

Who is Using my CPU's?



1 Introduction

At any z/OS site, the CPU is the most expensive component in terms of both hardware and software costs. This is the reason why some companies perform tuning cycles on a regular basis in order to:

1. Understand who is using the CPU;
2. Estimate possible CPU savings;
3. Estimate the corresponding tuning efforts;
4. Tune system and application components.

Understanding who is using the CPU may look like a simple task to perform but, unfortunately, this is not always true, especially in complex environments with multiple LPARs, each of which possibly hosting many CICS regions, DB2 subsystems, etc.

The major issue to face is that most of the available performance analysis and tuning tools don't provide a comprehensive view of the whole environment but only a lot of partial views each focusing on a system or subsystem.

So it may happen that a task running in multiple systems and/or subsystems doesn't appear to be an intensive CPU consumer when focussing on each specific instance, but the situation can be very different when analyzing its global CPU usage.

In this paper, we will discuss a simple technique which will allow you to easily perform the first essential step of the CPU tuning cycle and to find major CPU consumers by taking the whole z/OS environment into consideration.

2 Defining objects

Most z/OS workloads are composed of many different workload types, all of which present different characteristics:

- CICS and IMS applications use a transaction UOW - identified by the transaction ID provided in SMF 110 (TRAN) and in the IMS log records;
- DDF users use a transaction UOW - identified by different SMF 101 fields, such as the user authorization ID or location (QWHCAID, QLACLOCN);
- JOB, OMVS and STC applications use address space UOW - identified by the address space ID which is provided in SMF 30 (SMF30JBN);
- TSO users use address space UOW - also identified by the address space ID which is provided in SMF 30 (SMF30JBN).



WebSphere transactions could also be included in the analysis, but only if you collect SMF 120 subtype 9 records¹.

In the technique we are describing here, we will consider any transaction, user and address space, as identified above, as tuneable “objects”.

Figure 1 below summarizes this concept.

OBJECT TYPE	OBJECT UOW	OBJECT ID
IMS	Transaction	transaction ID
CICS	Transaction	transaction ID
DDF	Transaction	user authid/location
JOB	address space	address space ID
OMVS	address space	address space ID
TSO	address space	address space ID
STC	address space	address space ID

Figure 1

As you can see, we will consider 7 object types; inside each of them there will be many object IDs.

The only required metric is the number of CPU seconds² used as provided by collected data (we will not take into account capture ratio issues), which can be summarized at any temporal scale (shift, week, month). In the case study presented in this paper we will use a daily summary.

Summarized data will be sorted by descending values so that the most CPU-intensive object will show at the top of the rank.

3 Results in a real case study

The result of an analysis performed at a customer’s site on August 1st showing the TOP 20 OBJECTS is presented in Figure 2 on the next page.

The first object in the rank is an IP address. Unfortunately this customer identifies DDF users by their location instead of authorization ID when collecting data in the EPV database.

In this case, the IP address belongs to a server acting as a gateway to DB2 for most of the DDF requests coming from outside z/OS. This choice should be reviewed in order to get more meaningful information.

However, you can see all these DDF requests were served by 6 DB2 subsystems (SS column), residing on 2 systems (SYSTEM column).

¹ All the other SMF 120 subtypes provide cumulative CP time including CPU, zAAP, and zIIP.
² This technique can easily be extended to zAAP and zIIP.



OBJ TYPE	OBJ ID	CPU sec	SYS-TEMS	SS
DDF	10.120.3.26	28,010	2	6
JOB	USER1A	15,462	1	
STC	NETVIEW	13,346	4	
JOB	OPCJ2401	13,104	1	
TSO	USER2	12,218	1	
JOB	OPCJ6172	11,059	1	
STC	MAINVIEW	9,836	4	
JOB	OPCJ6101	8,518	1	
STC	CATALOG	7,898	4	
STC	BETA92	7,770	2	
JOB	USER2Q	6,670	1	
JOB	OPCJ2150	5,708	1	
JOB	OPCJ1302	5,623	1	
IMS	IMSA7M10	5,440	1	1
JOB	OPCJ1202	5,207	1	
JOB	OPCJ4004	5,196	1	
JOB	OPCJ6401	5,066	1	
JOB	OPCJ1331	4,457	1	
JOB	OPCJ6152	4,178	1	
OMVS	USS42724	3,822	1	

Figure 2

The second object is a user batch job, which consumed more than 15,000 CPU seconds. Going deeper in the analysis we discovered that all these consumptions were in a single DB2 package; DBAs are now working on this.

The third object is the NETVIEW automation address space; note that it runs in 4 systems.

The fourth object is a scheduled batch job; it presented two issues: it used to run in the daily time shift (between 8:00 a.m. and 6:00 p.m.), instead of during the night, it used more than 13,000 CPU seconds. We discovered a third issue when we tried to fix a meeting with the operation team to re-schedule and tune the job: the job was obsolete. So it was simply removed from the scheduling plan altogether.

The fifth and eleventh objects belong to the same TSO user who runs very complex and CPU-intensive QMF queries, both in TSO (USER2) and in batch (USER2Q); DBAs are now working to provide him with more efficient (and controlled) queries to run.

OBJ TYPE	OBJ ID	SYSTEM	SS	CPU sec
DDF	10.120.3.26	SYSB	DBD2	25,637
DDF	10.120.3.26	SYSB	DBD1	1,639
DDF	10.120.3.26	SYSA	DBDD	428
DDF	10.120.3.26	SYSB	DBD4	304
DDF	10.120.3.26	SYSB	DBDP	1
DDF	10.120.3.26	SYSB	DBDN	1
JOB	USER1A	SYSB		15,462
STC	NETVIEW	SYSB		8,634
STC	NETVIEW	SYSA		2,730
STC	NETVIEW	SYSC		1,772
STC	NETVIEW	SYST		211
JOB	OPCJ2401	SYSB		13,104
TSO	USER2	SYSB		12,218
JOB	OPCH6172	SYSB		11,059
STC	MAINVIEW	SYSB		5,989
STC	MAINVIEW	SYSC		1,903
STC	MAINVIEW	SYSA		1,861
STC	MAINVIEW	SYST		83
JOB	OPCJ6101	SYSB		8,518
STC	CATALOG	SYSB		6,039
STC	CATALOG	SYSA		1,198
STC	CATALOG	SYSC		653
STC	CATALOG	SYST		8
STC	BETA92	SYSB		7,707
STC	BETA92	SYST		63
JOB	USER2Q	SYSB		6,670
JOB	OPCJ2150	SYSB		5,708
JOB	OPCJ1302	SYSB		5,623
IMS	IMSA7M10	SYSA	IMSD	5,440
JOB	OPCJ1202	SYSB		5,207
JOB	OPCJ4004	SYSB		5,196
JOB	OPCJ6401	SYSB		5,066
JOB	OPCJ1131	SYSB		4,457
JOB	OPCJ6152	SYSB		4,178
OMVS	USS42724	SYSB		3,822

Figure 3

More details are provided in Figure 3 to the left, which shows the split of CPU consumption by system and subsystem.

As you can see, most of the DDF consumption is in the DBD2 subsystem, hosted in the SYSB system.

Also note that NETVIEW, MAINVIEW and CATALOG consumptions are much higher in SYSB than in the other 3 systems.

4 Conclusions

The information provided in Figures 2 and 3 were only the starting point for a much deeper analysis, which produced a lot of tuning ideas and actions which are still in progress.

We have briefly described some of them in this paper. The goal was to show how this simple “top objects” technique can provide useful and sometimes unexpected results.

SEGUS Inc is the North American distributor for EPV products

For more information regarding EPVB for z/OS, please visit www.segus.com or call (800) 327-9650