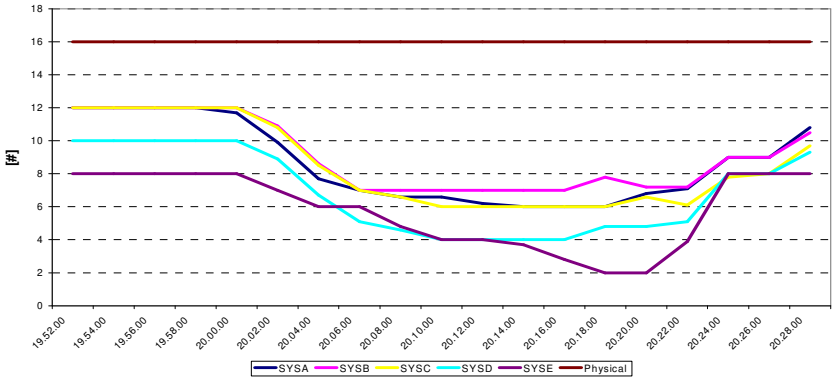
Service Consumed



LPAR Weights

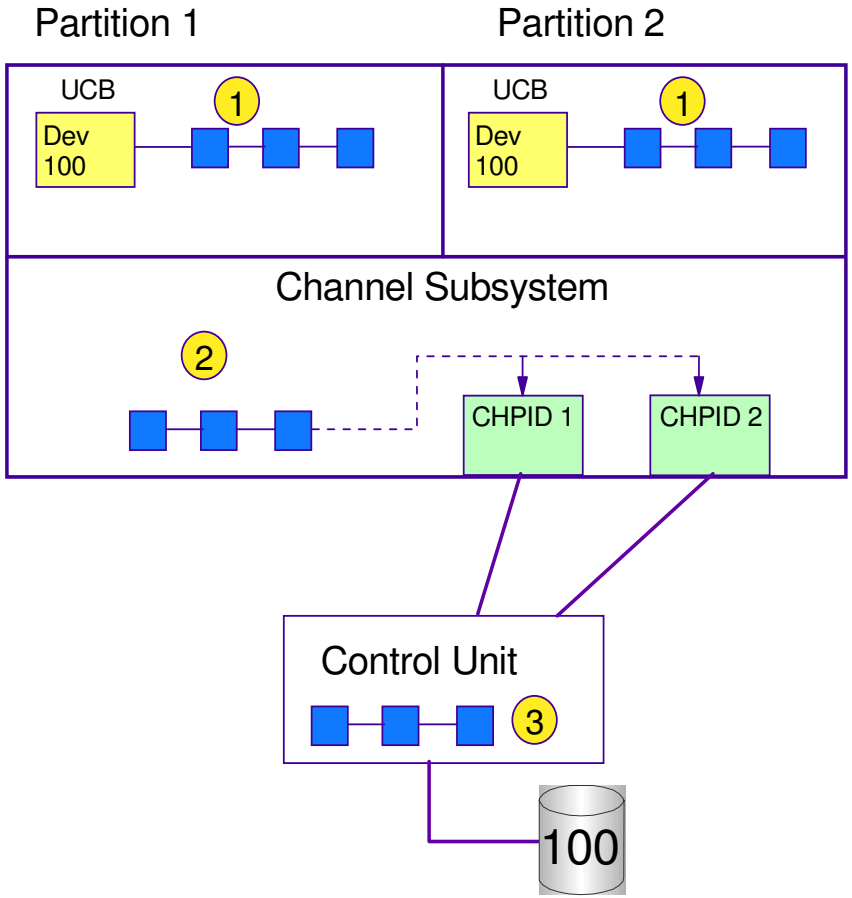


#Logical Processors



Channel Subsystem I/O Priority Queuing

- Allows WLM to assign a priority to an I/O request
- Channel subsystem can now use a priority managed queue
  - Prior to this time the channel subsystem used a FIFO queue
  - Effective when there is contention in the CSS
- Complements Priority Queuing in other parts of the I/O Subsystem
  - IOS UCB Queue
  - In control unit



CSSPQ Controls

Change LPAR I/O Priority Queuing

Input/output configuration data set (IOCDS): A0
Global input/output priority queuing: Disabled
Maximum global input/output priority queuing value: 15

Logical Partition	Active	Minimum input/output priority	Maximum input/output priority
SCLM1	Yes	0	0
SCLM2	Yes	0	0
SCLM3	Yes	0	0
VM9	Yes	0	0
COM1	Yes	0	0
COM2	Yes	0	0
COM4	Yes	0	0
IRL1	No	0	0
IRD1	Yes	0	15
IRD2	Yes	0	15
IRD3	Yes	0	8
CFIRD	Yes	0	0

Note: WLM I/O Priority Management must be enabled in the WLM Service Definition for CSSPQ

1. Image profiles:
   - Range of I/O priorities that can be used by the partition
     - WLM maps priority to available range
     - 0-15 can be specified; range of 8 is recommended
     - Specify same range for all members of an LPC
2. Reset profile:
   - Global switch to activate CSSPQ

Optimizing parallelism – WLM-managed Batch Initiators

- Dynamic, goal-oriented management of the time that jobs spend waiting for an initiator
  - Multi-system workload balancing
    - This does NOT imply balancing for equal CPU utilization!
  - Reduced operational demand
  - Improved reporting of job response time and pre-execution job delays

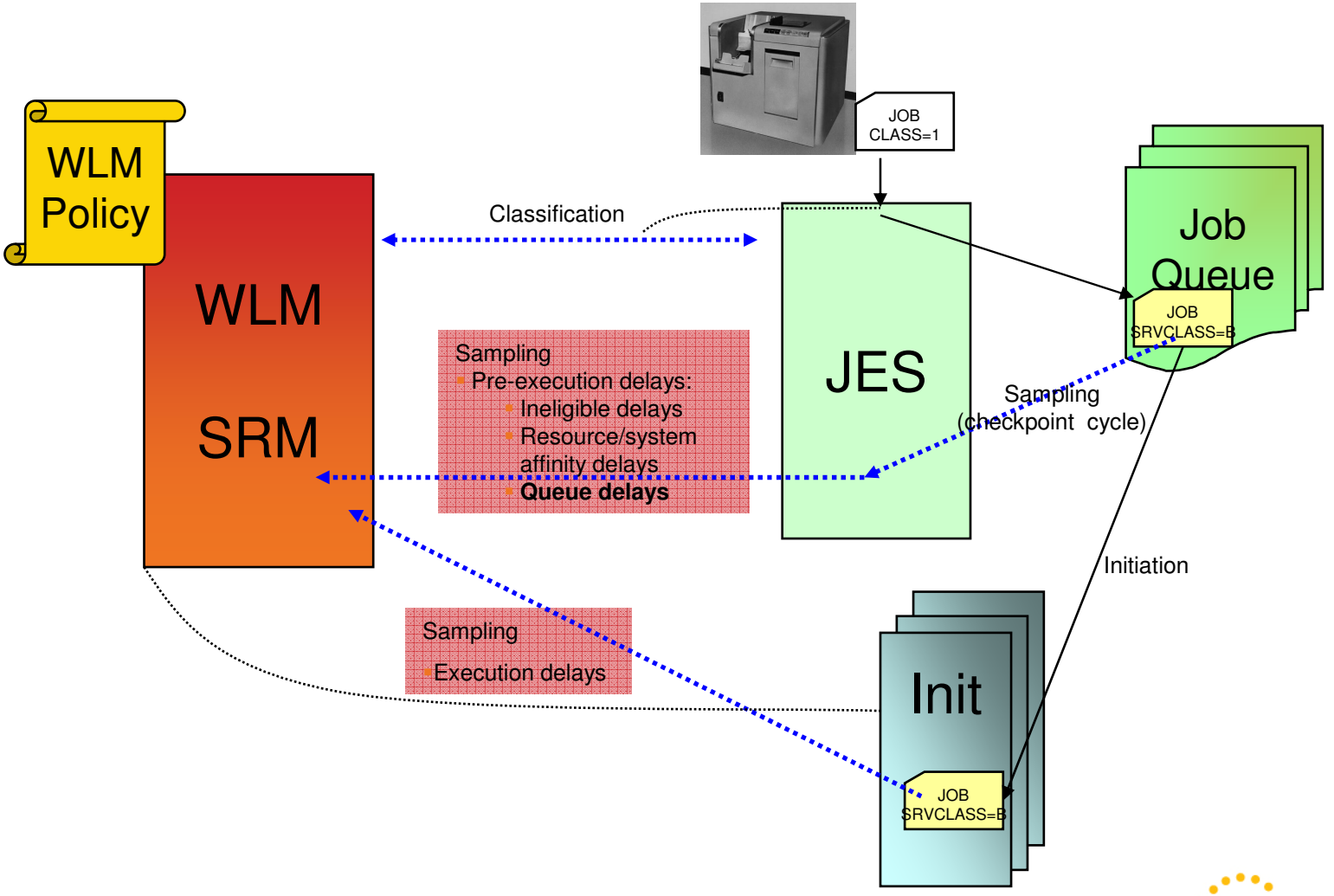
- WLM does not
  - Deadline scheduling
  - Job history management
  - Consumable or multi-state resource

WLM-managed Batch Initiators
What Does WLM Manage?

- Number of initiators
  - Queue delay samples identify delays due to lack of initiators
  - Adjustments are made based on goals and importance
- Placement of initiators
  - Select system with “largest available” capacity
  - Awareness of system affinities
  - Work is displaced if necessary ...
    ... Importance is used to minimize impact
- Exploitation / migration on job class bases

➔The previously independent processes of initiation and performance management are now linked together

The Big Picture – WLM-managed Batch Jobs



WLM-managed Job Classes

- **WLM management is done on a job class basis**
- **Each job class has a mode assigned via the JES2 init deck or the $T JOBCLASS command**
  - MODE=JES is the default
  - MODE=WLM designates WLM-managed job classes that run jobs in WLM-managed initiators
- **Initiator definitions do not have to be updated to remove references to WLM-managed job classes**

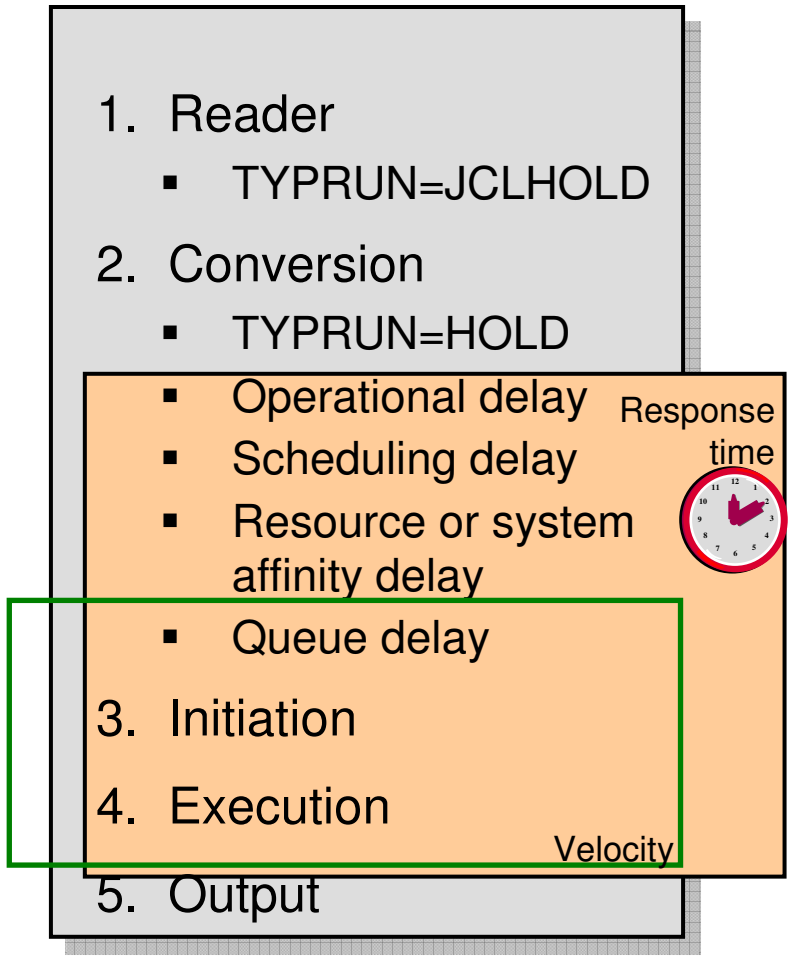
```
JOBCLASS(A) MODE=JES
JOBCLASS(B) MODE=WLM
JOBCLASS(C) MODE=WLM
:
INIT1 ABC
INIT2 BC
```

JES Initiators vs. WLM Initiators

- JES initiators
  - Started via INIT procedure
  - Select jobs from JES-managed job classes only
  - Started and stopped by operators
  - Consume job numbers
- WLM initiators
  - Started via INIT procedure
  - Select jobs from WLM-managed job classes only
  - Started and stopped by WLM based on
    - Available capacity
    - Goals
  - Started under MSTR subsystem, so they do not consume job numbers

Pre-Execution Job Delays

- User-specified delay
  - TYPRUN=HOLD and TYPRUN=JCLHOLD
- Operational delay
  - Job held
  - Job class held
- Scheduling delay
  - Job class execution limit
  - Duplicate jobnames
- Resource or system affinity delay
  - Scheduling environment is not available
  - System is not available
- Queue delay
  - Waiting for an initiator to select a job



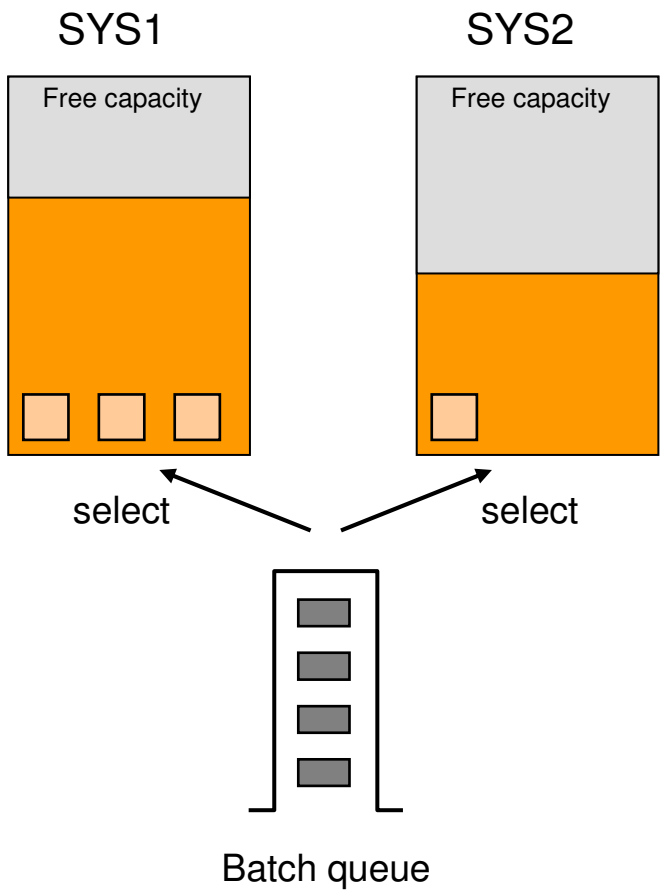
RMF Support

- RMF shows
  - Job delay data (Included in response times)
  - Execution time
  - Overall response time
- In several reports
  - Monitor III: SYSSUM, SYSRTD, and GROUP reports
  - Postprocessor: WLMGL report

```
REPORT BY: POLICY=POLPPPO2   WORKLOAD=WLMNORM     SERVICE CLASS=BAT_P_HI   RESOURCE GROUP=*NONE      PERIOD=1 IMPORTANCE=2
                                                  CRITICAL     =NONE

-TRANSACTIONS-  TRANS-TIME HHH.MM.SS.TTT  --DASD I/O--  ---SERVICE---   SERVICE TIME  ---APPL %---  --PROMOTED--  ----STORAGE----
AVG       1.30  ACTUAL                7.623 SSCHRT 342.1  IOC      41520  CPU    73.582  CP      24.72  BLK     0.000  AVG      7083.22
MPL       1.30  EXECUTION             6.979 RESP     2.3  CPU       2559K SRB     0.428  AAPCP    0.00  ENQ     0.001  TOTAL    9236.28
ENDED       18  QUEUED                 643 CONN     0.3  MSO          0  RCT     0.018  IIPCP    0.00  CRM     0.000  SHARED      2.81
END/S     0.06  R/S AFFIN                0 DISC     1.8  SRB      14894  IIT     0.129                LCK     0.000
#SWAPS       3  INELIGIBLE               0 Q+PEND   0.2  TOT       2616K HST     0.000  AAP       N/A                 -PAGE-IN RATES-
EXCTD        0  CONVERSION           1.048 IOSQ     0.0  /SEC      8719  AAP       N/A  IIP      0.00                SINGLE       0.0
AVG ENC   0.00  STD DEV             11.584                              IIP     0.000                                 BLOCK        0.0
REM ENC   0.00                                          ABSRPTN  6687                                              SHARED       0.0
MS ENC    0.00                                          TRX SERV 6687                                              HSP          0.0

GOAL: EXECUTION VELOCITY 60.0%     VELOCITY MIGRATION:   I/O MGMT  68.8%     INIT MGMT 66.4%

         RESPONSE TIME EX    PERF  AVG    --EXEC USING%--  -------------- EXEC DELAYS % -----------  -USING%-  --- DELAY % ---     %
SYSTEM                 VEL% INDX ADRSP  CPU AAP IIP I/O  TOT CPU I/O                                  CRY CNT  UNK IDL CRY CNT  QUI

SYSA        --N/A--    68.8  0.9   2.2   11 N/A 0.0  14   11 8.6 2.7                                  0.0 0.0   28  36 0.0 0.0  0.0
```

WLM-managed Batch Initiators Algorithms: The Short Version

- Policy adjustment:
  - Initiators are dynamically started by WLM to meet service class period goals by reduction of batch queue delays
  - WLM selects the system based on
    - Available system resources based on the job's importance
    - Availability of waiting batch jobs
- Resource adjustment:
  - Start additional initiators if jobs are waiting and unused capacity is available
    - Based on service class history
    - Up to 5 at a time



- Initiators are dynamically stopped by WLM
  - When significantly more initiators exist than needed
    (> 1.5 times avg. (active+queued) jobs)
  - In case of CPU or memory shortage
  - After one hour of inactivity
  - As part of Initiator re-balancing

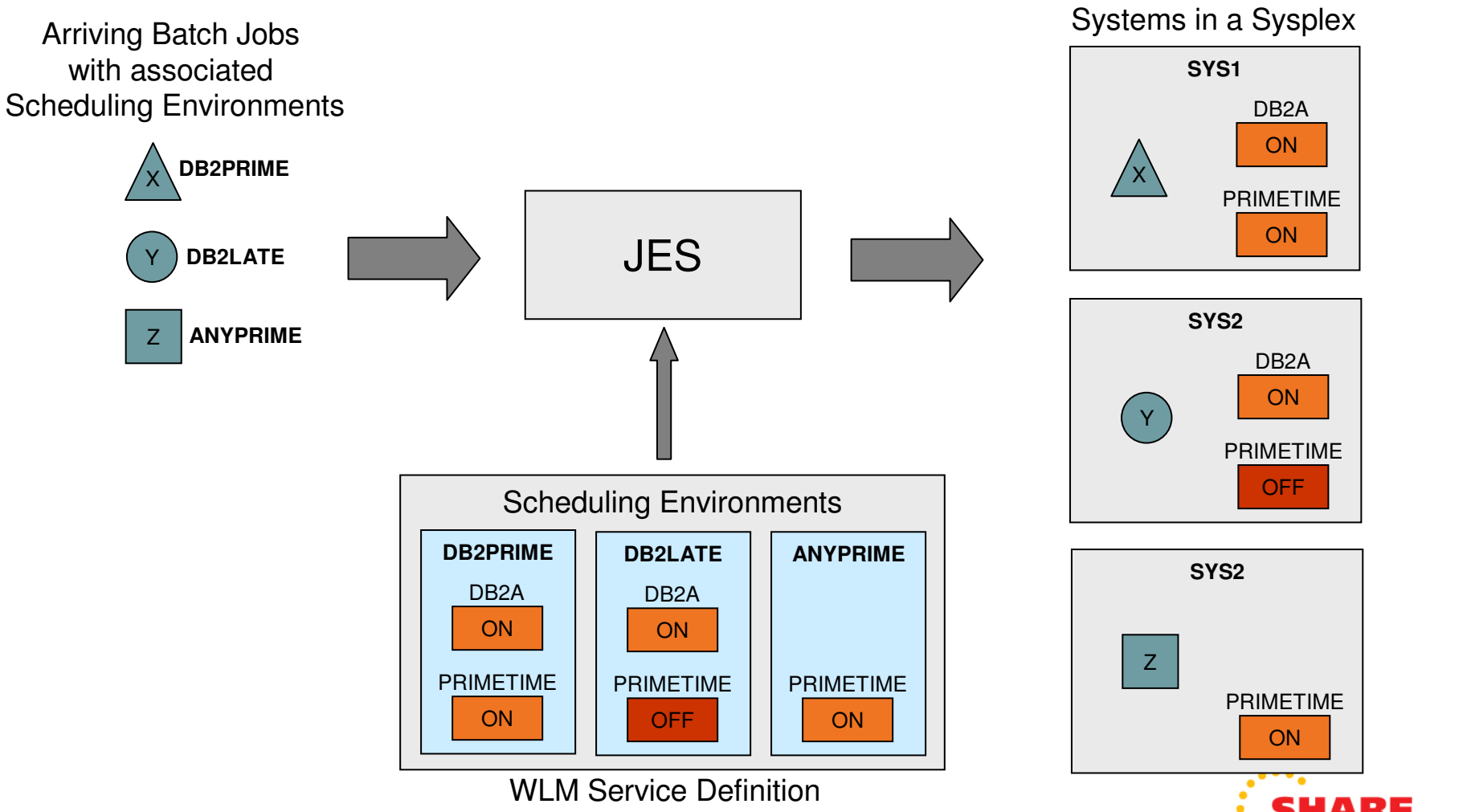
WLM-managed Batch Initiators Good Guys, Bad Guys…

- Some job classes might not be good candidates for WLM goal managed initiators
  - No relationship between response time and number of initiators
    - Too low arrival/ending rate or consumption
    - No good correlation between goal attainment of job and amount of available resources
    - Other delays
  - Need for immediate initiation
    - Jobs released “just in time” by scheduling systems
    - Jobs that are critical for interactive work

Increasing operational efficiency – Scheduling Environments

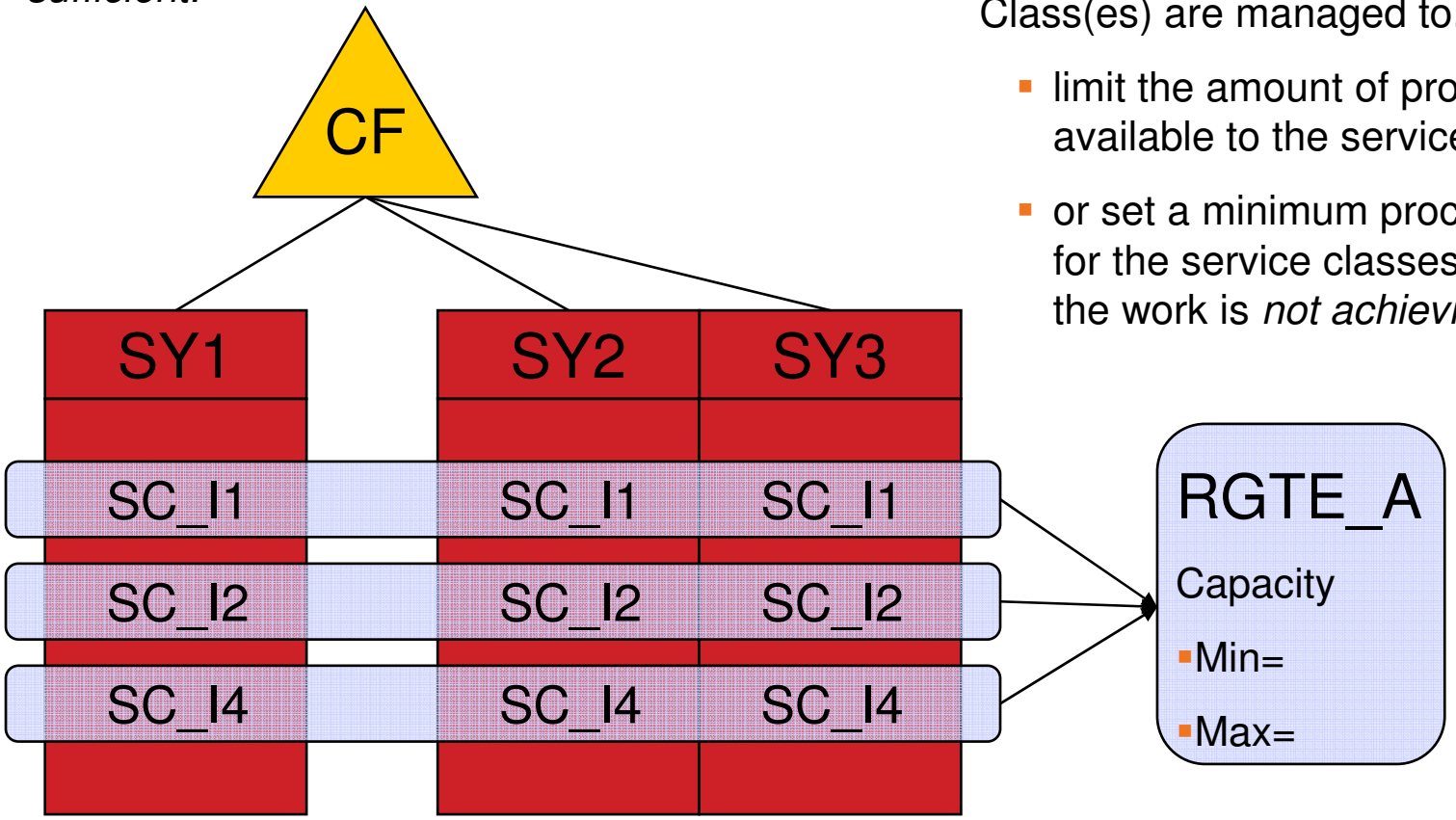
- Scheduling Environments help ensure that batch jobs are sent to systems that have the appropriate resources to handle them
- A Scheduling Environment is a list of Resource names along with their required states
- Resources can represent
  - actual physical entities, e.g. a data base or peripheral device
  - intangible qualities such as a certain period of time, e.g. second shift
- Scheduling Environments and Resources with their required state are specified in the WLM service definition
- Operator or automation sets actual Resource state on each system
- JES checks the Scheduling Environment associated with each arriving batch job and then assigns the work to a system where the actual Resource states match the required Resource states

Increasing operational efficiency – Scheduling Environments Example



Protecting workloads – Resource Groups

Resource groups are a means to protect work w*hen proper classification, goals and importance won't be sufficient.*



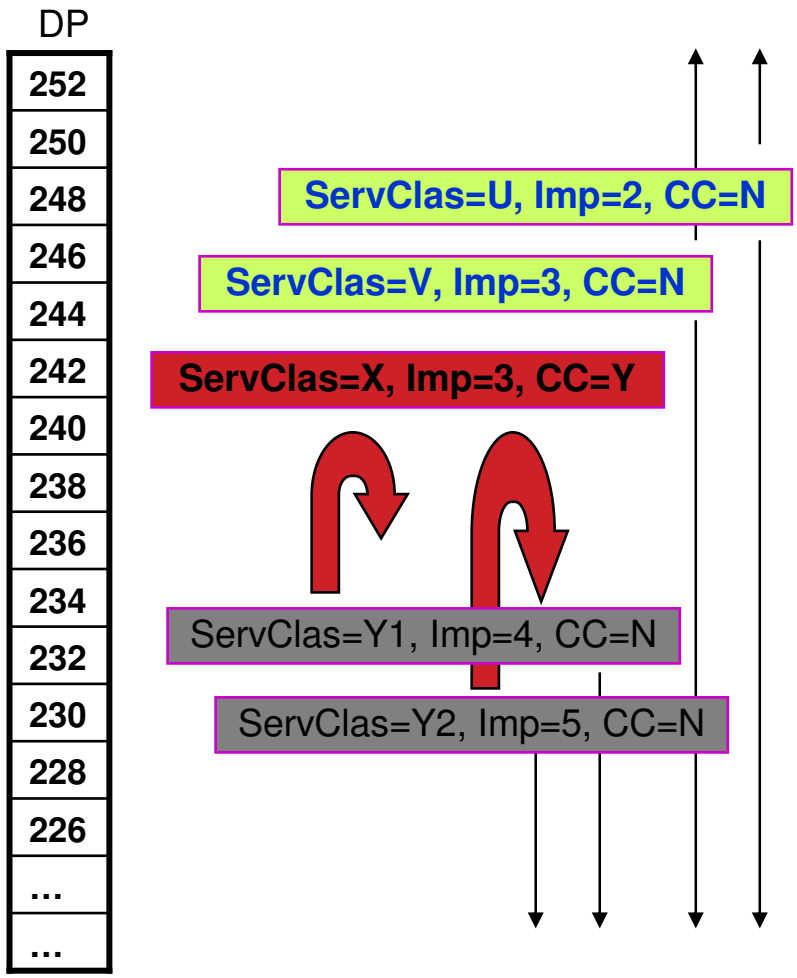
- A Resource Group is associated to one or more Service Classes
- Defines the service that the related Service Class(es) are managed to. Either
  - limit the amount of processing capacity available to the service classes,
  - or set a minimum processing capacity for the service classes in the event that the work is *not achieving its goals*

Protecting workloads – Comparison of Resource Group Types

Definition	Scope	Advantages	Possible Disadvantages
Type 1 RG: Service Units	Sysplex	• Allows balancing of resources across members in a sysplex	• Requires adjustments for migration • Difficult to monitor • Not applicable to constrain work on a single system in a sysplex environment
Type 2 RG: % of LPAR	System	• Allows to control work on single members in a sysplex ☑ Stable for migrations	• Need to understand what LPAR capacity really is
Type 3 RG: # of LCPs	System	• Allows to control work on single members in a sysplex ☑ Easy and straight forward definition	• Requires adjustments for migration

Protecting workloads – CPU Critical – Long Term CPU Protection

- Specified at the service class level
- Ensures that lower important work never gets a higher dispatch priority than the higher important work defined with the CPU critical attribute
  - You still need to define appropriate goals to CPU Critical work
  - It is not a full protection against everything
    - The work is still managed towards a goal



Documentation

z/OS MVS Planning: Workload Management (SA22-7602)
http://publibz.boulder.ibm.com/epubs/pdf/iea2w1b0.pdf

z/OS MVS Programming: Workload Manager Services (SA22-7619)
http://publibz.boulder.ibm.com/epubs/pdf/iea2w2b0.pdf

Redbook – System Programmer's Guide to: Workload Manager (SG24-6472)
http://www.redbooks.ibm.com/redbooks/pdfs/sg246472.pdf

Internet Links

WLM	http://www.ibm.com/systems/z/os/zos/features/wlm//
IRD	http://www.ibm.com/systems/z/os/zos/features/wlm/documents/ird/ird.html
Redbooks	http://www.redbooks.ibm.com/

Additional Documentation

z/OS R12 DFSMS Manuals
http://www-03.ibm.com/systems/z/os/zos/bkserv/r12pdf/#dfsms

Redbook – “Batch Modernization on z/OS” (sg24-7779)
Redbook – “Parallel Sysplex Batch Performance” (SG24-5952)

Internet Links

DFSMS	http://www-03.ibm.com/systems/storage/software/sms/index.html
OM XE for Stg	http://www-01.ibm.com/software/tivoli/products/omegamon-xe-storage
Redbooks	http://www.redbooks.ibm.com/