

Performance Testing Your SAP System — Best Practices for Preparing Your Test Environment

Tim K. Nguyen



Tim K. Nguyen is Managing Principal of Orchid Consulting Group, Inc., a consultancy that focuses on SAP and e-Business systems performance and business continuance issues and solutions. With over 20 years of experience building successful IT professional services practices, Tim oversees most of Orchid Consulting's projects worldwide.

(complete bio appears on page 54)

If you are an SAP project leader or an IT manager involved with SAP systems, then you know that testing is actually much more than a footnote to your urgent day-to-day tasks. It is a vital aspect of your job, with urgent priorities of its own. One underappreciated aspect of SAP performance testing is that, while it is obviously a technical exercise, it is ultimately motivated by the need to answer specific business questions and guide decision-making at many levels.

Long after the dry run and go-live, performance testing must continue through production to answer a wide range of business and technical “what-ifs.” What if we merge the operations of subsidiaries that do not use SAP into our company’s SAP instance? Does the system have enough capacity? What if our rapidly growing business doubles in size? Can order processing keep up? What if we add faster disk subsystems with faster disk cache? How much faster would the system run? Would the increased performance justify the investment?

With crucial business decisions depending on your tests and, ultimately, the profitability of your enterprise at stake, knowing how to conduct accurate and meaningful SAP tests is clearly a very important skill. In this two-part article series, I draw on my experience in working at diverse SAP installations to provide you with a framework of what works and what doesn’t in every aspect of performance testing.

Here in the first installment, I share best practices for test preparation. I help you make sense of the myriad testing and performance tools currently available, and I provide pros and cons to guide you in selecting what may be appropriate for your enterprise.

You'll get guidance on how to form your ideal testing team and how to set up the test itself, from defining your test scenarios to defining and setting up the test bed and various hardware configurations. You'll learn how to obtain meaningful test data, which can make or break the value of your test results. And finally, I discuss the options available for involving resources outside your company and how to evaluate the pros and cons for your situation.

In the second installment, I will cover test execution. You'll learn what to do when things go wrong (because they will, despite your careful preparations), how to minimize problems during tests, and how to prepare for various contingencies. I'll also give you some tips on evaluating results and preparing data for the all-important management review.

By the end of this article series, you will know what to test and how to test it. Most important, by learning the answers to the most critical questions regarding business processes and technical operations, you will be able to ensure that your test results support your company's business objectives.

Setting Up Your Performance-Testing Project for Success

As with any project, successful performance testing of your SAP system will depend on solid project management and contingency planning. Based on my experiences in leading performance-testing projects at a variety of organizations, I have learned that there are several things you can do at this early stage to set up your project for success while avoiding the kinds of problems that can derail it:

1. **Understand your objectives.** Although you will be conducting a performance test, as opposed to a functional test or an integration test, be sure that you and your team are clear on the business objectives of the systems and applications you are testing. There are two kinds of performance testing:

- *Business process testing* answers questions related to changes in business processes and how these changes might improve overall business efficiencies. For example, you might test whether a change in how your system processes orders can save system resources.
- *Technical and operational testing* answers questions related to changes in hardware configurations, weighing hardware costs against system speed and capacity. For example, how much faster would your system run if you added more CPUs and more RAM? Would the investment be worthwhile in terms of performance?

Both types of testing ultimately come down to answering questions of capacity, scalability, performance, and stability in order to support whatever business scenarios your company is contemplating. Being able to identify the questions your testing can answer *in terms of your company's business objectives* will put you in a good position to ask for the necessary budget to conduct these tests.

2. **Achieve buy-in from all stakeholders.** While a core technical team will conduct the actual execution of tests, all of the relevant people in your company must understand and support the goals of the test and be willing to participate as needed.

One challenging but absolutely necessary element of a performance test is getting stakeholder support for the large expense involved in setting up a test environment that adequately mirrors your production environment. The right test environment is critical to getting reliable results. I discuss how to meet the challenge of obtaining the approval and support of all stakeholders in the sidebar "Achieving Management Buy-In" on pages 34-35.

3. **Define the testing budget and time frame.** With the input of both your technical experts and senior managers, you will need to agree upon a budget and time frame. Obviously, these two groups will rarely agree on what costs and

schedules are realistic. The ideas in the sidebar on pages 34-35 should help you pull everyone together, however.

The testing budget should account for internal costs (people and equipment) plus the cost of hiring additional facilities or outside expertise. Make sure the budget is complete and that there are no surprises. Later, I will discuss how to evaluate whether or not to test offsite or to use external experts, as well as how to define a test environment that will yield accurate and meaningful results.

✓ *Investing in Performance Testing*

The more closely your test system mirrors your production system, the more meaningful and valuable your results will be. If you need to upgrade your test environment so that it is a realistic approximation of your production environment, the investment in testing can be significant. The ability to answer “what-if” questions without putting your production system at risk and without investing fully in a new, untried direction is invaluable, however. Demonstrating the business value of the test system may help you obtain funding for a system that mirrors your production system. For a discussion of the business benefits of such a system, see the sidebar “The Business Benefits of a Test Environment That Mirrors Your Production Environment” on page 48.

4. **Define test scenarios that yield realistic and meaningful results.** If you’re testing business scenarios, you’re interested in how changes to processes such as order entry, inventory updates, financial data consolidation, or the payroll run will affect your systems. If you’re testing technical scenarios, you’re interested generally in how changes to your IT infrastructure will affect your system — the effects of a bigger database server, more CPU and RAM in your application servers,

faster internetworking, or faster and better cached storage subsystems, for example.

To define your test scenarios, you will first need to define your company’s “Business Day Profile.” The Business Day Profile consists of SAP business transactions that are run in a typical business day.

✓ *Note!*

The Business Day Profile is one of the most important keys to successful performance testing. You should plan to make it one of your first orders of business and to invest whatever effort is necessary to build the most accurate profile possible. The section “Test Scenarios” later in this article provides concrete guidelines for defining your Business Day Profile.

5. **Define acceptable results.** All stakeholders must agree on what constitutes acceptable results. In terms of SAP testing, this means they need to agree on the following:

- *Baseline performance* — Create a baseline so that you can tell whether a scenario you tested improved or worsened system performance. After defining the various test scenarios, you will need to choose tools appropriate to the test scenario for capturing the current performance data from your production system. Then you will need to get everyone on the team as well as senior management to agree that this is a valid baseline performance for your systems.

SAP’s built-in CCMS (Computing Center Management System) provides a relatively simple method for capturing baseline data. You can use some of the SAP performance transactions to capture SAP performance and execution data without incurring any additional cost. Some of the key transactions for

Achieving Management Buy-In

Since your testing results will ultimately affect the course of your enterprise, it makes sense that the testing project should involve a broad cross-section of staff at your company. In addition, a significant portion of the expertise needed to create the test scenarios and scripts must come from the business and functional folks. How can you obtain the wholehearted support of everyone involved so that the will exists to fully fund the project and carry it through to completion?

To gain full support for testing, you must involve every potential participant early in the planning process, making your goals and their roles completely clear and communicating the benefits of the test to all other stakeholders. Once members of senior management are convinced of the value of the performance-testing project, they can impress upon the people in departments throughout your company why it is important to support the project.

In the early planning stages...

- ☑ **Clearly outline the goals of the test.** Be both realistic and specific in quantifying the benefits of the test to each of the involved parties. Remember which benefits are most important to which parties. Senior management wants to hear about how you'll save money in the long run. Your technical staff wants to know how much faster your network will run.
- ☑ **Ask for the support of each team member.** Be sure people are aware of (and agree to) what is expected of them, whether it's sharing their expertise and time, allocating budgets, providing access to hardware, allowing use of network bandwidth, or just agreeing to respond immediately if a problem or question arises during testing. For example, a database administrator might not need to participate directly in testing, but if a problem with the database occurs, he or she needs to be

this purpose are *ST03*, *ST04*, *SM50*, *SM51*, *DB02*, and *AL08*. Later, in the section "Evaluating and Selecting Testing Tools," I'll describe how you can use these transactions and other tools to measure the test.

- *A formula for measuring cost vs. performance* — Clarify your testing goals by quantifying how your testing will answer the questions that have been raised. For example, if you are seeking better performance and are willing to invest a bit to get it, what performance level are you seeking? How much is the additional performance worth to you in terms of dollar savings or increased business?
- *The amount of tuning that will be done as part of the test* — It is highly unlikely that you will run just one test. Every performance test project that I have ever been a part of has required significant tuning of everything from the hardware to the operating system, including SAP profile parameters and database parameters. Unless you define in advance what will be tuned, and more important what will *not* be tuned, your project will easily exceed its allotted time frame and budget. The second installment of this article series will include a detailed discussion on what should be tweaked and tuned and who should do the tuning and tweaking.

immediately available to address the problem. On-call specialists need to know about the project and agree to their role well in advance of when you desperately need their services!

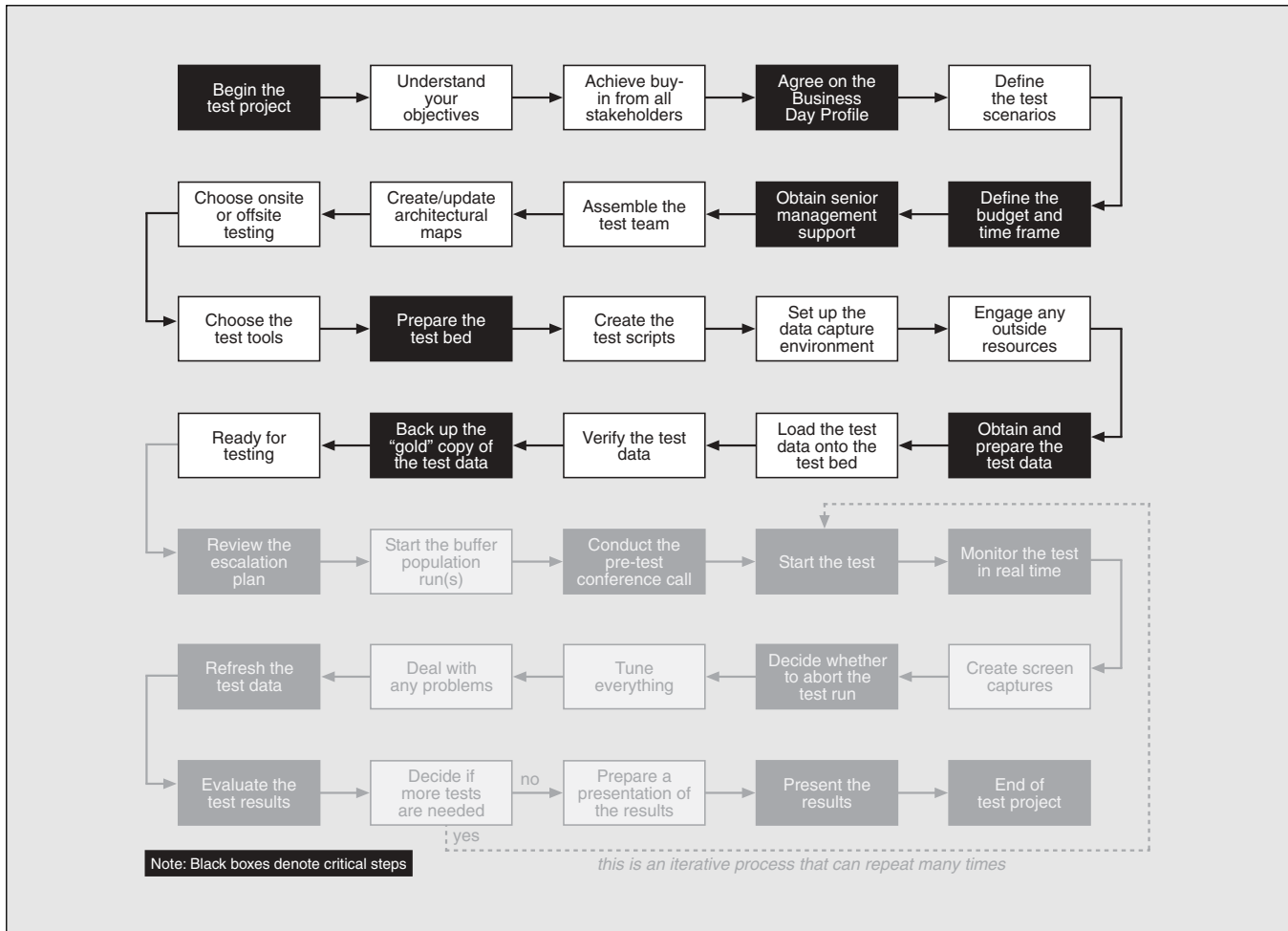
- ☑ **Develop a formal organizational chart.** Identify the person who will be reporting test results to senior management (the test director or the overall project manager), the person who will handle daily decision-making (the test manager), the technical leads who will be preparing and performing various aspects of the test, and the pool of on-call specialists within your company whose expertise may be tapped if needed. Add cell phone numbers and email addresses to the chart and distribute it to everyone involved, including the assistants of your more senior team members. I have seen many companies put up an intranet just for this sort of project. The intranet web server can house the project's organizational chart, the project goals and objectives, test-related job descriptions for each team member, project milestones and updates, project reporting, and even a discussion group for team members.
- ☑ **Make sure that upper management has a good picture of the magnitude of the cost involved in testing.** As a first step toward gaining financial support, make sure that the budget is complete. The budget should include internal costs of people and equipment as well as costs to hire additional facilities or outside expertise. If upgrading the test bed is necessary (and it probably is), it should not come as a surprise later. Senior management needs to understand what is required and why.
- ☑ **Demonstrate the value of the investment in testing.** This is where your understanding of how testing supports business objectives comes into play. Because the cost of upgrading your test bed could be quite significant, you might also need to underline the value of the test environment that will be created by demonstrating its usefulness in disaster recovery or business continuance. For a discussion of this point, see the sidebar "The Business Benefits of a Test Environment That Mirrors Your Production Environment" on page 48.

Getting Ready to Test

In my experience, the logistical efforts of getting everything and everyone ready for testing are far greater than the efforts of testing itself. Preparing your environment and your team to run SAP tests involves the following tasks:

- Selecting and preparing your internal testing team
- Creating or updating the architectural map of your SAP environment
- Deciding where to test — onsite or offsite?
- Evaluating and selecting testing tools
- Defining test scenarios
- Setting up the environment for capturing test data
- Defining hardware configurations
- Setting up your test bed
- Obtaining meaningful data to use for testing
- Planning for efficient execution of multiple test runs
- Weighing the pros and cons of involving external resources

Figure 1 Performance Test Project Tasks



In the sections that follow, I discuss best practices and provide recommendations for each of these tasks. **Figure 1** provides a graphical overview of the steps involved in preparing your test environment and the order in which they should occur, highlighting the critical points along the way. (Note that the shaded steps in the lower portion of the diagram cover the test execution, which will be discussed in the second installment of this two-part article series.)

Selecting and Preparing the Internal Testing Team

Testing teams can vary widely in size, composition, and combined experience. I've worked with teams

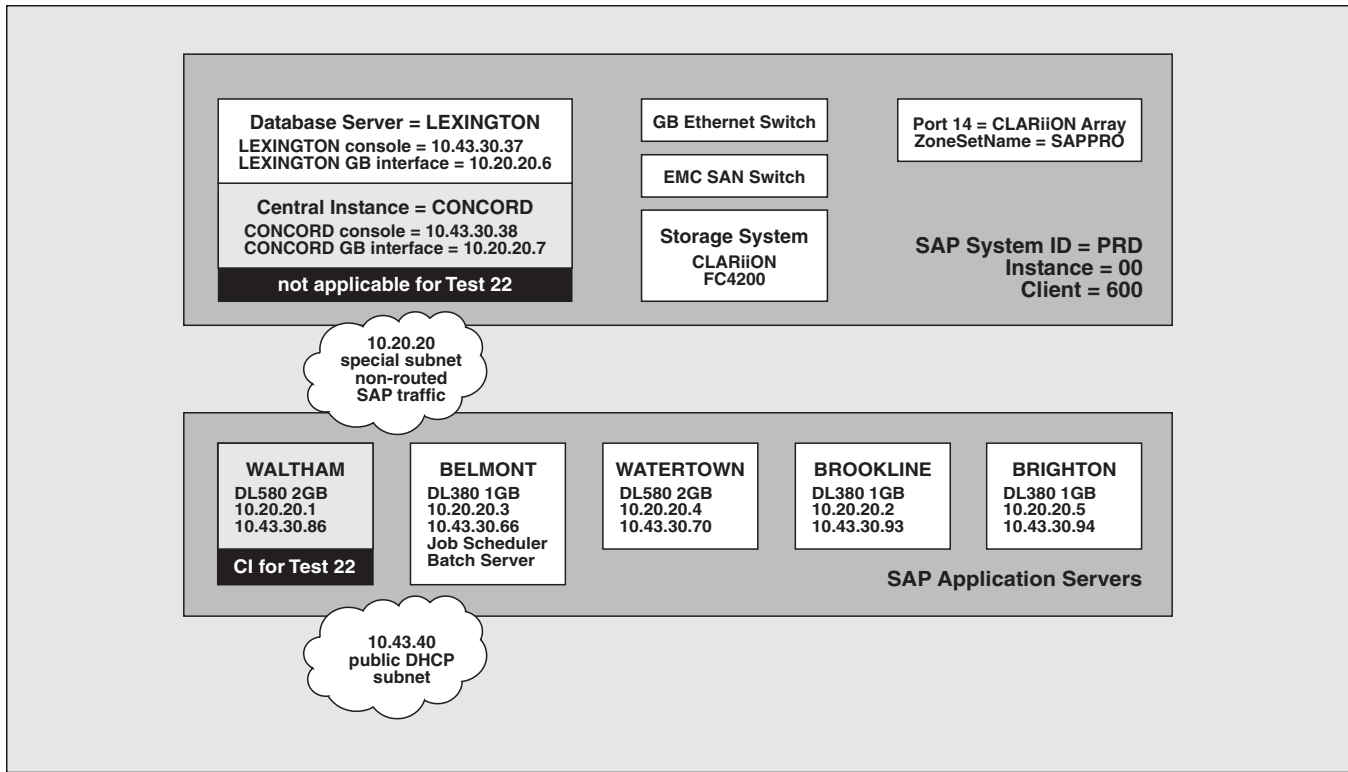
that had as many as four test script developers and twelve databases, along with SAP and/or hardware experts on standby. At smaller companies, members of the testing staff typically handle multiple roles. No matter what size your project is, however, the following people should be on the testing team:

- **Overall project manager (required):** This person is responsible for all aspects of the project, from justifying its business value to defining the test scenarios, as well as gathering and analyzing the test results, interpreting the final results, and presenting the findings and recommendations to management.
- **Test script creation team leader (required):** This team leader is also the person who will be in

charge of starting the test load once the project manager gives the go-ahead. On smaller teams, this person will create the test scripts.

- **Test run manager (required):** This is the go-to person for anything that can and will go wrong during the test run. In many of the projects I have been involved in, this role was assumed by the project manager. If you are considering having this role filled by your project manager, weigh the decision very carefully. The test run manager needs to devote his or her full attention to the test runs in order to monitor tests in real time and make decisions on whether or not to abort a test should a problem arise. If the test run manager is a member of senior management (as project managers often are) and there are time and priority conflicts with other responsibilities, he or she might not be available when needed.
- **One or more test run analysts (at least one required):** Test run analysts work with and support the test run manager to monitor the runs in real time and report any problem or abnormality. In some cases, there might be only one analyst, and it could be the test manager. In some very large tests that I have been involved in, there were so many things to watch that as many as half a dozen test analysts were needed, with each one of them responsible for watching as many as ten different specific test result areas.
- **Hardware and OS specialist (required part of the time):** This person takes care of all the hardware requirements and OS setup for the myriad servers and workstations needed for the test. Your current test bed most likely is not ready for this type of test, therefore you need this person's time and expertise, even if only part-time. The hardware and OS specialist may also be called upon to fine-tune the performance of the hardware and the OS as a result of aborted or unsatisfactory test runs. Of course, this person needs to be on standby in case of a hardware or OS failure while the tests run.
- **Database administrator (required part of the time):** The DBA is crucial to the test because the database always requires some fine-tuning between test runs. The challenge you face is that you *must* secure this person's prompt response to any issue or problem, or else you will incur significant delays — but there is usually not enough work to keep this person around full-time.
- **SAP Basis administrator (required part of the time):** Everything about the DBA in the previous bullet item applies to the SAP Basis administrator as well, almost word for word — except that this person is called upon somewhat more frequently, since he or she has to make changes to the SAP instance and other profiles as the results of each test run are evaluated.
- **LAN/WAN network specialist (optional):** To deal with network problems that arise while the tests run — such as domain name system (DNS) issues, network congestion due to an imbalanced Ethernet switching configuration, and of course the dreaded network hardware failure — you need an on-call network specialist. The LAN/WAN network specialist may also need to work with the hardware expert to configure the initial SAP test bed network, but that should not be a major effort.
- **Data backup and restore specialist (required initially, then optional):** This person may double as the hardware specialist. Refreshing the test data is a crucial operation, and the data backup and restore specialist sets up all the processes and procedures for the data refresh. Later, he or she will need to be available on a standby basis in case a major problem arises with the data refresh.
- **Storage specialist (required if applicable):** If your company uses specialized storage (such as EMC Symmetrix, CLARiiON, Network Appliance, or IBM Shark), and you plan on using such specialized storage as part of your test bed, then you need to have the appropriate storage specialists available during both the initial configuration of the test bed (which can be quite costly) and during the test runs (in case there are any problems with the specialized storage systems). These experts usually also possess the skills

Figure 2 Sample Architectural Map — Networked Servers



required to deal with the complicated SAN (Storage Area Network) switching fabric that their hardware uses. If you do not enlist storage specialists, you will need to secure the expertise of an SAN expert.

- **Test script creation analysts or consultants (optional):** For test script analysis, you might recruit the power users who helped create the test scripts or the hired consultants who fine-tuned your SAP CATT (Computer Aided Test Tool) or Mercury LoadRunner and WinRunner scripts, for example. Consider having both the power users and the hired consultants watch as tests are run. For the power users, watching tests can be a useful learning experience, as they may get a better picture of why and how excessive loads are put on the SAP system. For the hired consultants, being able to observe how test scripts are executed with real load on a large set of real data allows them to produce more effective and relevant scripts — an especially important consideration if, going for-

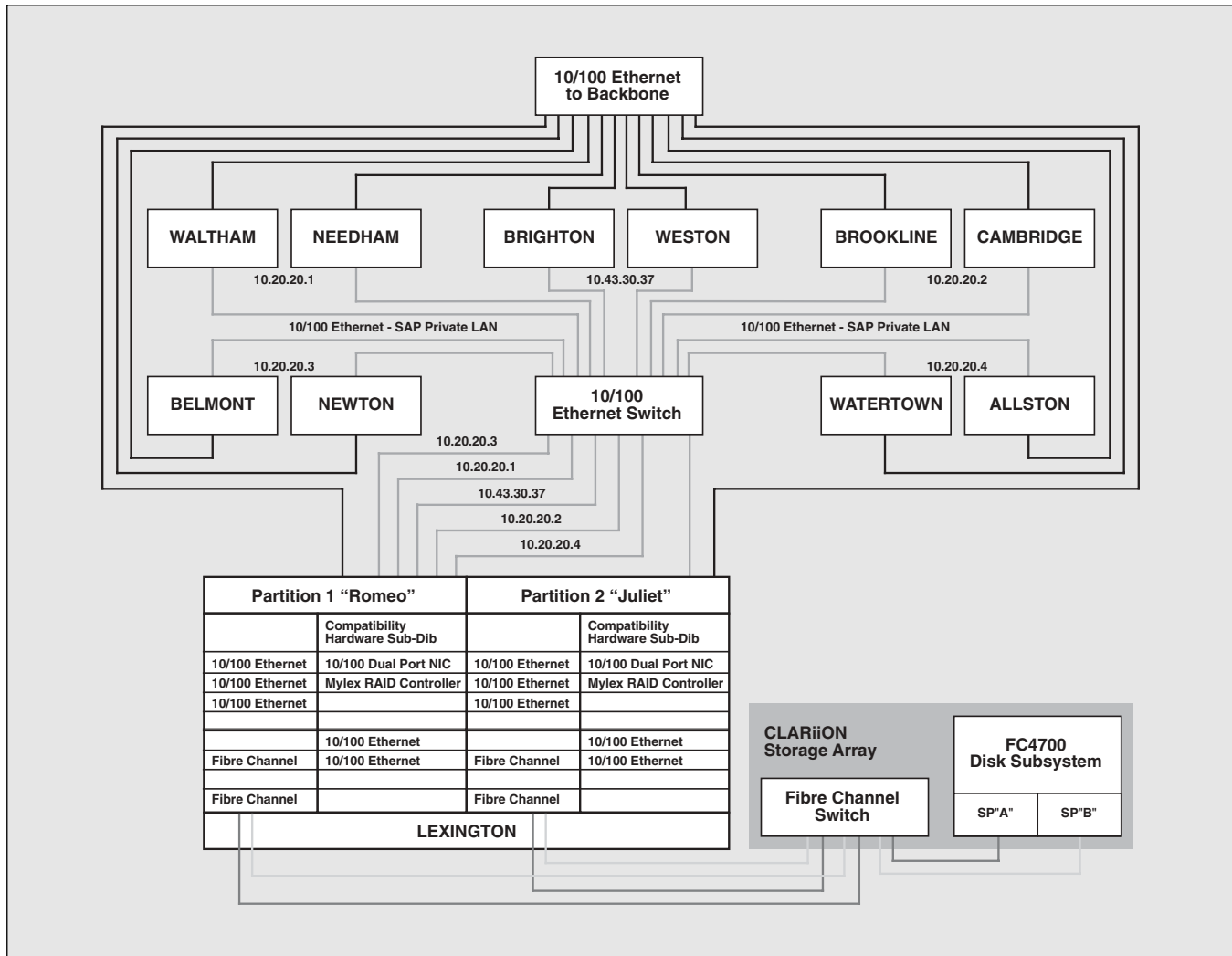
ward, your strategy is to have ongoing testing to support production.

Creating or Updating the Architectural Map of Your SAP Environment

With the initial planning accomplished, the first step in actually setting up your test environment is to create or update a detailed map of your current SAP production system architecture. Documenting your system architecture is essential in order to configure a test environment that properly represents your production system, but it also offers many benefits that extend beyond the tests.

If you don't know for sure how much RAM each of your application servers currently has or what esoteric parameters are contained in your SAP instance

Figure 3 Sample Architectural Map — Ethernet Switching Layout



profile, this is an excellent opportunity to obtain that critical information and make it readily available on your intranet. Further, examining a map of your environment provides an opportunity to identify and analyze weak points in your network architecture and/or your SAN switching fabric.

Even if you already have system architecture documentation, carefully review and update it. System documentation becomes outdated rapidly, and it's easy to let mapping slide. Take the time now to thoroughly update all maps — otherwise, your test results may not correlate to results that would occur on your production system.

Figure 2 is a sample architectural map that gives a high-level view of networked SAP servers in one hypothetical company. Note that this map indicates an alternative configuration to be used in one of the performance tests planned by this company. To see the effects of having an SAP central instance (CI) on the same machine as an application server, one of the application servers (Waltham) will serve as the central instance in Test 22. For that test only, server Lexington will not be the CI server, even though it is labeled as such on the map.

In **Figure 3**, you see a more technically detailed map for the same company, one that will be necessary

for planning and executing tests. It shows the Ethernet switching layout. When creating your own architectural maps, you should plan on getting to this level of detail for both your Ethernet switching layout and your RAID group disk layout. These are just two simplified examples, and they are by no means the only maps that you will need.

In-House or Offsite?

Your next major decision is to determine whether to conduct the tests at your own facility or at an offsite location. I've participated in both types of tests, and there are pros and cons to each approach.

Conducting tests in-house is easier than offsite testing and, if you already own the space and the equipment, is cheaper as well. Duplicating your production system can be easier in-house, but I strongly recommend that you find a way to segregate the test network's LAN traffic from the production network.

Obtaining data backups with which to populate your test servers is simpler in-house; it's also easier to summon people with the expertise required to solve the problems that arise. If you suddenly need the services of your DBA, for example, he or she is probably right down the hall. Also, downtime isn't wasted, since team members who are not immediately needed can simply go back to their desks.

But testing in-house isn't always best. For one thing, it has the potential to adversely affect users. Having the test system completely removed from your production system can help ensure that no one suffers any ill effects from testing. Offsite, your team can also devote its full attention to testing, without the distractions of the daily work environment. If you're testing new hardware or a significantly different configuration, starting fresh in an outside test environment can be easier than making major changes to an existing in-house test or development system.

However, if your SAP systems have lots of links

to other external non-SAP systems, then going offsite to test may not be very practical. To even partially duplicate the links to external systems that your production system uses might be too complicated. In addition, if your test data contains highly confidential information or if your test scripts contain highly sensitive business processes and procedures, conducting the tests anywhere but inside your firewall is not even an option.

I strongly recommend discussing with your hardware partners (or even with one of their competitors) whether or not to test in-house. In many cases, your hardware or software partners will welcome the opportunity to let you use their SAP Competency Center for testing. If you're considering upgrading your hardware or software, it's helpful to give it a test run using your own data before making any commitments or changes to a production system that works. Using a partner's SAP Competency Center also gives you direct access to their hardware and OS expertise (or even networking expertise), which can help you make the most of your system architecture and configuration. Whether or not there will be a cost associated, and what that cost will be, depends on several factors, such as your relationship with the partner, whether or not they expect to be able to sell you new products, and so on.

Another place to go for offsite testing is the data center provided by your disaster recovery company. Companies such as SunGard and Comdisco can set up entire data centers where you can conduct tests, though the costs of doing so may or may not be part of your contract with them. It is not unusual, however, for business continuance contracts to contain clauses allowing the client to come in for a disaster recovery drill once a year — you can use that clause to have a mirror image of your production environment prepared for testing purposes.

There are no hard and fast rules for determining when the advantages of outside testing outweigh the costs. Each case must be judged on its own merits. **Figure 4** summarizes the conditions for which each choice is optimized.

Figure 4 *In-House vs. Offsite Testing*

Consider In-House Testing If You:	Consider Offsite Testing If You:
<ul style="list-style-type: none"> <input checked="" type="checkbox"/> Have many links to external systems <input checked="" type="checkbox"/> Have confidential data or business processes to protect <input checked="" type="checkbox"/> Have the available space <input checked="" type="checkbox"/> Have most of the equipment already in place <input checked="" type="checkbox"/> Need constant access to your in-house expertise 	<ul style="list-style-type: none"> <input checked="" type="checkbox"/> Need to set up lots of new hardware <input checked="" type="checkbox"/> Want to evaluate new hardware <input checked="" type="checkbox"/> Are not concerned about what happens to your test data <input checked="" type="checkbox"/> Want to avoid major disruption to your in-house systems and users

Figure 5 *SAP Transactions That Are Useful for Testing*

SAP Transaction	Description	Usefulness in Testing
ST03 and ST03N	Workload analysis	Capture key performance data of SAP application servers.
ST04	Database performance analysis	Get a quick snapshot of database performance.
SM50	Work process overview	See if any process is incurring an unduly long wait, a sure sign of something failing or a performance bottleneck.
SM51	Instance overview	Determine whether all application servers are present and participating in the test run.
DB02	Database performance, tables and index	Determine whether there are problems in the database, such as missing indexes. Check to see if the database has grown because the test runs have added transaction data.
AL08	Current active users in system	Check to see if the appropriate test users are active and if any unauthorized users are somehow logged in.

Evaluating and Selecting Testing Tools

The testing and measurement tools available to you are numerous, and a typical test might involve several tools at once. Some of the most appealing tools are those that are available at no additional cost. You will probably want to license some other tools because of the ease with which they allow you to capture test data.

Tools Available at No Additional Cost

The tools available at no additional cost include SAP transactions, OS features, and tools you already have that are not designed specifically for testing.

- **Test measurement tools:**
 - Several transactions within SAP itself capture and display SAP performance and execution data. These include the SAP transactions detailed in **Figure 5**, which are the ones that I

use most often. These SAP-provided CCMS tools should be familiar to most Basis administrators and consultants.

No matter what additional tools you use, data collection for your test should in all cases include results from SAP's own performance monitoring transactions, which provide real-time feedback on the system's behavior. If something goes wrong, you'll know about it right away. You won't have to wait until the end of the run, needlessly incurring costs and time delays.

- If you're running Windows NT or one of its successors, you can use the Windows PerfMon feature to capture system indicators and performance data. PerfMon can capture a wide range of system performance data such as CPU utilization, swap file activities, and network traffic load — all of which can help you to evaluate test results.
- If you use Unix, the "sar" tools and a whole range of system performance tools are also available as a part of the operating system.
- **Test script creation tools:** You may also make use of CATT and eCATT (extended Computer Aided Test Tool), SAP's test script creation tools. Your existing test scripts (those prepared for go-live) likely were done using CATT, so it would be natural to use CATT to further enhance these scripts if they were to be reused. eCATT is the successor to CATT, and it is used with the SAP Web Application Server (Web AS) to test not only SAP native ABAP transactions, but also the Java transactions that can be developed in the Web AS. Since eCATT is still quite new, I have not had much experience with it as of this writing.¹

¹ For more information on eCATT, see Jonathan Maidstone's articles "Extend the Range and Reduce the Costs of Your SAP Testing Activities with eCATT" and "Ready to Start Leveraging eCATT? Take Your Existing CATT Tests with You!" in the January/February 2003 and September/October 2003 issues of this publication, respectively.

- **Other useful tools:** Other tools in the "no additional investment" category that are useful during testing include several that are not directly related to measuring performance. For example, you will likely use the data loaders that are part of your database to load and reload test data. To facilitate the necessary data refreshes between the test runs, your backup software and hardware will be used heavily. The Microsoft Office suite (specifically Excel, PowerPoint, Word, and Access) is useful for organizing, documenting, and presenting test results and methodologies. (You can use similar or competing products for these purposes if you do not have Microsoft Office.)

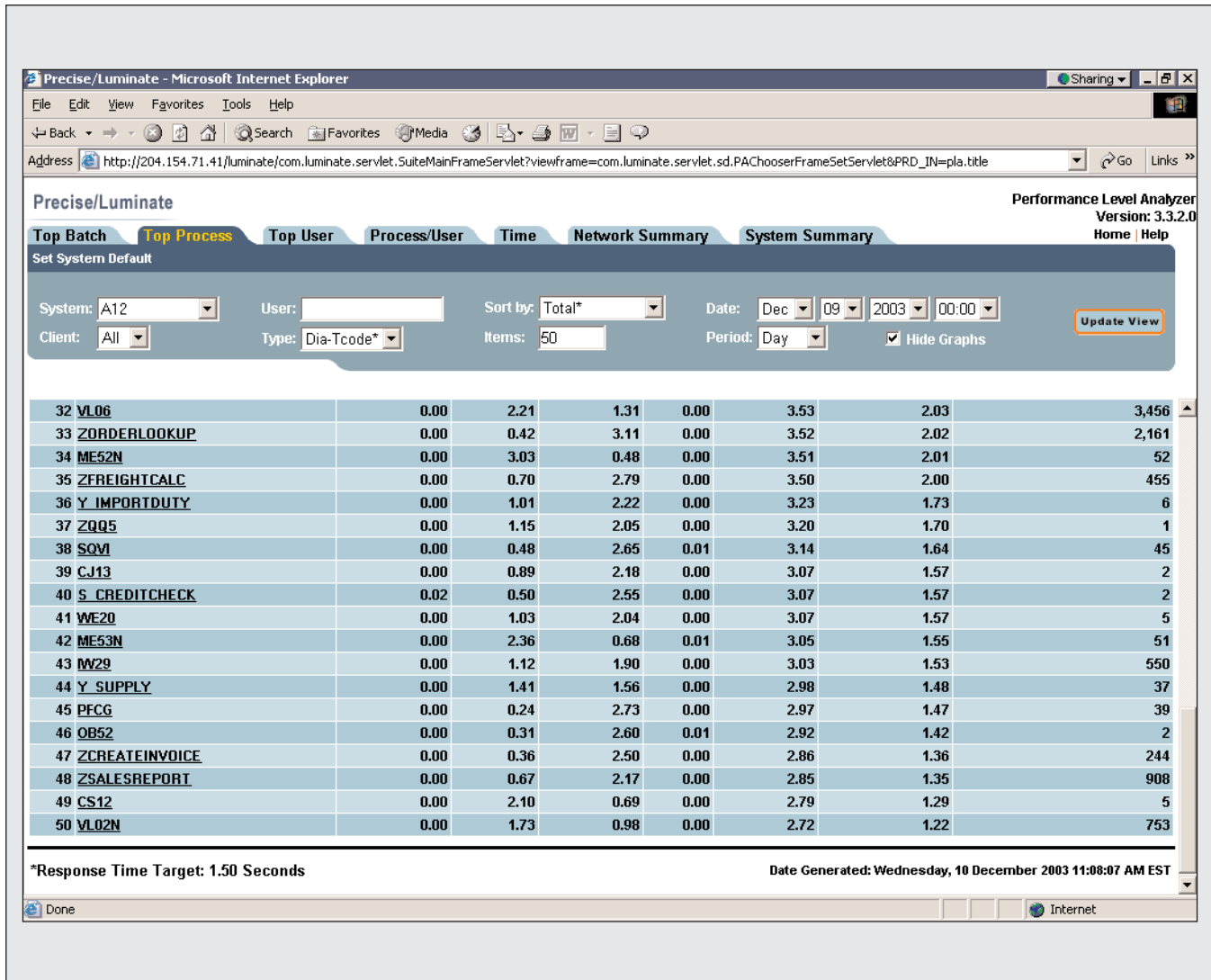
Licensed Tools Available at Additional Cost

Here are some tools that may be worth licensing:

- **Test measurement tools:**
 - VERITAS Luminate is now part of the VERITAS I³ for SAP product suite, and it is an excellent tool for detailed service-level reporting. I have used it many times to conduct rigorous performance testing at several different installations. It has real-time data capture capabilities that nicely complement CCMS, and its Oracle-based data warehouse gives you the capability for further analysis of all aspects of your SAP system's performance. If you can only afford one licensed tool, I recommend VERITAS Luminate.
 - VERITAS Insight is another valuable tool for measuring end-to-end application performance response time. What VERITAS Insight offers over the "free" tools is the ability to rapidly identify the source of performance degradation. This capability can be a real time-saver. The challenge is to justify the expense of licensing this product; you'll need to weigh the cost against the time saved in the analysis process.

Figure 6

Luminate View of Top Processes



- Several tools in the VERITAS Indepth product suite provide more detailed data collection and results analysis.

While the licensing costs for such tools can be considerable, the tools may prove invaluable if performance testing becomes a standard practice to support your company's business goals. In many cases, VERITAS may be willing to provide you

with a demonstration or loaner license to use in one or more test runs, with the hope that you will subsequently license its products. It is worth exploring this option with the VERITAS sales and marketing folks.

Figure 6 is a Luminate view of a performance matrix, showing "top processes" (transactions that required the most processing time).

Figure 7 Luminate View of Application Servers

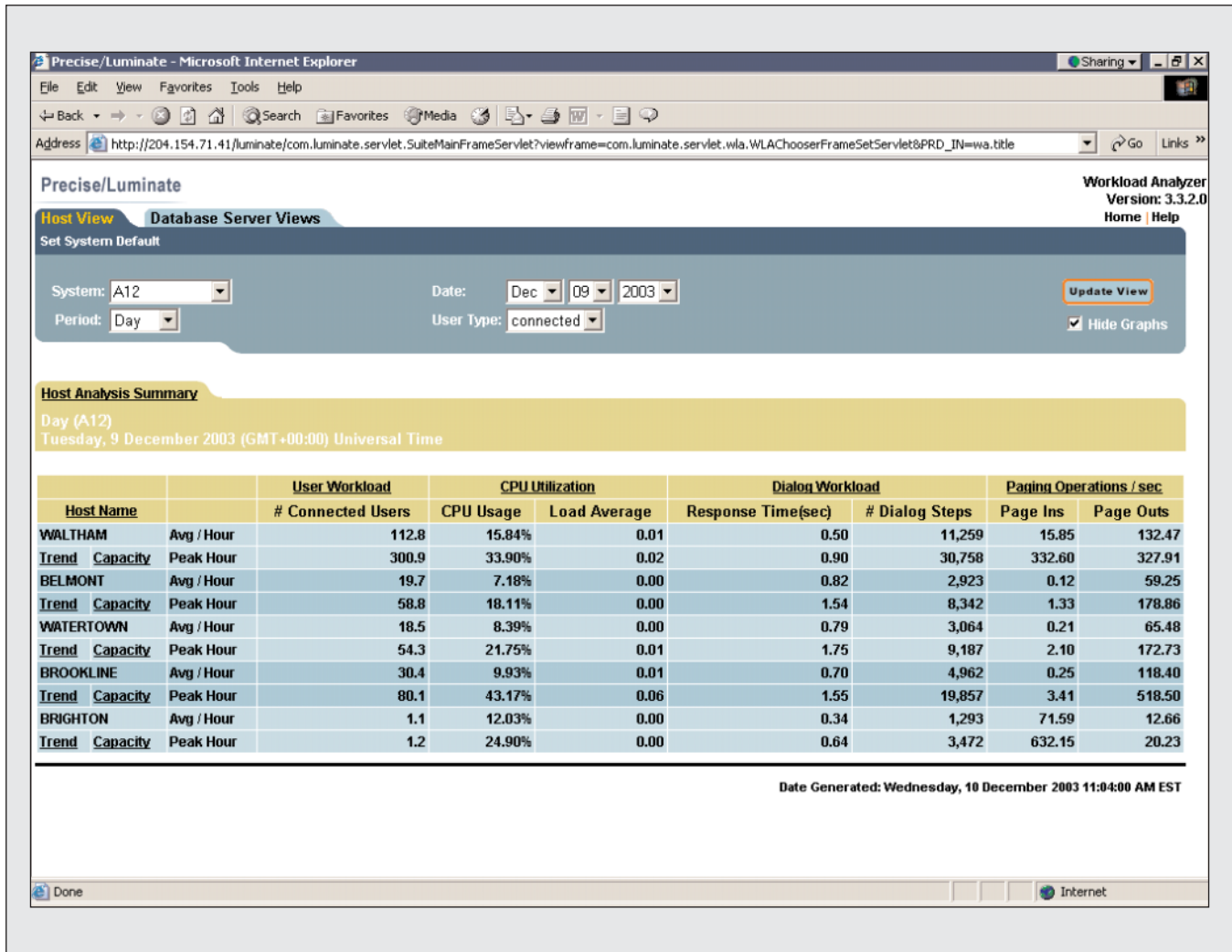


Figure 7 is a Luminate view of application servers that shows, for each server, the number of connected users, the load average, and so forth.

- **Test script creation tools:** Mercury LoadRunner and WinRunner are effective tools for creating and executing test scripts and scenarios. I have had good results using them to streamline the test development phase. One of the strengths of LoadRunner and WinRunner is the ability to capture actual screen and keystroke activities as generated by an SAP user. This allows you to drive a very realistic load through the SAP systems you're testing. These tools have some drawbacks,

however, not least of which is the cost of licensing them, which can be quite high. In addition, they require an expert to create the script, and rather beefy hardware to generate the loads.

- **Other useful tools:**
 - As mentioned earlier, you may need to schedule your batch jobs to execute as part of your test runs. Therefore, you may also benefit from implementing job scheduling software, such as Tivoli Workload Scheduler (formerly Maestro) or Computer Associates AutoSys.

- A screen capture tool such as SnagIt can also be very helpful throughout testing. You can use it to collect images that will be needed during documentation. SnagIt may already be in use at your company, or you can license it for a reasonable fee.

Test Scenarios

Whether you are setting up a business scenario or a technical scenario, you need to determine exactly what transactions or processes must run with the new configuration of the test system, and for how long, in order to determine the impact to your overall SAP system.

You don't have to test every single SAP transaction to execute an effective test. You do, however, need to test an authentic cross-section of transactions and processes in order for the test to be realistic and meaningful. One of your first steps, then, is to define your company's "Business Day Profile," which is composed of the commonly run SAP business transactions in a typical business day. If you're testing a business scenario, your Business Day Profile must include the batch jobs that are regularly run to support the business process you are testing.

The Business Day Profile

The easiest way to capture a list of regularly used SAP transactions is to use *ST03* (Workload Analysis) to choose one or more application servers, review either today's workload or the previous day's workload, and click on the *Transaction profile* button to run the Transaction Profile Report. The report will enable you to see all transactions by code and all batch jobs that were executed for that particular time period.

But this is just a starting point, as you will need to review these Transaction Profile Reports with your functional team and your business users in order to

decide which transactions should be part of your company's Business Day Profile. In my experience, this is a highly interactive and iterative process. The result should be that your test script includes all of the following:

- The most frequently executed transactions during a business day
- The transactions that consumed the most system resources during a business day
- The agreed-upon order in which these transactions will be executed
- The agreed-upon duration of the test run

✓ Tip

While it would be nice to run the test for an entire day, it is highly impractical to do so. I have found that results from a test run of one or two hours, which includes all relevant transactions and batch processes, can be extrapolated to provide a good picture of your SAP system's performance.

Once you know which SAP transactions are executed in the course of each day, identify the top 15 (the ones most frequently executed). The response time for these transactions constitutes the performance baseline, which will be especially important in helping you get a clear picture of how your overall system will perform under the increased workload of the new business or technical scenario you are testing.

Establishing your most commonly used SAP transactions is just one part of defining your test scenario. You should also consider batch jobs. Which are your longest-running and most expensive batch jobs in terms of processing capacity? Whatever your major batch jobs, they should be included in the test scenario to provide an adequate picture of system capacity and response.

Finally, one of the more complex aspects of defining a test scenario using a Business Day Profile is incorporating the links your SAP system has to external systems. Are you linked to TIBCO, EDI, webMethods, Actuate, Business Objects (formerly Crystal Decisions), mainframe computers, SAP Java Connector (JCo), Remote Function Calls (RFCs), or the like? Are those systems involved in the business or technical scenario you are testing? If they are, you have to find a way to either actually connect to those external systems during the test run or, at the very least, find a way to simulate connections to them.

What Are the Effects of Varying SAP Instance Profile Configurations?

SAP instance profiles define how resources are allocated on a particular SAP application server. Changing the values of instance profile parameters, adding new profile parameters, or using entirely different instance profiles can yield impressively different performance results. I have been involved in projects where the right combination of the central instance (CI) with the database server in an appropriately configured system has resulted in a dramatic increase in performance and capability. Clearly, making changes to SAP instance profiles will require the involvement of highly skilled SAP system tuners and/or the SAP EarlyWatch team, but the end results may be well worth the effort.

I would like to be clear that if you want to test using various SAP instance profiles, you must maintain the *same* hardware platform, operating system, and database. Otherwise, you will be introducing too many new parameters, and the resulting improvement or degradation in performance may be harder to analyze.

While a complete discussion of the possible configurations of instance profiles is outside the scope of this article, I highly recommend that you fully understand the performance benefits of varying your CI and database configurations.

Testing for Completely New Conditions or Environments

If you have the business requirements and the means to do so, you can test scenarios that were not previously feasible, such as a much larger database server with more CPUs and memory, a larger number of application servers with various memory configurations, and so on. Just keep in mind that creating these new operating conditions and environments can be very costly, so the business benefits must be able to justify the additional costs and efforts of this type of test.

In conducting tests like these, you should look not only at how the new configuration will affect your current operations, but also at how it will function under probable conditions in the future. For example, perhaps you are testing a technical scenario, such as altering the configuration of your database server. If it's likely that your online orders will significantly increase in volume in the short term, then it's important to include that new volume in your test, even though your main goal is testing your hardware, not testing a new business process.

Leveraging Your Existing Test Routines and Scenarios

It's likely, especially if you've been running SAP for a while, that your company has existing test routines and scenarios. These may be useful for your current testing purposes, and it's always an advantage not to have to reinvent the wheel. Carefully consider, though, whether these routines are applicable to your current environment. If the routines were developed for use during go-live, they may be very outdated and require serious modifications before being used. Examine whether they make sense for your current hardware and software configuration and whether they properly reflect how business is currently conducted at your company using SAP, especially in light of your newly defined Business Day Profile.

Setting Up the Test Data Capture Environment

Clearly, you will be observing a test in real time as it is running, but to allow for more detailed analysis once the test runs are completed, you will need a way to capture all the results. I strongly recommend investing in the necessary hardware and software to capture the test data.

The basic data capture equipment can be workstation-class machines running workstation-class operating systems, although you could use a server-class OS (such as Windows 2000 or Linux/Unix) as well. My workhorse data capture machines have typically been Pentium 3 or 4 processors running at 500 MHz and above, with 512 MB of RAM, 10 GB or larger hard disk, and Windows 2000 Server or Advanced Server. Thus this discussion will be somewhat skewed toward Microsoft Windows.

At a minimum, you will need to deploy the following equipment in your data capture environment:

- A machine to capture the Windows PerfMon data from the database server and SAP application servers. In fact, I recommend two machines in order to have redundancy, since it would be extremely costly to rerun the test should your data capture machine fail.
- One or more machines to serve as the real-time SAPGUI-based monitoring console, where you can quickly and easily take screenshots of key transactions such as *ST03*, *ST02*, *OS06*, and so forth, while the test is being run. Remember that things happen very fast during a test run, so having more than one person and more than one machine observe the run and make screen captures in real time can be highly beneficial later when you sit down to review the test results.
- An appropriately configured Oracle server, if you use a third-party tool such as VERITAS Luminate or the VERITAS I³ for SAP product suite. In the past, I have used Luminate servers with dual

Pentium processors running at 700 MHz, 2 GB of RAM, 60 GB or more hard disk, and Windows 2000 Server (please check with VERITAS for the latest hardware requirements). You may need more than one server if you choose to run the complete VERITAS I³ for SAP product suite. The investment in hardware, software, and setup is considerable, but the results will be well worth the cost and effort, as you will have a real treasure trove of information to work with.

- A machine for using sophisticated network monitoring tools such as HP OpenView, if you want to monitor network traffic.
- A machine to capture runtime statistics from your third-party batch job scheduler (if you use one). Though you can use the PerfMon capture machines for this purpose, they can be overloaded with large amounts of data, and some critical information may be lost. If you decide to capture data on the server that hosts your third-party batch job scheduler, you will need to make clear to the server's owner that substantial log files will be written to it.

Defining Hardware Configurations

The most important consideration when defining the various hardware configurations for SAP performance testing is whether your test environment is truly a mirror image of your production environment. Unless your test and production systems are nearly identical in disk capacity, size of the database server, and number of application servers, you do not have a suitable environment to conduct the types of tests discussed here.

Results obtained on disparate systems cannot be completely reliable. To obtain reliable results, you need to have approximately the same amount of disk space in your test environment as in your

✓ **Tip**

Once your test environment is as similar as possible to the production environment, you will need to address the naming conventions of your test SAP instances. This is especially true if the test instances are in-house and on the same network. Keep in mind that you can save a great deal of time by choosing a naming convention that will minimize the work involved in preparing the test data. For example, if your production instance is called E54, calling your stress test instance T54 would save you time in renaming the data for testing, since the last two digits of the instance remain the same. If you have more than one production instance to test, a naming convention like this one is especially helpful, as it allows you to easily change the instance name in the test data set.

production system. If your test and production environments are not similar in terms of their storage systems, I advise upgrading your test environment before proceeding. You may have to ask for a bigger budget, rent or borrow the needed equipment, or find some other way to deal with the extra expense of the additional storage. One way to manage the expense is to use lower-end and/or slower storage subsystems for your test environment. The results will be reliable as long as the two systems are of similar size, since you know that the results achieved with the lower-end subsystems would only be better on production-class storage. For example, you can have lower-end disk subsystems such as EMC CLARiiON substituting for EMC's higher-end Symmetrix, or you can have

a run-of-the-mill RAID 5 configuration with lots of disk cache enabled.

Creating a perfect replica of the production system is a big problem for many companies. It takes a considerable budget and a considerable amount of time and effort to create such a replica, and time and money are always in short supply. It's especially difficult if your database is large (i.e., anything over 300 GB) and your storage subsystem is a high-end one such as EMC Symmetrix or IBM Shark. Having a spare system around for testing purposes is usually viewed by the keepers of the budget as an unnecessary luxury. (For a discussion of how you might justify the cost, see the sidebar below.)

The Business Benefits of a Test Environment That Mirrors Your Production Environment

- Disaster recovery:** Organizations with well-developed test environments that mirror their production environment are far ahead in disaster planning and recovery. If disaster strikes, the test system can take on production processing relatively smoothly, providing invaluable business continuity benefits. A test environment that can double as a disaster recovery system may justify to management the high cost of duplicating the storage capacity and processing power of your production environment.
- A safe environment for testing upgrades:** Once test environments and testing software are established, you have a comprehensive means of testing upgrades to new SAP releases and functionalities without having to fiddle with your live system.

If you haven't executed SAP tests since go-live, your test environment may not only be quite different from your production environment, it may be completely unsuited to the types of tests discussed in this article. Often, only a "mini" test environment is set up to handle SAP functional testing, and stress tests are performed on the actual production system prior to go-live. If that was the case at your organization, you may have a lot of work to do to get your test system caught up with your production system.

The Need for Multiple Configurations of the Test Hardware

Technical testing generally requires multiple hardware configurations since, by definition, you're testing the differences in performance between different configurations. For example, you might be testing the effects of adding more RAM or CPUs to your database server configuration, having a dedicated server for processing batch jobs, or having a dedicated report server. Or perhaps you are testing different combinations of the CI and database. If you want to analyze different parameter values of instance profiles, you also may need multiple hardware configurations.

Whatever the details of your test, preparing your test environment includes carefully defining each hardware configuration to be tested. You also need to know how many configurations there will be so that you can plan your budget and time frame accordingly.

Setting Up Your Test Bed

Setting up your server test bed requires careful preparation. In addition to duplicating the hardware and network configuration on your production network as much as possible, you should plan on adding more hardware and network capacity or reconfiguring your existing hardware to see if these new configurations can improve the overall performance of your production system.

Based on my experience in a variety of performance testing projects, here are a few rules for setting up the test bed:

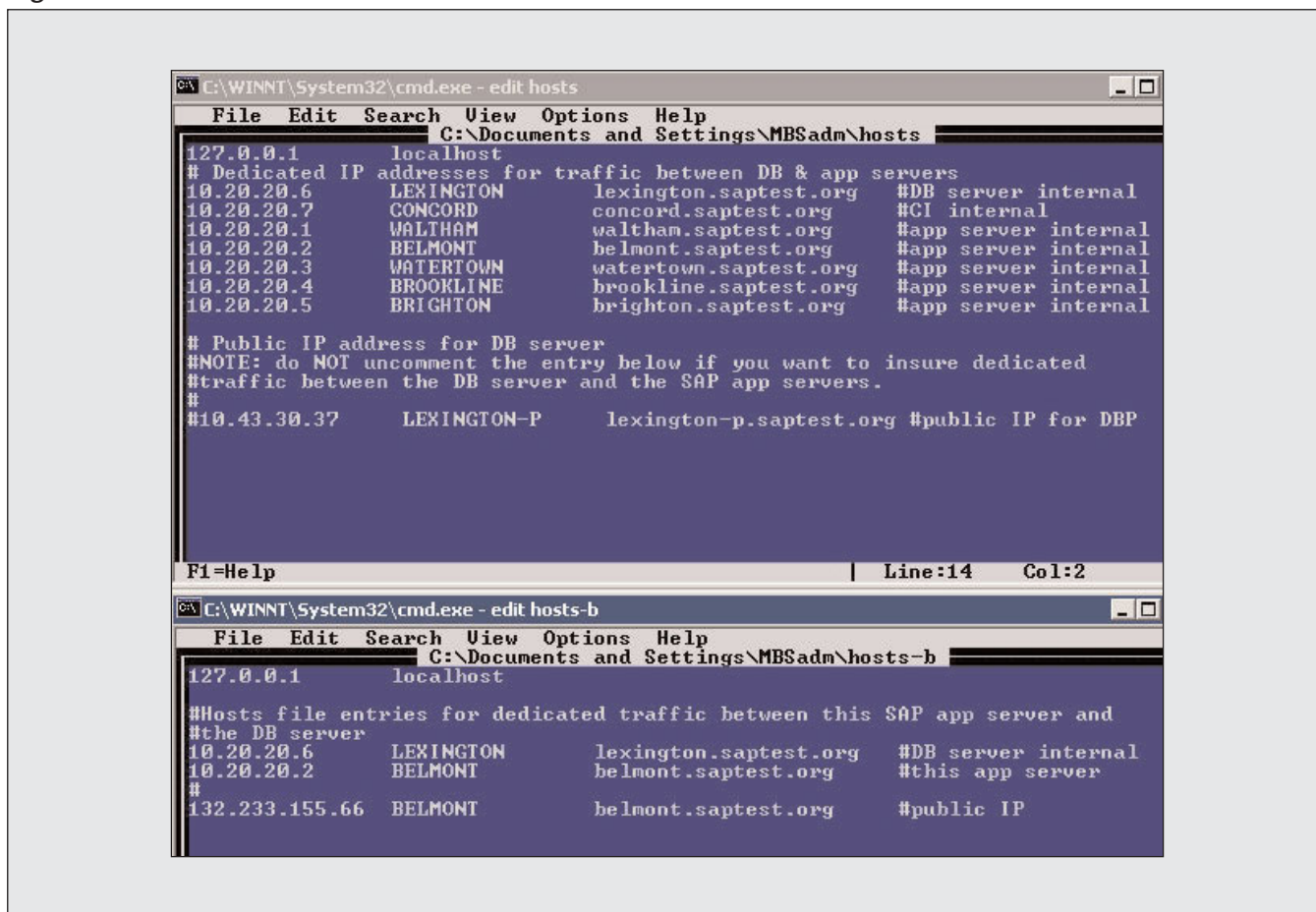
1. **Dual Ethernet is a must.** According to SAP's configuration guidelines, all application servers must be able to communicate with the database server over a private network. This rule requires application servers to have a second Ethernet card (100BaseT or Gigabit) to communicate with the database server. I have followed this rule even in a switched Ethernet environment in order to eliminate any question or doubt about network latency. (Refer back to Figure 3 for an example architectural map of a switched Ethernet.)
2. **Private IP addresses are a necessity.** This rule is an extension of Rule #1. Using private IP addresses for the private network between application servers and the database server, you can completely separate the private network traffic from the general SAPGUI and other network traffic.

✓ Important!

If multiple hardware configurations are involved, you will need to plan for the reinstallation of SAP, the database, and the operating system, all of which can be very time-consuming. In addition, these tasks can involve complications, such as the need to schedule downtime so that more CPUs or RAM can be added. If you are in a Windows NT/2000 environment, you can use a tool like Norton Ghost to quickly restore a server to its original configuration, but the key element is to be able to refresh your test database data very quickly. See the section "Efficiently Executing Multiple Test Runs" for details.

Figure 8

Hard-Coded IP Addresses



3. **Add several more Ethernet cards into the database server.** Preferably, you will add one Ethernet card per application server into the database server (it doesn't matter whether the server is dedicated or not). I have found that this technique increases performance fairly substantially under certain conditions, such as when there is heavy database I/O, when an application server is dedicated to running batch jobs, or when login groups are used to segment user classes. If heavy database I/O is required, having a dedicated network channel for each application server significantly alleviates traffic bottlenecks. The downside, of course, is the increased cost — in addition to the extra network cards, you will need to support the additional Ethernet ports with a larger switch, which could be fairly expensive.
4. **Hard-code the IP address in *lmhost.sam* and in the host files for traffic segmentation.** This technique only works if you have multiple Ethernet cards installed in the database server and each Ethernet card is dedicated to an application server (Rule #3). It essentially guarantees that network traffic between application servers and the database server will be routed to the Ethernet card to which the hard-coded IP address is assigned, since no DNS translation will be required. See **Figure 8** for an example of hard-coded IP addresses for multiple Ethernet cards installed in a database server.
5. **Load up on cache memory if your test storage environment is less expensive than your production storage.** If your SAP environment uses specialized storage subsystems, such as EMC

Symmetrix or IBM Shark, and they cannot be duplicated in your test environment, you will have to make certain adjustments to your server test bed. For example, if you have to use EMC CLARiiON instead of Symmetrix, you should try to obtain the largest CLARiiON system possible, one with multiple frontend processors and large amounts of built-in disk cache. (This is where a good relationship with your hardware and/or storage partner will pay handsome dividends!)

Obtaining Meaningful Test Data

You know that your tests are only as good as your data. Because you are conducting a performance stress test and not a functionality test, you can assume that the SAP software is running correctly and yielding the appropriate data as a result. But how do you make sure you have obtained meaningful *test* data?

Just as a test hardware configuration that mirrors your production system gives the most reliable results, using the entire production database for your test data is the best way to get realistic and meaningful results. A subset of the production data cannot reflect your Business Day Profile. There are always exceptions, of course, but I find that there is no substitute for using the entire production database as the basis for your test data.

Obtaining, Preparing, and Loading Test Data

There are many ways to make a good copy of your production system database, but by far the easiest and surest method is the database system copy method or the homogeneous system copy method, which copies *everything* over.²

² For a complete guide to performing a database system copy on Windows, please refer to Giovanni Davila's article "Database System Copies Made Easy — A Guide for Copying an Entire R/3 System to a New Windows Platform" in the May/June 2003 issue of this publication. For details on performing a copy on the Unix platform, see Bert Vanstechelman's article "A Homogeneous System Copy in 60 Minutes? It Can Be Done!" in the January/February 2002 issue.

Another approach is to export the database and then import the data back into a newly installed "starter" database, similar to how you installed your first SAP system. The trick, however, is to be sure to adjust the .R3S file(s) to accommodate the larger sizes of the tablespaces on your production system.

Before you load the data obtained from your production system into your test system, you will need to change the SID, or instance information, so that you do not have a duplicate SAP instance on your network. You will also need to perform several other tasks — in addition to cleaning up the database table, you will need to adjust background jobs, operation modes, the spool server and printer, system profiles and parameters (if needed), and perhaps even more, depending of course on your unique environment.³

You will also need to make a decision regarding the SAP license:

- If your performance test project is a one-time operation, you will need to check with SAP to be sure that you are in compliance with the terms of your license agreement. You should be able to use the temporary 30-day license that SAP grants to all its customers. Again, please check with SAP regarding this matter.
- If you plan on performing ongoing testing as part of production support, then you will need to apply for a new SAP license key and relicense your test servers.

Performing a Complete Backup of Your "Gold" Copy of the Test Data

The "gold" copy of your test data is the one that you will use to refresh your database server after every test run in which the database has been significantly altered. If you do not have a reliable backup of the test database, you will need to repeat the time-consuming task of cleaning up the data from

³ See the system copy articles mentioned in footnote 2 for details on performing these tasks in Windows and Unix.

the homogeneous system copy, and that will considerably slow down your test schedule, not to mention introduce potential errors and omissions into the cleanup process.

In addition to having a reliable backup mechanism and procedure for the gold copy of your test data, it also helps immensely if the backup system is fast. I've seen the restore of a 60 GB database take over eight hours because the backup and restore was performed using a single digital linear tape (DLT) drive, and the second tape had to be manually mounted by a tape operator before the restore could be completed. Under these conditions, my team was essentially limited to one test run per day, and our project took three times longer than necessary.

I recommend that, whenever possible, you use hard disks instead of tape subsystems to back up and restore test data for the purpose of refreshing your database. If your database server runs Windows 2000, I propose one of the following alternatives, both of which are inexpensive:

- **Alternative 1:** Use a FireWire-based external disk subsystem. You can purchase a 250 GB capacity subsystem, including the necessary controller card, for well under \$1,000 (USD). The FireWire controller card will be installed in one of the slots on your database server, and the drive unit will be attached externally to the server. The resulting data transfer is probably the fastest that you can achieve for this minimal investment — I have been able to back up and restore 80 GB in about 2.5 hours. You can substitute a USB 2.0 controller card for FireWire, but I have not had direct experience with that.
- **Alternative 2:** Use an inexpensive data backup server equipped with high-capacity yet inexpensive Advanced Technology Attachment (ATA) disks. The big benefit of this approach is that you can back up multiple servers to an ATA disk. A simple Intel server equipped with multiple 160 GB or 250 GB ATA disks can be built for under \$2,000, excluding the Windows 2000 Server license. To avoid network traffic, you

can attach the data backup server to the private Ethernet network that joins the SAP application servers and the database server. With this configuration, you may be able to achieve a backup and restore of 80 GB of data in roughly 3.5 hours. (Please note that all these figures are estimates.)

Both of these solutions should work in a Linux/Unix environment as well, though I have not had direct experience with them under Unix.

Efficiently Executing Multiple Test Runs

A reality of testing is that you will spend about 20 percent of your time refreshing data between tests, another 40 percent of your time running the tests, and the remaining 40 percent troubleshooting and fixing the problems that have arisen in each test run. Therefore, consider creating multiple test beds to speed up your testing schedule, so that all testing doesn't have to grind to a halt when it's time to refresh data.

For example, if you have an SAN environment or an EMC Symmetrix environment, you could have one or more application servers acting as backup database servers, so that you can keep multiple copies of a database on the Symmetrix or the SAN-based storage subsystems. This technique should result in data refresh times of as little as one or two hours. Activating backup database subsystems is not a trivial task, though you can automate this activity with scripts. If you have high-end storage subsystems available, you certainly should try to take advantage of them using this technique.

Involving External Resources

The final step in setting up your tests and test environment is determining which external resources, if any, to involve during testing. In many cases, you will need hardware or software vendors, including SAP itself, to be present during configuration changes so

that you don't void warranties or service agreements. This is particularly true if you have EMC Symmetrix as your storage system.

It can be advantageous to involve these experts anyway, even if their presence is not required, given their access to in-depth test results conducted at their own test sites during product development. The downsides of involving hardware and software vendors are that there may be a cost involved (which can be significant), and your project's schedule will be subject to the availability of the vendor's experts, who may be busy with other projects.

It may also be advantageous to involve your consulting partners during testing. If outside consultants have been deeply involved in the configuration and tuning of your system, their experience with your company, as well as their view of best practices with other clients, could be invaluable in your testing effort. Consulting partners can also be very helpful in the delicate task of interpreting and presenting test results.

Here are some of the external resources that you should consider engaging in order to have a successful performance test project:

- **Storage system experts:** If you have EMC, these experts are a necessity. They can also be useful in achieving measurable improvements to your database I/O.
- **Hardware tuning experts:** If you have a very large and complicated database server, it may be very helpful to involve that vendor's performance-tuning experts. They have compiled best practices during their SAP hardware certification runs and through their efforts to achieve high watermarks for their servers, and they can help you achieve performance gains that might even help pay for the project — whether through productivity improvements or a delay in the acquisition of a new server.
- **Test script creators:** The learning curve for Mercury LoadRunner and SAP eCATT can be

quite steep. Serious productivity improvements can be achieved if you bring in seasoned test script creators to create the first few scripts and train your staff to use these script-creation products. You will need to weigh the cost of hiring these experts against the potential productivity gains and time-savings that can be achieved.

- **Test run analysts:** Unless you have previously been through a performance-testing project and witnessed one or more test runs, you may be surprised at the complexity and the chaos that a test run can create. Having an experienced hand on board is not a bad idea if your performance test is a highly complex one. Part of the analyst's agreement with you should include the obligation to share his or her expert knowledge with you and your staff.

In the end, involving external resources requires a careful and realistic analysis of what capabilities you have in-house, what skills or specialties are necessary to ensure meaningful and useful tests, and what options your budget and schedule allow.

✓ Tip

In my experience, assistance from outside experts may cost a lot of money, very little money, or no money at all. It depends on how you structure the relationship: if the hardware or software partner can sell you something once the testing project is complete, then they should be willing to "invest" in the test project.

Conclusion

By basing your test preparations on the principles, ideas, and suggestions set forth in this article, you can

be confident that your SAP testing is poised to provide results that are both accurate and meaningful. You have the right people, the right tools, the right data and test environment, and the right testing strategy. All that remains is to execute the tests, analyze the results, and present the data to management. That's what we'll get into in the next installment of this two-part article series, which presents best practices and strategies for test execution, analysis, and reporting.

Tim K. Nguyen is Managing Principal of Orchid Consulting Group, Inc., a consultancy that focuses on SAP and e-Business systems performance and business continuance issues and solutions. With over 20 years of experience building successful IT professional services practices, he oversees most of Orchid Consulting's projects worldwide. He regularly conducts SAP knowledge transfer and guides SAP market strategy for leading product and service providers, in addition to performing his hands-on technical work. A graduate of the University of Texas at Austin, Tim has authored two computer-related books published by Simon & Schuster and QUE Corporation, as well as numerous articles in publications such as Computer Reseller News, PC Week, and Lotus Magazine. He is fluent in English and French, and is also comfortable working in Italian. Tim can be reached at tim@orchidgroup.com.