

# Fine Tuning LPAR for Maximum Performance and Minimum Intervention

BY LYNETTE LENZ

THE CONCEPT OF LOGICAL PARTITIONING (LPAR), which allows one physical system to be split into two or more logical systems, was first introduced by IBM in 1988 for the S/370 architecture. LPAR became available on the AS/400 with OS/400\* V4R4 in the summer of 1999.

LPAR's benefits include server consolidation, a single point of administration and the efficient use of system resources.

However, while LPAR is conceptually simple, implementing a successful LPAR environment can be complex. With V5R1, IBM announced dynamic LPAR, which offers the capability to dynamically move system resources from one partition to another without restarting the adjusted partitions. Dynamic LPAR provides greater flexibility and better distribution of system resources. By implementing stringent requirements for gathering and planning, it's possible to define an LPAR implementation plan that provides maximum performance and optimal use of resources with minimal administrative intervention. (Note: This article is based on V5R1 functionality.)

### Understanding Partition Requirements

Before taking advantage of dynamic LPAR, it's a good idea to first analyze your current environment—documenting each partition's system requirements, monitoring system activity and then analyzing the results.

Following are brief descriptions of the steps required to analyze partition requirements:

- Describe each partition's critical jobs, job

streams and processing functions.

- Understand the processing windows needed for each partition.
- Gather performance data using Collection Services. Data should be gathered by partition, by day, by week and by month. (More information on using Collection Services can be found in the December 2001 article "A Better Way to Collect Performance Data.")
- Consolidate and analyze data using a tool such as Performance Monitor to create an overall picture of system resources.

The analysis performed will be used to devise a plan for timely, dynamic movement of resources.

### Understanding Configuration Options

LPAR configurations entail defining the amount of system resources assigned to each partition. Resources that can be allocated include memory, CPU power and interactive performance. Dynamic LPAR takes the partition definitions one step further and allows you to predefine specific movement of system resources from one partition to another at specific times or days. Before we examine dynamic LPAR, let's look at the basics of configuring a partition.

The first consideration is determining what function the primary partition will play in the overall system-use plan. Although on the surface all partitions seem to be created equal, the primary partition has special responsibilities. It's responsible for all partition management functions. Likewise, the pri-

*By implementing stringent requirements for gathering and planning, it's possible to define an LPAR implementation plan that provides maximum performance and optimal use of resources with minimal administrative intervention.*

primary partition must be active before any other partition can be available for use. Restarting the primary partition also restarts all secondary partitions.

This increases the importance of thinking about how the primary partition will be used. Using the primary partition as a test environment, for example, may not be well advised if your test environments tend to require frequent, and often unplanned, system restarts. It's advisable to create a small primary partition that's strictly used to manage the overall system relegating other functions and activities to the secondary partitions.

Next, consider how processors will be allocated to the partitions. There are two options. They can be distributed as dedicated processors (whole) or as shared processors (fractional increments). Both methods have advantages and disadvantages.

Shared processors offer greater granularity in the distribution of processing power as processors can be distributed in increments as small as one-hundredth of a processor. A system with one or two processors can be configured to act as several logical systems and offer better overall use of available processing units. The downside is that shared processors increase the system overhead because the processing cache is being cleared.

The concept of shared processors doesn't mean that the system physically splits the processor among partitions. Actually, each partition uses 100 percent of the processor for its allo-

cated time slice. For example, a partition that's assigned one-quarter of a processor will have full use of the processor 25 percent of the time. The process of switching from one partition to the next forces the processor cache to be cleared, which, in turn, can cause as much as a 10-percent overall system performance decrease.

Any partition using shared processors must have at least one-tenth of a processor assigned to it. Although it's possible to allocate such a small portion, IBM recommends a minimum of one-quarter of a processor be assigned to each partition.

Dedicated processors offer the converse—more efficient use of processing power. Again the performance gain is due to the processing cache. Yet, the use of physical processors can leave unused CPU in one partition while another experiences high CPU usage. Additionally, it's often difficult to spare an entire processor for a partition that is using large amounts of CPU.

As partitions are configured, minimum and maximum values must be set for memory, processors and interactive. Setting these values isn't as easy as you may think.

All hardware resource assignments are maintained in the hardware page table within the primary partition. The page table size is based on the potential need for memory if all partitions operate at their maximum settings. Consider the following: If each partition is configured with a maximum amount of memory equivalent to all

memory available on the system, not only will the page table be excessively large, it would be physically impossible for all partitions to use the defined maximum memory at the same time.

This illustrates the importance of understanding the requirements of each partition and of not overestimating the maximum amounts of resources. In addition, redefining a partition's minimum and maximum values requires the partition to be restarted, so it's better to define the range accurately from the beginning.

As you configure processor and interactive minimum and maximum values, you'll find their values are tightly integrated. Each partition must have a certain amount of interactive and CPU assigned to it. As you increase the amount of interactive, the system may require additional processing units, or dedicated processors, to support the assigned interactive percentage. A general calculation (see Calculation 1) is used to determine the amount of processing units required to support a given amount of interactive.

### **Footprint Total CPW Dynamic LPAR**

Let's briefly examine two simplified scenarios showing dynamic resource movement can be of value.

**Scenario 1**—Company A has offices in the United States, Japan and France. The majority of system processing for each location occurs during standard business hours (9 a.m. to 5 p.m.). Likewise, each location experiences peak processing requirements during mid-day. Figure 1 is a chart showing system performance data for each location. Time has been charted using "Coordinated Universal Time."

Company A must maintain separate operating environments to accommodate both language and division accounting requirements. This company could benefit from dynamic LPAR by

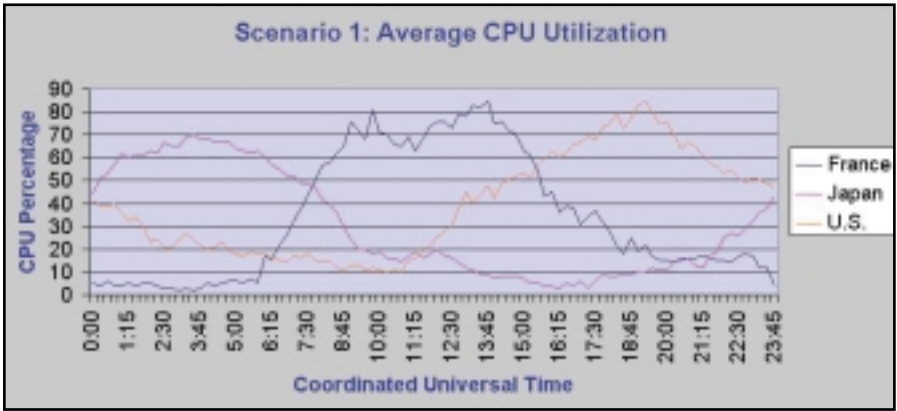


Figure 1

systematically moving system resources from one partition to another as needs change from one time zone to the next.

**Scenario 2**—Company B requires separate operating environments based on specific business processes. The company has one partition for all financial processing (e.g., A/R, A/P, G/L), all BI and data mining activities in another and a third partition that's used as a Domino\* server. Figure 2 (below) shows the average processing needs of each of these environments.

Examining Figure 2 shows that Company B can improve performance by moving resources to the Domino\* partition early in the day, mid-day and again around 5 p.m. Although the financial applications are used throughout the day, system needs peak later in the day and early evening when day-end processing is completed. Movement of additional resources could reduce the end-of-day processing time.

If you plan to implement an LPAR environment that includes dynamic-resource sharing, consider the following (answering “yes” to one or more of these questions may make you a candidate for dynamic LPAR):

- Are there common time or processing-period boundaries where resources can be moved?
- Do certain partitions have peak processing times?

- Do certain partitions have peak resource needs (e.g. CPU or interactive)?
- Are there off-peak processing times when a partition can relinquish resources?
- Do certain partitions have processing trends (e.g. daily flash reports) that require a particular type of resource?

### Dynamic Resource Sharing

Dynamic resource movement can be configured through System Service Tools (SST), Dedicated Service Tools (DST), Operations Navigator or programmatically. In a well-defined operating environment, resource utilization can improve dramatically. Unfortunately, a haphazard plan also

could cause a dramatic reduction in overall system performance.

Resource movement can cause a certain amount of tax on your system. Before devising a plan for resource sharing let's examine some of those taxes.

In general, regardless of the method to move resources or the resource to be moved, it's important to avoid quick overreactions to a partition that needs immediate resources. Moving resources from one partition to another when the resource need is short-lived causes unnecessary churn as the system moves resources. This, in turn, causes a reduction in resource efficiency, the opposite goal of most LPAR users.

Indiscriminate memory movement from one partition to the next can affect performance. When the system moves memory resources from one partition to the next, several events occur. First, all memory to be removed from a partition must be cleared of data. This memory is moved to the machine pool and then saved to disk. Depending on the amount of data and the disks/disk arms available to write the data, this could take several minutes to complete. The Work with System Status (WRKSYSSTS) screen will show an increased amount of page faults as this

### Calculation 1

$$\text{Necessary Processing Units} = \frac{\text{Total Interactive CPW Needed} * \text{Total Footprint \# of Processors}}{\text{Footprint Total CPW}}$$

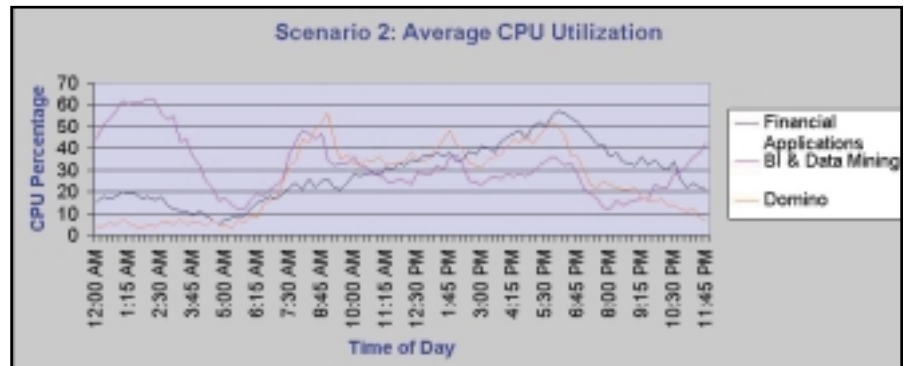


Figure 2

## Technical Corner

*Regardless of the method to move resources or the resource to be moved, it's important to avoid quick overreactions to a partition that needs immediate resources.*

process is completed. Next, memory is moved to the base pool and becomes available to other partitions. Finally, the now-available memory is moved to a different partition. Don't be surprised if this memory movement causes a brief performance degradation in the partition that's losing the memory. If not carefully orchestrated, memory movement could cause performance degradation for several minutes.

Another small performance tax occurs in the re-creation of SQL access plans that have been stored in memory. Storing access plans in memory may force the system to rebuild those access plans in a partition where memory is being removed. As access plans are rebuilt, SQL response times may increase. Rebuilding the access plans usually takes less than 20 seconds.

Dynamic resource sharing can be accomplished through manual intervention, scheduled moves or through the use of programs that determine a valid need for additional resources.

Allocating resources for a partition through either SST or Operations Navigator is relatively simple—once you understand the configuration constraints. Operations Navigator provides a step-by-step wizard to create a new partition. Figure 3 (page 76) illustrates the screen used to define processor resources for a new partition. All processors are currently assigned to the primary partition. In addition, the partition's configuration is set to use dedicated processors, which means one of the two processors, currently being used by the primary, must be moved to the new partition. Fractional processor allocation could be specified if the partitions had been defined to

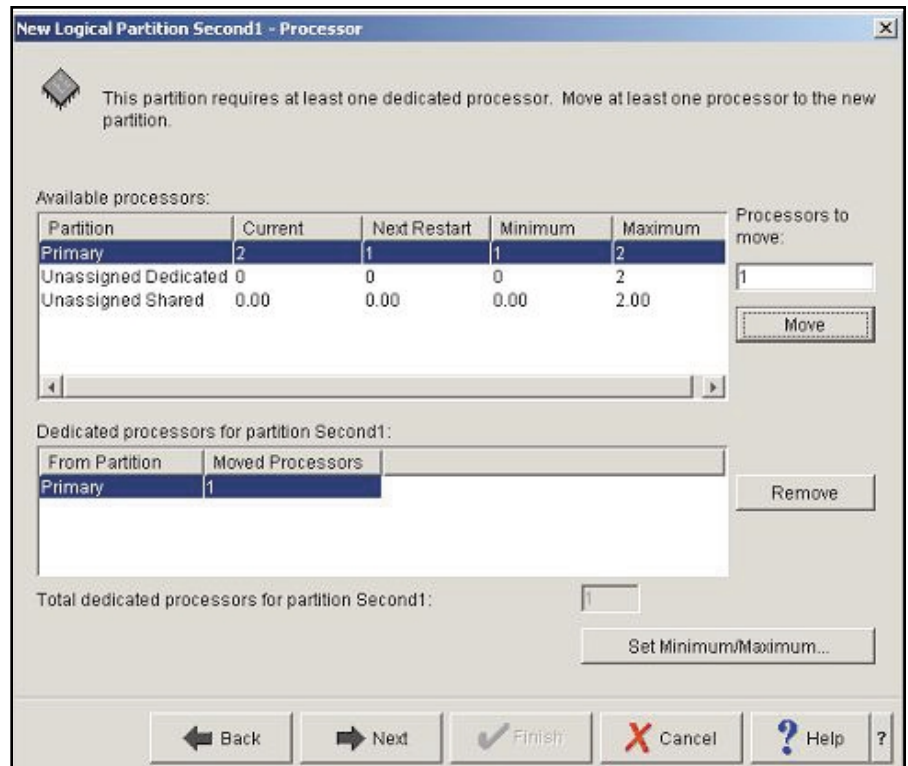


Figure 3

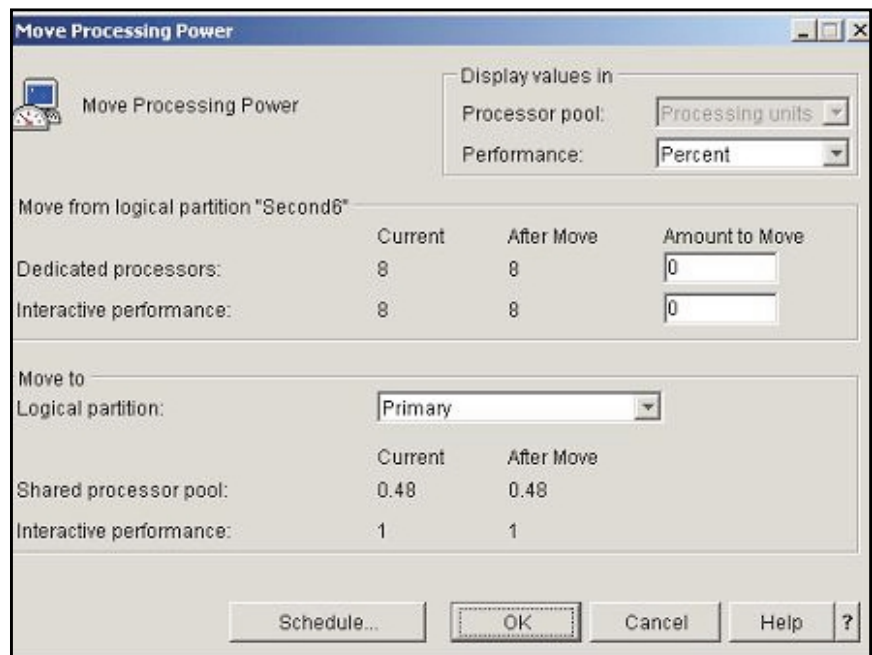


Figure 4

*Taking advantage of scheduled moves eliminates the need to have an operator constantly watching over the system to determine if resources should be moved.*

use shared rather than dedicated processors.

Reallocating resources from one partition to another is also simple with Operations Navigator. Figure 4 shows how processing power (processors and interactive workload) can be moved simultaneously, when necessary, from one partition to another. Use the Schedule button at the bottom of the dialog box to assign scheduled movements of the processing power.

Taking advantage of scheduled moves eliminates the need to have an operator constantly watching over the system to determine if system resources should be moved. Scheduling resource moves can be completed through Operations Navigator or programmatically. Figure 5 (below) is a sample screen showing how to schedule the movement of shared processing units from one partition to another using Operations Navigator. The scheduler can be used for a one-time resource movement or to setup recur-

ring resource movement on a system.

### Managing the Process

Once you've completed the initial processing requirements of each partition, partition configurations and planned resource movement, it's imperative that the process be documented. Documentation should include the requirements defined early in the process, performance analysis summary, partition configuration details, resource movement guidelines and performance metrics to be used for evaluation purposes. It also should include a plan for regular performance evaluation and the plan to update the process when necessary.

Lack of evaluation causes eventual, if not immediate, performance problems, as well as potential systems management complications. Detailed steps for requirements planning and LPAR implementation can be found in the following IBM Redbook publications ([www.redbooks.ibm.com](http://www.redbooks.ibm.com)):


"Slicing the AS/400 with Logical Partitioning: A How to Guide" (SG24-5439-00) and "Capacity Planning for Logical Partitioning on the IBM @server iSeries Server" (SG24-6209-00).

### Achieving Success

To successfully implement dynamic LPAR follow these simple steps:

1. Spend time evaluating each partition's hardware, software and processing requirements.
2. Monitor partition activity and look for common times when one or more partitions should be able to release resources that can be reassigned to other partitions that could take advantage of the additional resources.
3. Configure partition resources for initial settings. Remember to consider configuration options as this is done.
4. Create schedules or methods to move resources as necessary by using Operations Navigator, SST, DST or programs.
5. Set acceptable performance metrics.
6. Document requirements, performance analysis, metrics and the evaluation process.
7. Perform regular performance analysis for all partitions.

In other words, plan, implement, evaluate and modify, if necessary.

Information on LPAR is available on IBM's iSeries Web site ([www.ibm.com/iseries/lpar](http://www.ibm.com/iseries/lpar)). 



Lynette Lenz works on technical documentation for Barsa's Automatic Partition Resource Manager (APRM), a dynamic LPAR resource tuning application, at Barsa Consulting Group, LLC. She has also worked extensively in the iSeries BI arena. Lynette can be reached at [llenz@barsaconsulting.com](mailto:llenz@barsaconsulting.com).

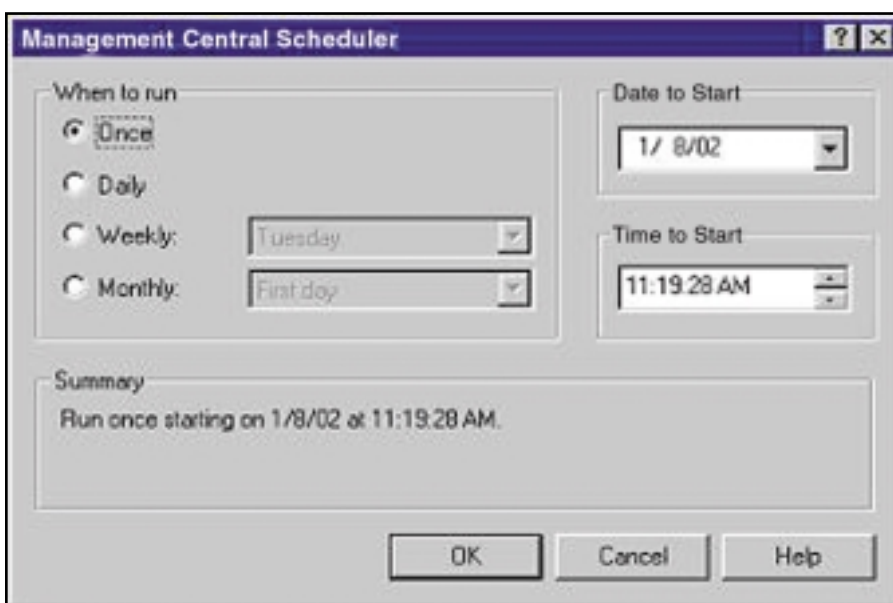


Figure 5