z/OS® Workload Manager
Sysplex Routing Services

Tuesday, 25 April 2011

Robert Vaupel
STSM, z/OS Workload Management
Schoenaicherstr. 220
D-71032 Boeblingen
+49-7031-16-4239
vaupel@de.ibm.com

The following document gives a brief description how WLM supports sysplex routing. The routing is performed by program products like IBM Communication Server or subsystems like DB2 or Websphere. WLM offers a set of application programming interfaces which assist the routing services to distribute work in a sysplex environment.

WLM Sysplex Routing Services

The sysplex routing services allow work associated with a server to be distributed across a sysplex. They are intended for use by clients and servers. A client is any application or product in the network that requests a service. The service could be a request for data, a program to be run, or access to a database or application. In terms of the sysplex routing services, a client is any program routing work to a server. A server is any subsystem address space that provides a service on an MVS image.

WLM primarily supports three external routing services:

1. IBM Communication Services or Sysplex Distributor
2. DB/2 Gateway
3. Websphere

The WLM programming interfaces for Sysplex Distributor (1) and DB/2 Gateway are described in [WLMSERV] chapter 8 and the programming interfaces are described in detail in the reference section of this manual. The programming interfaces for Websphere are not described externally.

WLM Routing Services for Sysplex Distributor

Sysplex distributor offers two modes how it obtains routing recommendations from WLM:

1. BASEWLM: This mode requires that all target instances register to WLM by using the IWMSRSRG service. WLM is then able to calculate a routing recommendation across all instances which have been registered with the same location name, network id and LU name. The recommendation is a value between 0 and 64 and the sum of all target instances is 64.
2. SERVERWLM: This mode is available to registered servers which pass the work to another processing instance. A registered server is for example the TCPIP address space and the processing server is a CICS Terminal Owning Region (TOR). The TOR just opens a port to TCPIP address space. WLM now offers a service (IWM4SRSC) which allows to query a routing recommendation for a specific server and the work which is processed by this server.

Sysplex Distributor Option: BASEWLM

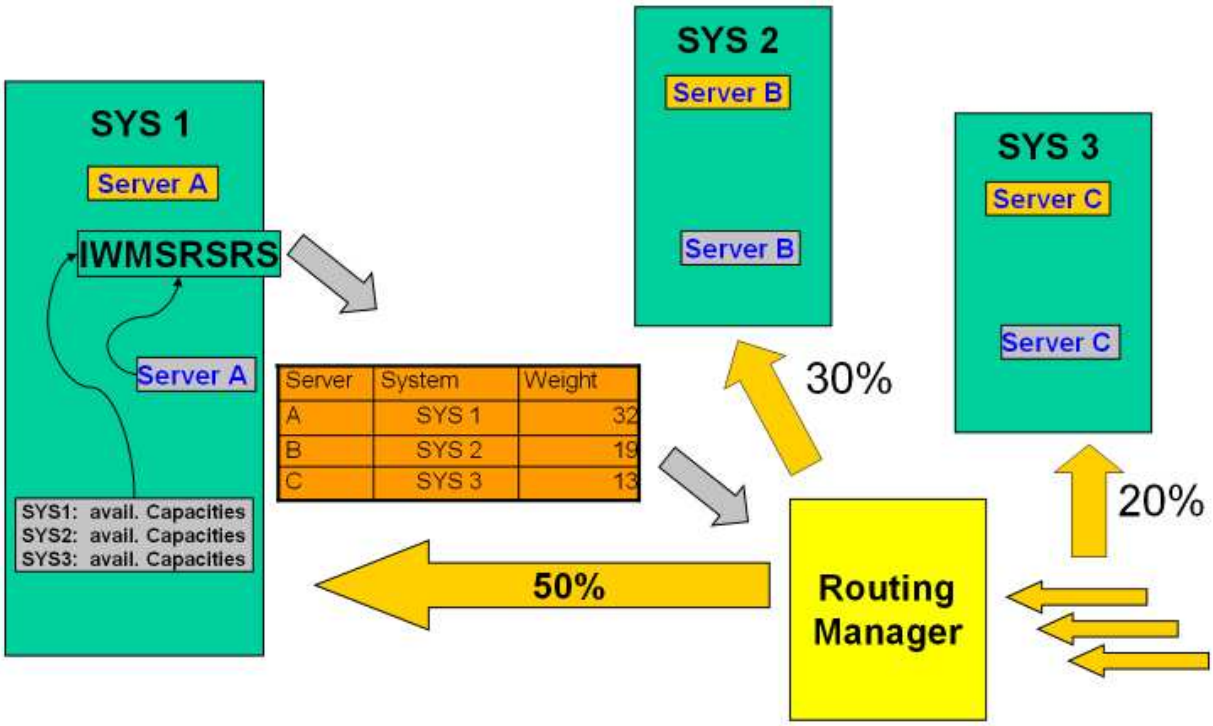
For BASEWLM Sysplex Distributor uses 3 services:

- IWMSRSRG lets a caller register as a server
- IWMSRSRS provides the caller with a list of registered servers and the number of requests that should be routed to each server
- IWMSRSRD lets the caller deregister as a server

One advantage of these set of services is that it is only necessary to call IWMSRSRS once in a sysplex to obtain a routing recommendation for all servers which are registered by the same location name, network id and LU name.

The following picture shows the usage of IWMSRSRS[1] service in a 3 system sysplex environment. WLM bases the routing recommendation on the CPU consumption for each importance level across all systems. It measures the CPU consumption for the importance levels on each system and sends this information to all systems in the sysplex. Then it is possible to calculate a routing recommendation for all servers in the sysplex environment and to return this information on any system to the caller of the IWMSRSRS service.

[1] Sysplex Distributor call IWMSRSRS with FUNCTION=SELECT, see also chapter „WLM Routing Services for DB2“ on page 8.



The importance table on each system consists of 8 rows:

- Row 0 is for system work (corresponds to service classes SYSTEM and SYSSTC)
- Row 1 to 5 corresponds to the importance definition for each service class in the service definition
- Row 6 corresponds to discretionary work or work without a specific goal of the service definition
- Row 7 corresponds to free capacity of the system

The algorithm now scans the table from top to bottom and until it finds a level where at least 1 system has at least 1% of cumulative service units of capacity. The it calculates a system weight by dividing the capacity of the table row with the cumulative capacity of all rows across all systems which have at least 1% capacity at this level. If more than one server is located on the system the system weight is divided by the number of the servers on this system. If a system does not have at least 1% of cumulative capacity at this level the weight is set to 0. The total results are scaled to 64 so that the sum of all weights for all servers totals at 64.

Formulas:

$$\text{Level}_k = \text{Selected Importance Level } k = \{0,..,7\} \text{ with at least } 1\% \text{ of cumulative capacity}$$

$$\text{Weight}(\text{System}_i) = \frac{\text{ServiceUnits}(\text{System}_i; \text{Level}_k)}{\sum_{i=1}^{N} \text{ServiceUnits}(\text{System}_i; \text{Level}_k)} \times 64$$

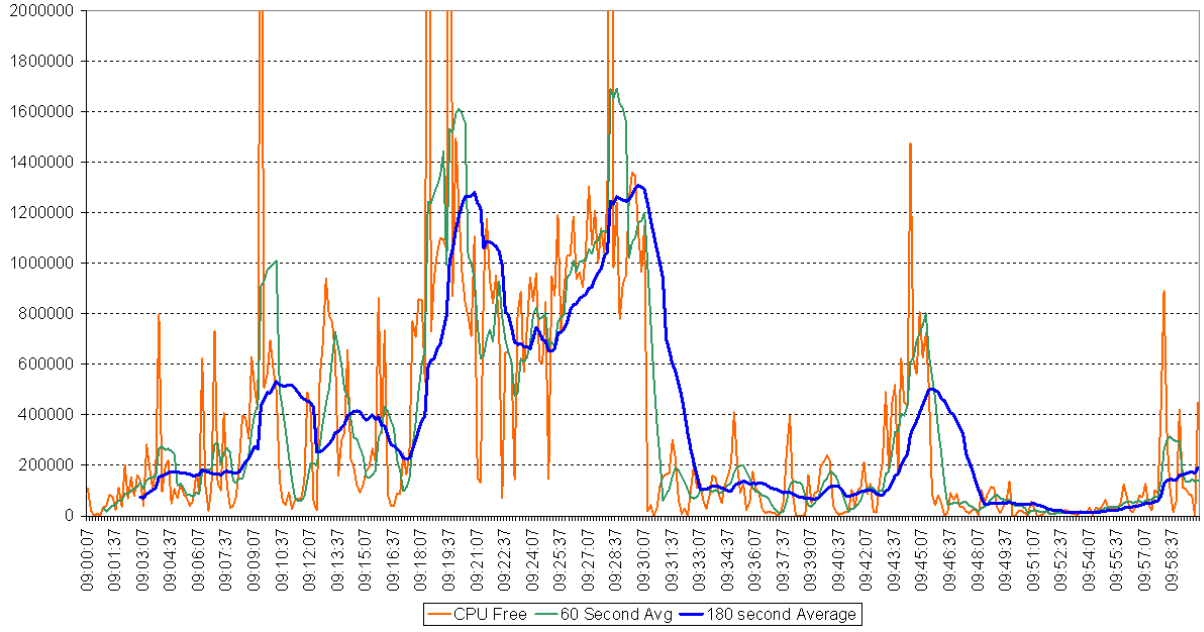
$$\text{Weight}(\text{System}_i; \text{Server}_j) = \frac{\text{Weight}(\text{System}_i)}{M} \text{ with } j = \{1,..,M\} \text{ and } i = \{1,..,N\}$$

Example:

The following example calculates the system weight for 3 servers located on three different systems. The three systems SYS1, SYS2, and SYS3 show different capacities 2000, 1500 and 1000 service units. WLM now scans the table from the bottom and finds the first capacity at importance level 5 on each system. Each system shows more than 1% of its capacity at this level. Based on the formula above the weights returned for the three servers A (on SYS1), B (on SYS2), and C (on SYS3) is 13, 32, and 19. The weight for server A is calculated as Server A = 120 / 600 * 64 = 13 with 600 the sum of the capacity at importance level 5 across all systems.

	SYS 1		SYS 2		SYS 3	
Level	SUs	% of system	SUs	% of system	SUs	% of system
0	2000	100	1500	100	1000	100
1	1800	90	1200	80	800	80
2	1600	80	900	60	700	70
3	1200	60	700	47	500	50
4	400	20	500	34	300	30
5	120	6	300	20	180	18
6	0		0		0	
7	0		0		0	

In order to provide stable information WLM calculates the CPU consumption for each importance level as a rolling average over a 180 second interval. Every 10 seconds the latest measurement data is added and the oldest 10 second interval is removed. The following picture shows this effect for measured free capacity (Row 7)[2] on z/OS system:



Note:
It should be noted that storage constraint systems are also eliminated from the list of eligible systems to which work is being routed to. This applies to all routing algorithms discussed in this paper.

Sysplex Distributor Option: SERVERWLM

The usage of services IWMSRSRS as discussed before has several major disadvantages:

1. The way the services calculates the routing recommendation based on CPU consumption for importance levels completely ignores on which importance level the work is being processed. That

[2] The blue line is the 180 second rolling average of free capacity. The orange line are the 10s measured free capacity. The green line is a 60 second average to depict the smoothing effect.

can result in eliminating a system which has a lot of low important work running while the work being routed could easily being processed.

2. The service does not consider how well the work is processed. No information whether work achieves its performance goal nor whether the work is processed properly is considered in the routing recommendation.
3. The TCPIP address space registers as a server but the TCPIP address space does not process the work request. Therefore it is not possible to consider any additional information related to the work execution.

In order to overcome these limitations a new service IWM4SRSC has been introduced which is used by Sysplex Distributor if the installation selects the routing option SERVERWLM. In order to depict the change of routing weights we use the same example as for BASEWLM and we assume:

- That the work is passed to a server environment which processes the requests, for example CICS with TORs and AORs.
- That the work is classified to a service class with an importance definition of 2.

	SYS1		SYS2		SYS3	
Level	SUs	% of system	SUs	% of system	SUs	% of system
0	2000	100	1500	100	1000	100
1	1800	90	1200	80	800	80
2	1600	80	900	60	700	70
3	1200	60	700	47	500	50
4	400	20	500	34	300	30
5	120	6	300	20	180	18
6	0		0		0	
7	0		0		0	

The table above now shows the same CPU consumption as before but now the IWM4SRSC service uses the information at importance level 2 on each system. In addition the service uses the capacity of biggest system which is SYS1 as denominator for calculating the weights. It now divides the cumulative service units at importance level 2 by the capacity of SYS1 and multiplies the result with 64. The result is the base weight for each server and depicted in the table below:

Server	SERVERWLM	BASEWLM
A	51	13
B	29	32
C	22	19

It can be observed that now the weight for server A on system SYS1 is much bigger than for the other two servers because SYS1 has 1600 SUs at importance 2 while SYS2 only has 900 and SYS3 only 700. The new weight now reflects much better the capability of the server and system to process the work at the corresponding importance level.

Note:
The sum of the weights is no longer 64. Only the weight of each server is in the range from 0 to 64.

Note:
IWM4SRSC must be invoked on each system. The caller is responsible for requesting the data and combining the results in order to obtain a sysplex routing recommendation. This is necessary because WLM does not know which servers are connected to the registered servers which issue IWM4SRSC.

After calculating the base weight the weight is divided by the goal achievement value (PI = Performance Index) of the service class to which the work is being routed if the PI > 1. Otherwise the weight is unchanged. The PI for the work is determined in the following way:

- If the work is routed to a CICS or IMS subsystem with response time goals, the service classes in which most transactions execute are used to determine the importance level and the PI.
- If the work is routed to an enclave server the enclave service class which show the highest completions are used.
- Otherwise the service class of the server is used.

Note:
The PI lowers the routing weight if the goal is not achieved. It is possible that this effect is not desired if an installation is not able to define the goals in a way that the PI really indicates a problem if it is above 1. Therefore it is possible to smooth the effect of the PI by specifying the OPT parameter RTPIFACTOR on all systems in the sysplex. Also RTPIFACTOR effects all routing calculations of IWM4SRSC for the system.

Using the WLM Health Service: IWM4HLTH

As a final possibility the subsystem, a monitor, or a supervisor function is able to influence the returned weight of the IWM4SRSC service by using the IWM4HLTH services for the target server[3]. The IWM4HLTH service allows the function to set a value from 0 to 100 which indicates how healthy the environment is. A value of 100 means that the processing environment and the server are healthy and the weight is not degraded. Otherwise the weight is multiplied with the defined value and divided by 100. This also allows a monitoring function to stop all routing recommendations to a server which encounters problems by setting the health indicator to 0.

Extended Options for IWM4SRSC

The introduction of offload processors (zAAP and zIIP) as well as customer feedback resulted in various extensions to the IWM4SRSC service. These extensions are exploited by Sysplex Distributor:

Parameter	Value Range	Description
WEIGHT	0 .. 64	Calculated weight for the server (specified by input parameter STOKEN). If offload processors are present on the system this is the combined weight across all processor types.
CPUWEIGHT ZIIPWEIGHT ZAAPWEIGHT	0 .. 64	Individual weights for each processor type.
CPUPROPORTION ZAAPPROPORTION ZIIPPROPORTION	0 .. 100	Proportion of weights in WEIGHT calculation
ABNORM_COUNT	0 .. 32767	Number of abnormal completions (currently only supported by CICS)
HEALTH	0 .. 100	Health value defined for the server environment
METHOD	EQUICPU, PROPORTIONAL	An optional parameter which allows to define the weight calculation method. PROPORTIONAL is the default but for some functions EQUIV is required
COST_ZAAP_ON_CP COST_ZIIP_ON_CP	1 .. 32767	An optional value which allows to influence the weight calculation by defining an additional cost if zAAP or zIIP capacity is exhausted and work must be executed on regular processors. The parameter is meaningful if zIIPs or zAAPs are not equally installed on all systems in the sysplex environment. In order to exploit this function METHOD=EQUICPU must be specified.
IL_WEIGHTING	0, 1, 2, 3	An optional parameter which allows to give work executing on lower importance levels a higher weight

[3] This is the server which opens a port to TCPIP address space and which is the target for the work request from a TCPIP point of view.

The parameter IL_WEIGHTING has been introduced to acknowledge the possibility that systems in a sysplex environment execute workloads at different importance levels. IWM4SRSC treats all work at lower importance levels as a subject to be displaced. It doesn't distinguish between the lower importance levels anymore. With IL_WEIGHTING a distinction is possible. The parameter supports 4 values

0. Constant, this is the default and lower importance levels will not be distinguished
1. Logarhythmic, low distinction
2. Linear
3. Quadratic

The effect of IL_WEIGHTING is illustrated in the table below. The same example as before is used and the routing weights for the IWMSRSRS service are also added:

IWM4SRSC	ILWEIGHTING	SYS1 Server A	SYS2 Server B	SYS3 Server C
Base	0	32	19	14
Logarhythmic	1	30	21	14
Linear	2	28	22	14
Quadratic	3	23	27	14
IWMSRSRS		13	32	19

It can be observed that the weight for server A on SYS1 is reduced and the weight for server B on SYS2 increased because more work is executed at importance 5 on SYS2.

Extended Options for IWMSRSRS

IWMSRSRS also supports the extended options as described for IWM4SRSC[4]. The options are either returned by the return data area of IWMSRSRS or via optional output parameters on the invocation of the interface. Please refer to [WLMSERV], chapter 61 IWMSRSRS.

WLM Routing Services for DB2

DB2 uses the IWMSRSRS service in two flavors:

- FUNCTION=SELECT: This is identical to the description for option BASEWLM for Sysplex Distributor
- FUNCTION=SPECIFIC: This is an extension which also includes additional information for the returned weight.

IWMSRSRS FUNCTION=SPECIFIC

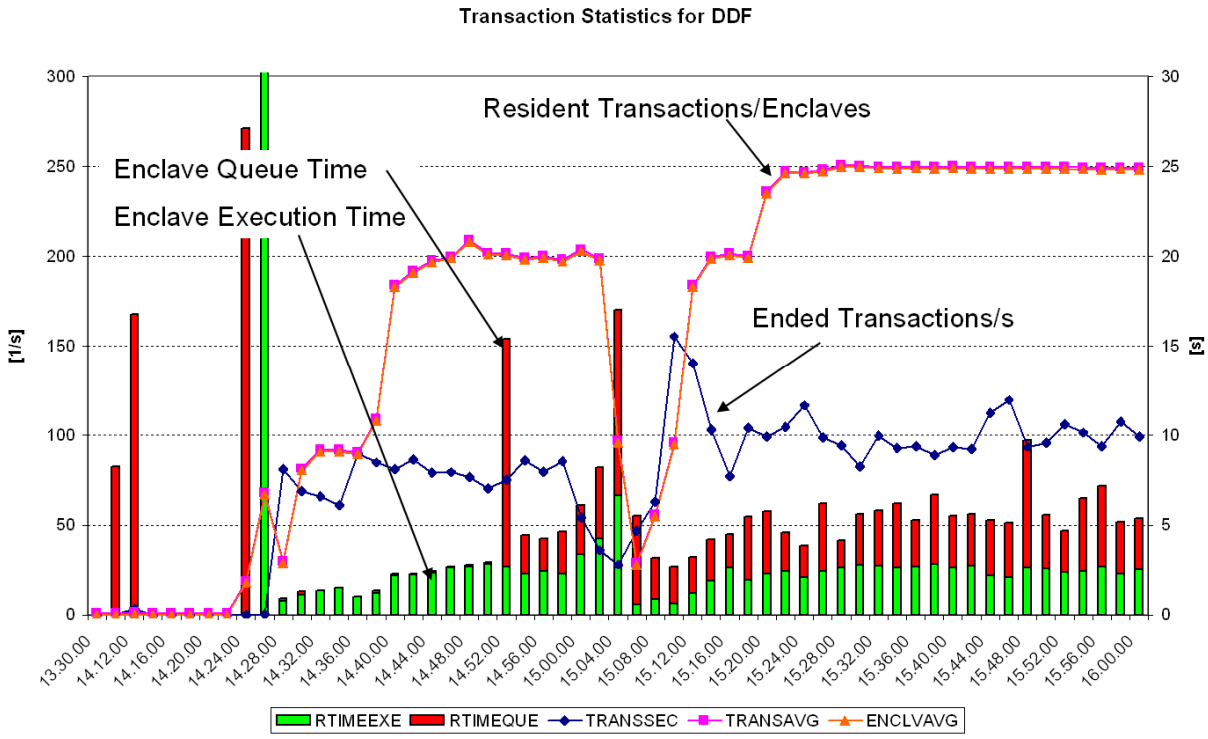
This function also includes additional information related to the server environment and goal achievement of the work similar to service IWM4SRSC. The main difference is that the base weight is calculated as described for Sysplex Distributor for option BASEWLM. The additional information is

1. The goal achievement value (PI) in the same way as for IWM4SRSC
2. The health indicator as described for service IWM4HLTH and IWM4SRSC
3. The queue time proportion for enclaves

[4] Except the CPUPROPORTION, ZAAPPROPORTION and ZIIPPROPORTION.

For bullet 1 the service class is determined for the work being routed. This is accomplished in the same way as described for IWM4SRSC on page 6. The health indicator and the queue time proportion is derived for the registered server and send to all other systems in the sysplex.

For DB2 the registered server is the DIST address space of the distributed data facility component of DB2. The DIST address space functions as the work receiver which also classifies the work request and brings it into the system for execution. After classifying the work request the DIST address space creates an enclave which associates the work with a service class.
One possible contention point of the DIST address space are the number of threads to process incoming work requests. If for any reason a contention situation occurs it is possible that the threads can't process the incoming requests and the requests need to queued in the DIST address space. The DIST address space tells WLM the time difference from receiving the request until it creates the enclave and schedules an SRB to process the work request. This time difference is treated as queuing time and can be observed for the service class which is associated with the enclaves. If the queue time of the enclave service time is very large it can be determined that the DIST address space has a problem and the weight recommendation needs to be adjusted. The following graphic shows a scenario with very high queue times for the enclaves of the DDF work. This is a result of internal queuing in the DIST address space.



The proportion of queue time to enclave response time (queue + execution time) is used to reduced the routing weight:

$$\text{Returned Weight} = \text{Base Weight} \times \frac{\text{execution time}}{\text{execution time} + \text{queue time}}$$

WLM Routing Services for Websphere

WLM Routing Services for Websphere are not officially documented[5]. The routing algorithm is similar to the routing algorithm for IWMSRSRS FUNCTION=SPECIFIC, meaning:

[5] But they are official interfaces which can also be used from other products than Websphere if required.

- The algorithm uses the same capacity based routing algorithm to determine the preferred Websphere Application Control Region (ACR)
- The algorithm also uses the aggregated Performance Index (similar to the SPECIFIC function) to determine the routing weight.

The main differences of the algorithms are

- Websphere must invoke the programming interface (IWMSRCRI) to obtain a routing recommendation for each request which should be routed to an ACR
- WLM calculates the routing weights every 10 seconds and assigns the requests in a round robin fashion to the ACRs until all weights have been used. Then the original weights will be re-loaded until the weights are recalculated

Example of Selecting ACRs

Let's assume we have three ACRs: ACR1, ACR2, and ACR3 for which WLM calculated routing weights of

- ACR1 = 32
- ACR2 = 12
- ACR3 = 20

Please note that the sum of the weights is again 64[6]. The following table shows how WLM will select one of the 3 ACRs and hands the requests to them. After 64 calls have been processed the initial weights are re-loaded:

Request Call	Request send to ACR1	ACR2	ACR3	Remaining Weight of ACR1 32	ACR2 12	ACR3 20
1	x			31	12	20
2		x		31	11	20
3			x	31	11	19
4	x			30	11	19
5		x		30	10	19
6			x	30	10	18
7	x			29	10	18
8		x		29	9	18
9			x	29	9	17
10	x			28	9	17
11		x		28	8	17
12			x	28	8	16
31	x			21	2	10
32		x		21	1	10
33			x	21	1	9
34	x			20	1	9
35		x		20	0	9
36			x	20	0	8
37	x			19	0	8
38			x	19	0	7
49	x			13	0	2
50			x	13	0	1
51	x			12	0	1
52			x	12	0	0
53	x			11	0	0
54	x			10	0	0
55	x			9	0	0
63	x			1	0	0
64	x			0	0	0

[6] Deviations are possible due to rounding.

Determining the Aggregated Performance Index

The final weight for the ACR as shown in the table above is first based on the capacity of the ACR (Sysplex Weight), see determining the sysplex weight for IWMSRSRS for BASEWLM on page 3 ff. Then an aggregated PI is applied which reflects the ability of the ACR to process the work. This is similar to IWMSRSRS FUNCTION=SPECIFIC but the aggregated PI reflects the ability of the ACR to process more than 1 service class[7].

The following example assumes that an ACR processes work for 4 service classes:

Service Class	Importance	PI	# Transactions
A	2	1.6	25
B	2	1.4	175
C	3	2	100
D	4	1.8	100

WLM now uses data for at least 75% of the most important work for the ACR which is the information for the service classes A, B, and C of the example:

$$\text{Aggregate PI} = \frac{(1.6 \times 25) + (1.4 \times 175) + (2.0 \times 100)}{(25 + 175 + 100)} = 1.62$$

The final weight is then calculated as

$$\text{Final Weight} = (1 - (\text{Aggregated PI} - 1)) \times \text{Sysplex Weight}$$

But only if the Aggregated PI > 1 otherwise the Sysplex Weight is used as Final Weight. Also if the Aggregated PI is greater than 1.5 the Final Weight is set to 0. Example:

Control Region	System	Aggregated PI	Sysplex Weight	Adjusted Weight	Final Weight
ACR1	SYSA	0.8	16	16	26
ACR2	SYSA	1	16	16	26
ACR3	SYSB	1.4	12	7	12
ACR4	SYSC	1.6	20	0	0

Using Round Robin for Work Distribution

The described algorithm to distribute work between ACRs has been fully introduced since z/OS 1.9. Prior to this level WLM only distributed work based on a simple round robin algorithm between the available ACRs. It is still possible for an installation to select which routing algorithm should be used. An installation can specify WASROUTINGLEVEL=1 for round robin or WASROUTINGLEVEL=0 for the capacity and performance based algorithm which is the default, see also [INITREF].

[7] For IWMSRSRS FUNCTION=SPECIFIC the service class with the highest number of completions is selected.

Summary

The following table summarizes the exploitation of WLM routing services by Sysplex Distributor, DB/2 and Websphere. All functions as described in the table are fully available since z/OS 1.11:

Exploiter	Service and Function	Scope	Recommendation based on						Remarks
			System Capacity	Server Capacity	PI	Server Queue.	Abnorm. Termin.	Health Status	
Sysplex Distributor BASEWLM	IWMSRSRS FUNCTION=SELECT	Sysplex	X						Individual Weights by Processor Type Cost ZAAP and ZIIP on CP
Sysplex Distributor SERVERWLM	IWM4SRSC	LPAR(1)		X	X		X	X	(1) Caller must invoke service for each server Individual Weights by Processor Type Cost ZAAP and ZIIP on CP Importance Level Weighting
DB/2	IWMSRSRS FUNCTION=SPECIFIC	Sysplex	X		X	X		X	
Websphere	IWMSRCRI	Sysplex	X		X(2)				(2) Aggregate PI WASROUTINGLEVEL to determine routing algorithm

References

[AAMDDPW]	Adaptive Algorithms for Managing a Distributed Data Processing Workload – Aman et. al., IBM Systems Journal, Volume 36, Number2, 1997
[REDBWLM]	Systems Programmer's Guide to: Workload Manager, IBM Redbook – SG24-6472-xx
[WLMPLAN]	MVS Planning: Workload Management – SA22-7602-xx
[WLMSERV]	MVS Programming: Workload Management Services – SA22-7619-xx
[WLMWORK]	z/OS Workload Manager; How it Works and How to use it
[INITREF]	z/OS Initialization and Tuning Reference – SA22-7592-xx