# Rotting ROTs

For years, performance and capacity planners have used Rules of Thumb (ROTs) as a starting point for tuning and validation of performance. In our last Hot Flashes and in our TUNING Letter 2006 No. 6, we suggested that it was time to create some new ROTs and provide our recommendations for them. We also encouraged people to join our SHARE EWCP project to work on the ROTs. We received many emails during that time and looked at several data center reports.

Immediately, we started to see some problems with just the few ROTs we included. Our suggestion that you run with CPU Busy at 100%, for example, caused headaches in some installations. We had made the assumption that both WLM and your operators were well-managed. In two cases, the operators simply kept flooding the system because users were complaining that their jobs weren't executing, so new initiators were started. At 100% busy, they may be executing, but very slowly! In another case, the WLM goals were not well defined, so that at 100% processor busy, the important online systems suffered while the lower importance batch jobs took over the machine. And we found that there were exceptions to every ROT that we published.

Our conclusion is that it's not really currently possible to create a ROT that applies to even a small percentage of installations. When ROTs first became popular, most installations had similar types of workloads, similar machine sizes, and similar needs. We see that with our customers today, the variation is so wide in these areas that ROTs usually don't work well. You simply wouldn't expect the same paging rates, DASD response times, and out-and-waiting users on a 25 MIPS uniprocessor running legacy batch work and a little TSO that you would see on an 18,000 MIPS 54-way processor running huge WebSphere and CICS applications. So we're retracting our initial recommendation and will instead provide you with two preferred methods of monitoring your system.

The theory behind both of these suggestions is that nobody has time to review all of the important metrics on your system and that you need some type of exception processing to handle any system problems. There are several monitors that provide exception processing, but you need to know how to set the parameters that define exceptions. That's where ROTs used to come in handy. So the purpose of these two methods: "Top Ten Lists" and "Knee of the Curve" is to provide those values tailored to your specific installation.

## Top Ten Lists

One of the methods we described in the initial ROTs article was using 'Top Ten' lists. The theory here is that each installation will have unique 'outliers' or 'exceptions' when it comes to any performance metric. You need to find out what's an outlier for your installation. First pick some metrics that you want to analyze (e.g. CICS response time, DASD response time). For each day, sort them in descending (usually) size and list only the top ten. This will only take a day or two for you to find your own ROT (usually the value of the tenth item) for each metric.

This will also provide a lot of knowledge for your performance staff (even if it's only you!) about what is reasonable and to be expected in your installation. You can use these ROTs to set exceptions for your online monitors and/or background reporting.

### *Performance Metrics*

We'll provide a short list of the performance metrics that we think are the most important. Performance analysts are used to looking at dozens of metrics (for example, we used to look at about a dozen paging metrics), but there simply isn't enough time. The purpose of exception reporting is to let you know that there might be a problem. Once you identify that there is a problem, you will then need to obtain more metrics and more detailed data in order to fully analyze the situation. So let's look at the key performance metrics on your system:

RESPONSE TIMES -

For this discussion, our use of the term 'response time' refers to both online response time and batch turnaround time. They both measure how well you are providing service to your users. If your response times are acceptable, then the other performance metrics really don't matter to you from a performance point of view.

You are meeting your objectives, so who cares if you have long DASD response times? Nobody but the capacity planner (which might be you too). You may be coming to the end of some of your resources where response times and batch turnaround times will start to become unacceptable. Not only do you need to know when you will reach that point, but you need to know which resource is causing you the most delays.

There are several ways to view response time, and each method provides a different view:

- You should always identify exceptions from some type of online monitor so that the operators or automated tasks can respond. If the average CICS response time is .4 seconds and it spikes to 2 seconds, then someone needs to know about it immediately and take action.

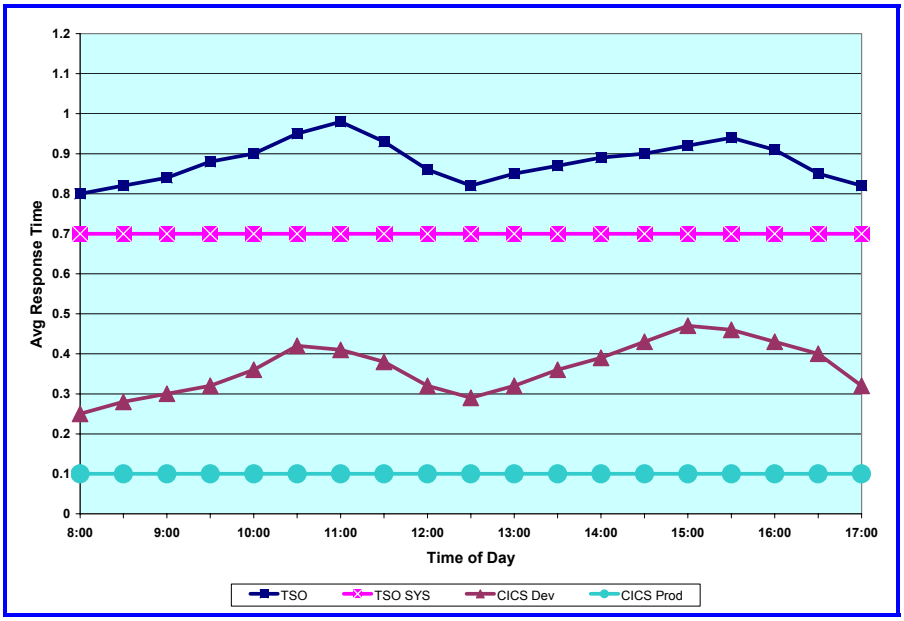
- You can look at different types of response times, and each will provide a different view.
  - Although most installations look at average response times, we think it's best to look at two types of response times for the same subsystem (such as CICS). The first type of response time is that for your high importance work. These response times should be well managed by WLM and never fail to meet your goal. If they do fail the goal (as evidenced by a Performance Index or PI over 1.0), then that exception is very noteworthy and indicative that action needs to be taken.
  - The second type of response time is that for your low importance work. This is the work that will be punished first when you start to run out of resources. We consider this to be a much better indicator than the high importance work response times, because these are early indicators that your loved ones will soon experience pain.
  - If you only view the average of all response times, then you won't see the early warning of the low importance work.
  - So you will probably set your exceptions to the PI for important work and the actual response time (plus a bit) for your low importance work.

- One of the most important ways to view response times is by trending. The problem with using exception reporting for response times is that you probably want to experience zero exceptions. If that's true, how are you going to tell when they are getting bad? The easiest way is to view the lengthening of the response times over time. We recommend a graph that shows the average response time of high importance work and the response time of low importance work on the same graph. Figure 5 is a sample that shows time of day measurements for both high and low importance transactions. You can use a similar trend by day of the week.

- Another form of exception reporting for response times is to monitor your Performance Indices (PIs), where the exception is any PI that is over 1.0.

RESOURCES -

**Figure 5- Average Response Times**




Although the response time metrics are going to tell you all you need to know about the service you are providing your users, the reason to monitor resources is to be able to identify which resources are becoming the most strained. We don't believe that you need to monitor all metrics available, but just one key indicator for each resource. Once you identify any type of problem with a resource, then you will need to look at other metrics, but there's no reason to have indicators for several metrics when one or two will do.

- DASD Response Times (sort by response time, and list the volume, I/O rate, and time of day for the top ten volumes). We use 'response time', which includes IOSQ time and not 'service time' because it is simply one of the components of response time. Once you identify poor response times, then you can determine the cause of that response time, whether it be IOSQ time, path busy, or device busy. You will tend to see the same volumes every day, and can start investigating whether you can improve them. There are some volumes that you can never improve, and you'll simply learn that they'll always appear on the report. You may even want to eliminate them from reporting once you identify that they can't be improved. Your ROT here is probably the response time for the tenth worst volume.

- DASD Space (trend the total DASD space and the temporary storage volumes). This will show you the rate of growth and help you determine when to obtain more storage (or set archive requirements differently).

- Storage Activity (sort by paging activity rate by peak interval and locate the highest paging rates and the times of day). This will let you pick a paging rate for your ROT for online or background monitoring. When paging spikes, investigate the workloads causing the spikes by service class (RMF or CMF).

- Storage Usage. You could trend the storage usage to find some high watermarks, but we think that identifying the large storage users (see the next section) is more useful. If you trend storage usage, you should track the following: fixed pages below the line, CSA/SQA below the line (used and allocated), total fixed pages, total storage used.

- CPU Usage (sort total, not average, CPU busy by time of day for each LPAR and the total CEC). The peak is almost always 100% of all CPs (e.g. 600% for a 6-way), but what's most important is to determine how often this occurs during the day (two intervals or 20 intervals?). We provided an extensive article on CPU utilization in our TUNING Letter 2003, No. 6 (pages 20-30), so please review that article for more suggestions on dealing with CPU busy times.

  What's more important in CPU usage is the latent demand. There is no way to identify the actual latent demand on your machine, but there are indicators that show whether the latent demand is increasing. We'll provide some of those in our next TUNING Letter.

After you create your top ten lists, you can then set exceptions in your monitors to only display anything that's an exception. As you resolve some of the problems (e.g. moving data sets to reduce volume contention), your list of exceptions will go down. But as soon as a new problem appears, your monitor will now tell you about it.

### *Largest Users*

In addition to performance metrics, you should also be able to identify the largest users of the system, in terms of either the number of transactions or the amount of resources used. The usual analysis is done on the following:

- Jobs using the largest amount of CPU
- Jobs using the largest amount of DASD I/Os
- Jobs using the largest amount of DASD space
- Jobs using the largest amount of storage (fixed frames, storage, etc.)
- Online transactions used the most (ten highest CICS, ten highest IMS, etc.). You might need to do this by application, but usually the top ten are the ones you care about the most.
- TSO users consuming the most resources during the day

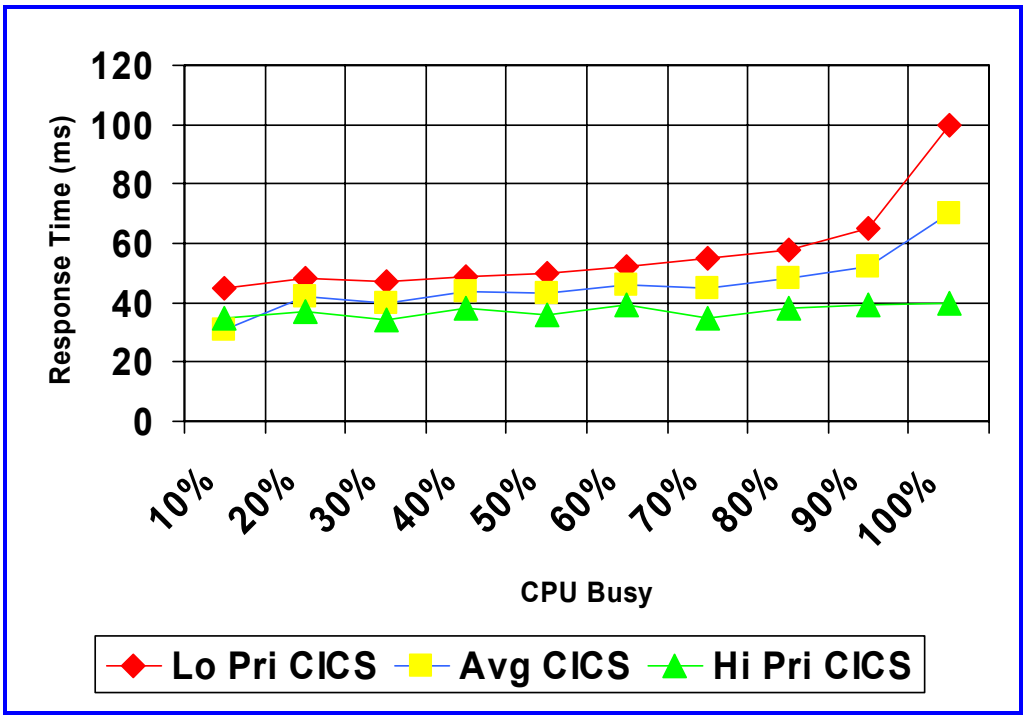
After you create your top ten lists of users, you immediately become more knowledgeable about your system. If, for example, the 'SLRT' CICS transaction accounts for 25% of all transactions, then you may be able to simply track the response time of the SLRT transaction. Likewise, if a new transaction suddenly shows up on the report, then you know that you need to start keeping an eye on the new transaction.

# Knee of the Curve

There is usually a point in time where a constrained resource will quite suddenly start negatively impacting your response times. This can be seen in many 'knee of the curve' reports. Figure 6 is an example showing the effect of CPU busy on CICS response times. As you can see in the figure, the high priority transactions seldom experience any increase as the CPU utilization increases. But the low priority transactions start being affected when the CPU utilization exceeds 70%. The knee of the curve in this figure is between 90% and 95% when the low priority transactions increase dramatically. That 90% busy becomes your ROT for CPU utilization based on CICS response times.

You will need to experiment to see which resources affect your response times most significantly. But a good start is to plot the response times against each of the metrics we listed in the sections above. These plots will often validate that you are using the correct ROTs for your installation.

**Figure 6 - Knee of the Curve Report**




# Summary

As we stated in our original ROTs article:

> *Once upon a time, the typical tuning effort of a performance analyst would be to take the RMF reports for the peak interval and review all of the data elements for problem areas. Problem areas were identified by exceptions to ROTs that the analyst knew by heart. That method is no longer available due to the increase in data elements available to the analyst, the variety of new subsystems that need to be monitored, the increasing size of each installation, and the emergence of a new generation of analysts. The only hope we have is to al-*

*low the machine to monitor itself with exception processing. If no exceptions are found, then the machine is performing as efficiently as possible. The analyst can simply concentrate on the exceptions.* ***Norman Hollander*** *from CA sums it up best when he says that "The best monitor is a blank screen." You really don't need dials and graphs - all you really need to know is whether anything is exceeding your guideline. It's up to you to define the exceptions and that's where ROTs can be used as a starting point.*

*We'd be interested in learning which metrics you have found most useful*

### *Future of ROTs*

We think that setting ROTs is an important step for any performance analyst, but the best ROTs are developed from your own installation, not a published standard. We would be interested in hearing which metrics you have found to be the most useful in your own analysis. Please send us an email with your suggestions to **technical@watsonwalker.com**. ■