What is a Resource Group (RG)?

A *resource group (RG)* is an amount of processor capacity that governs specific workloads and is optional when you define service level objectives. Although, other than you having a special need to limit or protect processor capacity for a group of work, you should avoid resource groups letting workload management to naturally administer all of the processor capacity to meet performance goals.

Keep in mind your WLM service objectives if you decide to assign a service class to a resource group. Given the combination of the goals, the importance level, and the resource capacity, some goals may not be achievable when capacity is restricted. RGs are a last resort to protect work when alternate measures do not suffice and can cause latent demand[1] representing poor service making capacity planning cumbersome.

If you do need to use a Resource Group, it can:

- Limit the amount of processing capacity available to one or more WLM service classes
- Set a minimum processing capacity for one or more service classes in the event that the work is not achieving its goals
- Define a minimum and maximum amount of capacity sysplex-wide, or on a system-level.

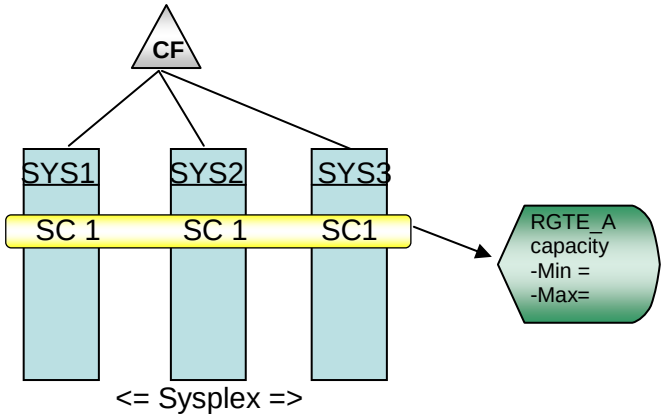
It is recommended to evaluate whether using a resource group fulfills your requirements best, or whether you let workload management take care of managing the resources. For a resource group, you specify either a minimum or a maximum amount of sysplex capacity in *unweighted* CPU service units per second, or both.

If work in a RG is consuming CPU service units[2] (SUs) above the specified maximum capacity, the system throttles back (*caps*) the associated work to slow down the rate of resource consumption. WLM calls the zOS Supervisor to mark the work unit nondispatchable. Using resource groups places additional burden on the policy adjustment (PA) algorithms. Since PA needs to monitor each service class period on how well it is meeting its goal, in addition to RGs.

The system may use several mechanisms to slow down the rate including swapping the address space, changing its dispatching priority, and capping the amount of service that can be consumed. Reporting information for a service class reflects that the workload may not be achieving its goals because of this capping and will appear that it is running out of CPU.

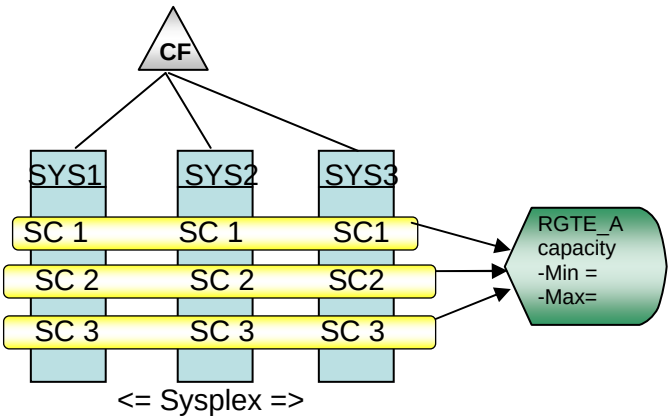
By setting a processing capacity, you create an overriding mechanism to circumvent the normal rules of importance. If the work in a resource group is not meeting its goals, then workload management will attempt to provide the defined minimum or maximum amount of CPU resource to that resource group.

NOTE: Generally, resource group capping, as a goal enforcement, is performed at the sysplex level. The CPU service rate values are accumulated first on local systems, and then across the sysplex for total. The total value of the consumed service rate is used to determine if it exceeds the resource group maximum service rate.



SC = Service Class

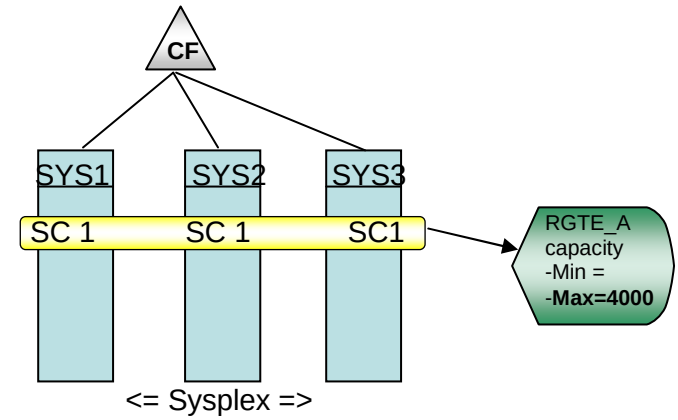
It's also possible that many Service Classes connect to a single RG as illustrated on the next page.



SC = Service Class

If a service class does not execute on SYS1 due to transaction routing or otherwise, then only SYS2 and SYS3 share the 4000 "unweighted" service units (SUs).

Weighted service units: A fundamental aspect of performance characteristics is the rate at which an address space is receiving service relative to other address spaces competing for resources. Service Definition Coefficients are **weights** that are applied to Service Units[2] (SU) that calculate the degree of how much service is consumed. These weights affect CPU, SRB, IOC, and MSO (main storage occupancy) consumption. Example - CPU=10.0,IOC=5.0,MSO=3.0,SRB=10.0.



SC = Service Class

To implement capping, the elapse time is divided into 64 slices and each cap slice then represents 1/64th of the total elapsed time. Dispatchable units (AWAKE SLICE) from address spaces or enclaves belonging to a RG are made nondispatchable (CAP SLIDE) during some slices in order to reduce access to the CPU to enforce the RG maximum.

16 Awake Slices
48 Cap Slices

1	9	17	25	33	41	49	57
2	10	18	26	34	42	50	58
3	11	19	27	35	43	51	59
4	12	20	28	36	44	52	60
5	13	21	29	37	45	53	61
6	14	22	30	38	46	54	62
7	15	23	31	39	47	55	63
8	16	24	32	40	48	56	64

Active slices are distributed equally over the pattern

Every 10 seconds the policy adjustment rules are evaluated for the RGs and will regulate the cap pattern accordingly. The forecast for the next 10 seconds is based on the average data from the last minute passed.

NOTE: Because of the 1 minute average for history gathering, during a ramp up period, the max may be exceeded and skew until a history pattern is formed but eventually it balances out.

CAVEATS: Under certain conditions work may continue consuming service even while capped.

- Any locked work will continue to be dispatched as long as the lock is held
- The address space will not be marked nondispatchable until the next dispatch.

In some cases, however, you might want to use a resource group, for example, to limit the service that a certain service class can consume. It is recommended that each resource group is assigned to only one service class. Using a resource group is appropriate, for example:

- If you have an SLA that charges for a fixed amount of CPU capacity and you want to ensure that no more is used.
- If you want to ensure that some work with discretionary goals receives some minimum amount of capacity.

Resource Group (RG) Management Types

Resource Group Type 1 (original)
The capacity is specified in unweighted CPU service units per second, the value must be between 0 and 999999. Minimum and maximum capacity applies sysplex-wide, that is, WLM ensures that the limits are met within the sysplex.

If an RG is not meeting its minimum capacity and work in that RG is not meeting its goals, WLM will attempt to give CPU resource to that work:

- Even if more important work may miss its goals
  (**EXCEPTION** – Discretionary work, WLM will only assist this type of work in an RG that is not meeting its minimum capacity if it does not cause other work to miss its goals)
- The minimum capacity setting has no effect when work in the RG is meeting its Goal

NOTE: A disadvantage of RG Type1 is WLM will 'attempt' to balance MSUs across the sysplex, but there is no assurance. It is possible that a service class might receive most or all of its capacity limit on one system and very little service on other systems. As a result, using Type 1 is not easy to understand how much is consumed on which system and depends highly on the capacity of each z/OS image especially when workload changes.

Other problems are that many SCs can share one RG and can have different levels of importance. In the case of creating an RG for WAS, use a standalone SC and place all address spaces and enclaves grouped together in that definition. Service units are intended to be a processor independent measure of service, so MSUs (millions of service units) values require a particular workload should remain reasonably consistent (but not exactly the same) across hardware changes. If you change hardware, you will need to recalculate new MSUs values. See WLM Homepage for STSI Model name and SU/SEC:

http://www-03.ibm.com/servers/eserver/zseries/zos/wlm/

The value basically depends on the number of CPUs in the LPAR, not he number in the CEC, but this can vary due to multiprocessing (MP) effect.

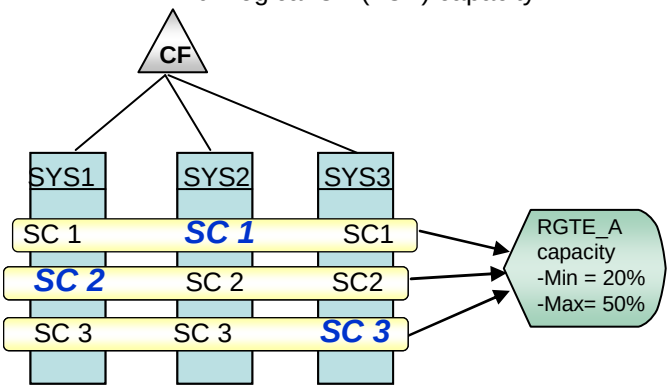
NOTE: The ASCBUWND bit is set in the address spaces and the ENCBUWND bit is set in the enclaves (i.e. WAS Transaction) to indicate that the unit of work is not dispatchable due to RG settings.

New Types of Resource Groups - definition applies to each system.

General Considerations for Type 2 and Type 3

Multiple service classes can be assigned to an RG, but this has no sysplex wide effect anymore.

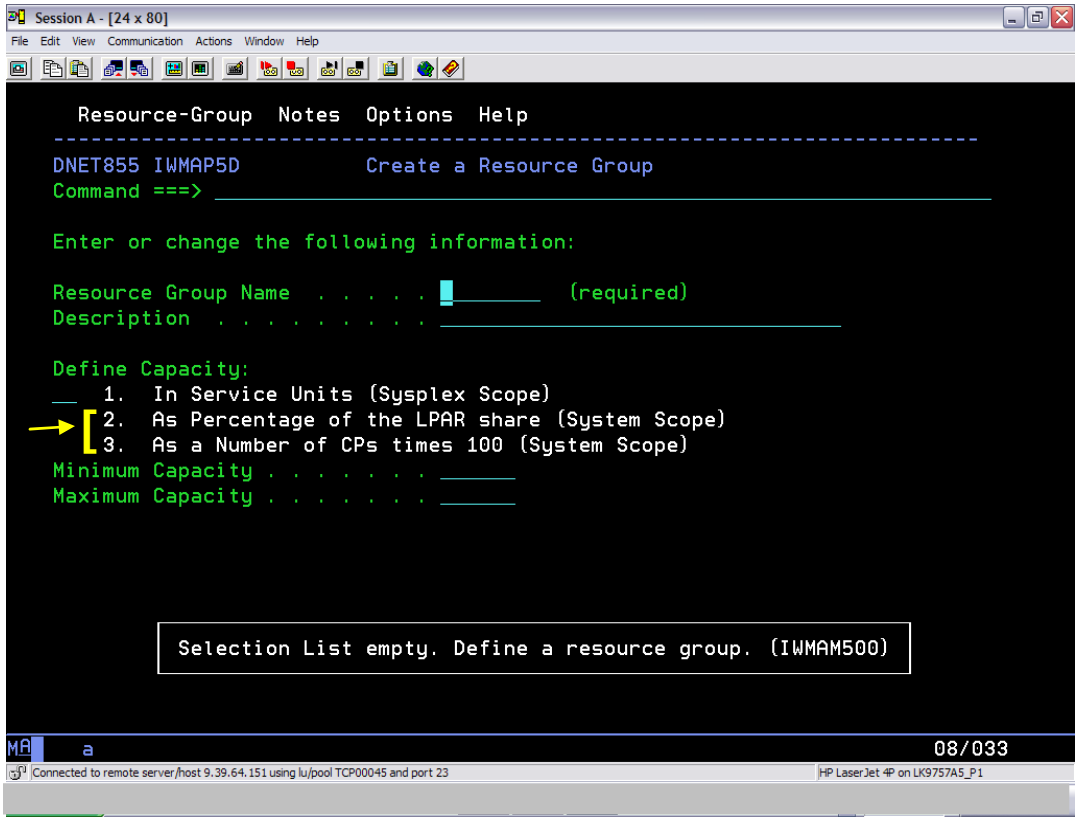
- Definition is based on one of two possible units
  a. LPAR capacity: based on LPAR weight **<= Type 2 RG**
  b. Logical CP (LCP) capacity **<= Type 3 RG**



Results

- New RGs are managed by the image thus they are evaluated on a per system bases
- These RG types grow automatically when systems are upgrade
- Easier to understand

The Workload Management panels have been updated to use the new RG types.



Type 2: Capacity is specified as a percentage of the effective LPAR share.
- Range 0…99
- Effective LPAR share is the minimum of:
  a. CPC shared pool capacity * (LPAR Weight / SUM of Weights)
  b. Logical Processor capacity
  c. Defined capacity limit (when software capping is in effect)

Type 3: Capacity is expressed as an equivalent number of general purpose processors
- Scaled by 100, i.e. 100 equivalent to 1 CP (range: 0….999999)

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

Resource Group Considerations with zAAP and zIIPS

- Resource Groups capping is determined by consumption for the resource group of its general processor usage
  NOTE: zAAPa and zIIPs are ignored for determining when capping starts and stops
- Capping means to set the work units running in service classes associated with resource groups non-dispatchable
  - This affects all processor types
  - Therefore RG capping *also* stops the work running on specialty processors

Other Considerations
- Consider scope: Sysplex v.s. System
- Not valid for transaction oriented work with CICS or IMS transactions
  - In order to assign a minimum capacity to CICS or IMS transactions, the regions (address spaces) can be assigned to a resource group
  - Given the combination of the goals, the importance level and the resource capacity, some goals may not be achievable when capacity is restricted
  - Unless there is a specific need for limiting or protecting capacity for a group of work, it is best *not* to define resource groups and to let workload manager do it job to manage processor resources to meet performance objectives.

1. Ever wonder why your CPU utilization still hits 100% the day after an upgrade. Further concern, does the 100% busy mean 100% busy or more than 100% busy? Latent demand is the true demand, both satisfied and unsatisfied of a z/OS workload. The latent demand model uses standard queue lengths from the IN&READY queue (a.k.a. true ready queue or TRQ) and develops an estimator of the true demand expressed in units of percent busy, where demand over 100% indicates latent demand is present. For example, a value of 80% busy indicates all demand was satisfied, where as a value of 120% indicates 20% was unsatisfied and a configuration that could supply 1.2 times as many MIPS as the current system would be needed to satisfy the true/total demand. This is extremely important when it comes to comprehending CPU sizing.

2. **Service Unit:** The amount of service consumed by an address space and is computed by the formula:

service = (CPU x CPU Service Units)
+ (SRB x SRB Service Units)
+ (IOC x I/O Service Units)
+ (MSO x Storage Service Units)

where

CPU Service Units = task (TCB) execution time, multiplied by an SRM constant which is CPU model dependent.

SRB Service Units = service request block (SRB) execution time for both local and global SRBs, multiplied by an SRM constant which is CPU model dependent.

I/O Count Service Units = measurement of individual dataset I/O activity and JES spool reads and writes for all datasets associated with the address space.

Main Storage Service Units = (central page frames) x (CPU service units) x 1/50, where 1/50 is a scaling factor designed to bring the storage service component in line with the CPU component.

- - -