

Specialty engines such as zIIPs and zAAPs are an integral part of IBM zSeries hardware. Reporting their use accurately can be confusing, especially if you are not running full size general purpose processors. This presentation will walk through what a specialty engine is, and how their use is reported in TYPE70 and TYPE72 records. It will cover the confusion that can result when general purpose engines are not full size – how a transaction can say it used more CPU time than the transaction actually took. It will then clear the confusion by showing various methods to present the data, and show how to tell the story you want your data to tell.

What We Will Cover



- Definitions
- What we can measure
 - TYPE 70 information
 - TYPE72 information
- What the information tells us
- What we can learn from it



We will review how mainframe CPU use is reported, and how sub capacity processors affect reporting specialty engine use. We will do this by looking at the RMF TYPE70 and TYPE72 records for the data pertinent to today's environment. We will then examine what these measurements tell us; what we can learn from them; and what we can conclude about what and how to measure and report on specialty engine use when we have sub capacity processors.

Definitions



- **Specialty Engines**
 - zIIPs, zAAPs
 - Ignore other types as not applicable to z/OS
- **GPs**
 - general purpose processors
- **Kneecapped processors**
 - Processors whose GPs do not run at full speed



We start by defining terms so we understand what we want to talk about. The term “specialty engines,” in this paper, refers only to the IBM zIIP and zAAP processors which can run a specialized subset of z/OS work. These engines have been described in other papers and will not be explained again here. While there are other types of specialty engines such as coupling facility engines (ICFs) and the Internal Facility for Linux (IFLs) they do not run in partitions running z/OS so they will not be included in this paper.

The term GP, first used by Cheryl Watson of Watson and Walker, will be used to refer to a general purpose engines to distinguish them from specialty engines. That will leave the generic term CP available to refer to all engine types.

Sub capacity processors, or “kneecapped” processors as Kathy Walsh of IBM Advanced Technical Skills (formerly Washing Systems Center) has called them, are mainframes in which all CPUs do not run at the full speed they are capable of running. This only applies to the general purpose CPs, not to specialty engines. Specialty engines always run at full speed. This can cause confusion in reporting use – it is that confusion this paper will address

What We Can Measure TYPE 70



- SMF70NRM normalization factor for zIIP
- SMF70CIN CPU type (CP, IFL, AAP, IIP, ICF)
- SMF70CTN CPs of type SMF70CIN online at End of Interval (EOI)
- SMF70CAN accumulated CPs during the interval
- SMF70DSA number of diagnose samples
- SMF70IFA number of online zAAPs at EOI
- SMF70SUP number of online zIIPs at EOI



Let us first look at the TYPE70 information we want to use. Since we are looking specifically at kneecapped processors, field SMF70NRM is very important to us. This field gives us a number which we can divide by 256 and see how much faster the specialty engines are than the GPs (it is identical to fields R723NFFI and R723NFFS which have been available in the TYPE72s for years). Field SMF70CIN tells us what kind of CP we are running – GP, zAAP or zIIP. Field SMF70CTN tells us how many CPs of this type were online at the end of the RMF interval.

Field SMF70CAN tell us how many processors have accumulated during the interval. If we divide SMF70CAN by SMF70DSA (the number of diagnose samples) we can see the average number of processors online during the interval. If you are still using Intelligent Resource Director (IRD) instead of HiperDispatch for CPU management this number may be of more interest to you than SMF70CTN.

For specialty engines, field SMF70IFA tells us how many zAAPs were online to this LPAR at the end of the interval while field SMF70SUP gives the same information for zIIPs.

What We Can Measure TYPE 70



- SMF70PAT CPU parked time
- SMF70WAT CPU wait time
- SMF70PDT LP dispatch time
- SMF70EDT LP effective dispatch time
- SMF70CIN (again)



To get actual use for individual processors, the following fields are of value. SMF70PAT gives us parked time if HiperDispatch is active, SMF70WAT gives us wait time, SMF70PDT gives us dispatch time, SMF70EDT gives us effective dispatch time and SMF70ONT gives us online time. These fields, used with SMF70CIN (processor type) can be summed by LPAR for all processor types in each LPAR. This is what RMF does when it builds reports. If you have MICS or MXG, they take this raw data and build it up into accumulated values so you do not have to do the grunt work yourself. This just tells you where the numbers originate.

What We Can Measure TYPE72



- R723MADJ Adjustment factor for CPU rate (service units per second)
- R723MCPU CPU service coefficient * 10,000
- R723MSRB SRB service coefficient * 10,000
- R723NFFI zAAP normalization factor
- R723NFFS zIIP normalization factor



But capacity measurement and reporting is not all about what happens at the processor or even the LPAR level. We also want to know what happens at lower levels. The TYPE72 records offer us a look at service and report class level reporting. Since all supported z/OS levels now require Workload Manager to be in goal mode, we will only discuss the measurements we can see about service and report classes.

The basic unit of reporting in the TYPE72 records is the service unit. Field R723MADJ tells us the number of service units per second so we may convert service units to time. We also need to be aware of the service coefficients R723MCPU, the TCB CPU service coefficient, and R723MSRB, the SRB CPU service coefficient. These are important because the RMF CPU use numbers are weighed by them. If you use a value other than 1 for these service coefficients you will need to take it into account in order to see actual use. This paper will assume that both coefficients have been set to 1 since this is the IBM recommended value. They will not appear in any formulae in this paper.

Fields R723NFFI and R723NFFS are the zAAP and zIIP normalization fields, and should be identical to each other. Just like the newer SMF70NRM field in the TYPE 70 records, you divide this number by 256 to arrive at the actual normalization factors for specialty engines. If R723NFFI ever disagrees with R723NFFS in the same RMF record, call IBM. They will probably want to know.

What We Can Measure TYPE72



- R723CSUC Service Units (SUs) on GPs for zIIP eligible work
- R723CIFC SUs on GPs for zAAP eligible work
- R723CSUP SUs on zIIPs
- R723CIFA SUs on zAAPs



The fields we want to look at for CPU consumption are as follows. R723CSR_B tells us the service units consumed on a GP by SRB processing. R723CCP_U tells us the GP service units consumed by TCB processing. We get into the numbers for specialty engines. Work that is able to run on specialty engines may run there or it may run on GPs. While all GP work is shown in R723CCP_U or R723CSR_B, we want to know how much of that work could run on a zIIP or a zAAP. Field R723CS_UC tells us how much zIIP eligible work ran on GPs and field R723C_IF_C tells us how much zAAP eligible work ran on GPs. Field R723CS_UP tells us how much work actually ran on zIIPs and field R723C_IF_A tells us how much actually ran on zAAPs. Be aware that if you have turned on the zAAP on zIIP feature IBM offers that fields R723C_IF_A and R723C_IF_C will always be zero. All work will be reported as zIIP or zIIP eligible. There is no way to see what might have been able to run on zAAPs when this feature is enabled.

What the information tells us



- TYPE70
 - Actual engine use
 - Nothing about eligible specialty engine use
 - Kneecapping factor
- TYPE72
 - Service class engine use
 - Actual and eligible specialty engine use
 - Kneecapping factor



The TYPE70 records tell us actual engine use. They do not tell us anything about potential specialty engine use that actually ran on GPs, they only tell us what actually ran where. This information by itself may not seem very useful when we are trying to size specialty engine growth but it gives a baseline we can use to see how much potential and actual use we have in the TYPE72 records. If this sounds familiar, that is good. It is capture ratios, extended to specialty engine use. While this paper will not deal specifically with capture ratios, we need to be aware that the information is there and available for use.

The TYPE72 records tell us about service class engine use. This is broken out into GP use, specialty engine use and specialty engine eligible use. It does not matter if the specialty engines are zIIPs or zAAPs, they are handled the same way. Kneecapped processors bring the difference in specialty engine use and specialty engine eligible use into special focus. Remember that kneecapped processors GP engines run at reduced speed but the specialty engines run at full speeds. An approximation of this can be determined by dividing the ratio of SRM seconds to real seconds for various processor families to the same ratio of their full size relatives. Table 1 shows us selected values of this of these ratios. For exact information on your processor, examine the SMF70NRM value from TYPE70 information, or either R723NFFI (zAAPs) or R723NFFS (zIIPs) from your own RMF data and divide the number you see by 256. That will give you the exact ratio for your processor. Remember that full speed processors will have a ratio of 1.

What the information tells us



- Kneecapping factor
 - xxxxxxxx / 256
 - TYPE70
 - SMF70NRM
 - TYPE72
 - R723NFFI (zAAP)
 - R723NFFS (zIIP)
- Service units to seconds
 - R723MADJ



We can use field R723MADJ, the service units per second, to convert the other TYPE72 fields we have discussed, that express CPU use in terms of service units, from service units into time. Products such as MICS or MXG will do this conversion for you. If you need to perform it yourself, just divide a value such as the one for R723CSUC (zIIP eligible work on GPs) by R723MADJ to get time used. All further discussion of CPU use in this paper will talk about time and will assume you have done the conversion, either by hand or with the aid of a product such as MICS or MXG.

Table 1 - Selected Processor Ratios



Processor	Full Size Ratio
2097-4xx	4.21
2097-5xx	1.95
2097-6xx	1.44
2097-7xx	1.00
2817-4xx	4.89
2817-5xx	2.00
2817-6xx	1.53
2817-7xx	1.00

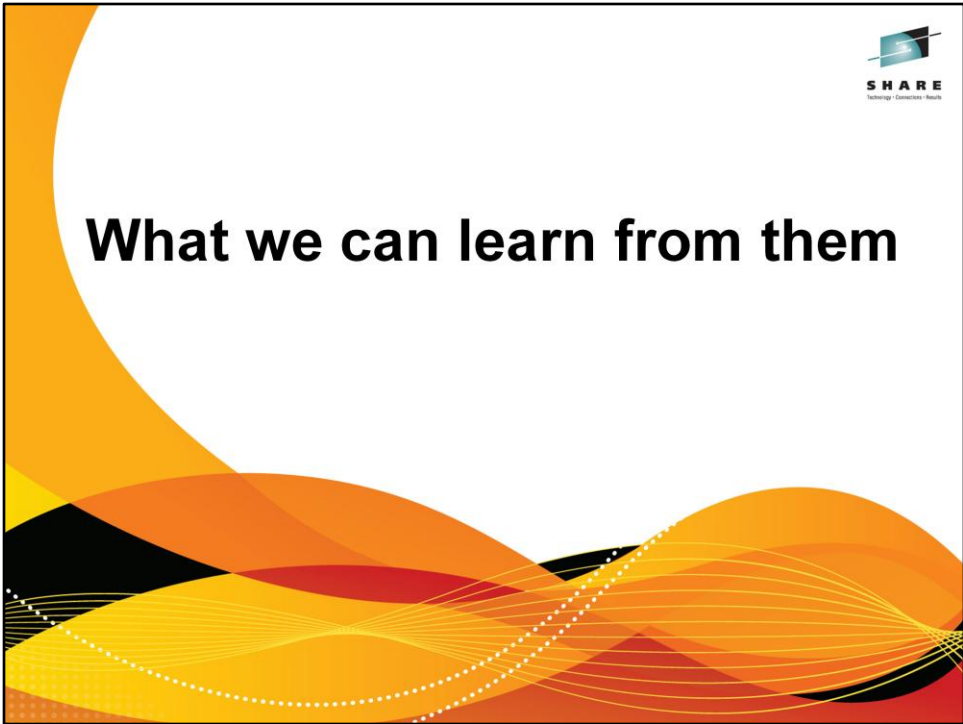


This is a table of selected kneecapped processors and the ratios of GPs to specialty engines. Not all models are included because data was not available to the author to confirm actual ratios but the formula of

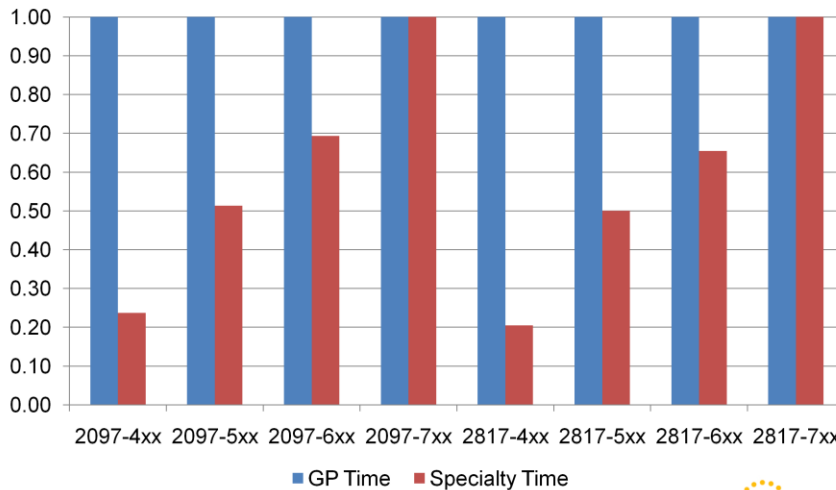
$$\text{RATIO} = \text{SMF70NRM} / 256$$

will hold true for all processors. As you can see from the chart, the ratio is a constant within a processor sub-family group. This is because the SMF70NRM, R723NFFI and R723NFFS values stay the same within the sub-family groups.

What we can learn from them

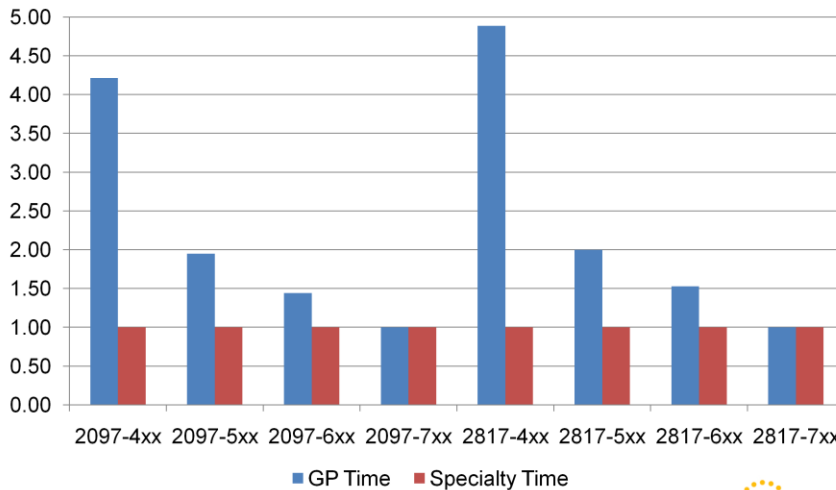


One CPU sec on GPs v. Specialty Engines



If we do not have any specialty engines yet, life is simple. We can see that it is possible for over four engines worth of work on GPs to run on a single specialty engine if we are on the smallest processors in our table. On the larger processors in our tables, it could be around one and a half engines worth of work on GPs able to run on a single specialty engine. This is useful information when we are trying to determine whether to buy a specialty engine. It does not help us with arrival rates or many other details we may want to know but it does help us find a starting point. For detailed analysis we should use IBM's zPCR tool – but this gives us an idea of where to start. Armed with a few simple facts we now know how to get, we can easily get an idea of the potential savings a specialty engine can offer us.

One CPU sec on Specialty Engines v. GPs



As you can see, the specialty engines consume much less actual CPU than the same work run on GPs. But let us look at work on specialty engines versus work on GPs. This figure shows us the same one second of actual CPU time – but this time it is on a specialty engine and we can see how much equivalent time would be required on GPs. We know it is the same data as in the previous picture, but the impact is different when we see the graph.

When we already have specialty engines it gets a little cloudier. The data stays the same but presentation becomes more important in helping us determine what the data mean.

Engines



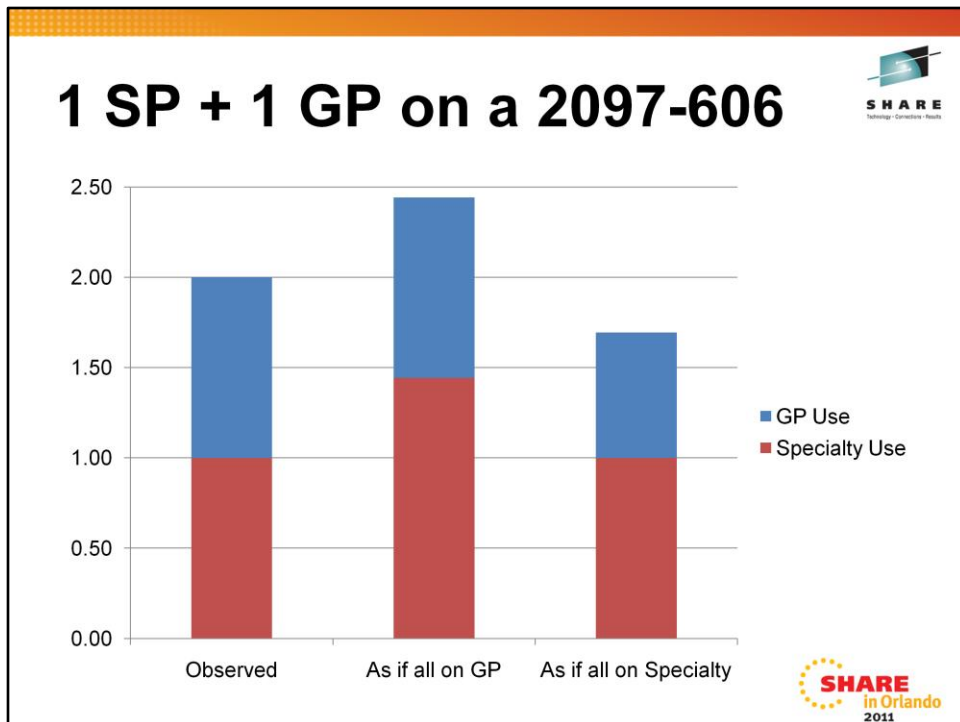
- Management buys engines
- Present data to them in terms of engines
- How to calculate:

$$(CPU\ time) / Duration = Engines$$



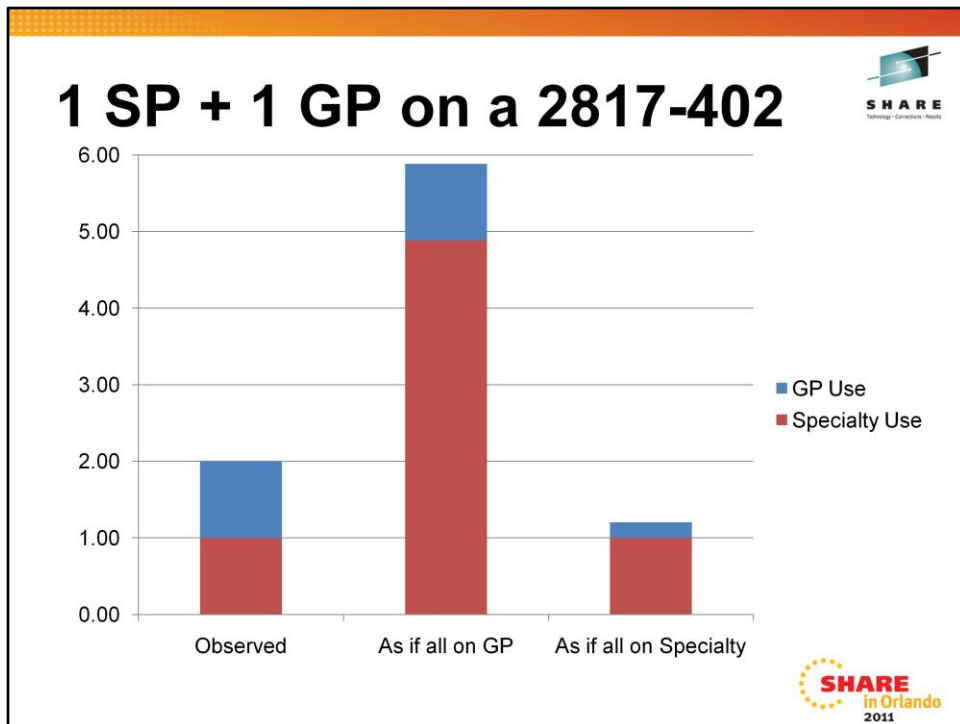
So let us look at a situation where we already have specialty engines running. We are still using units of one, but now we are going to talk about them as engines being used rather than a single CPU second. The main reason for this is simple. We buy CPU hardware by engines. We do not buy an additional five percent over what we currently have; we buy the number of engines that gets us close to the five percent that management has decided we need. So why not present things to management in terms of engines in the first place? Do they care about 90th percentile this and 15 percent that, or do they want numbers that clearly say, “Buy this. It is the amount we need”? So why not give them the information they must have to make the decisions that only they can make, but give it to them in a form that makes sense from a buying perspective? Give them the numbers in engines.

Providing number in engines is easy. In any RMF interval, the number of engines used can be expressed by dividing the amount of CPU time used by the duration of the interval. That is,. It is that simple.



This figure shows us the consumption of one specialty engine and one GP engine in a single RMF interval on a 2097-606, but it shows it to us in three different ways. As we saw in Table 1, a specialty engine is 1.44 times as fast (or “bigger”) than a GP on a 2097-6xx. Our first view shows us the actual CPU use, one engine on specialty processors and one engine on GPs. The second view shows us how this would look if all the time had been on GPs. The amount of time that was actually on a specialty processor would take almost one and one half times as long if it was on a GP while the time on a GP still takes the same time. The net result is that the whole thing would require almost 2.5 processors instead of only two. While this may not seem like a lot, it is enough of a difference that short transactions could appear to take more CPU time than they do wall clock time.

The last part of this figure shows us how many engines would be used if all of their time was on specialty engines. Not surprisingly, the specialty engine amount remains the same, one engine. But the remaining time, the one engine’s worth of GP use, now takes much less than one engine.



While none of these views is radically different from any of the other ones, they do show that there is a difference. This figure takes this to an extreme, by showing us one engine of specialty and one engine of GP as they appear on a 2817-402.

You may think this is blown out of proportion but each view has its own story to tell. The measured view tells what is actually happening. When you are telling management what the hardware is actually doing, this is the one to use. The second view, the one showing the work as if it was all on GPs, can help illustrate the software savings specialty engines make possible. It can help when trying to sell management on buying more specialty engine, by showing them what the current ones save.

Something to be aware of about this view for MXG users is that this is the view you will see if you present the numbers MXG gives you in its TYPE72GO dataset. All specialty engine use numbers have been normalized to the GP processor speed before being saved. If you want to show actual use, you must first unnormalize specialty engine use. This is not mentioned as a knock against MXG, but it is something you should be aware of. When I first presented processor use on a kneecapped processor to management, we were all surprised when the numbers showed more actual zIIPs being used than were present in the hardware. It was easy to correct but it was completely unexpected.

The last view, the one showing all work as it appears on specialty engines, is one that you can use to show management how many specialty engines may be needed to run the work eligible to run on them. While zPCR will give you more accurate numbers, this is an easy way to start the conversation. It can be especially useful when no specialty engines are in the hardware yet, even though the examples shown here show specialty engines already in use.

Final Thoughts



- We need to understand kneecapping and its effect on reporting specialty engine use
- We need to understand our data and what it is telling us
- We need to show what is actually happening
- We need to be able to discuss “What if...?”
 - How many GPs we might save
 - How many specialty engines we might need



We have looked at the type 70 and 72 records and seen the information in them available for showing current and potential use of specialty engines. We have examined the difference between full sized specialty engines and kneecapped GP engines, and shown that reporting their use can be confusing. And we have shown how to clear up the confusion, and make the most of your data – and hopefully your hardware. Specialty engines running on kneecapped processors can offer tremendous performance boost. But to truly understand it, you have to know what you have and how to report it.