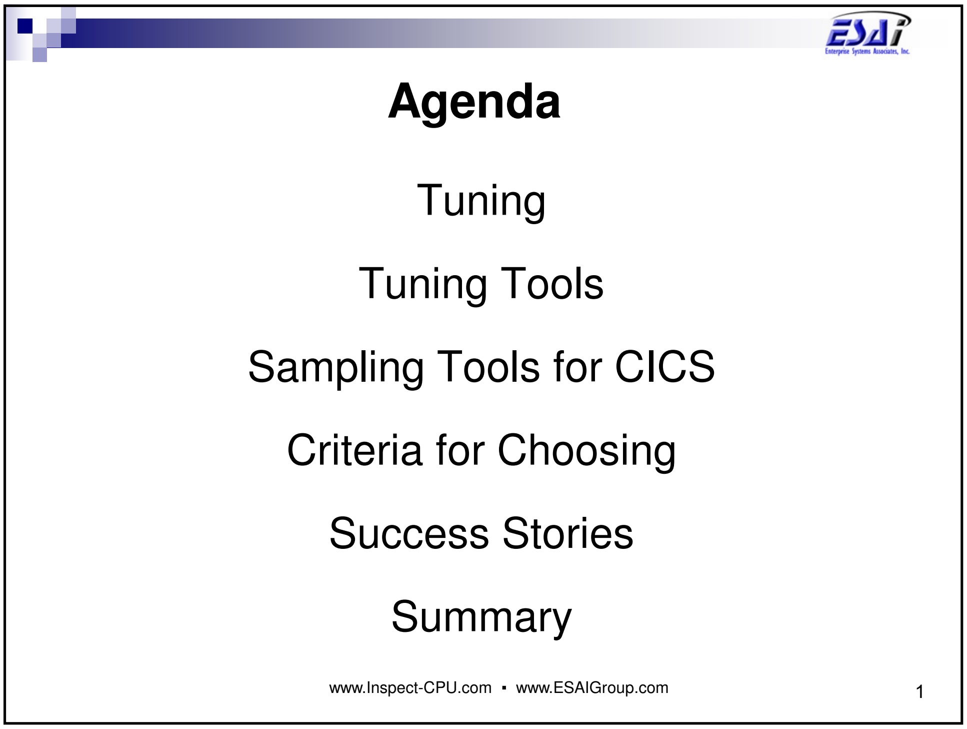


Agenda

Tuning

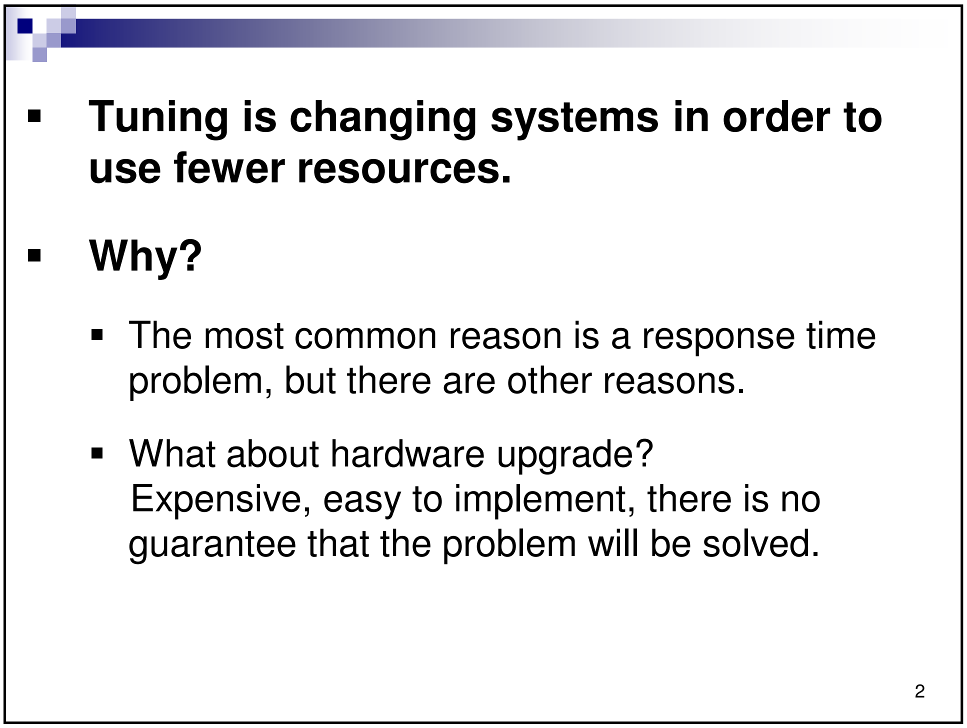
Tuning Tools

Sampling Tools for CICS

Criteria for Choosing

Success Stories

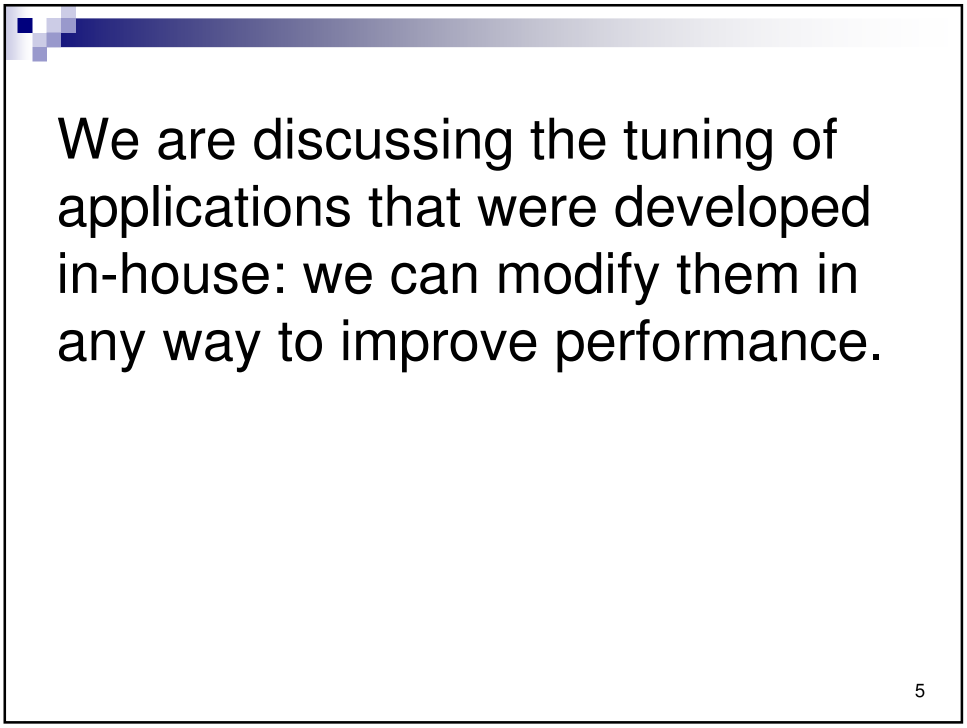
Summary



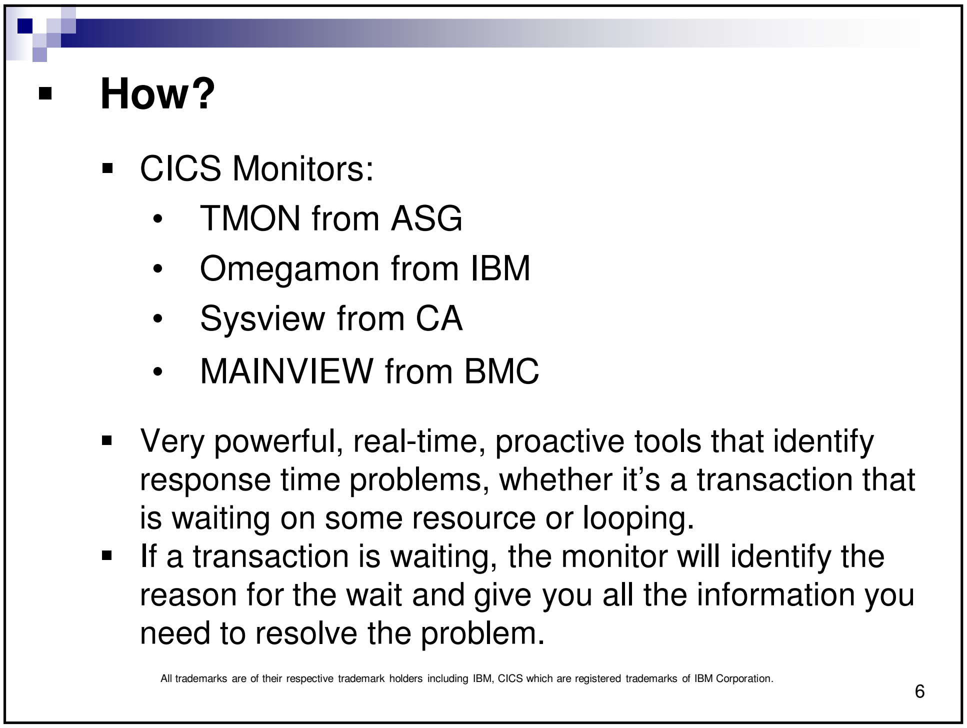
- **Tuning is changing systems in order to use fewer resources.**
- **Why?**
  - The most common reason is a response time problem, but there are other reasons.
  - What about hardware upgrade? Expensive, easy to implement, there is no guarantee that the problem will be solved.

- **Who?**
  - Tuning usually relies on expensive, hard to get experts who don't have too much time to spare. But what about the programmer?
- **When?**
  - When there is a problem. Why not tune regularly to avoid production problems?

- **What?**
  - What are we 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.

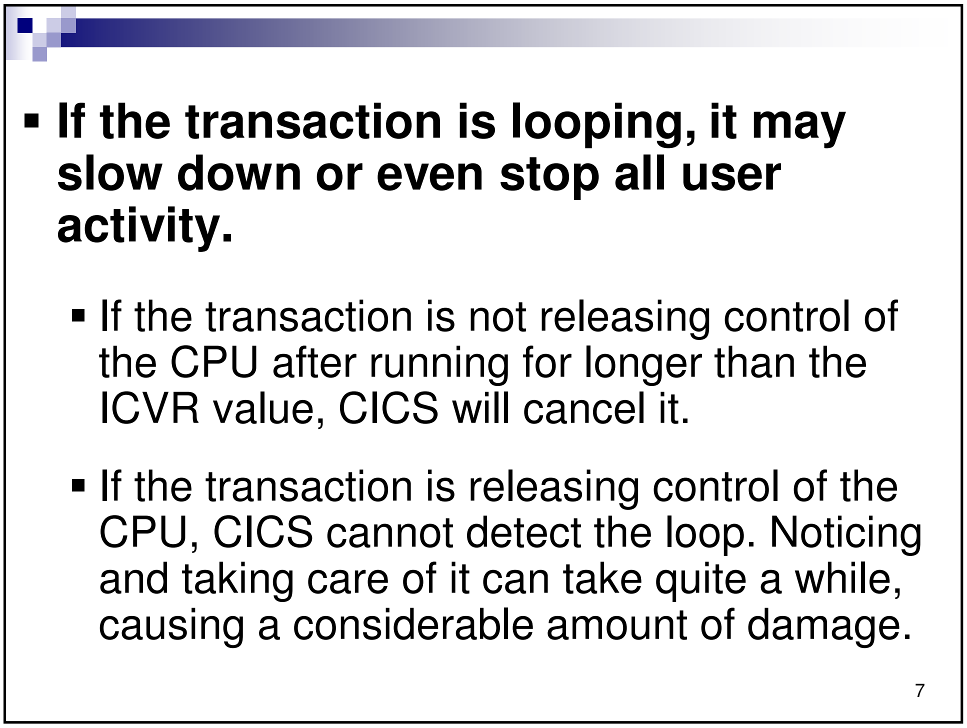


We are discussing the tuning of applications that were developed in-house: we can modify them in any way to improve performance.

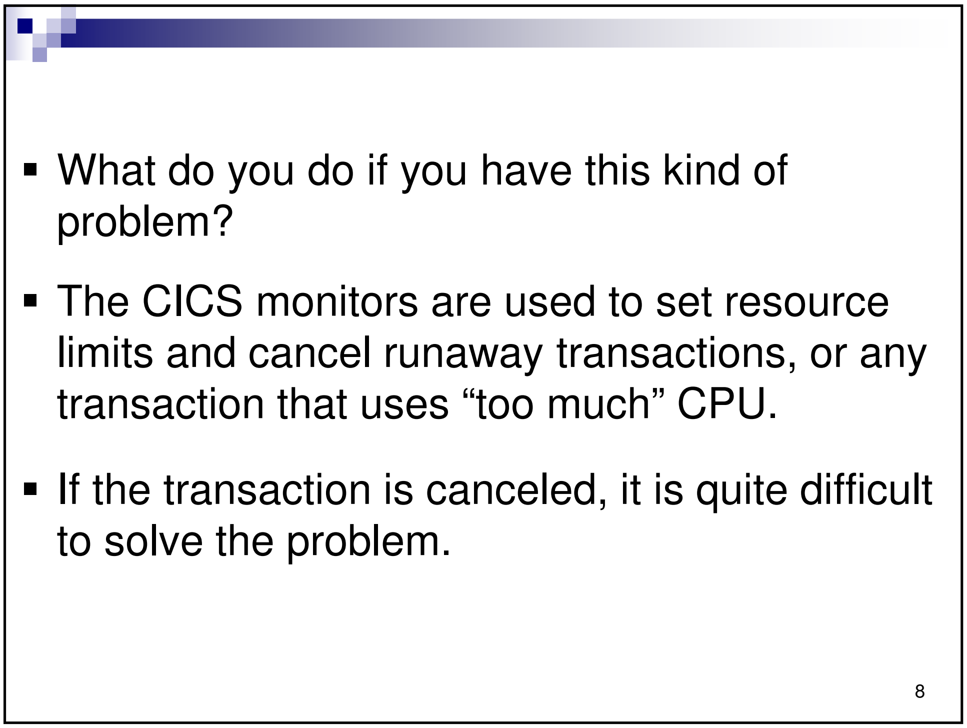


- **How?**
  - CICS Monitors:
    - TMON from ASG
    - Omegamon from IBM
    - Sysview from CA
    - MAINVIEW from BMC
  - Very powerful, real-time, proactive tools that identify response time problems, whether it’s a transaction that is waiting on some resource or looping.
  - If a transaction is waiting, the monitor will identify the reason for the wait and give you all the information you need to resolve the problem.

All trademarks are of their respective trademark holders including IBM, CICS which are registered trademarks of IBM Corporation.



- **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 canceled, it is quite difficult to solve the problem.

The only resource which commonly used CICS monitors don't provide enough information about is the most expensive and critical resource:

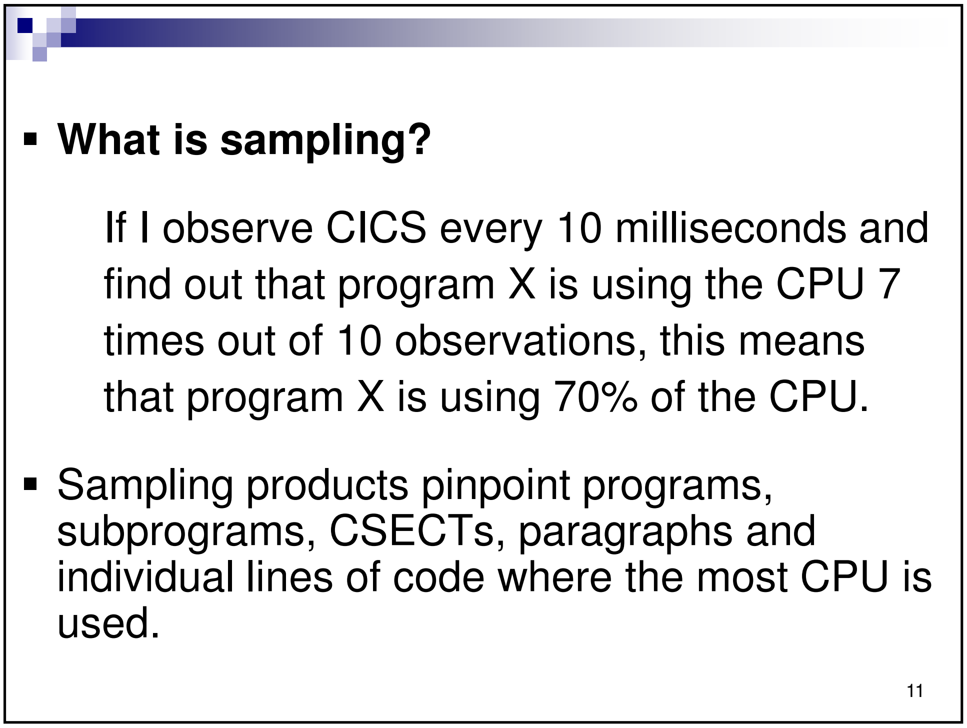
the CPU.

- **Will these monitors help you solve a hogging transaction issue?**

  To solve this 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?**

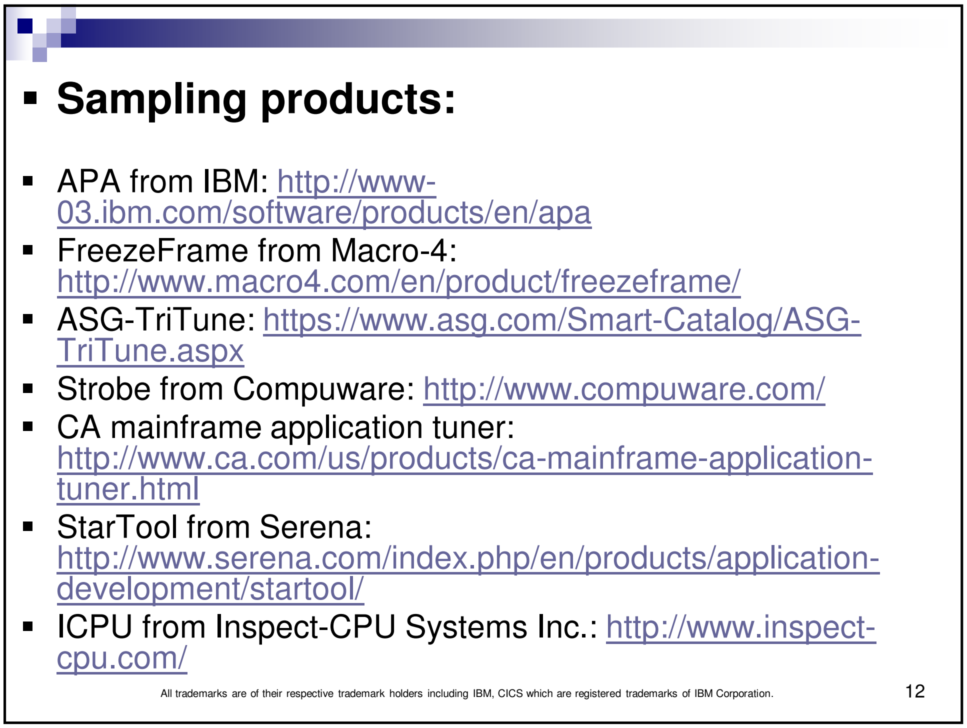
  To get this kind of information, you need a sampling tool.



- **What is sampling?**

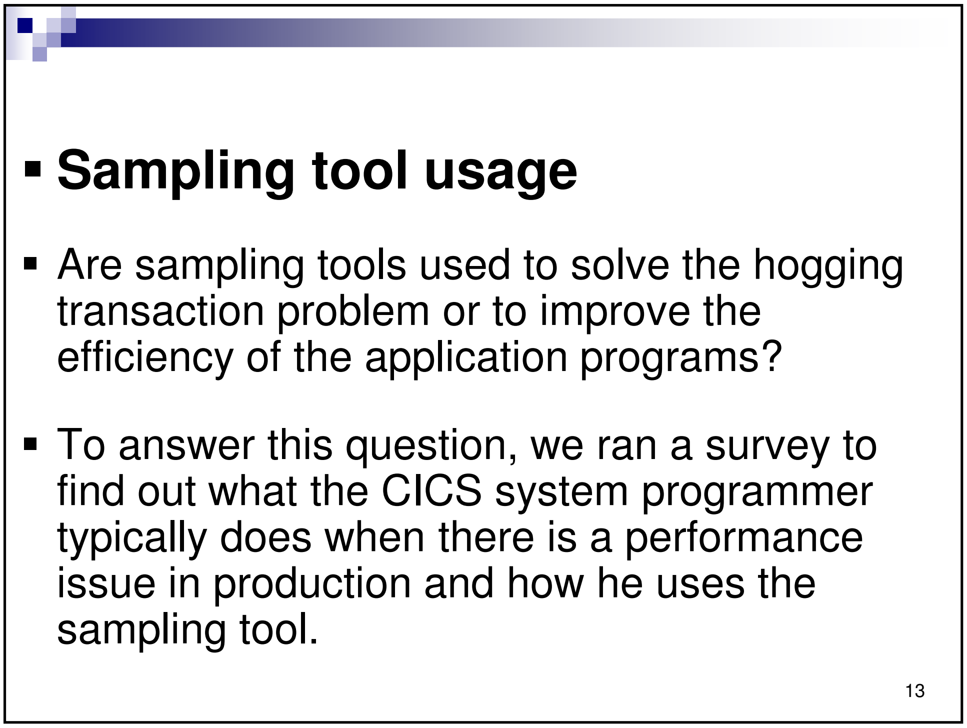
  If I observe CICS every 10 milliseconds and find out that program X is using the CPU 7 times out of 10 observations, this means that program X is using 70% of the CPU.

- Sampling products pinpoint programs, subprograms, CSECTs, paragraphs and individual lines of code where the most CPU is used.



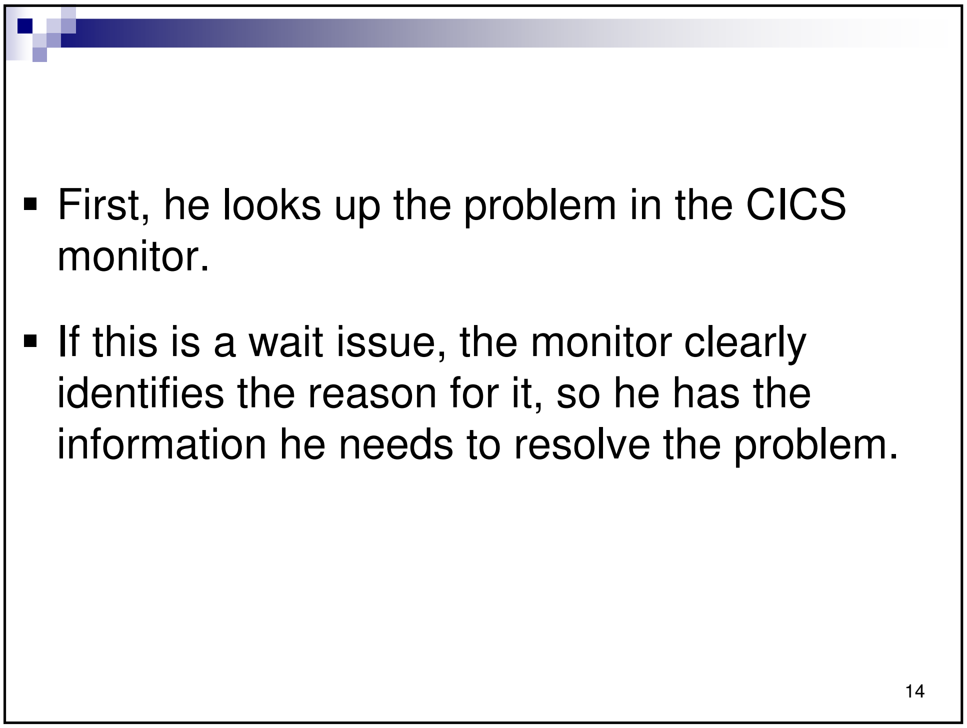
- **Sampling products:**

- APA from IBM: http://www-03.ibm.com/software/products/en/apa
- FreezeFrame from Macro-4: http://www.macro4.com/en/product/freezeframe/
- ASG-TriTune: https://www.asg.com/Smart-Catalog/ASG-TriTune.aspx
- Strobe from Compuware: http://www.compuware.com/
- CA mainframe application tuner: http://www.ca.com/us/products/ca-mainframe-application-tuner.html
- StarTool from Serena: http://www.serena.com/index.php/en/products/application-development/startool/
- ICPU from Inspect-CPU Systems Inc.: http://www.inspect-cpu.com/

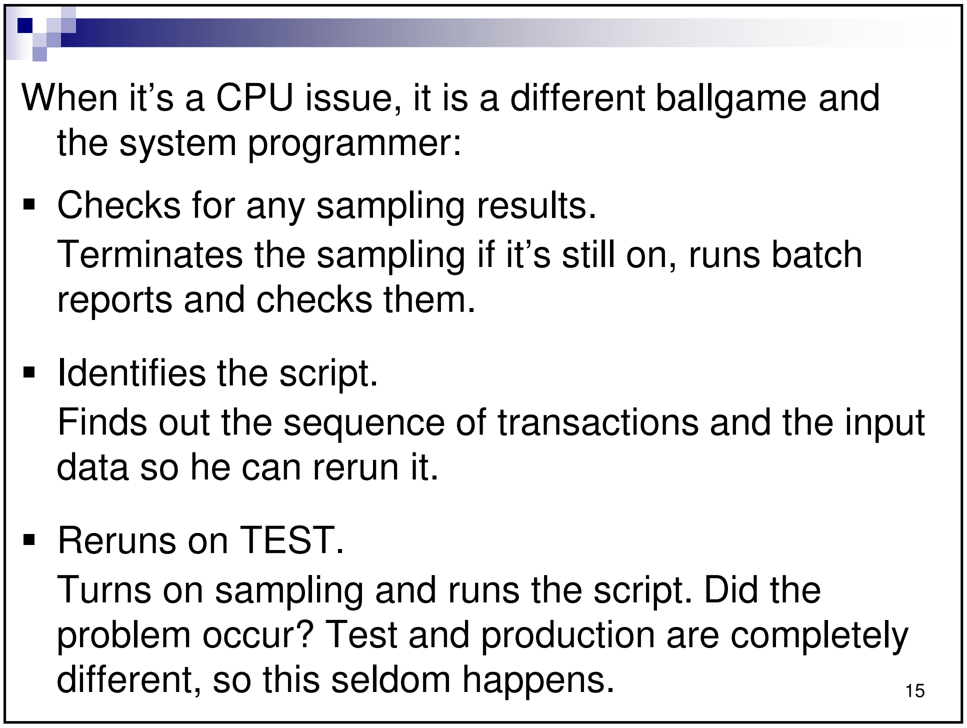


Sampling tool usage

- Are sampling tools used to solve the hogging transaction problem or to improve the efficiency of the application programs?
- To answer this question, we ran a survey to find out what the CICS system programmer typically does when there is a performance issue in production and how he uses the sampling tool.



- First, he looks up the problem in the CICS monitor.
- If this is a wait issue, the monitor clearly identifies the reason for it, so he has the information he needs to resolve the problem.



When it’s a CPU issue, it is a different ballgame and the system programmer:

- Checks for any sampling results.

  Terminates the sampling if it’s still on, runs batch reports and checks them.

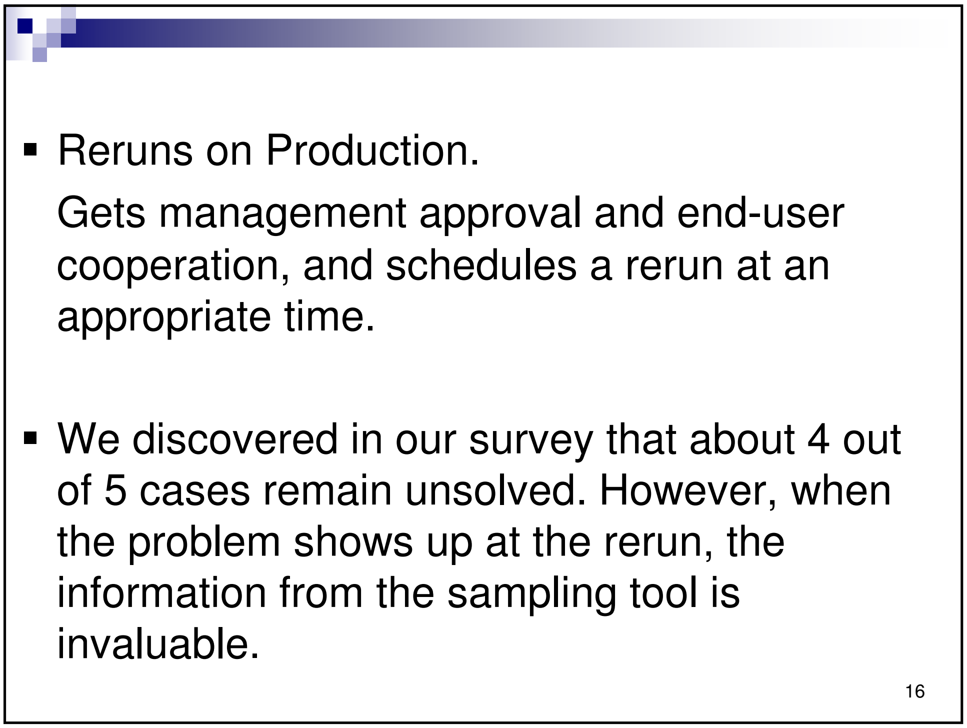
- Identifies the script.

  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. Did the problem occur? Test and production are completely different, so this seldom happens.

15

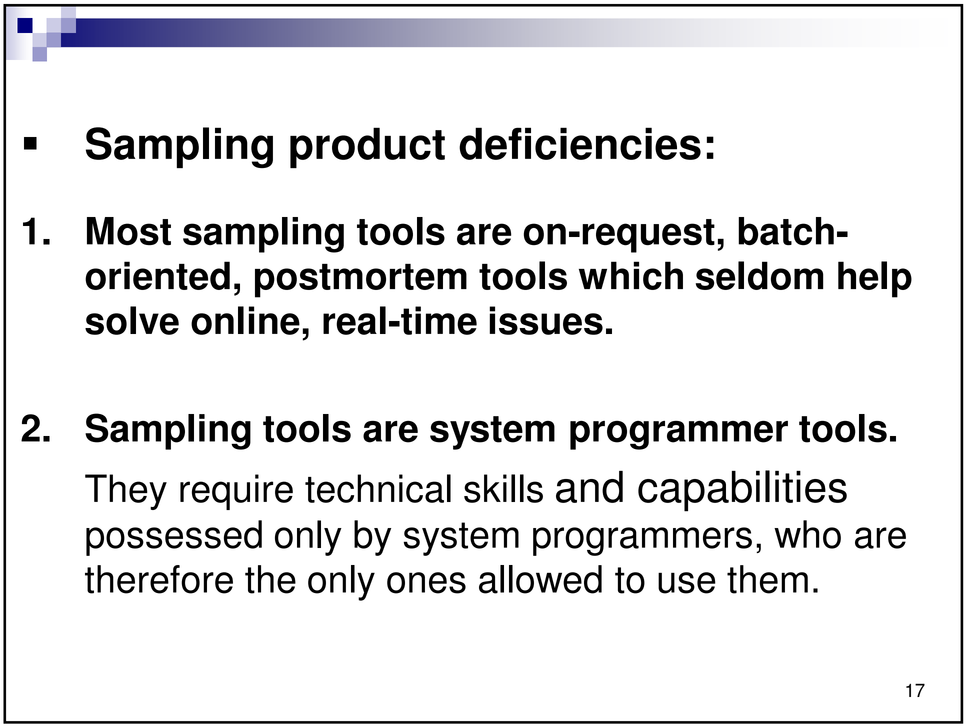


- Reruns on Production.

  Gets management approval and end-user cooperation, and schedules a rerun at an appropriate time.

- We discovered in our 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.

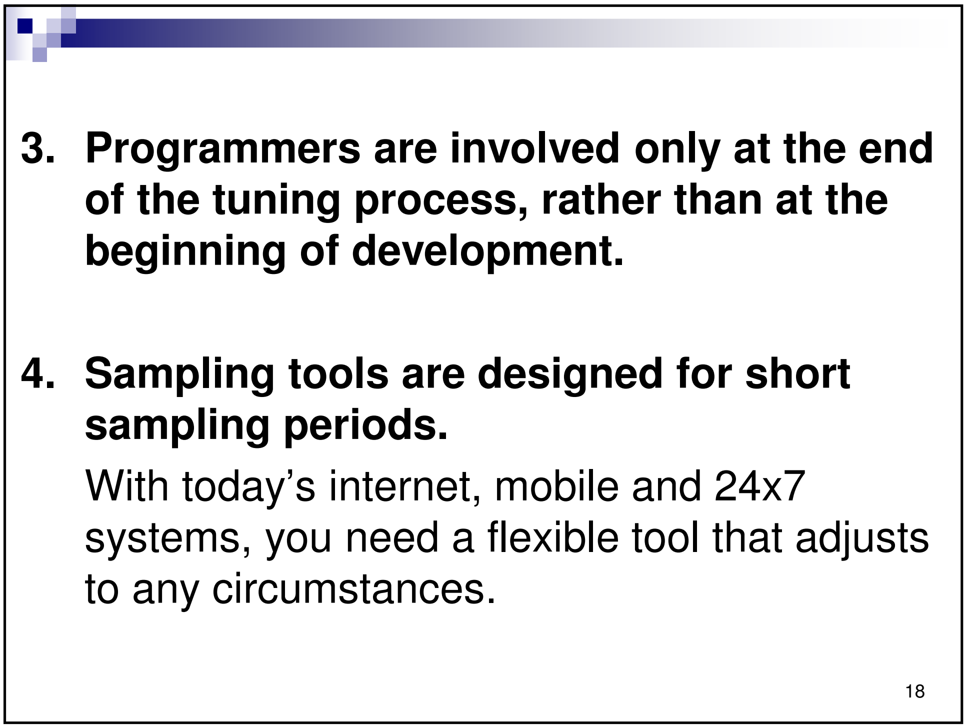
16



- **Sampling product deficiencies:**

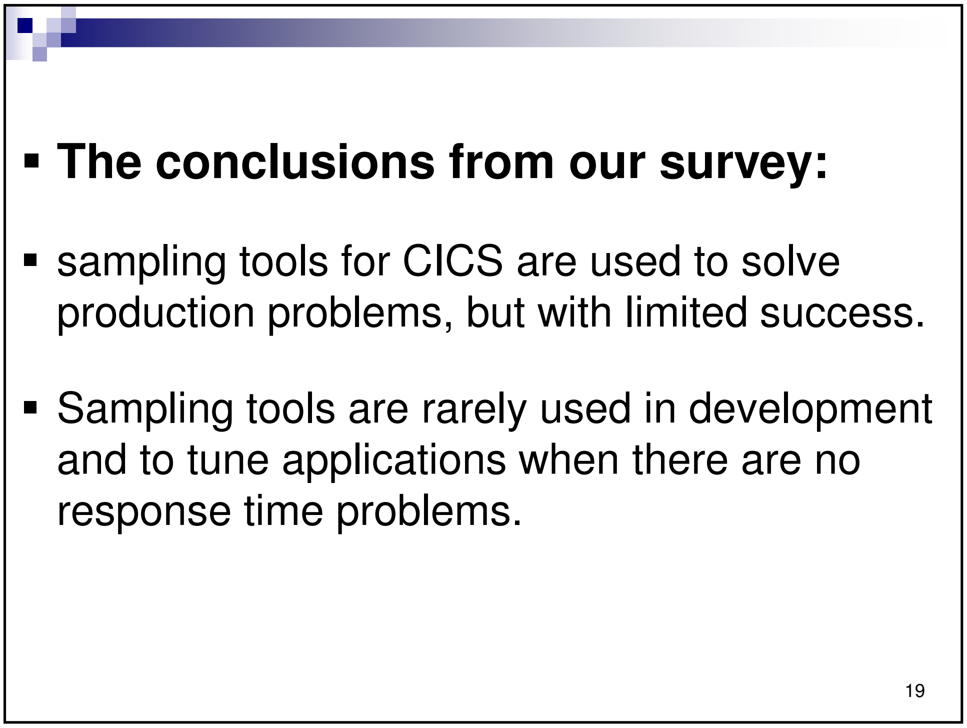
1. **Most sampling tools are on-request, batch-oriented, postmortem tools which seldom help solve online, real-time issues.**
2. **Sampling tools are system programmer tools.**

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

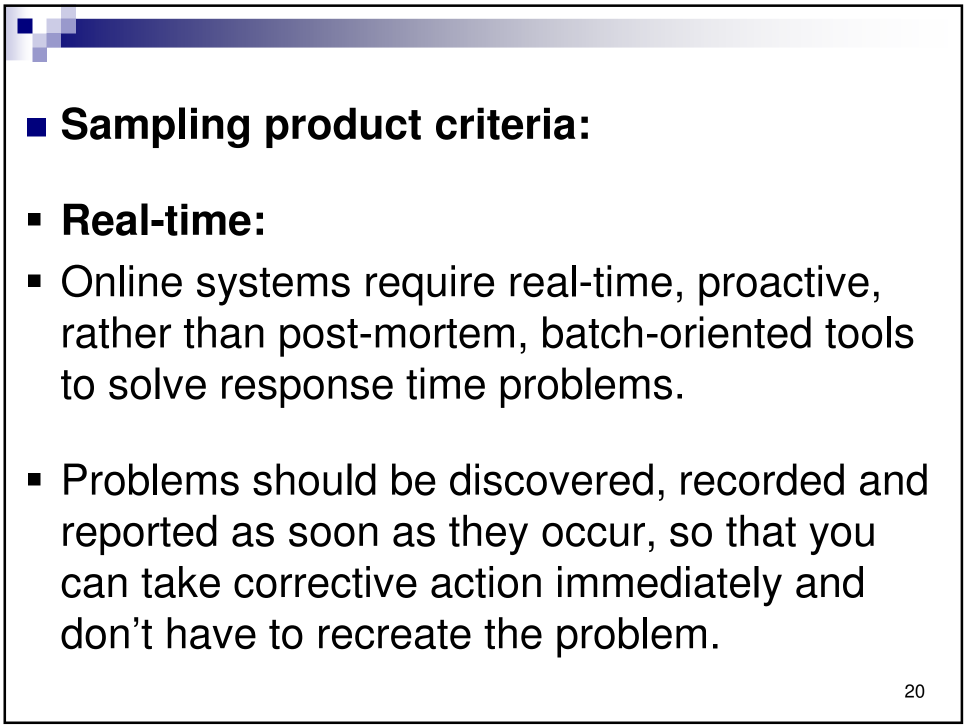


3. **Programmers are involved only at the end of the tuning process, rather than at the beginning of development.**
4. **Sampling tools are designed for short sampling periods.**

   With today’s internet, mobile and 24x7 systems, you need a flexible tool that adjusts to any circumstances.



- **The conclusions from our survey:**
- 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.



- **Sampling product criteria:**
- **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.

- **Integrate with development life cycle:**
  - 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.

- **Unlimited**:

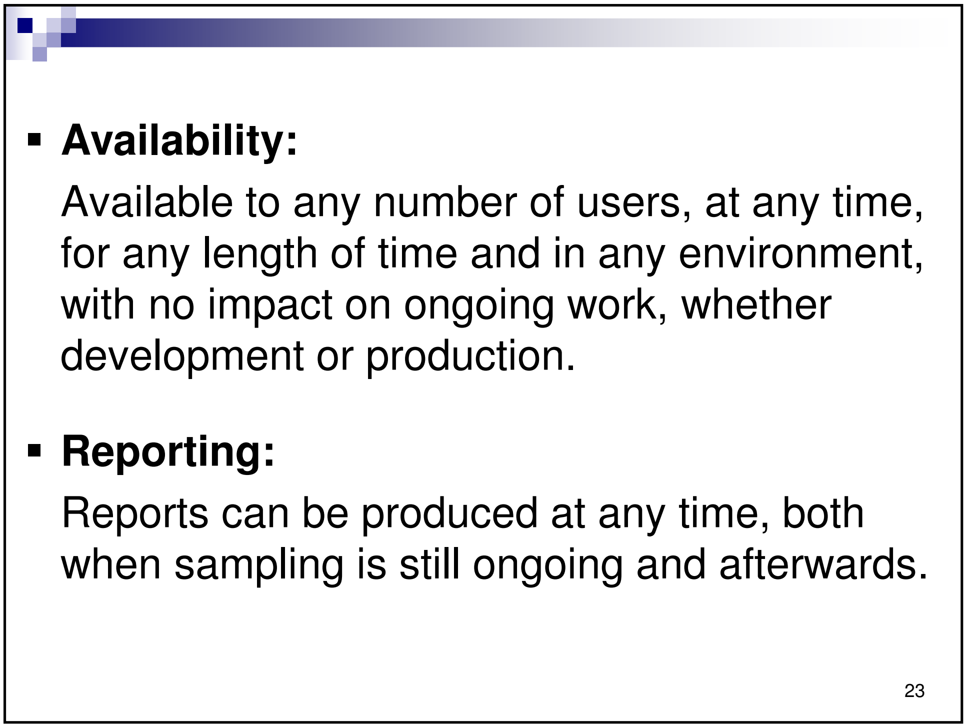
  A sampling tool that can be used when needed, without worrying about its impact and having to go through bureaucratic hurdles.

- **User-friendly:**

  Easy to use, easy to understand and meaningful results for everyone at any level.

- **History repository:**

  Allows you to evaluate the affect of modifications and trends in your applications over time.

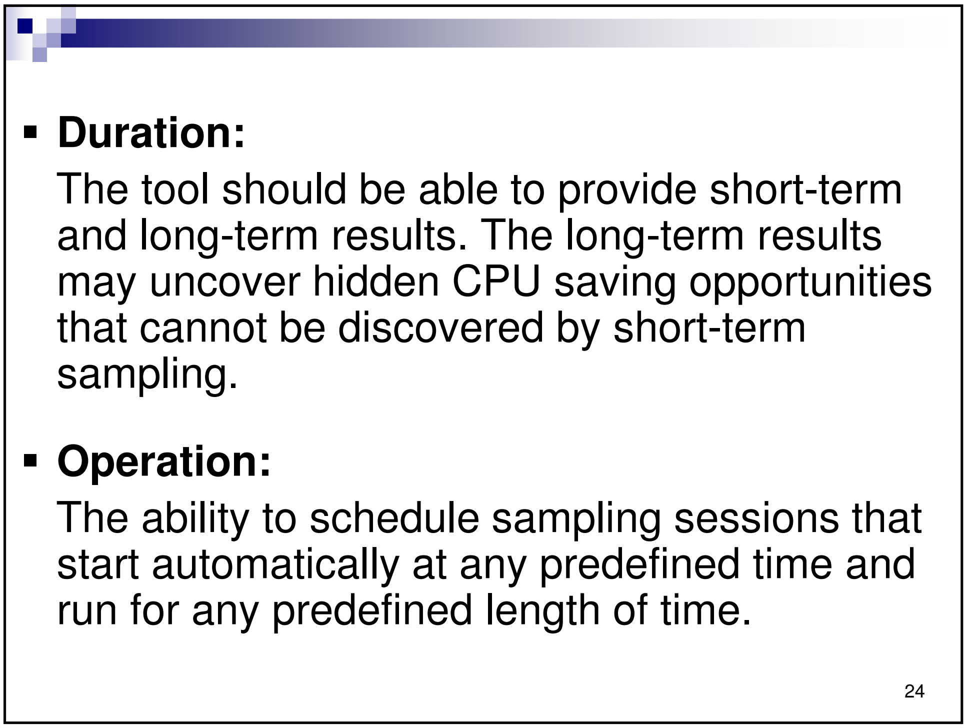


- **Availability:**

  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.

- **Reporting:**

  Reports can be produced at any time, both when sampling is still ongoing and afterwards.



- **Duration:**

  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.

- **Operation:**

  The ability to schedule sampling sessions that start automatically at any predefined time and run for any predefined length of time.

- **Cost Considerations**

- 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.

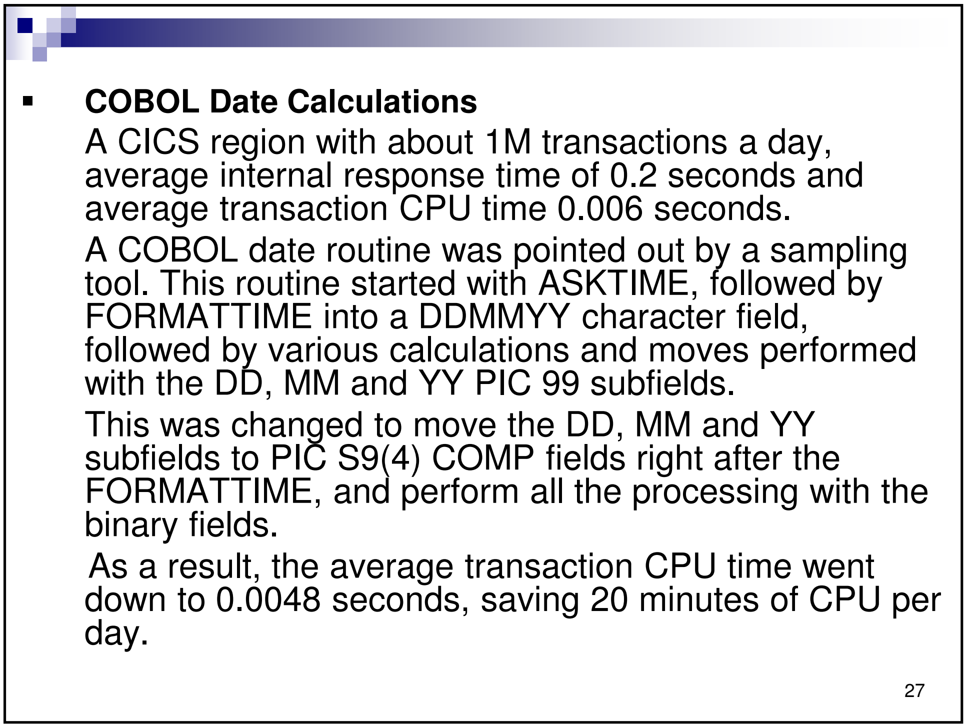
- **Two success stories:**

- **MQ/Syncpoint**

  A triggered MQ transaction was reading all the messages on a queue, and was invoked again when a new message arrived.

  The program executed a Syncpoint after every message and the sampling tool showed that the Syncpoints take most of the CPU.

  The program was changed to do a Syncpoint only after a set number of messages, specified by a parameter, and the CPU consumption went down dramatically.



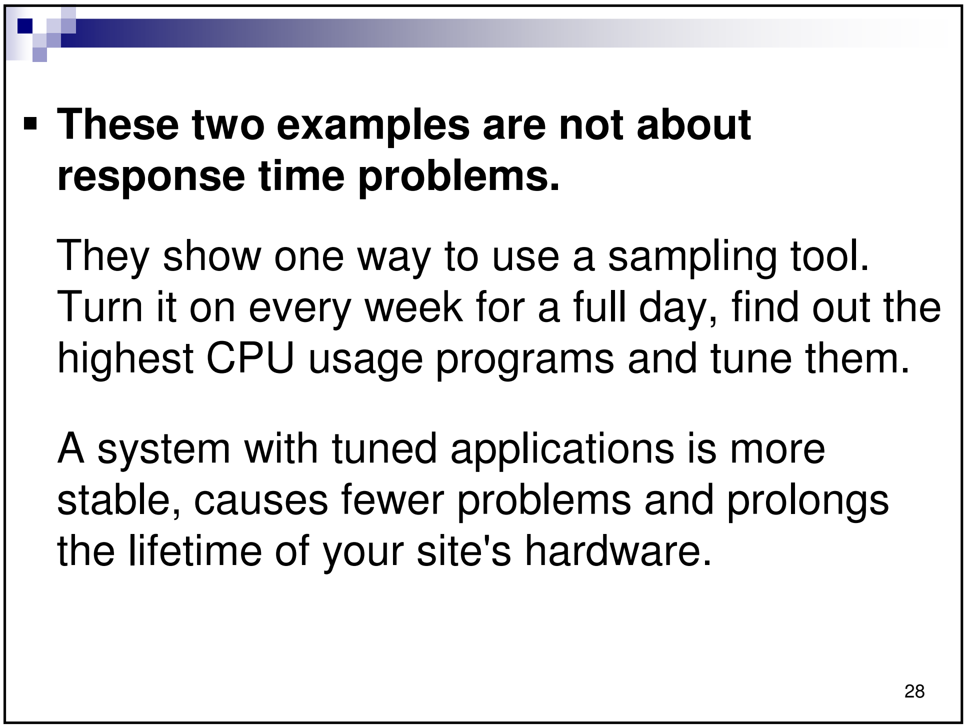
- **COBOL Date Calculations**

  A CICS region with about 1M transactions a day, average internal response time of 0.2 seconds and average transaction CPU time 0.006 seconds.

  A COBOL date routine was pointed out by a sampling tool. This routine started with ASKTIME, followed by FORMATTIME into a DDMMYY character field, followed by various calculations and moves performed with the DD, MM and YY PIC 99 subfields.

  This was changed to move the DD, MM and YY subfields to PIC S9(4) COMP fields right after the FORMATTIME, and perform all the processing with the binary fields.

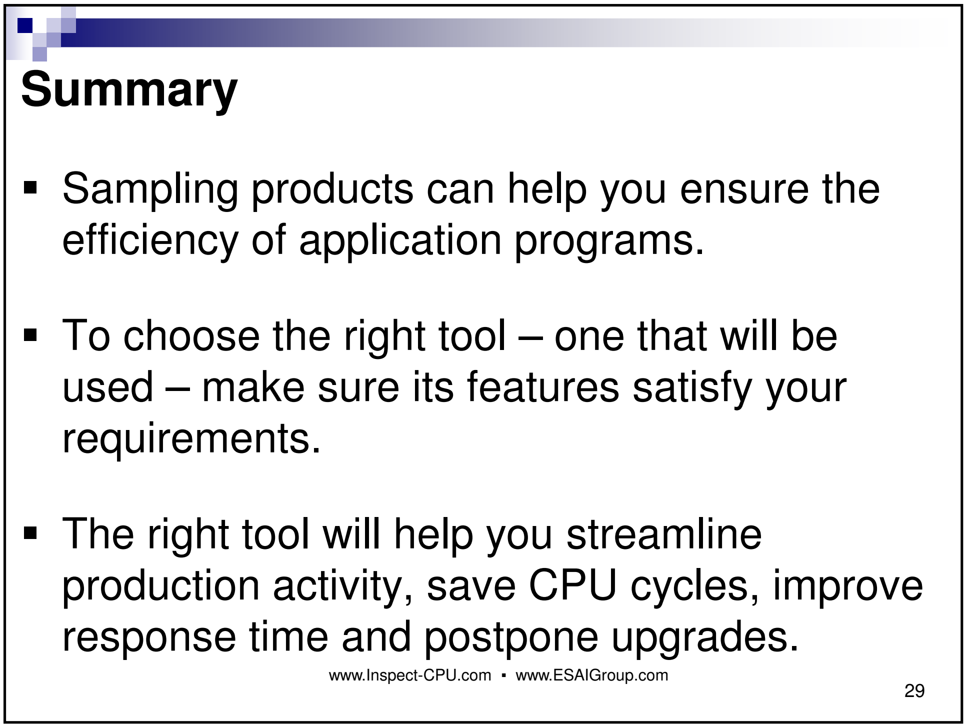
  As a result, the average transaction CPU time went down to 0.0048 seconds, saving 20 minutes of CPU per day.



- **These two examples are not about response time problems.**

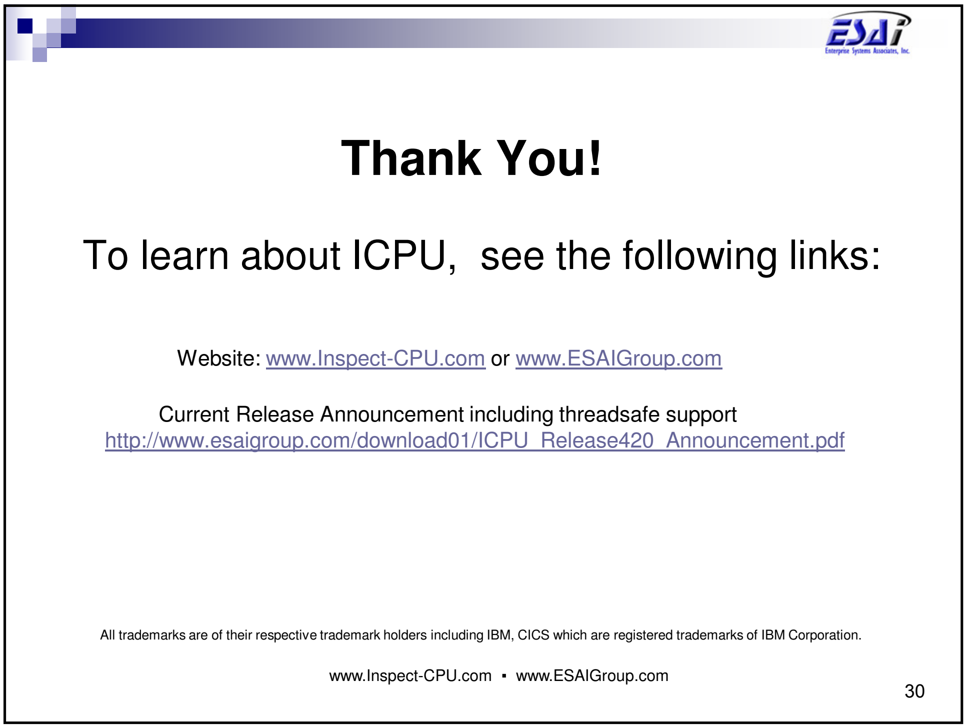
  They show one way to use a sampling tool. Turn it on every week for a full day, find out the highest CPU usage programs and tune them.

  A system with tuned applications is more stable, causes fewer problems and prolongs the lifetime of your site's hardware.



Summary

- 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.



Thank You!

To learn about ICPU, see the following links:

Website: www.Inspect-CPU.com or www.ESAIGroup.com

Current Release Announcement including threadsafe support
http://www.esaigroup.com/download01/ICPU_Release420_Announcement.pdf

All trademarks are of their respective trademark holders including IBM, CICS which are registered trademarks of IBM Corporation.