

Inspect-CPU

Performance Tuning Brief for IBM® CICS® Applications

Tune or Buy More CPU ?

Suppose someone asks if your CICS® system is tuned? Think about it for a second. It is not about response time. Is there any reason to tune when you have a very good response time where 90% of the transactions complete in less than a second? Whenever your processor limit is reached - Isn't it cheaper to buy more hardware, especially now with the lower hardware prices? This is a debate where more and more installations choose hardware upgrade over tuning – it is easier to buy hardware and tuning is an iterative process that needs a lot of technical knowledge, time, and testing.

CICS System Parameters and...

We are not going to take a side in this debate, but we want to throw some light on it with a small example. Let's take a look at tuning at a CICS level – for example the CICS SIT parameter, ICV. The default is set to 1,000 milliseconds, which means that CICS will check every second if there is work to do. However, CICS will “wake” up anyway when there are tasks to work on. So assuming a non-swappable CICS, setting ICV to 20,000 milliseconds will free CPU cycles and will not hurt your response time. There is no doubt that this kind of tuning is worthwhile. The time required to apply the change is minimal, and it is a one-time effort, which saves you CPU cycles over the years to come.

Application Tuning...

The above was an example of system tuning, but we want to focus on application tuning, which is more difficult to perform, but carries similar characteristics as the example above. Application tuning can have a dramatic effect on the behavior of your CICS systems. As experience has shown, the long-term and short-term gains of well-tuned applications are worthwhile and is cost-justified to spend the effort to achieve this goal. However it is not straightforward and the number of possible interactions between programs – the transaction mix – and the number of changes applied to the applications does not make it easy to “catch” the offenders. So what can you do when you suspect your application program is causing a performance problem, or you want to assure that a badly written program won't go into production?

CICS Statistics

CICS does provide information about the programs and tasks that are running and we can use this information, analyze it and get to the programs which are the problematic ones. We can get CICS hourly statistics and find the programs that run the highest number of times and do a code review of those programs. We would look for CPU intensive commands – in Cobol for example Initialize, String, Unstring – or for coding loops which would retrieve information from a database and provide MORE than a screen full of information to the user. This approach is very slow, very tedious and most of the time we will not know if a certain bad coding sequence is executed many times – thus making it a good target for improvement.



AUXILIARY Trace

CICS provides more information with the Auxiliary trace – we would need to know which trace points to activate first – run the transaction which invokes the program(s) in question – print the Auxiliary trace with a CICS utility and then go over the CICS commands issued in the transaction to find problematic commands – like Getmain/Freemain, numerous Links or XCTL to other programs etc. This approach is worthwhile once we have set our sight on a certain program and we need further investigation of that specific program. But what if we are in the dark completely – it is impossible to run an AUXTRACE on a complete CICS system for practical reasons – never mind the enormous amount of information provided that would have to be analyzed.

Debuggers

There are debuggers available that can be used to step through program logic and find excessive loops – again a good approach once we have a culprit that warrants a closer look – not practical if we want to get an overview over the whole set of programs run and their behavior. If examining a program in production, interactive debugger use is typically not allowed.

Record and Replay

Some installations have gone through the effort (and pain...) to define a typical transaction mix for a given CICS system and for certain time-of-day behavior – they are able to backup/restore the databases/files needed and then run scripts in order to mimic the production environment. This enables a customer to run the script more than once – and use sampling software to find out if a certain application release can provide the required performance. The effort involved in setting up such a "record and play" environment is very big and needs to be adjusted constantly – there are also limits on how accurate a transaction mix can be run to simulate the production environment as closely as possible. Other limitations are that you would have to run this environment solo – run load modules compiled in the same manner like in production – setup all the CICS parameters exactly as in production and and and...

Traditional Sampling Software

Sampling software tools are another approach - can be run in the production environment – only for very, very short periods and at a very high impact on the system resources and the CICS system itself. Because of that, NOBODY runs sampling software at peak hours – because NOBODY can afford the overhead associated with it.

The New Approach to Sampling Software

Imagine if you have sampling software that can run at a FIXED OVERHEAD – in your PRODUCTION environment – which will give you the MOST CPU intensive PROGRAMS and point to the OFFSETS in those program where

the most CPU is spent? It is now possible to do exactly that with ICPU/CICS and receive the BEST information possible – because you would:

- get the real life transaction mix
- get it during your most busiest times during the day
- get the PROGRAM that uses the most CPU – never mind which transaction uses it
- pay at most 5 % overhead for the information
- get the offsets in the programs where most CPU is spent
- know what CICS functions were invoked by those programs
- get the information online – immediately – or batch for later analysis (for off-time sampling).

Inspect-CPU (ICPU) for IBM® CICS®

Is a new software tool designed to give you accurate information about CPU usage in a Production environment at an acceptable overhead in an easy to understand format.

***For more details watch the demo video at
www.esaigroup.com/products/Inspect-CPU/DemoICPU.html***

**Call us for more info and a FREE 14 day trial.
1-866-GO-4-ESAI (1-866-464-3724)
info@ESAIGroup.com**



All trademarks are of their respective trademark holders including z/OS, CICS, DB2 which are registered trademarks of the IBM Corporation.