Virtual CICS user group: Newsletter 32

Welcome to the Virtual CICS user group newsletter. The Virtual CICS user group at www.fundi.com/virtualcics is an independently-operated vendor-neutral site run by and for the CICS user community.

Virtual CICS user group presentation

The latest webinar from the Virtual CICS user group was entitled, “Sampling products for CICS”, and was presented by Dan Eilam, CTO of Inspect-CPU Systems Inc, who also works with Enterprise Systems Associates Inc.

Inspect-CPU Systems developed the ICPU sampling product for CICS. Dan has more than 40 years’ experience as a CICS system programmer and specializes in performance and tuning. Dan has developed several CICS system tools, among them CICS/Replay.

Dan Eilam started his presentation by giving his definition of tuning, which he said is changing systems in order to use fewer resources.

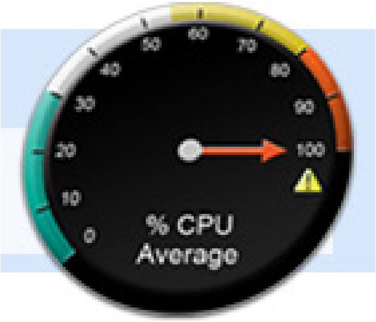

Figure 1: Measure your programs’ CPU usage

He went on to say that the most common reason for tuning is following a response time problem, but there are other reasons, eg a hardware upgrade. Dan suggested that, often, tuning is expensive, easy to implement, but there’s no guarantee that the problem will be solved.

Tuning usually relies on expensive, hard to get, experts who don’t have too much time to spare. And they’re usually called on to do it when there is a problem. Dan asked: “Why not tune regularly to avoid production problems”.

Contents:

Dan wondered what we are tuning: our applications or the system infrastructure on which our applications run? Tuning a system is changing parameters and external resources, such as increasing the number of I/O buffers to reduce I/O, or increasing the CICS ICV SIT parameter to save CPU cycles. In this presentation, he wanted to discuss the tuning of applications that were developed in-house. He suggested that we can modify them in any way to improve performance.

The way to monitor them is usually with CICS monitors such as: TMON from ASG; Omegamon from IBM; Sysview from CA; or MAINVIEW from BMC. These are very powerful, real-time, proactive tools that identify response time problems, whether it's a transaction that is waiting on some resource or looping. If a transaction is waiting, the monitor will identify the reason for the wait and give you all the information you need to resolve the problem.

If the transaction is looping, it may slow down or even stop all user activity. If the transaction is not releasing control of the CPU after running for longer than the ICVR value, CICS will cancel it. If the transaction is releasing control of the CPU, CICS cannot detect the loop. Noticing and taking care of it can take quite a while, causing a considerable amount of damage.

What do you do if you have this kind of problem? The CICS monitors are used to set resource limits and cancel runaway transactions, or any transaction that uses "too much" CPU. If the transaction is cancelled, it is quite difficult to solve the problem.

The only resource that commonly-used CICS monitors don't provide enough information about is the most expensive and critical resource – the CPU.

Will these monitors help you solve a hogging transaction issue? To solve that kind of problem and find the real CPU culprit, you need a sampling tool. Will these monitors help you analyze your programs and find opportunities to reduce CPU MIPS? Again, to get this kind of information, you need a sampling tool.

If I observe CICS every 10 milliseconds and find out that program X is using the CPU 7 times out of 10 observations, this means that program X is using 70% of the CPU. Sampling products pinpoint programs, subprograms, CSECTs, paragraphs, and individual lines of code where the most CPU is used.

Examples of sampling products include: APA from IBM; FreezeFrame from Macro-4; ASG-TriTune; Strobe from Compuware; CA mainframe application tuner from CA Technologies; StarTool from Serena; and ICPU from Inspect-CPU Systems Inc.

Dan Eilam asked whether sampling tools are used to solve the hogging transaction problem or to improve the efficiency of the application programs?

To answer this question, he ran a survey to find out what CICS system programmers typically do when there is a performance issue in production and how they use the sampling tool. First, they look up the problem in the CICS monitor. If this is a wait issue, the monitor clearly identifies the reason for it, so they have the information needed to resolve the problem. However, when it's a CPU issue, it is a different ballgame. The system programmer: checks for any sampling results; terminates the sampling if it's still on; runs batch reports and checks them; identifies the script; finds out the sequence of transactions and the input data so he can rerun it; reruns on TEST;

turns on sampling and runs the script; because test and production are completely different, this seldom works; reruns on Production; gets management approval and end-user cooperation, and schedules a rerun at an appropriate time.

He discovered in his survey that about 4 out of 5 cases remain unsolved. However, when the problem shows up at the rerun, the information from the sampling tool is invaluable.

It's not all good news, sampling product have their deficiencies:

1. Most sampling tools are on-request, batch-oriented, post-mortem tools that seldom help solve online, real-time issues.

2. Sampling tools are system programmer tools. They require technical skills and capabilities possessed only by system programmers, who are therefore the only ones allowed to use them.

3. Programmers are involved only at the end of the tuning process, rather than at the beginning of development.

4. Sampling tools are designed for short sampling periods. With today's Internet, mobile and 24x7 systems, you need a flexible tool that adjusts to any circumstances.

The conclusions from the survey were:

- Sampling tools for CICS are used to solve production problems, but with limited success.
- Sampling tools are rarely used in development and to tune applications when there are no response time problems.

Dan's suggested sampling product criteria were:

- Real-time.
- Online systems require real-time, proactive, rather than post-mortem, batch-oriented tools to solve response time problems.
- Problems should be discovered, recorded, and reported as soon as they occur, so that you can take corrective action immediately and don't have to recreate the problem.

They should also integrate with the development life-cycle. That way, a programmer has debugging tools to ensure the functionality of his programs. He should also use a tool to ensure the efficiency of these programs. Catching inefficient code before it goes into production saves a great deal of money, time and effort and reduces production outages.

A sampling tool should be able to be used when needed, without worrying about its impact and having to go through bureaucratic hurdles. It should be easy to use, easy to understand, and produce meaningful results for everyone at any level. And it should allow users to evaluate the effect of modifications and trends in their applications over time.

It should be available to any number of users, at any time, for any length of time and in any environment, with no impact on ongoing work, whether development or production. Reports should be produced at any time, both when sampling is still ongoing and afterwards. The tool should be able to provide short-term and long-term results. The long-term results may uncover hidden CPU saving opportunities that cannot be discovered by short-term sampling.

It should provide the ability to schedule sampling sessions that start automatically at any predefined time and run for any predefined length of time.

The cost of the sampling

product includes the price of the product itself. It also includes the cost of the resources and overhead used to collect the information. In addition, it includes the cost of experts needed to use the product, maintain it, and analyze the results. Sometimes you get the product for “free” as part of a package deal. But this discourages you from looking at other sampling tools that can do the work.

Dan Eilam gave a couple of examples of using a sampling product.

In conclusion, he said that sampling products can help you ensure the efficiency of application programs. To choose the right tool – one that will be used – make sure its features satisfy your requirements. The right tool will help you streamline production activity, save CPU cycles, improve response time, and postpone upgrades.

A copy of Dan Eilam’s presentation is available for download from the Virtual CICS user group Web site at www.fundi.com/virtualcics/presentations/CICSsamplingMay16.pdf.

You can see and hear the whole user group meeting by downloading the WMV file from www.fundi.com/virtualcics/presentations/2016-05-03meeting.wmv.

Meeting dates

The following meeting dates have been arranged for the Virtual CICS user group:

- On 12 July 2016, we have Mark Cocker, Senior Software Engineer, CICS Development, IBM talking about “Meeting the challenges of agile development with DevOps, CICS, and UrbanCode Deploy”.

  Mark Cocker reviews how CICS teams are meeting the challenges of agile development and automated deployments using the combination of DevOps principles, the new build toolkit and DFHDPLOY utility, and UrbanCode Deploy. To illustrate these in action, he will show you how a change to a CICS bundle can be built and deployed to CICS regions in test then production environments.
- The following meeting is on 13 September 2016, when Y-Innovate’s John J Sissing will discuss, “Accessing your mainframe data directly from your browser or mobile device”.

We will be using Citrix GoToMeeting for the user group meetings.

Recent CICS articles

Developing Liberty applications for CICS: Part 3 – Deploying your application by Michael Jones on developerWorks (31 May 2016): You can find the article at https://developer.ibm.com/cics/2016/05/31/developing-liberty-applications-for-cics-part-3-deploying-your-application/

Communicating with CICS TS for z/OS by Mike Brooks and Andy Wright in *Enterprise Tech Journal (issue 2 2016).* You can find the article at: http://ourdigitalmags.com/publication/?i=301400#{%22issue_id%22:301400,%22numpages%22:1,%22page%22:10}

Using CPSM RTA to monitor critical tasks in CICS by Greg Todaro on developerWorks (13 April 2016). You can find the article at https://developer.ibm.com/cics/2016/04/13/using-cpsm-rta-to-monitor-critical-tasks-in-cics/

CICS news

Enterprise Systems Associates Inc (ESAi) has announced Version 4.2 of Inspect CPU for CICS. ICPU is a low-overhead CICS Application Performance Management product that identifies applications programs and the lines of code causing issues or spikes in CPU usage. The new release provides TCB switching information, which, in addition to the current CPU usage information, provides a complete performance picture of the application programs. ICPU now identifies the programs that are the best candidates for threadsafe conversion opportunities. Full details can be found at www.esaigroup.com/download01/ICPU_Release420_Announcement.pdf.

Like us on Facebook
facebook.com/VirtualCICS

On Twitter use the hashtag #virtualcics

About the Virtual CICS user group

The Virtual CICS user group was established as a way for individuals using IBM's CICS TS systems to exchange information, learn new techniques, and advance their skills with the product.

The Web site at www.fundi.com/virtualcics provides a central point for coordinating periodic meetings (which contain technically-oriented topics presented in a webinar format), and provides articles, discussions, links, and other resources of interest to IBM CICS practitioners. Anyone with an interest in CICS is welcome to join the Virtual CICS user group and share in the knowledge exchange.

To share ideas, and for further information, contact trevor@itech-ed.com.

The Virtual CICS user group is free to its members.