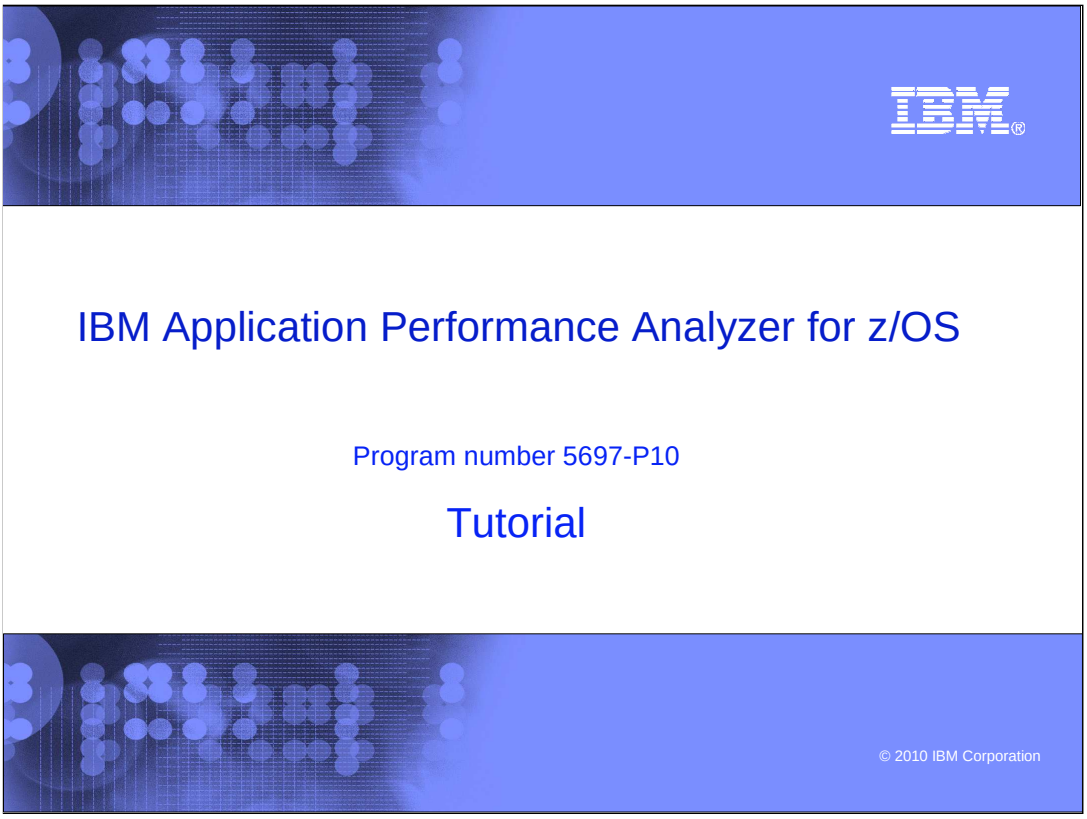
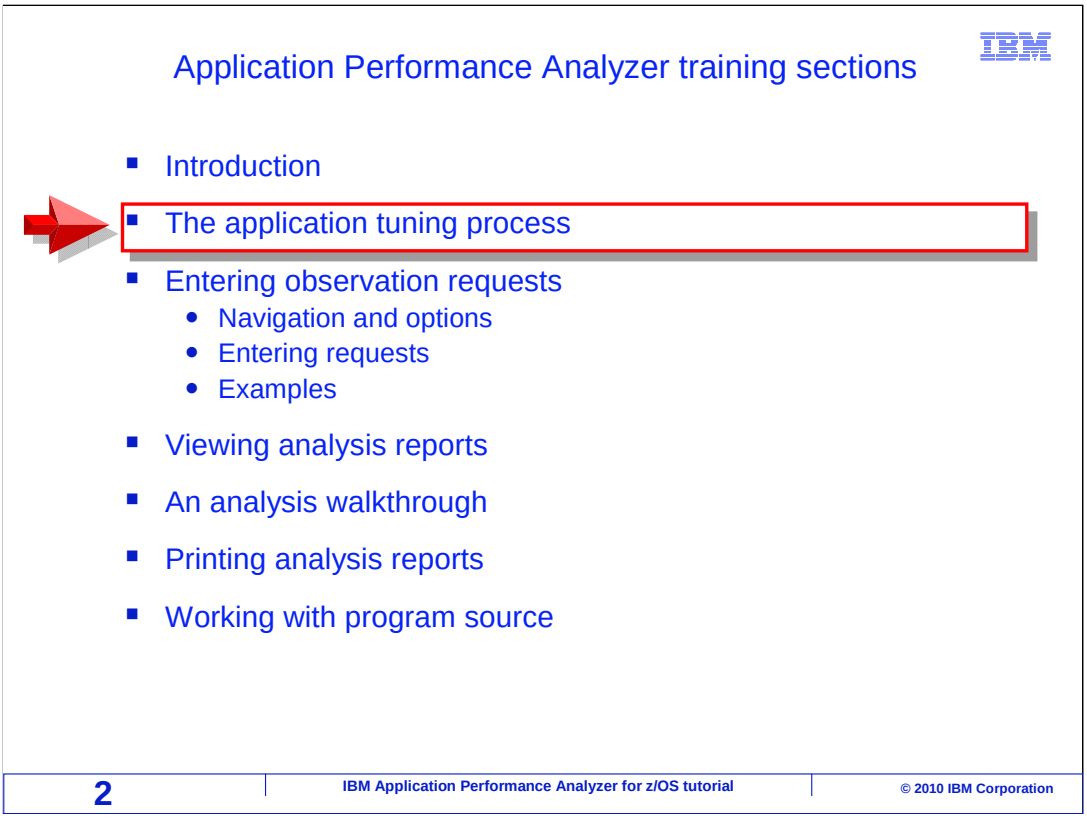
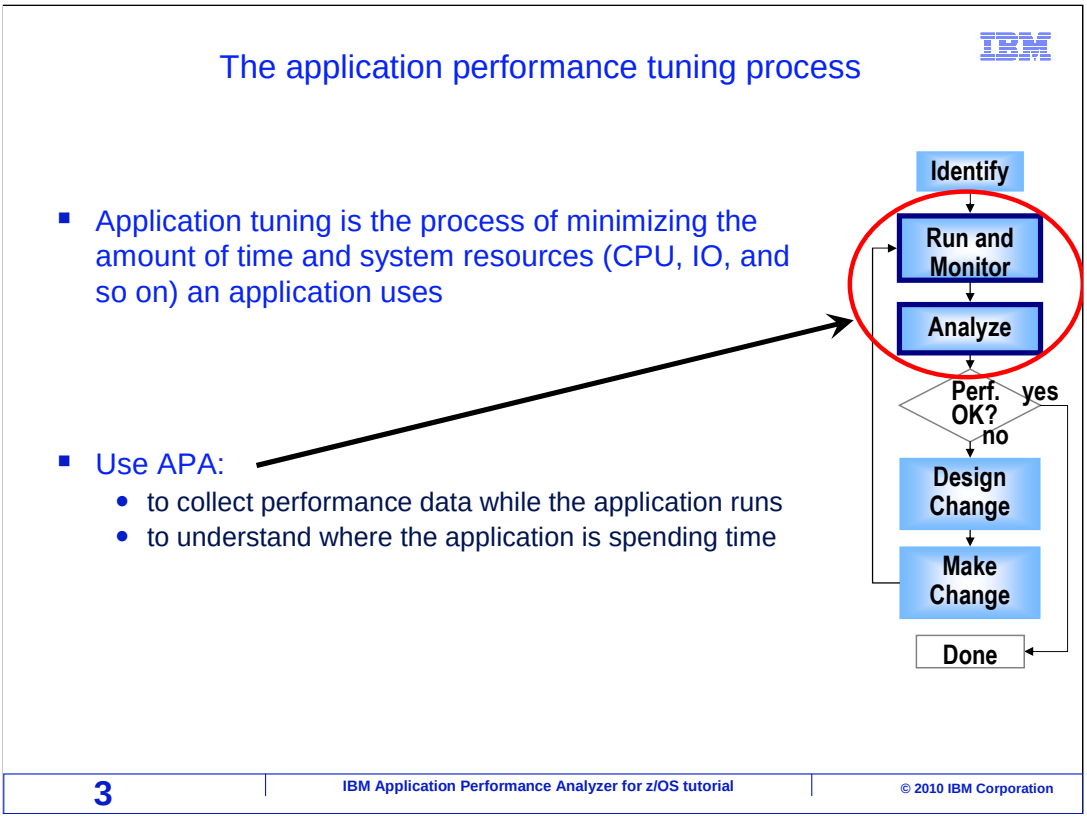


This is the tutorial for IBM's Application Performance Analyzer for z/OS®, one of the IBM zSeries® problem determination tools.



In this section: “The application tuning process”, you will learn about the steps that you might go through to identify performance bottlenecks in your applications. You will learn how you can use APA during the process to identify the programs, and even the source statements, that are having the greatest performance impact.



Application tuning is the process of minimizing the amount of time and system resources (such as CPU time, I/Os, and so on) that an application uses. As you are taken through the process, keep in mind that these are just the basic steps that you might go through to tune an application. Working through the steps can be a relatively informal process that you iterate through quickly and naturally. Or it can be a formal project worked on by a team of people.

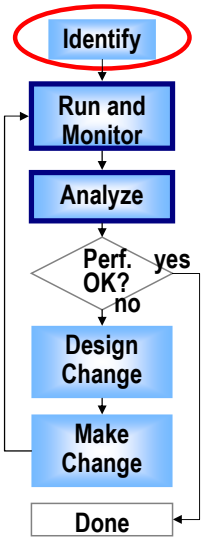
Here are the steps. First, identify the performance issue. After that, run the application to monitor it while it is running, and capture its performance characteristics. Then analyze the collected performance data. Based on your analysis, determine whether the performance of the application meets your objectives. If it does, you are done. If not, then design a change to the system or the application, and make the change. If you have made a change, then go back to the second step and test the performance impact of the change. Run through these steps iteratively until the application meets your performance goals.

During this process, you can use APA during the second and third steps. Use it to collect performance data while the application runs, and to understand where the application is spending its time.

The application performance tuning process: step 1

STEP 1: Identify the Performance Issue

- What is the issue?
  - Batch job running too long
  - Online response time is slow
  - Excessive CPU utilization (looping?)
  - Want to test performance before promoting a change to production
- Consider:
  - What level of performance is required/acceptable?

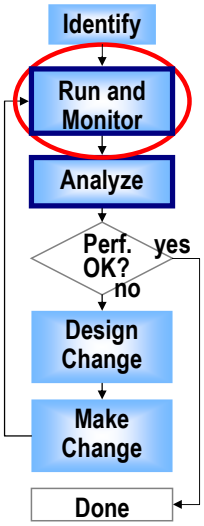


Step one in the process is to identify the performance issue. The goal of this step is to answer the question" "What is the issue?". That is, "is a batch job running too long?", "is online response time too slow?", or "is there an application using an excessive amount of CPU time?". Also you might ask, "do you just want to re-test the performance of an application before promoting a change to production?". An important consideration is "What level of performance is required or acceptable?". When it comes to judging application performance, there never seems to be an "ideal" level of performance. As you set a performance objective, you are typically trying to set a bar that is "good enough" based on your user's requirements.

The application performance tuning process: step 2

STEP 2: Run and Monitor the Application

- Use APA to collect performance data
  - Enter an APA observation session
- APA Options:
  - Start an observation session for a running application, or
  - Schedule observation sessions for applications that will run later
  - Specify the duration of the observation session
  - Specify the sampling rate
  - Specify whether to capture in-depth DB2®, CICS®, MQ, and IMS™ data

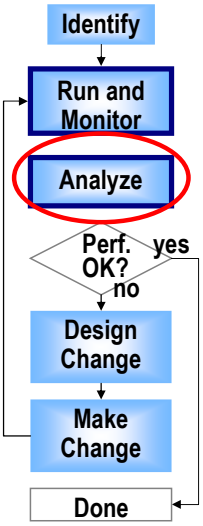


The second step is to run the application, and monitor it while it is running. Use APA to collect performance data. You enter an observation session request to tell APA to monitor an application. You have several options here. You can start an observation session for an application that is already running, or you can schedule one for an application that will run later. You specify the duration of monitoring, the sampling rate, and whether to capture detailed information for DB2, CICS, MQSeries®, and IMS.

The application performance tuning process: step 3

STEP 3: Analyze performance data

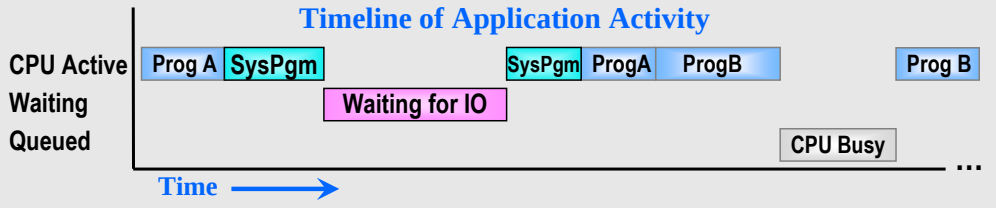
- Use APA to analyze performance
- The goal of this task is to answer the questions:
  - Where is my application spending most of its time?
  - What did the application (or system) do that caused it?



In the third step, you analyze the performance data collected by APA. The goal of this task is to answer these questions. First, where is my application spending most of its time? And secondly, what did the application or system do that caused it? You will use the reports that APA produces to help you answer these questions.

The three states of an application

- At any time, an application is in one of these three states:
  - **CPU Active**
    - An application or system program is executing instructions
  - **Waiting**
    - The application is waiting for the system to complete an activity that it requested, such as a file or database read or write
  - **Queued**
    - The system is busy processing other tasks
- You can tune the application to affect **CPU** and **WAIT** times



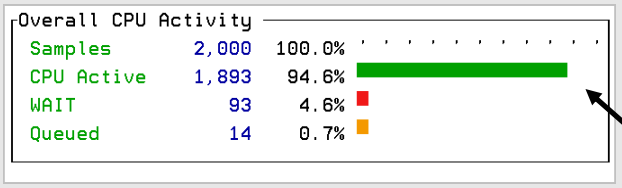
As you are analyzing the performance of your application, keep in mind that at any point in time an application can only be in one of three states. It is either CPU active, waiting, or queued. CPU active means that the application or a system program is actively executing instructions. Waiting means that the application is waiting for the system to complete an action that it requested, such as a file I/O or database read or write.

Queued means that the system is busy processing other tasks, and the application is not being dispatched. You can tune an application to affect CPU or wait times. During your analysis you will want to determine what percentage of time the application is spending in each of these states, and then for the states where a lot of time is spent, determine why.

APA reports percentages for each state

- APA will report the percentage of time spent in each state

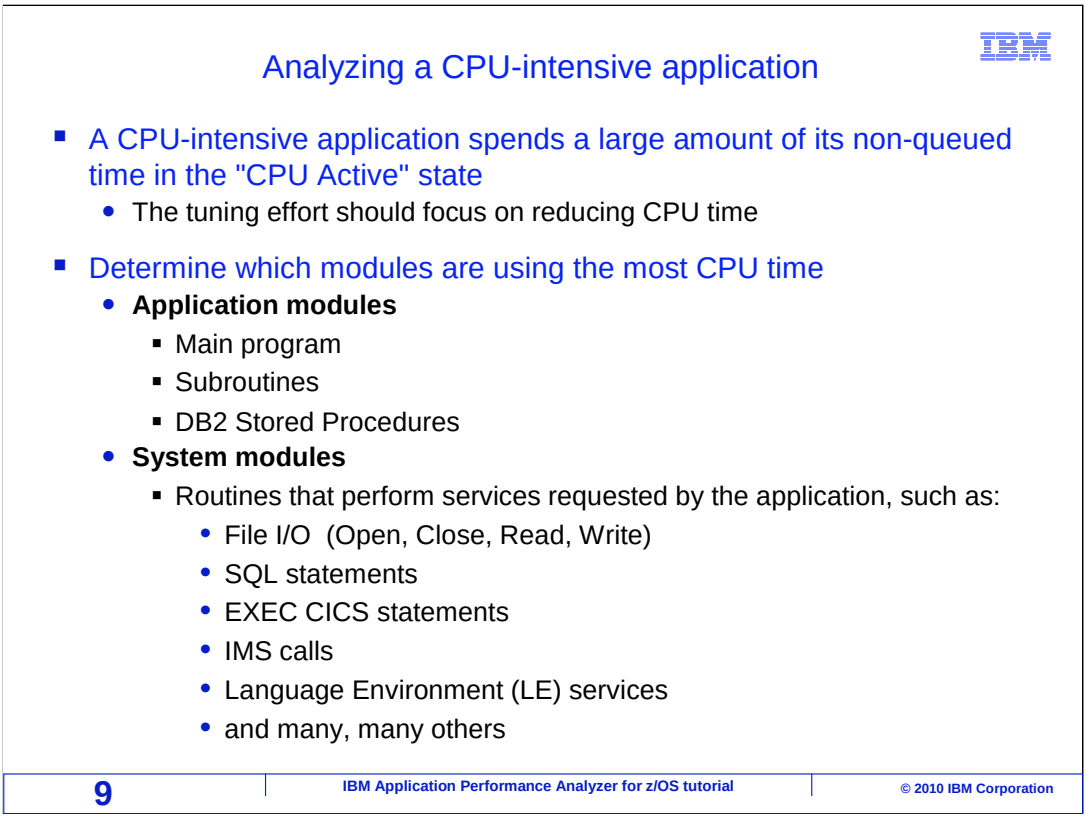
Excerpt of the APA "Measurement Profile" report



In this example, most of the time is spent in CPU Active

- To analyze the application:
  - Determine which is the issue: **CPU or WAIT** (or both)
  - Drill down into detail reports to find out what the application did to cause the CPU or WAIT time

APA will report the percentage of time spent in each state. In this example, the application spent about 94% of its time in the CPU active state, about 4% of its time in the wait state and about 1% of its time in the queued state. To analyze an application, first determine which is the biggest issue: CPU, wait, or both. In this example, most of the time (about 94%) was spent in the CPU active state, so clearly that is where you want to focus the research. You can drill down into detail reports to find out what the application did to cause the CPU or wait time.



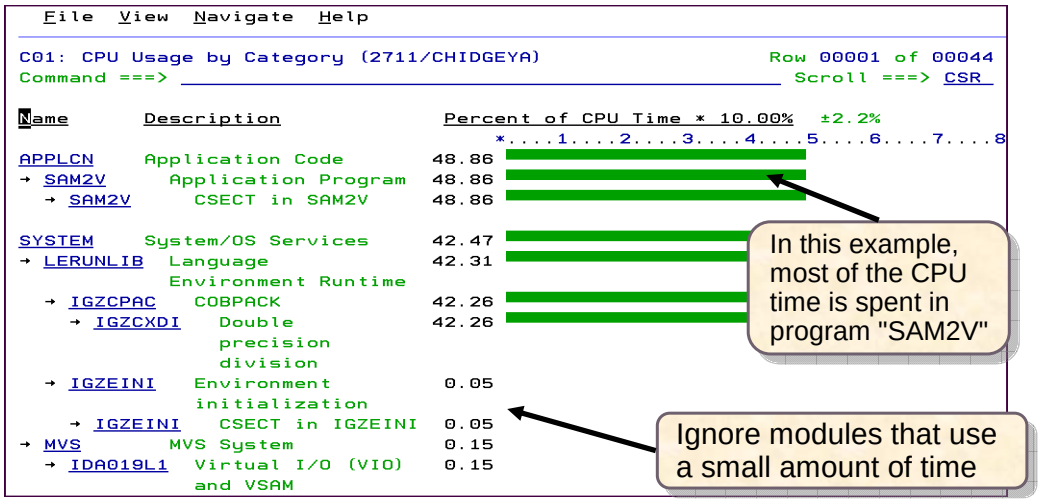
To start your research, it can be helpful to categorize your application as either CPU intensive or wait intensive. That will help you determine where to focus your efforts. A CPU intensive application spends a large amount of its non-queued time in the CPU active state. You should determine which modules are using the most CPU time. Is most of the time spent executing logic in application modules? Is it mostly in the main program? In subroutines? In DB2 stored procedures?

Or is most of the time being spent in system modules, routines that perform services that were requested by the application. These are services such as, file I/O's, SQL statements, EXEC CICS statements, IMS calls, language environment services, and so on.

APA provides “CPU” reports

- Determine which modules are using the most CPU time

The APA "CPU Usage by Category" report

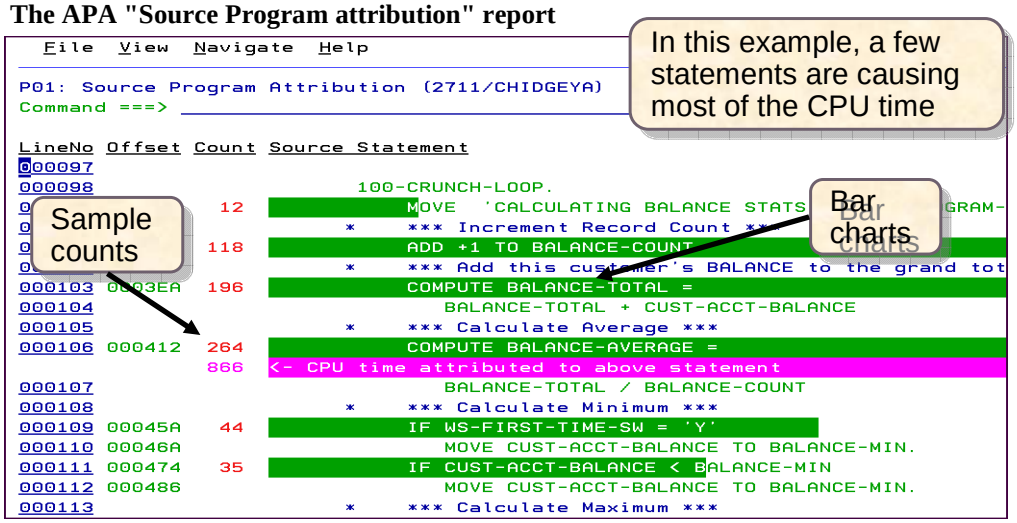


APA provides reports that will help you determine which modules are causing the most CPU time. In this example, most of the CPU time is spent in an application program named SAM2V. In its reports, APA will show information for all of the programs that it caught executing as it performed its sampling. However, you might identify programs listed that used a very small amount of CPU time. Generally, you can ignore these low-usage programs, because you want to focus your efforts on the programs that use a lot of CPU time.

APA can show source statements

- For application programs, you can drill down to the source statements

The APA "Source Program attribution" report

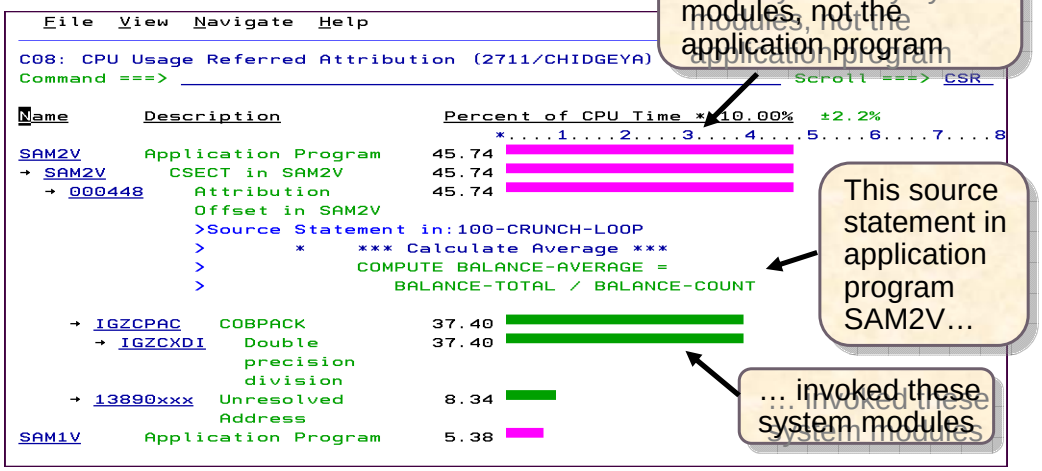


For application programs, you can drill down to the source statements. In this example a few program statements are causing most of the CPU time. In its reports, APA can provide visual clues such as the bar chart that is overlaid on the source code. It will show you sample counts for each statement, so you know the relative CPU impact of each statement sampled.

APA can “attribute" CPU time

- For system programs, you can identify what application program statement invoked them

The APA "CPU Usage Referred Attribution" report



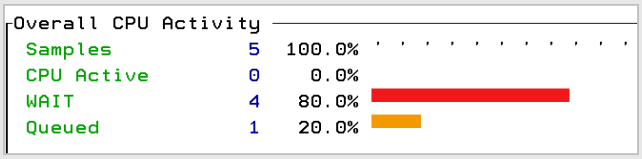
Very often, most of the CPU time is not caused directly by statements in your application programs. Most of the time might be caused by executing instructions in system modules that are performing services requested by the application. The report shown on this slide is the APA “CPU Usage Referred Attribution” report, which shows you some very important information. It shows you the application statement that ran that caused the lower-level system routines to be invoked. With this information, you can pinpoint exactly what your application programs did that resulted in system-level CPU time. In this example, the system modules that are near the bottom of the screen were invoked by a Compute statement in an application program called SAM2V.

At this point, you should be aware that APA can help you understand what is causing an application to use most of its CPU time.

Analyzing a wait-intensive application

- A wait-intensive application spends most of its non-queued time in the "WAIT" state
  - The tuning effort should focus on reducing wait time

Excerpt of the APA "Measurement Profile" report



- Determine why the application is waiting, for example:
  - Waiting for file or database I/Os to complete
  - Tape Mounts
  - Supervisor Calls (SVCs)
  - and many, many other possible reasons

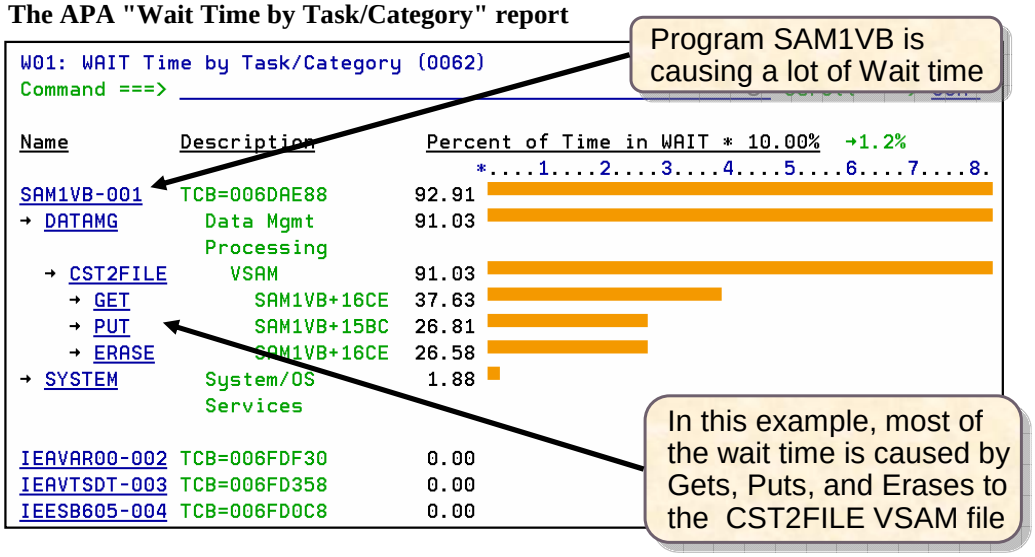
In a lot of applications, CPU time is not the biggest problem. The problem might be that the application is spending most of its time in a wait state. You might identify your application as being wait intensive, which means that it spends most of its non-queued time in a wait state. For these applications, of course, your tuning effort should be focused on reducing wait time.

During your analysis, determine why the application is waiting. For example, is it waiting for file or database I/Os to complete? Is it waiting for tape mounts? Or supervisor calls? Or any of the other numerous reasons why an application might be waiting.

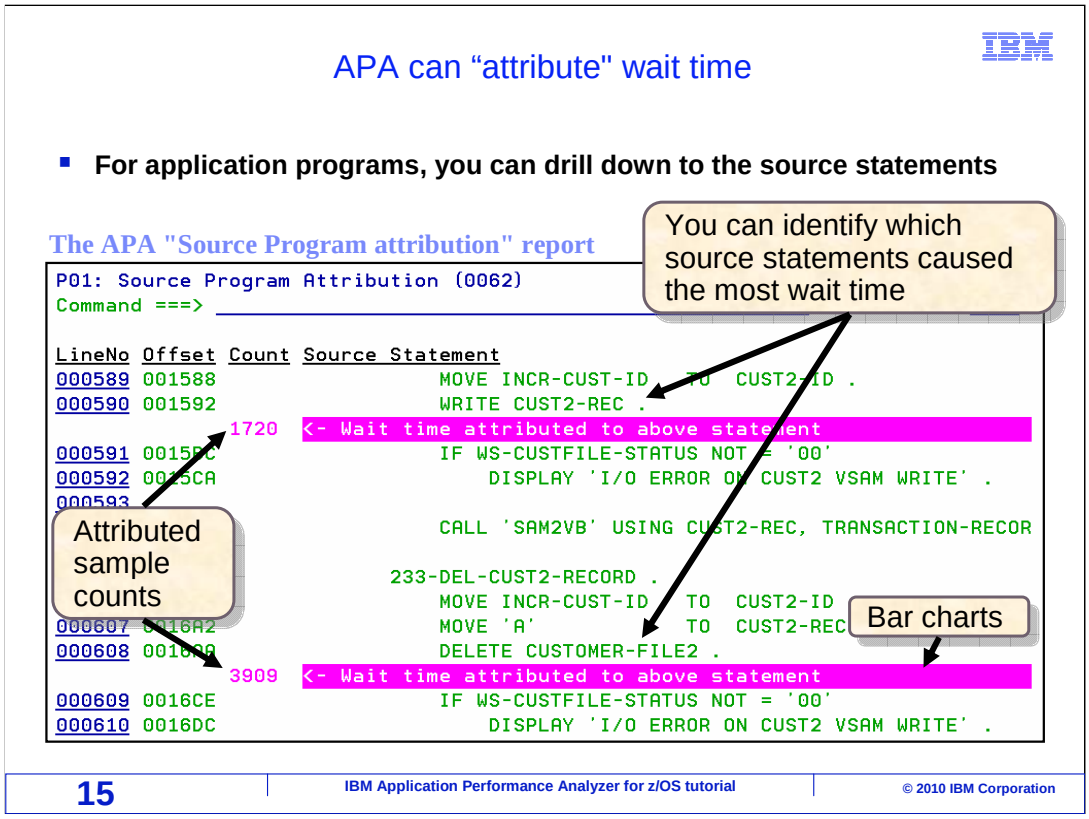
APA provides “Wait” reports

- Determine which modules are causing the most wait time

The APA "Wait Time by Task/Category" report



For wait intensive applications, determine which modules are causing the most wait time. In this example, most of the wait time is caused by Gets, Puts, and Erases to the CST2FILE VSAM file. A program named SAM1VB is causing a lot of the wait time.



For application programs, you can drill down to the source statements. Wait time can be attributed directly to the program statements that caused it. In this APA report, you are given visual clues in the form of bar charts, and sample counts so you know the relative impact of each statement that caused significant wait time.

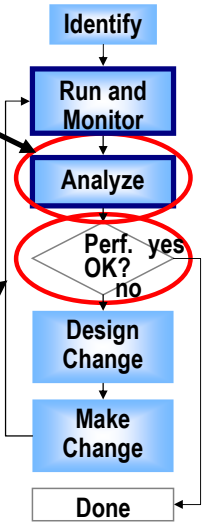
The application performance tuning process: step 3

STEP 3 (continued): Analyze performance data

- On the first iteration, select and capture "baseline" performance metrics
  - job duration
  - transaction response time
  - CPU utilization %
  - time to process 1,000 records
  - ???

Use a metric that is meaningful for your application

- On subsequent iterations, re-capture and compare the performance metrics
  - Is the performance desirable / acceptable?
    - Is the performance metric in the required range?
    - When it is, you are done!



Back to the application tuning process. In the third step, you will analyze the performance data using APA reports. As you have been shown, you can use APA to help you isolate performance issues. If you are working towards a performance goal, this will probably be an iterative process. On the first iteration, identify and capture baseline performance metrics. As a performance metric, select a simple measurement that is meaningful for your application and your users. For example, batch job duration or transaction response time. In subsequent iterations, recapture the performance metric, and you can compare it to the baseline to determine if you have improved it.

After you have monitored an application, and analyzed the results, then you are ready to ask the question: “Is the performance desirable, or acceptable?” If it is, you are done. If not, you will move to the next step.

The application performance tuning process: step 4 & 5

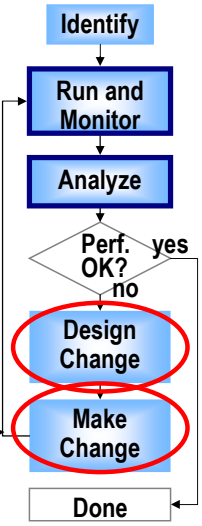
- Once you know where the application spent its time, you can design changes to reduce CPU or WAIT time

STEP 4: Design a Change

- Examine your application and system, and formulate a change to improve performance

STEP 5: Make the Change

- Code and implement your change
- Get ready to re-run the application to test the performance impact of your change



Once you know where the application spent most of its time, you can design changes to reduce CPU or wait time. In the fourth step, examine your application and system, and formulate a change to improve performance. In the fifth step, implement the change, and then get ready to rerun the application so you can test the performance impact of your change.

At this point, you have gone through how you can use APA to collect performance information about your application. You have also been through how you can use it to identify specific areas of your application code that are causing the most CPU or wait time.

That is the end of this section, the application tuning process.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_APAv10s02TuningProcess.ppt

This module is also available in PDF format at: ../APAv10s02TuningProcess.pdf

You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, copyrights, and disclaimers