What's Really Driving Peak CPU Usage?

In many cases, batch processing accounts for as much as 40 percent

By Martin Wills

Many installations using some form of subcapacity pricing believe their peak CPU usage is driven by their online workload. But is that really true?

During my tenure with MVS Solutions, I often asked customers what they thought drove their usage peaks. Typically, they'd say the peaks were entirely dependent on online processing. However, I've long suspected that batch processing also plays a significant role. So I decided to see if my suspicions were justified.

Some colleagues and I devised a way to analyze monthly usage and calculate both the overall four-hour rolling average and the four-hour rolling average for batch. The results among the participants in our study indicated that batch processing accounted for a surprising 30 to 40 percent of the peaks in many cases. That's a significant contributor to usage and ultimately results in increased costs.

At the Source

Instantaneous spikes in usage, however, aren't really the issue, other than the fact they drive the four-hour rolling average higher. Peaks on a given LPAR aren't usually the problem either, because most costs are based on the highest four-hour average usage of the month on each physical machine or central electronic complex (CEC). Of course, all LPARs contribute to this average, so any approach to controlling your peaks has to take this into account.

To help control four-hour rolling averages, IBM provides a mechanism, known as soft capping, in which you can specify a limit for the four-hour rolling average. All workloads running on the CEC are affected to some degree when capping causes usage to be

ILLUSTRATION BY DANIEL BEJAR / THEISPOT.COM compressed below the level the demand requires. So it's important to plan your cap level carefully. If you set it too low, it might interfere with your ability to meet service objectives for online and high-priority batch work. But if you set it too high, you don't get the savings you might be seeking.

Is there a way to reduce your peaks safely, without lessening the service needed for really important work? Recognizing that a business's online systems are what they are and must get what they need, the focus falls on batch processing.

Ideally, low-importance batch workloads would be slowed down as the four-hour rolling average approaches its capped level. This prevents overloading the machine with work that can be run later. It also minimizes the impact of running into the cap, thus avoiding any negative effects on your online and critical batch workloads. Unfortunately, that's difficult to detect, especially given that instantaneous peaks and four-hour rolling average peaks often don't coincide.

Setting Priorities

The first step is to define the batch workload in terms of importance. This can be done using existing service-level agreements. Another good source of de facto service-level information is your disaster recovery (DR) plan. A DR plan typically details the work that must be run in the event of a major problem. That workload is your highest priority.

A DR plan is likely to detail a second level of work that should be run, if possible. That level is your next highest priority. Those two levels of priority most likely include all or most of the batch workload critical to the business and to supporting the online processes.

Clearly, you want those high-importance workloads to get good service constantly, even when capping is in effect. The remaining workload can be subdivided, too, but any lower-importance work that doesn't directly impact the business can be grudgingly, but safely, sacrificed, if necessary. Typically, such lower-level batch workloads include testing, which is considered both less urgent and less important—except to application developers, of course.

Manual Approaches

To control usage, some rely on manual approaches that are either static or require operator actions.

One such approach is to structure LPARs and the workload to separate the low-importance work from the higher-importance work as much as possible and make sure the balance is correct for your needs. This isn't optimal, but if your main goal is to reduce costs, it might be a worthwhile trade-off.

For example, a CEC with four LPARs might have two of them running online and high-importance batch, one running medium-importance batch and one running low-importance batch. The available CPU would be shared out by setting the weights for the LPARs in such a way that the LPAR running low-importance work has a low share of the CPU when the demand on other LPARs is high.

However, when the demand from other LPARs is low, it has access to more shared logical processors than the weight requires so it can use more. Setting weights and shared logical processors in this way, while avoiding limitations on the higher-importance LPARs, is complex but, in the absence of other tools, might be necessary.

Given that workload separation and LPAR structuring is in place, some operator commands can then be used to favor the higher-importance workloads during monthly peak periods.

If you use Job Entry Subsystem (JES) initiators, you can use JES commands to limit the number of initiators of a given class—typically on the low-importance LPAR—during periods of heavy activity. If you use Workload Manager (WLM) initiators, you can limit the number of concurrent jobs on an LPAR in a job class or stop selecting jobs in a particular service class or job class on a specified LPAR.

THE COMPLEXITY OF A MODERN DATA CENTER is simply too much to realistically control manually, no matter how efficient and knowledgeable your operators and support staff are.

If your installation has a predictable monthly peak, such as three days of afternoons and evenings for month-end processing, you might consider putting these limits on for those times and those days, rather than leaving the decision to the judgment of operators.

The problem, of course, is that these manual approaches are severely limited by the information and the timeliness of that information available to operations in order for them to make a reasonable decision. For example, as we noted, there's no easy way to know when the four-hour rolling average on an LPAR or a group of LPARs is approaching the cap level, so you might be restricting batch unnecessarily or at the wrong times.

If your primary objective is to provide good service to your critical online systems and batch but you'd also like to save money, you can still do so to some degree by setting the capping level fairly high so that you rarely, if ever, run into it during critical business processing.

A Better Way

However, none of these options is satisfactory. So what else can you do? A more effective solution requires intelligent automation software that can make decisions on the fly with full knowledge of what's going on at that specific moment in terms of capacity and demand. A modern data center typically runs thousands of batch jobs each day, spread over a number of LPARs, in addition to numerous online transactions and Time Sharing Option sessions. The complexity of such an environment is simply too much to realistically control manually, no matter how efficient and knowledgeable your operators and support staff are.

Automation software can measure the four-hour rolling average, know when it's approaching the capping level, recognize peaks before they become peaks and take appropriate action before the usage becomes a problem. These actions can smooth out usage spikes and, in fact, allow you to run at a lower capping level than otherwise practical, without sacrificing your service levels. Therefore, you get the full benefits of subcapacity pricing. Z

ON THE WEB

IBM links to ISV products that help with subcapacity pricing. See "ISV Pricing Tools" at www.ibm.com/systems/z/resources/swprice/.

Martin Wills is a 40-year veteran of batch processing who consults on batch practices. He has worked as a product specialist for MVS Solutions Inc.