

10

Developing the SAP Data Center

Introducing the SAP Data Center

In this chapter, I will discuss in detail the process inherent in building a new data center facility, or transforming your current data center into a foundation capable of supporting a mission-critical enterprise SAP application and its requisite solution stack. The goal is clear—to create a stable and highly available facility for hosting your SAP implementation. I take this process from building out the power, network, and rack infrastructure through installing servers, configuring disk subsystems, and installing server operating systems. In essence, by the time you have completed the activities described in this chapter, the SAP system landscape should be ready for the SAP Basis installations to be performed.

Although you may have already decided on many of the factors that will drive the design and deployment of this facility, including location, hardware and software vendors, and SAP products and components, “availability” should typically represent the single most important consideration driving the data center build-out. Remember, this application and its resident data will ultimately prove critical to your company’s well-being. Should this application and data become unavailable, even for a short period of time, costly and otherwise huge ramifications could result:

- Thousands of users may sit idly by, waiting for the system to come “back.”
- Trucks may stack up in the loading docks, waiting for bills of lading and shipping orders.

IN THIS CHAPTER

- Introducing the SAP Data Center
- The SAP Data Center “Big Picture”
- Data Center Physical Requirements
- Rack Planning for Data Center Resources
- Network Infrastructure for SAP
- Network Server Preparation
- General Storage Considerations
- On the Road to Implementation
- Tools and Techniques

- Customers may call in or “click in” looking for status updates on their orders, only to be told to “try again sometime later, the system is down.”
- Manual processes may need to be invoked to keep new orders coming in. And to make it worse, eventually these manual orders will need to be keyed into the system when it again becomes available, further impacting the users’ time to place new orders.
- Reports will be unavailable, impacting decision making from the boardroom down to the assembly line, and everywhere in between.

► For more details on the impact that high availability plays in the SAP Data Center, see “A Final Look at the SAP System Landscape,” p. 217 in Chapter 6.

So, keeping “availability” uppermost in our minds, let’s move on, noting that the data center in some way affects all layers of the SAP Solution Stack. For example:

- At the lowest layer of the stack we find power requirements. I have worked on a number of mySAP.com implementations where as many as 80 new servers and related disk resources are deployed over the course of a year, for example. Such a formidable collection of hardware pulls a considerable amount of amps and volts, not to mention the raw power infrastructure requirements needed to simply keep all of this gear running. An incorrectly architected power infrastructure will bring down an otherwise highly available clustered SAP solution in a heartbeat—all of the high-availability offerings at other layers in the solution are “powerless” without a well-architected Uninterruptible Power Supply (UPS) solution or power distribution system.
- Similarly, cooling requirements must be addressed at one of the lowest layers of the stack. Those same 80 servers I mentioned not only pull a tremendous amount of power, they also generate a considerable quantity of BTUs (British Thermal Units, or units of heat). An inadequate cooling and air handling system can wreak havoc with availability statistics.
- Servers, disk subsystems, network infrastructure, and other SAP-related infrastructure needs to all be neatly racked and cabled. Lack of attention to detail, poor planning, and more can quickly become contributing downtime factors. In more than one case, I have seen racks of servers cabled neatly and in an organized manner, only to have all of this work ripped out after someone finally thought to pull out a server for servicing. Why? Because the cabling did not allow enough “slack” for the servers to be actually pulled out more than a few inches! In other cases, I have seen high availability compromised simply because otherwise redundant pairs of cables were routed through the same cable conduits, and the conduit itself failed or was damaged.

- SAP's tiered architecture requires forethought in regard to network planning. As you will see later in this chapter, neglecting the impact that a three-tiered architecture can have on public and private network segments can impact not only availability, but also overall performance. In addition, we cannot forget about network firewall security and other network infrastructure considerations that will impact availability when it comes to Internet-facing SAP Web servers, eCommerce/procurement systems, and so on.
- Server infrastructure and design directly impact availability, both "in the box" via single points of failure, and "out of the box" when it comes to higher-layer solution stack matters like failover clustering.
- Your disk subsystem, whether direct-attached SCSI, fibre channel arbitrated loop, switched fabric, or network attached, tends to be the single most important performance factor in my experience, outside of really bad coding. But it is also one of the most easily misunderstood solution components when it comes to high availability.
- Operating system installation, configuration practices, and more can easily impact availability. I will note how and why this seems to be so often overlooked, and help you to conform to best practices in this regard.

As you have just seen, the design and implementation of a company's specific SAP Solution Stack through a well-planned data center deployment affects availability at all levels of the stack. Single points of failure (SPOFs) abound everywhere. A lone power source, single power distribution unit, nonredundant power supplies, single network segment to a server, single server hosting a database, and more all represent opportunities for downtime. And I have not even begun to discuss the database and SAP application layers!

THE SAP SOLUTION STACK OR THE OSI MODEL?

If you prefer to look at things from an OSI model perspective, you'll be happy to note that many of the key layers of the SAP Solution Stack map nicely to a layer in the OSI model. Therefore, every aspect of planning for a highly available SAP Data Center also tends to "snap into" one of those seven OSI layers, from the physical layer—power, through the data, network, and transport layers, up to the session, presentation, and application layers—where your end users are provided with the SAPGUI interface.

In the remainder of the chapter, I will take a closer look at each of the availability factors in the bulleted list provided earlier, as well as operational and other processes that will ultimately impact the net availability of your SAP solution to your end-user population.

The SAP Data Center “Big Picture”

From a “big picture” perspective, you need to ensure that no detail is overlooked when planning for your SAP Data Center facility or facilities. Availability, after all, is all about details. Each layer in the SAP Solution Stack presents you with single-point-of-failure challenges; each layer must therefore be thoughtfully considered with an eye toward eliminating these SPOFs, or at least noting and mitigating risk to the point where such risk becomes financially acceptable.

First Things First—Standardization

Before we dive into identifying and addressing single points of failure common to SAP Data Center designs and implementation, it first makes sense to discuss the role of standards and standardization in your data center. Standards impact everything. Fortunately, taking a close look at standards early in the SAP Data Center planning process forces you to think ahead, and ultimately avoid many pitfalls potentially lurking in the future. Consider the following:

- Server naming conventions must be descriptive enough to promote manageability, but short enough to be technically supported by the particular SAP component version (and any other applications that might need to reference this name, including systems management applications like HP Openview or BMC Patrol).
- IP naming conventions should help identify what type of server network connection is being made. For example, standards that help to identify public, private, clustered, and internal/other network connections are quite prevalent in the world of SAP infrastructure.
- Disk naming (and drive letter, for Windows 2000/NT) conventions should be published and leveraged for consistency. Such consistency by its very nature impacts availability as well, as the chance of someone “accidentally” bringing a disk resource offline is less likely to occur when the disk name or disk drive letter for the database (for example) tends to be the same throughout the SAP landscape.
- Even something as simple as color-coding network cables and power cables can improve system availability. Similar to IP naming conventions, a good color-coding scheme helps avoid unplanned downtime due to inadvertently uncabling or miscabling a vital network or power connection. I have witnessed companies leveraging factors other than simply color, too; the number and thickness of bands in a cable, and even the cable thickness itself can also be used to differentiate otherwise similar cables.
- Standard “high-availability server configurations” are normal in most SAP shops. This usually equates to servers configured with redundant power

supplies, fans, power distribution units, processors, RAID-protected disk drives, RAID or ECC-protected RAM, network cards, and so on, in a server cluster configuration with dual-host bus adapters in each server.

- As with standard highly available server configurations, most SAP shops also promote an equivalent “high-availability disk subsystem.” This usually involves a standard frame or disk chassis, standard drive size (in example, 36GB 15,000 RPM 1” drives), standard redundant disk drive controllers, and redundant disk interconnects back to the server infrastructure.
- Next, a standard operating system built for high-availability deployments is typically developed. This would include specific OS release levels, patch or service pack levels, any patches or bug-fixes required, other software drivers and their versions, and so on. Nowadays, standard methodologies for deploying customary server images are often employed, leveraging OS-build approaches ranging from traditional disk imaging to custom scripting, deployment of imaging servers, hardware vendor-specific approaches, and so on.
- Finally, standardized processes regarding managing all of the aforementioned resources help to minimize downtime across the board.

Other standards exist, of course, but the preceding list should prove useful in identifying the key areas within each layer of the SAP Solutions Stack that must be addressed before a single data center floor-tile is ever pulled up, or server mounted, or operating system installed.

Data Center Physical Requirements

Physical requirements mean different things to different people. In the context of building your data center, I could go into minute details regarding actual construction of the facility, for example. In such case, physical requirements would exist with regard to the following:

- Physical construction materials and related factors (fire codes, weight-bearing members for roof-mounted AC/Environmental Units, load-rating factors for the raised floor construction, cable risers or trays, and so on).
- Location (avoid the first floor or top floor of buildings, and avoid flood zones, areas where access may be limited in terms of single stairs, elevators, roads, and so on).
- Environmental systems (cooling, heating, humidity levels), including access to published thermal specifications for each component to be housed in the data center.

- Accessibility (loading dock/freight elevator access, as well as double-protected public access points).
- Physical security/access, including monitoring systems (card or other systems for doors and window access, and attention to vertical security above the dropped ceiling and below the raised floors).
- Controlling systems (temperature/fire suppression, smoke, water sensors, and so on).
- Lighting and plumbing, as required.
- A central operations/monitoring station.
- Access to high-bandwidth multi-path data communications circuits or network/Internet connections.
- General dual power infrastructure, in terms of 208 volts AC versus other options, including the availability of generators, battery backup, and so on. This can also include access to two discrete city or state power grids, should high-availability or disaster recovery requirements dictate such a robust power infrastructure.

These details are best left up to the experts who design and build data center facilities. Ensure that at a minimum the preceding points are addressed, however.

Power Requirements

Power problems can plague an otherwise bullet-proof solution architecture and therefore power needs to be planned for the long term, not merely for the demands of Go-Live—your SAP environment will grow, grow, grow. When addressing the power needs of the SAP data center, it is helpful to first analyze each specific server, disk subsystem, network, or other *hardware* component requiring power, and then work “back” to the ultimate power source. For maximum availability, ensure the following:

- Where the highest levels of availability are necessary, each hardware component must support redundant power supplies (otherwise, the remainder of this list is not of any use). Preferably, these power supplies should also be “hot swappable” or “hot pluggable.” In this way, in the event of a power failure, not only would the server remain available and powered up on its second power supply, but the failed power supply could be pulled out and replaced without incurring downtime.
- Each power supply alone should be capable of keeping the hardware component up and running. That is, if the average load being pulled from one of the power supplies in a dual-power supply configuration exceeds 50% of its rated

capacity, you actually don't have protection from failure of the other power supply! The alternatives are clear—lower the capacity requirements by reducing the number of disk drives, CPUs, and so on, or increase the number of power supplies to three or more, or in rare cases simply replace the current power supplies with higher-rated alternatives.

- Each power supply must have its own unique power cable. This is a very common oversight with some of the second-tier server and disk subsystem vendors, where highly available systems might be touted, but reality differs. These high-availability wannabes often provide only a single power cable receptacle even in their “redundant” power supply configurations. A single *anything* represents a single point of failure, and should be therefore avoided. Besides, I have actually seen a couple of power cables fail in the real world. As silly or unlikely as that sounds, it happens. And besides, even more likely is the potential to pull out a single power cable, effectively bringing down the most available server or disk subsystem.
- As I indicated previously, color-coding or otherwise differentiating power cables makes it very clear to everyone when things are cabled correctly. The most common implementation of this involves a black cable cabled to the primary power supply, and a gray or white cable cabled to the redundant power supply.
- Each power cable must be routed to dedicated separate power distribution units (PDUs, or power strips, and so on)—whatever is used by the company to centralize many power feeds into fewer larger-capacity connections. Each PDU needs to be analyzed to ensure that the load placed on this single component, should the other PDU fail, can still be addressed by the remaining PDU. And as I said earlier, the most common implementation of this is black cables to one PDU, and gray or white cables to the redundant PDU.
- Each PDU must in turn be cabled to redundant uninterruptible power supplies, or UPSes. Like the PDUs, these need to be regularly tested and analyzed to ensure that they are indeed “redundant.” Note that a UPS tends to only be equipped to handle short-term power losses, thus necessitating our next power component.
- Primary power for each redundant power supply should culminate in redundancy at the breaker boxes as well. That is, each power supply should ultimately receive its power from a dedicated breaker panel, like that illustrated in Figure 10.1.
- The “back-up generator” is a necessity for mission-critical SAP shops. Whereas the UPS provides short-term relief from blackouts and brownouts, a generator can conceivably provide power for days, as long as fuel is available. Select a generator that runs on whatever is most easily accessible or readily available, including diesel fuel, propane, or natural gas.

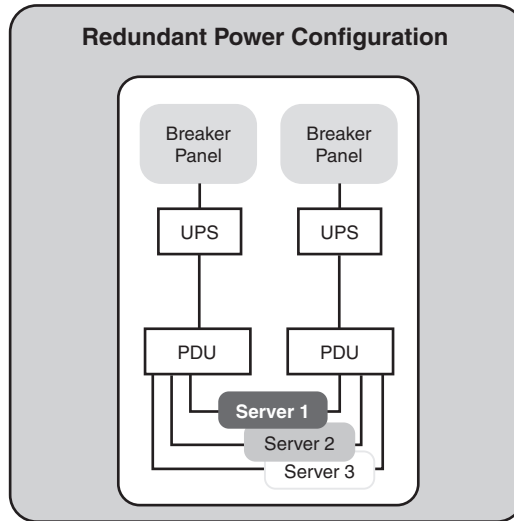


FIGURE 10.1 Here you see a completely redundant power infrastructure, from the servers/hardware components all the way back to the breaker panels.

It is of utmost importance that the generator and the UPS be properly sized to handle the loads placed upon them. Generators must leverage an Automatic Transfer Switch (ATS) to allow them to tie into both the power company (the power utility, or “utility power”) and the SAP data center, as you see in Figure 10.2. Critical systems need to be identified and earmarked for generator backup. These systems typically include emergency lighting, emergency environmental and safety controls, the critical SAP data center gear, and in some cases the contents of the entire data center.

Best practices also dictate that critical computing systems be isolated from main facility/operational power, but that each source of power “back up” the other. In this way, operational failures do not impact the enterprise system, and vice versa, and both are protected from failure by two power sources.

USING KVA FOR ACCURATE UPS SIZING

UPSes are rated by KVA or Kilo Volt-Amps. The formula to calculate KVA is $\text{amps} \times \text{volts} / 1000 = \text{KVA}$. So if your rack is capable of pulling $69.5 \text{ amps} \times 240 \text{ volts} = 16680 / 1000 = 16.6 \text{ KVA}$. Never allow your UPS to run above 80% capacity. 16.8 KVA is 80% of 21 KVA. So at a minimum, the rack should have 21 KVA worth of UPS.

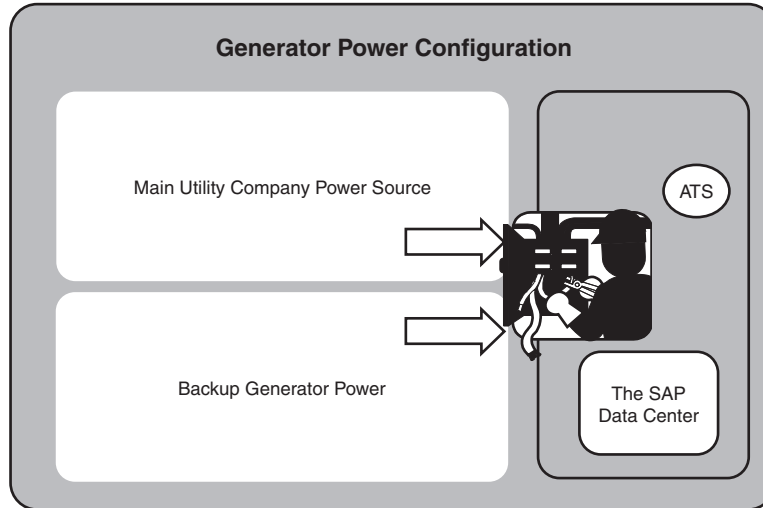


FIGURE 10.2 Without an Automatic Transfer Switch, the actual usefulness of a backup generator is questionable, thereby impacting high availability.

Power Oversights in the Real World

One of my favorite enterprise SAP servers, the HP ProLiant, serves as my next example. A wealth of information is available on the ProLiant, in the form of something HP calls “quickspecs.” These technical specifications have been published and updated for years, and describe in great detail much of the minutia of little interest to anyone but hardcore techies once the SAP system landscape is in place. Prior to that time, though, these quickspecs fulfill a number of critical roles. First, quickly perusing this document reveals that the ProLiant DL760 8-CPU server will draw a maximum of 10 amps on a 208- to 240-volt line, while producing a moderate 5309 BTUs every hour. These little tidbits of information will help ensure that the data center facilities folks understand how many and what type of power circuits to run. The BTUs, on the other hand, should be fed into a simple model that will determine the minimum rating of the air handlers (cooling/heating system).

Another bit of information is provided in the quickspecs as well, the power plug connector. As with most servers, the power connector from the server to the PDU or UPS is ordered as an option with the PDU or UPS. The real challenge then becomes matching the PDU’s power plug with the appropriate power receptacle. Typically, either an L6-20P or L6-30P is called for—in the case of certain PDUs deployed at one of my particular customer’s new SAP data center sites, the L6-30P (a 30-amp circuit) was specified by the quickspecs.

However, the customer got ahead of themselves and in the interest of meeting deadlines for power, laid enough L6-30P power cables for the new Storage Area Network, too, which was to arrive shortly after the servers. When the SAN cabinets showed up, though, they couldn't plug them in. The connectors were different—they required the L6-20P receptacles. To this day my colleagues and I are still not quite sure how our motivated customer managed this, but they actually forced their million-dollar SAN power cables into the wrong receptacles, and ran the system this way for perhaps a week before someone noticed that “something just didn't look quite right” under the subfloor. Understand that these two connectors are quite different, and this little engineering feat will begin to sink in—they somehow managed to squeeze and twist the male plugs most of the way into the female receptacles. Not only did this pose a potential safety hazard, but they also risked blowing up their 20-amp SAN gear with 30 amps of juice.

They got lucky. The SAN had very few drives actually installed and spinning (and therefore drawing power) that first week. However, this simple oversight caused a one-week delay in the project plan, while the system was effectively shut down awaiting the proper wiring. In the end, the lost time was made up in the operating system and SAP basis installations, and neither gear nor people were any the worse. But the moral of my little story should be clear. Not only should the technical specifications for each piece of equipment be checked and verified, but we should never solve our power problems by brute force. Besides, because all of this information is just a click or quick-spec away, there's really little excuse for misconfiguring or underallocating power and cooling resources.

Another common mistake illustrates how the redundancy of power-related components can be rendered useless through lack of attention to cabling and overall power architecture. My customer in this instance wanted to factor in redundancy at the physical layer of their SAP deployment. Their high-end servers, disk subsystem, and network equipment all supported redundant power supplies, so they took advantage of them. Each power supply on the back *left* side of each server and disk subsystem drive shelf was the recipient of a black power supply cable. This in turn was carefully routed along the left side of the rack enclosures to a power distribution unit dedicated for this purpose. Similarly, each power supply located on the back *right* side was fitted with a gray power cable, and these gray cables were also carefully routed to their own PDU. So far, so good—no single points of failure existed, in that half of the power supplies, cables, and PDUs could fail, and the system could still remain up and powered.

However, all of the careful preparation and planning that went into this phase of the project was tossed out the window after another few minutes of work. My customer plugged both PDUs into the same UPS, which then was cabled to two large redundant data center UPSes. Like the 1975 Chevrolet Corvette's exhaust system, a dual-power approach to high availability is just “smoke and mirrors” after everything

merges into a common pipe. The Corvette never realized its peak power potential that year, and my customer lost any hopes of achieving 100% reliability, even though the solution “looked good” from many angles. By plugging the PDUs into the same UPS, they defeated the purpose of redundant power—if their single-point-of-failure UPS were to fail, the entire system would grind to a halt.

Like power, the next layer in the solution stack also represents a basic necessity for supporting your SAP enterprise—cooling.

Cooling and Other Environmental Controls

One of the biggest causes of hardware component failure is heat. Luckily, planning for cooling requirements has become a lot easier with the popularity of the World Wide Web. That is, nearly every hardware vendor out there today publishes BTU/thermal specifications. Your job is then to simply pull down and “add up” these technical specifications on every piece of equipment you plan to deploy. Don’t forget to allow for future growth, either—with server and disk form-factors shrinking every year, the heat generated per cubic foot of data center space just continues to grow and grow. To conservatively address the next three years in your data center planning efforts, determine the average BTU output per cubic foot, and then double that number and apply it to any remaining floor space that could conceivably house incremental SAP gear. Don’t forget to factor in the fact that air might not move uniformly through your data center and any other cooling dynamics inherent to your facility. In doing so, you will be eminently ready for the day the VP of Operations says, “Hey! We’re gonna go ahead with that SAP PLM project, so make room for 20 new servers and a couple of SAN cabinets in the next few weeks.”

As with cooling, air handlers exist that allow for controlling and exhausting heat, monitoring and fine-tuning humidity, and so on. Ensure that new hardware additions to the SAP data center are plugged into the BTU model as soon as possible, so that any new requirements for cooling will be given an opportunity to be addressed. Air handlers and other large environmental gear like this require significant lead times when new components or upgrades/replacements loom in the future.

Don’t forget to load the proper OS-drivers or applets that may be required by your hardware system to shut itself down in the event of overheating or loss of cooling. In the case of the ProLiant, the Compaq System Shutdown Service shuts down the server when the heat exceeds a predefined threshold, acting in response to commands from the integrated management features inherent to the ProLiant server platform.

And consider some of the newer trends in air handling and monitoring. For example, HP recently developed a robot that literally rolls around your data center floor looking for hot spots. Upon finding a hot spot, the robot analyzes the conditions and may, for example, signal your cooling system to increase airflow to the

area. Or it may instead communicate with your hardware systems to relocate workloads from one system to another. Utilizing a combination of these approaches, HP believes that it can reduce cooling costs for its customers in the neighborhood of 25 percent.

Rack Planning for Data Center Resources

With your highly available power infrastructures laying the foundation for your SAP Data Center, and attention to environmental requirements already addressed, you are ready to proceed with the next “physical” data center infrastructure layer—the rack mounting systems for housing all of our enterprise computing gear. In this section, I will cover:

- Rack layout and design considerations
- Optimizing rack “real estate”
- Rack mounting and related best practices
- Cabling and cable management
- Rack placement in the data center

What exactly is a rack? One of my customers describes a *rack* simply as “furniture for computers.” In most cases, racks are somewhere between four and seven feet tall, 19 inches wide, and something like 34–39 inches deep, depending on requirements. Racks allow for computer gear—servers, disk subsystems, network components and more—to be easily mounted, cooled, and serviced in what amounts to a “stacked” configuration. That is, the gear appears to be stacked one on top of the other. In reality, of course, none of it should literally be stacked one on top of the other, as this makes servicing and cooling the gear quite difficult.

Often, each component is mounted on sliding rails (though less advantageous fixed rails are still quite popular with some server and rack vendors). These sliding rails facilitate rapid access to the top and sides of each hardware component, where service areas are easily accessed. And something called *cable management arms* make this serviceability possible, allowing hardware systems to be pulled out for service without requiring cables to be disconnected or rerouted. Cabling is discussed later in this chapter.

Rack Layout and Design Considerations

Before you order a truckload of racks, you need to step back and develop a plan for deploying and laying out these racks. There are a number of best practices to be considered, but nearly as important is achieving some sort of consistency in deployment. This can be accomplished by working with your hardware vendors, and

demanding detailed deployment guidance as it relates to how servers and disk resources should be mounted, how many racks are required, how they should be optioned, and so forth. And there are questions related to rack placement in terms of metrics and standards that must be answered:

- Determine the airflow design for the equipment to be mounted. Then check to ensure that the racks allow for this design. For example, the HP ProLiant server line has always been designed for front-to-back airflow, thus mandating the use of a rack that supports this. In other cases, servers pull air from the data center subfloor and vent it out the top of the rack. It is therefore necessary to ensure that the servers and racks “match” each other—do they?
- It is highly recommended that the racks be arranged in a front-to-front, or back-to-back manner. Picture standing in between two rows of racks that either face toward each other, or face away from each other, and you will understand front-to-front and back-to-back, respectively. For maximum front-to-rear airflow, the *air registers* (the floor tiles with the little holes in them that force out cool air) should be placed in line with the front of the racks. Similarly, air returns located inline with the rear of the racks are a must, too, as you see in Figure 10.3. In the case of bottom-to-top airflow, the air registers are instead placed such that the rack sits on top of them. Are your air registers positioned for the best airflow?
- Racks need space. Is there enough room both in front of and behind each rack (25” in front and 30” in back is suggested, though your specific rack documentation may indicate otherwise) to open and close the doors, and to provide the cooling airflow necessary to keep everything running? If not, then reposition the racks, or move the offending gear/walls.
- Would purchasing a top-mounted fan option or split rear door (both enhance cooling to some degree, regardless of airflow direction) make sense, given the volume of computing gear to be racked? I generally recommend top-mounted fans in the densest of racks, regardless of the direction of airflow otherwise; these fans help draw component-damaging heat out of the rack as quickly as possible.
- Are there overhead fire protection sprinkler devices that must be considered? Check local building codes for acceptable clearances before installing racks underneath such devices.

At this juncture, it makes sense to actually diagram rack layouts for your specific data center needs. Again, I recommend working with your hardware vendor or an experienced systems integrator to get this right the first time.

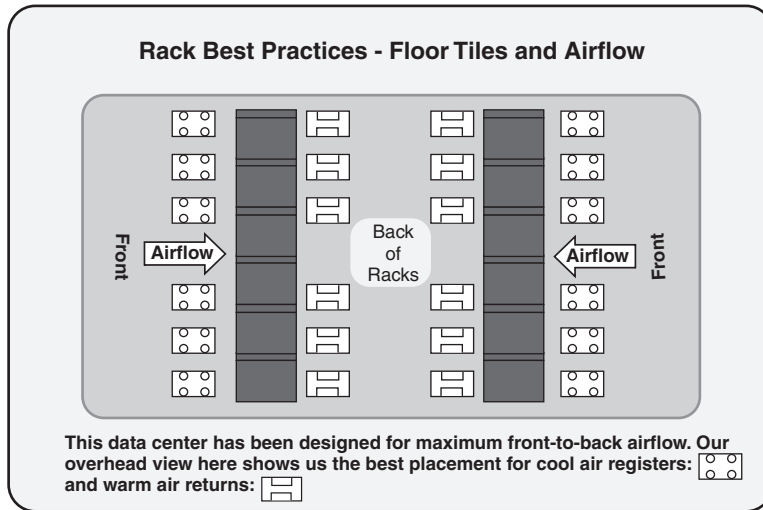


FIGURE 10.3 Placement and layout of racks and floor tile airflow registers/returns are key to maintaining proper air temperature in the SAP Data Center.

Optimizing Rack Real Estate

When loading equipment into the racks, observe the following best practices and observations:

- Put heavier items in the bottom, that is, UPSes and large servers.
- Address floor-loading weight limitations. For example, never load a rack with all UPSes, as the weight alone will almost certainly exceed the data center's raised floor load-bearing factor. Instead, distribute UPSes evenly across all racks.
- Locate monitors and keyboards for maximum ergonomic comfort (based on your particular sitting/standing arrangement). Do NOT locate them on the top of each of your 7-foot racks, like one of my customers did! It's funny reading about this after the fact, but it was no laughing matter at the time.
- Attempt to distribute monitors and keyboards such that access to a particular environment is available in two ways. For example, if an SAP Staging landscape consists of eight application servers, it would make sense to attach four of these to one monitor/keyboard, and the other four to the second monitor/keyboard. In this way, performing maintenance, upgrades, applying patches, and so on is possible from one monitor/keyboard, while the other remains available for business-as-usual.

- Rack blanking panels must be used to “fill in” spaces greater than 1U in size (where a “U” is a standard rack unit of measurement equivalent to 1.75 inches). This is important to ensure that good airflow is maintained within the rack enclosure itself.
- All items of gear housed within the rack—servers, disk subsystems, everything—should be installed with their covers and sides on. Running a server without the top cover or side panels, for example, disturbs the flow of air through the computer such that it is not cooled off as effectively as it otherwise would. In some case, running a server without the cover and sides will actually invalidate the unit’s warranty, too.

Other best practices exist as well. Refer to the Planning CD for a comprehensive “Rack Best Practices” document.

Rack Mounting and Related Best Practices

When connecting the racks to power, fault-tolerant considerations should be heeded. Your servers and disk subsystems already have dual redundant (or N+1) power, and to make the most optimum configuration, multiple (at least two) PDUs should be deployed. I recommend at least one mounted on each side of each rack. All right-side power supplies should be plugged into the right-side PDU, and the left-side power supplies into the left-side PDU. The PDUs should be plugged into separate circuits and breaker panels, as I have discussed previously. These circuits should be adequately sized to meet the potential load demand, which should rarely exceed 80% of their rated capacity if at all possible. In this configuration, if a circuit is lost, only half of the power supplies in each rack will be without power, leaving the other half to continue powering all of the computing equipment. And you will have no downtime.

Another thing to consider in rack planning is how the servers should be grouped. Ask yourself what makes sense. Some of my clients like to group all development resources together, all test resources together, and so on. Others prefer to group all servers together, all disk subsystems together, all network components together, and so on, similar to what you see in Figure 10.4. I suggest that you select either the approach that is used in your data center today, or the one that makes the most sense given the specific SAP system landscape being deployed.

Racking Clustered Servers

Regardless of the grouping approach, though, a best practice exists when it comes to clustering servers. Cluster nodes should always be mounted in separate racks, using separate keyboards, mice, and monitors, connected to completely different PDUs and network infrastructure, and so on. Why? Because if the cluster nodes share anything in common, this becomes a single point of failure. And given that you are clustering in the first place, avoiding single points of failure is key for these systems.

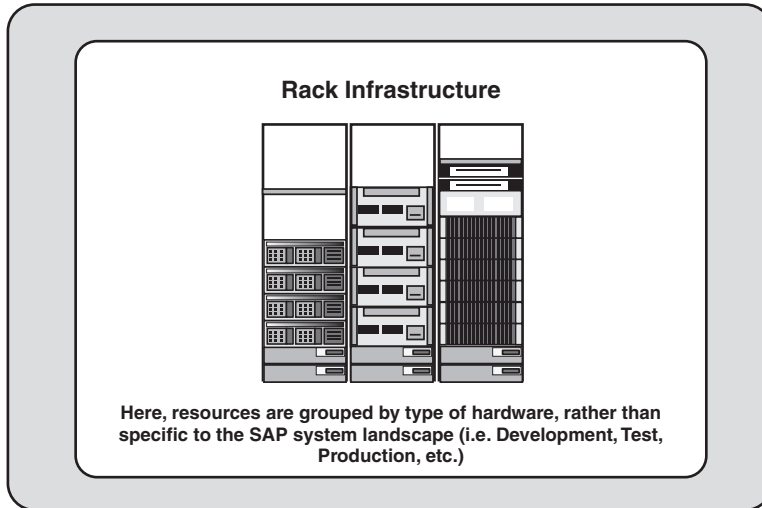


FIGURE 10.4 This grouping approach is hardware-centric, rather than SAP landscape-centric. Either approach is valid.

Racking Production

When it comes to mounting the Production servers in their racks, it is critical to maintain a separation of all infrastructure. That is, Production should ideally be isolated from all other SAP environments, from a power, network, rack, SAN, and server perspective. The Production racks therefore require their own power infrastructure, network infrastructure, and so on.

In the real world, as SANs continue to enable the highest levels of availability and scalability for mySAP customers, the line between production and other resources is beginning to blur. Expensive switched-fabric switches, SAN management appliances, SAN-enabled tape backup solutions, and the fibre cabling that ties everything together is often shared between production and development systems, for example. The important thing to address in these cases where the cost of a sophisticated SAN outweighs the risk of maintaining separate production resources is to provide for some kind of change management support. That is, you still need a test environment of sorts, be it a Technical Sandbox, Test/QA system, or whatever, to test the impact of changes to your SAN. In my experience, these changes are often firmware and hardware-related, though topology and design changes also need to be tested as well.

Cabling and Cable Management

If you have ever attempted to squeeze 21 servers into a 7-foot (42U) rack, the concept of cable management should evoke special thoughts of long days and late

nights. Cables have shrunk in diameter over the last five years, but not to the extent or in keeping with the trends we have all seen in hardware platform densities. As a result, one of the biggest immediate challenges facing new SAP implementations is how to best route the various keyboard, mouse, monitor, disk subsystem, network, and other cables required of each piece of gear. More and more, we see fat cables being replaced by smaller ones, and lots of little cables being replaced by fewer centralized cables. But in the end, there's still a whole lot of cabling to consider!

Alternatives to traditional KVM (keyboard, video, and mouse) cable connections abound today. The following represent two common methods of reducing the number and size of cables (a long-time favorite and a newer alternative):

- Installing a dedicated “computer on a board” into a PCI slot in each server, to facilitate out-of-band and in-band management of the server. Such boards typically require a network or modem connection only, and facilitate communications via a browser-enabled user interface. A fine example is HP's Remote Insight Board. Depending on the particular server, these boards may come standard with the system, and may even be integrated into the server's motherboard.
- Installing “KVM over IP,” which represents another way to shrink various cables into a small single network cable. This method also requires a small server of its own, not to mention licensing costs for the particular enabling product.

More often than not, I figure we will continue to see the widespread use of cable management arms, tie-wraps, Velcro-wraps, and similar inexpensive approaches. Regardless of the cabling techniques, though, continue to stay focused on *availability*. Single points of failure can exist in this realm like any other area—remember that a server may be effectively unavailable once it loses access to its keyboard, mouse, or video connections. Even in the best case, if the server remains up you may still be operating blind, saved only through your management and monitoring tools if so enabled.

Network Infrastructure for SAP

Given that SAP is architected to support a distributed three-tiered design, a slew of configurations exist that can potentially impact high availability. There are similar performance considerations as well, as the database, application, and Web tier layers are all affected. For example, a back-end network is recommended to interconnect the database and the application servers, another network interconnects the application servers to the Web/Internet layers, and a third public network addresses client requirements. If you plan on pulling backups across the network (rather than via a

disk subsystem that supports direct-attached SCSI or fibre channel tape drives), a separate back-end network subnet is highly recommended in this case, too.

In all cases, 100Mbit switched network segments are warranted, if not Gigabit Ethernet. Backups are very bandwidth-intensive, and if data is transmitted over otherwise crowded network lines, database disconnects between your application and database servers may result, for example. In this case more than any other, therefore, Gigabit is warranted. Beyond network-enabled backups, the second preferred subnet in which to leverage Gigabit includes the network connecting the application and database servers, where traffic can also be quite heavy. This is especially true as more and more enterprises grow their mySAP environment by adding new SAP components but insist on leveraging the same backend network.

The application/Web tier often consists of a single 100Mbit subnet. As the application servers/Web servers are usually not considered critical from a backup perspective, dedicated network backup segments are usually not warranted in this case—these servers usually contain fairly static data that lends itself to weekly or other less-regular backups.

The Web tier is typically split, though. In the case of SAP ITS implementations, for instance, an application gateway (AGATE) connects to the SAP application servers, a Web gateway (WGATE) connects to the AGATE, and the end users connect to the Web gateway. Thus, the WGATE is often housed in a secure DMZ for access via the Internet or an intranet, while the AGATE remains behind a firewall, safe in its own secure subnet, as you see in Figure 10.5. In this way, the network accommodates the performance needs of the enterprise without sacrificing security.

Network Fault Tolerance

As I indicated earlier, there are many ways to architect a network solution for SAP. Simply segregating each layer in the solution stack achieves minimum performance metrics but does not address availability. In fact, it actually increases the chance of a failure, as more and more single-point-of-failure components are introduced. Fault tolerance must therefore be *built* into the design, not looked at afterwards. To this end, I will next discuss the primary method by which network availability is designed into the SAP data center, using an approach I call *availability through redundancy*.

Just as a redundant power infrastructure starts with the servers, disk subsystems, and other hardware components, so too does a redundant network infrastructure start with the networked servers. Here, redundant *network interface cards*, or NICs, are specified, each cabled to redundant network hubs or switches, which are in turn connected to redundant routers. Figure 10.6 illustrates this relationship between the servers and a highly available network infrastructure.

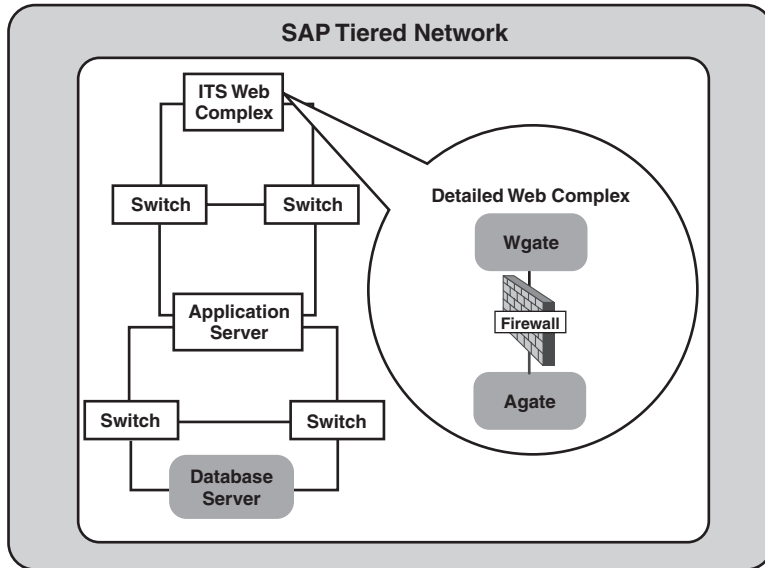


FIGURE 10.5 Network architecture involves every layer of the SAP Solution Stack, from the database server up to the application and Web layers.

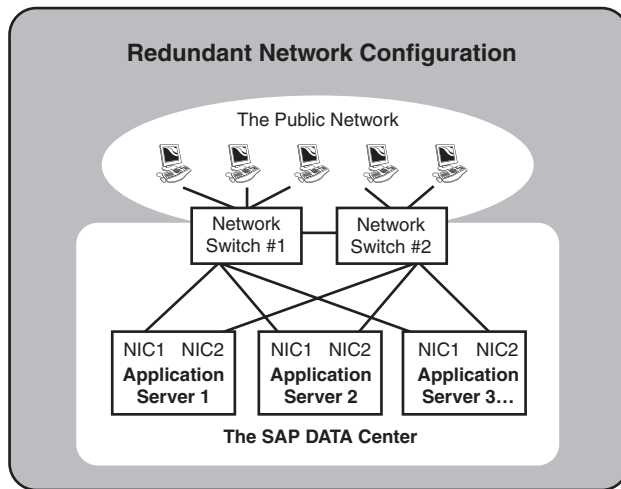


FIGURE 10.6 This example shows what a properly architected highly available network infrastructure may look like for SAP.

Network fault tolerance starts not only with two (or otherwise redundant) NIC cards, but also with OS-specific drivers capable of pairing two or more physical network

cards into one “virtual” card. Hardware vendors often refer to this as *teaming* or *pooling*, but other labels have been used for this important piece of the network puzzle. Regardless of the label, the idea is simple: Pair two or more network cards together, and use both concurrently for sending and receiving the same exact packets of data. Then, use the data on one network card/network segment (referred to as the “primary” NIC), and discard or ignore the (exact same) data on the other NIC as long as the network is up and things are going well. If a failure occurs with one of the NICs, network segments, or hubs/switches on the “redundant” path, continue processing packets as always on the primary path—business as usual. However, if a failure occurs with one of the NICs, network segments, or hubs/switches on the “primary” path, immediately turn to the redundant path and start processing the data moving through this path instead.

Today, most every operating system supports NIC teaming for high availability. Many OSes also support variations of NIC teaming geared toward improving performance, including network bonding, adaptive load balancing, FastEtherChannel, Gigabit EtherChannel, and more. Remember, though, that these latter variations do not necessarily improve availability; teaming must either be inherently supported by or used in conjunction with these different performance-enhancing techniques or approaches.

NOT ALL SERVER CONFIGURATIONS SUPPORT NIC TEAMING

Some operating systems simply do not support specific network cards when it comes to NIC teaming. Further, Microsoft’s cluster service specifically prohibits clustering the private high-availability server interconnect, or heartbeat connection, regardless of the type of NIC—Microsoft simply does not support teaming the heartbeat. Network availability in this case is gained by configuring the cluster service such that both the private and public networks can send “still alive” messages between the cluster nodes. As an aside, best practices suggest that the private network be preferred for this activity, and that the public network only be leveraged when the private network fails or is otherwise unavailable.

Let us return to our primary discussion on designing and implementing highly available networks for SAP. Remember, each NIC in the team should be connected to a different switch that in turn is connected to a separate router or separate card in a highly available router configuration (as required). Each redundant switch, router, and any other network gear must then be serviced by redundant power to indeed create an *NSPOF*, or no single point of failure, configuration. Insofar as best practices are concerned, I recommend that you also adhere to the following guidelines:

- All network interfaces must be hard-coded or set to a specific speed, for example, 100Mbit Full Duplex. Refrain from using the “auto configure” function, regardless of how tempting it is to avoid three or four mouse clicks per server to hard-code each NIC setting specifically. This is especially a problem if ignored in clusters or in network environments characterized by switches and

hubs servicing different network segments running at different speeds. Using the “auto configure” setting can easily mask network problems, including intermittently failing network cards, and ultimately add hours or even days to troubleshooting cluster issues.

- Keep in mind that NIC teaming is implemented via a software/driver-level function. Thus, anytime one or more NICs participating in a team are replaced, swapped out, or reconfigured, the team should be dissolved and reconfigured again. Failure to do so could create issues difficult to troubleshoot, including intermittent or unusual operation of the NIC team.
- Most servers manufactured in the last 10 years take advantage of multiple system busses. Each bus is normally capable of achieving a certain maximum throughput number. For a 64-bit 66MHz PCI bus, this number is something approaching 528 MB/second. To achieve even greater throughput, then, NICs need to reside in different busses. Taken one step further, the idea of multiple busses should also get you thinking about high availability. Remember, an NSPOF solution theoretically relies on redundant components everywhere, even inside the servers. So, multiple PCI busses fit the bill—as an added level of fault tolerance, place the NICs in different PCI busses.
- Finally, if multiple NIC teams are configured (for example, in the case of SAP Application servers, where redundant connections are preferred to both the database server and the public network or Internet/Web intranet layer), it is important to ensure that the MAC address of the first team represents the primary address of the server. The primary address of the server thus maps back to its name on the public network. This sounds a bit complicated, but hopefully makes sense in Figure 10.7. Here, Network Fault Tolerant Team #6 represents the public team. Further, the MAC address of NIC #1 of this team is the MAC address “seen” by other servers. In this way, both name resolution and failover work as expected.

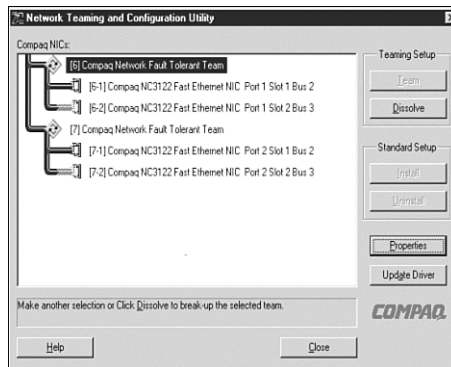


FIGURE 10.7 A unambiguous graphical user interface can help clear up otherwise complicated multi-NIC teaming configurations.

Central Systems and Optimal Network Configuration

Network configurations for SAP are also impacted by the type of server deployed in the system landscape. All of our attention thus far has been focused on three-tiered architectures, where the database, application, and client components of SAP are broken out into three different servers/hardware platforms. However, it is also quite common to deploy two-tiered SAP systems, or “Central Systems,” especially when it comes to small training systems or uncomplicated testing environments. In this case, the first question a network engineer new to SAP asks is “Do I cable the system to the public network, or the back-end database network?”

The question is valid. And because each option seems to make sense, there has been debate in the past on this very topic, to the point where SAP finally published an excellent paper on network recommendations for different system landscapes. Let’s assume that our Central System is an R/3 Development server, and take a closer look at each option in an effort to arrive at the best answer.

If our new network engineer cables the Central System to the back-end network (a network segment upon which other three-tiered SAP database servers reside), the following points are true:

- Given that the Central System is also a database server, the database resides where it is “supposed to”—on the back-end network.
- Any traffic driven by the DB server thus inherently remains on the back end as well. For example, if the SAP BW Extractors are loaded on this system, all of the network traffic associated with pulling data from our R/3 system to populate BW cubes stays off the public network. Again, our network traffic remains where it is supposed to.
- Any other DB-to-DB traffic, like transports, mass updates or data loads, and so on, also stays on the back-end.
- However, because our server is not on the public network, it is not as easily accessed by users and administrators. That is, name resolution becomes a challenge.
- Further, because the back-end network tends to be much busier than typical public networks (which from an SAP perspective only service low-bandwidth SAPGUI and print job traffic most of the time), our end-user response times will vary wildly, suffering across the board.

These are interesting points, but imagine how complex the issue becomes if your Central System now grows into an SAP cluster. In this case, because a cluster is implemented as a three-tiered configuration, you might be inclined to hang one

node (the database node) off the private network, and the Central Instance/Application Server Node off the public network. But what happens when the second cluster node fails over and the DB node becomes the CI/Application server as well as retaining its DB Server role? Or vice versa, and the CI/Application server takes on the role of the database server, too?

By now, you should understand that the best network solution in this case is not an either/or answer. Rather, the best solution is predicated on accessibility to both the private and public networks. This is accomplished by adding another network card into the server, NIC #2, and setting it up for connectivity to the back-end network. Meanwhile, the original NIC (now labeled NIC #1) is moved to the public network, where it is assigned a public IP address. Name resolution and other public network-related services regarding accessibility are now easily addressed. And by using static routing, all DB-to-DB traffic is retained on the back-end network segment as desired. Best of all, the cost to achieve this solution is minimal—a second NIC, another IP address, and a few hours of time are really all that is required.

To verify that your static routing indeed operates as it should after SAP has been installed, launch a SAPGUI session, log in with typical SAP Basis rights and privileges, and follow these steps:

1. Execute transaction **/nSMLG**.
2. Press F6 to go to the Message Server Status Area.
3. Verify that your PUBLIC subnet (under “MSGSERVER”) is listed under the Logon Group Name. If this is not the case, return to the previous screen and under the Instance column, double-click each application server, and then click the Attributes tab.
4. In the IP address field, enter the associated PUBLIC subnet for each server, and then click the Copy button.
5. Verify these settings again, by highlighting the application server and pressing the F6 key—the logon group will list the PUBLIC subnet.

By following these steps, you can in essence verify that the traffic associated with end users stays on the *public* network, thus ensuring that their response times will be as fast as possible—certainly faster than if the end-user traffic was routed over the back-end network.

With your highly available and optimized network infrastructure installed, you can now turn to the next few layers in the SAP Solution Stack—the network server and the server operating system.

Network Server Preparation

In the next few pages, I'll take a look at how to best set up and configure servers for their roles in the SAP system landscape. This includes optimum server hardware configuration best practices, details related to deployment and installation, and keys to configuring highly available and well-performing server-based systems.

Optimum Server Configuration Best Practices for SAP

When it comes to server configuration best practices, consistency and standardization are the keys. Consistent server configurations and standard server “builds” or software installations will eliminate a lot of future headaches. Your goals are simple:

- Minimize variety, which really equates to minimizing variables should you ever need to troubleshoot an issue
- Allow for rapid server replacement
- Support your systems management approach (discussed in detail throughout Chapter 14)
- Support the lowest total-cost-of-ownership model possible

I cannot say enough about minimizing the different server platforms in your SAP landscape. Two or three specific models of servers should be adequate for even the most complex installations or most cost-conscious customers. The benefits are great—fewer platforms for the SAP TSO to learn to support, less variety in the kind of hardware spare parts that might be stocked onsite, a simpler operating system stack (that is, from a software driver perspective), and so on. I recommend the following, considering these best practices regardless of which server platform is deployed:

- Only servers specifically certified by the hardware vendor to support SAP should be used.
- Maintain PCI slot consistency—whenever possible, keep all PCI cards in the same slots on all servers across the landscape. For example, if all servers house a public network card in slot 4, and a back-end network card in slot 6, the SAP TSO support staff will be less inclined to make false assumptions or mistakes later should a network issue arise.
- Standardize on a particular model of network card, disk controller, host bus adapter, and so on. In this way, not only do you minimize variables from a hardware perspective, you also minimize the variety of software drivers that need to be supported, and the variety of spares that needs to be maintained. In this way, you can even go so far as to swap NICs or HBAs from one server to another, to support troubleshooting without worrying about driver differences, for example. And this approach to standardization simplifies change control as well.

- Identify any PCI slot constraints, and ensure that these constraints are documented and followed. For example, particular HP servers only support the Remote Insight Board in a particular slot.
- Disk drives housed internal to each server should contain the operating system and Pagefile or Swap files. We'll discuss this in more detail later. The point here is that all other data types—database executables, SAP executables, database log and data files, and so on (unless otherwise required) should be housed externally.
- All internal disk drives should be protected in terms of availability. In some environments, this means attaching these drives to a hardware-based RAID Array controller (a high-availability and high-performing controller capable of withstanding the loss of one or more disk drives, depending on the specific configuration), and mirroring the drives. RAID stands for *redundant array of inexpensive disks*, and can be implemented via hardware or software (though in the latter case only if the operating system supports software-based RAID). Note that RAID implemented via software can be performed on standard disk controllers—no special controllers are needed beyond those supported by the OS. For the best availability, the controller and any necessary cables should be redundant as well. I will discuss RAID and RAID Array controllers in detail later in this chapter.
- All external disk drives should be attached to a separate RAID Array controller (or host bus adapter in the case of Storage Area Networks or other fibre-based storage systems, also discussed later in this chapter).
- If a disk controller does not support mirrored or otherwise protected cache (which is really just RAM that resides on the controller, to speed up reads and writes), or that cache is not backed up by a battery in case of loss of power, the controller should not be used for database or log drives. The reason is as follows. The cache allows writes to be “posted” or acknowledged by the controller, but not actually written to the physical disk drive. In this way, performance is greatly enhanced—the disk controller tells the operating system “I have the data” even when it's so busy that it does not have the time to actually write the data to the drive. Meanwhile, the operating system is then free to do more work. Later, when the disk subsystem has the time, it commits or writes the data in cache to the physical drives, and the operating system and database are none the wiser. However, if the controller loses power (usually the result of controller failure or losing power to the entire server), all of the writes sitting in the controller's cache never actually make it to the physical disk, and our database becomes inconsistent or corrupted. A battery backup found on the better controllers gets around this problem, though, by “holding” the data until the server comes back up. At that time, the controller then commits the writes, and again the operating system and database are none the wiser.

- Standardize on specific firmware revisions for all hardware—servers, disk controllers, disk drives, tape drives, and so on. Like software, firmware tells the hardware how to react or what to do, but it resides on a tiny read-only silicon chip on the hardware component itself. Also like software, firmware is subject to changes, bug-fixes, and so on. I recommend that a customer maintain a specific revision of firmware on each hardware component consistent with the vendor's SAP Competence Center's advice, testing planned changes in a Technical Sandbox first. Note that the "latest and greatest" firmware is not necessarily the best, and that a change even at a firmware layer requires the change to be tested and carefully promoted throughout the SAP system landscape (consistent with change control best practices discussed in detail in Chapter 13).
 - If it ain't broke, don't fix it—After a stable platform is assembled and supported by the vendor's SAP Competence Center, best practices dictate making no changes until forced to. This holds true for hardware components, firmware revisions, software drivers and patches, and so on. Think of it in this way—a lot of work went into assembling a collection of different SAP Solution Stack layers that actually work well together. Stay away from change *for the sake of change*, and your monthly reports reflecting unplanned downtime numbers will look that much better.
- To review the different levels of RAID, and learn more about configuration options, advantages, and features, see "Disk RAID Configurations and Features," p. 243 in Chapter 7.

With standardized hardware platforms behind us, let's embark on the journey to installing and optimizing an operating system for SAP.

Operating System Best Practices for SAP

Although many operating systems are supported for SAP, the following discussion focuses on the most popular OS underpinning new installations over the last three years, the Microsoft Windows family of OSes. In order to install an OS, you need to partition the hard disks upon which the OS will reside. Notice I said hard *disks*—I highly recommend mirroring the OS partitions or complete drives, such that if one physical drive fails, you can still continue to run from the second drive. With regard to installing the OS, you have two choices:

- If you are using a RAID Array Controller, you must run an array configuration utility to partition the drives. This readies them to be recognized by an operating system.

- If a standard SCSI or other controller is used (recommended in cases where software mirroring is warranted), nothing special needs to (or can) be done from a hardware perspective.

Create the OS Partitions and Logical Drives

In the case of a hardware array, create a mirrored pair (also called a RAID 1 set or mirror set). If possible, for maximum availability place the mirrors on drives that reside on separate SCSI busses. This is possible in the case of dual-channel controllers, or with servers where dual SCSI busses reside on the motherboard. In any case, by doing this, an outage can be avoided if a SCSI bus or cable fails. Not only that, but mirroring increases read performance as well, as both SCSI channels and both disk drives can later conduct work simultaneously.

After the disk drives are configured for a RAID set (as appropriate), create a logical drive. This is what will be “seen” by the operating system.

NOTE

If more than four partitions need to be seen by Windows on a single basic drive, you will need to create additional logical drives. This is because Windows is limited to creating four partitions on the primary bootable drive.

If you have started with a pair of 18GB drives, and assume you need only four or fewer disk partitions to be recognized by Windows, you are done—save the array configuration, and reboot the server to the Windows 2000 installation CD. Install the operating system, selecting to make the bootable partition 4GB or 8GB (or your standard boot drive size), and select the option to format it for NTFS. Finally, after the OS installation, carve up the remaining disks via Windows’s Disk Administrator utility into something just over 4 GB each—I recommend using these additional partitions for the Windows 2000 Pagefiles, discussed next.

Optimal Pagefile Configuration

Windows 2000 is still limited as to how large a physical pagefile can be made—4095 MB. And unless you take advantage of a published and fairly simple registry hack, you can still only place one pagefile on a disk partition (that is, one pagefile of 4095 Megabytes on the E: drive). As of SAP’s most recent Basis releases, total pagefile size should still equal about four times the amount of physical RAM in the server. Thus, a server with 3 GB of physical RAM should be configured with 12 GB worth of pagefiles. SAP further states, though, that a total pagefile exceeding something in the neighborhood of 10 GB offers little to no performance enhancements.

Keep in mind that Microsoft recommends a pagefile on the bootable drive (usually C:) equal to or greater in size than the amount of physical RAM, to support capturing full core dumps should the server succumb to a memory failure. Of course,

another simple registry hack allows us to get around this as well. Bottom line, though—if you have the disk space, create the pagefile.

If we return to my server example with 3 GB of RAM, the simplest supported manner of configuring pagefiles would look like this (accomplished by using Control Panel, System):

- The C: drive should be set up for 3 GB of pagefile.
- An additional D: drive should be formatted and configured for a 3.5GB pagefile.
- A final E: drive should be configured for a 3.5GB pagefile, too.

In this way, the server is configured for 10GB of pagefile total. And we have not had to do any registry hacks, nor sacrifice the ability to capture memory dumps. In addition, because we are not completely filling up the drive with a pagefile (we are leaving 500 MB of free space, assuming 4 GB partitions were created), we will avoid those pesky “disk at or near capacity” messages in the Windows Event Viewer. Finally, rather than wasting 2 GB of unnecessary space by building a 12GB pagefile, we instead took advantage of SAP’s concession in pagefile sizing, and created 10 GB total instead.

Another good example of pagefile sizing can be seen in Figure 10.8; note the partition sizes of just over 5GB each, allowing for maximum-sized pagefiles of 4095MB, with room to spare.

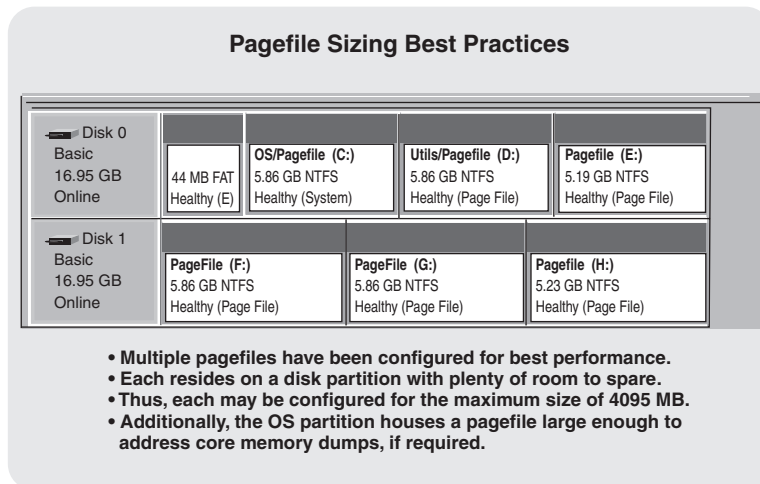


FIGURE 10.8 Pagefile sizing should adhere to the best practices noted here.

Other OS Configuration Requirements or Best Practices for SAP

The following configuration changes need to be made or considered prior to installing the database and mySAP component:

- Work with the SAP Basis team to select an SAP system name, and rename the server with this name. In the past, the server name could not exceed eight characters (releases prior to SAP Basis release 4.6C). Now, the limitation varies depending upon the specific basis release. However, given that SAP management utilities and applications have not all been updated to accept these new limitations, it is still a good idea to limit the name to eight characters until your own specific testing verifies that everything works properly with longer names.
- Create the admin/installation user (that is, P11adm or something similar) and give it domain administrator (or admin) rights—all SAP Basis installations, updates, and administration will be performed with this user ID.
- Ensure that all drives were indeed formatted for NTFS (Start Windows Explorer, select each disk, right-click, click Properties, and then select the General tab). Further, format each disk for 64KB block sizes (or allocation units), and label each appropriately (that is, Pagefile1, Pagefile2, and so on).
- If not already accomplished, label all remaining disk drive partitions via the Windows 2000 Disk Administrator. For example, label the C: drive *OS*, the D: drive *Pagefile1*, the E: drive *Pagefile2*, the F: drive *EXES*, the G: drive *LOGS*, the H: drive *DATA1*, the I: drive *DATA2*, and the J: drive *DATA3*. I also like changing the CD-ROM to drive letter Z: (or another drive letter “out of the way”).
- Set both the TEMP and TMP environment variables to point to C:\TEMP (by using Control Panel, System).
- Create the C:\TEMP directory if it does not exist already.
- Under Network Settings, ensure that the Maximize Throughput for Network Applications option is selected on the File and Printer Sharing tab. This impacts how memory/cache is allocated, and makes a significant performance difference.
- For each disk partition, grant “everyone” permissions to the root and all subdirectories.
- Load the SNMP service, in preparation for any systems management agents that will be installed later.
- Ensure that the appropriate Windows 2000 Service Pack is applied. Note that SAP and Microsoft will oftentimes indicate that a particular version of an OS fix, or patch, or Service Pack is supported. However, I really recommend verifying with your hardware vendor that the specific update in question is

recommended and supported on *their* specific server and disk subsystem platform. Work with the hardware vendor's SAP Competence Center to verify this information.

- Ensure that the vendor-specific software kit is approved by the vendor's SAP Competence Center, and then apply these updates on top of the latest service pack (unless instructed otherwise).
- Install the systems management agents (that is, Insight Manager, HP Openview, BMC Patrol, and so on per your standard). Refer to the appropriate management agent installation guide for any other details or prerequisites.
- Set up an alias for SAPTRANSHOST in the HOSTS file—edit C:\WINNT\SYSTEM32\DRIVERS\ETC\hosts and add an entry for SAPTRANSHOST (make it equivalent to the public TCP/IP address, and ensure that the SAP Basis team knows this has been set).
- Install Internet Explorer (via the SAP Presentation CD).
- Install the Active Dir Svcs Interface (ADSI), via the SAP Kernel CD (search the CD for ads.exe), if not already installed.
- Return to Control Panel, System, Advanced, and click the Performance Options button. Verify that performance has been optimized for background services. This setting ensures that SAP processes are given priority over locally logged-on users.
- Verify via the Windows 2000 Event Viewer that no issues exist from a hardware or OS perspective.
- Verify via your Systems Management Console (once installed) that no issues exist from a hardware or OS perspective.
- After the entire configuration process up to this point has been completed, I highly recommend that an automated or unattended installation script be created to easily set up a server that reflects all of the activities completed so far in this list. A disk imaging utility like Ghost is another good way to go. Other options include Microsoft answer files, and HP SmartStart scripts that load HP specific drivers as well as the OS. I suggest leaving out some of the specifics regarding pagefile sizing and partition labeling, so that the script is that much more useful in a generic "SAP server installation" manner. Barring all of these automated approaches, a good old-fashioned checklist will serve the same purpose, as long as it is followed precisely.
- Finally, to enable remote administrative access to each server, I suggest that a remote control application be tested and deployed. A number of very good solutions exist in the market today—Terminal Services for Windows 2000 is an excellent place to start.

SAP Server Configurations in the Real World

My colleagues and I have seen some pretty amazing SAP server configurations in our day. Think of these as more “Lessons Learned.”

More than once I have seen SAP production systems that are clustered in the name of high availability but forgo basic availability “in the box.” For example, one site chose not to mirror the internal disks located in each SAP cluster node, nor duplicate the requisite controllers and cables. Although this is not absolutely horrifying, it does mean that if a disk drive, controller, or controller cable were to fail, the server would die and the cluster would be forced to fail over to stay up. It has always been my view that it is preferable to never exercise your cluster failover capabilities unless absolutely necessary.

I ran into another cluster issue recently, too. It wouldn’t fail over. As it turned out, it was a one-node cluster.

In another case many years ago, my customer wondered why each server took a performance hit after 15 minutes of uptime. In their eyes, it was definitely a hardware issue. As soon as I walked up to the servers, the real reason was apparent, though—a very nice, and very processor-intensive, screensaver was set to start after 15 minutes of keyboard and mouse inactivity.

One of the biggest problems I see with server configurations regards the use (or misuse) of hosts files. As any network technician knows, the hosts file allows for host name resolution back into an IP address. However, if the contents of the hosts file is incorrect, or the host file itself is absent (that is, perhaps renamed to *host*, rather than *hosts*—another real customer issue of mine), and DNS is not being used, name resolution will simply not work. This problem can manifest itself in a “host” of ways—the SAP Basis installation will fail, the server will seem to be unavailable across the network, and so on. Best practices and experience tell us that the hosts file, if used, should reference all SAP servers in the system landscape, and their IP addresses. Each hostname should be listed twice, once in uppercase and once in lowercase letters. If applicable, cluster alias names should also be defined here.

Enough about configuring servers for SAP. Now, let’s move into one of the most challenging SAP Solution Stack layers, the disk subsystem.

General Storage Considerations

In my experience, perhaps 85% of SAP performance problems are considered to be storage-related. This is not to say that 85% of *all* SAP problems can be traced back to the storage system—but by the time I am called in, the easy fixes are typically taken care of. So I rarely run into simple profile parameter problems, or issues with other hardware subsystems, or problems with the network, either. Instead, after a quick

review of the entire solution stack I usually find myself drilling down into the following:

- The model and features of the disk subsystem servicing the systems that exhibit the slow performance
- Details as to how the disk subsystem has been configured
- Database statistics, to quantify how well the database itself performs on the given disk subsystem platform
- Specific transaction loads on the system (that is, batch processes or especially heavy online user transactions), especially the top 40 or so transactions identified by transaction ST03 as consuming the most database request time
- Finally, the ABAP programs (or other code) that is actually being executed by these top 40 transactions

Although a variety of disk subsystems are deployed today in support of SAP solutions, including direct-attached SCSI, Channel-Channel Arbitrated Loop-based storage, and a few NAS (Network Attached Storage) systems, the discussions that follow focus on the predominant disk subsystems being deployed today—Storage Area Networks, or SANs.

- ▶ To review the most common types of disk storage deployed for SAP environments over the last five years, see “Commonly Deployed Disk Subsystems for SAP,” p. 244 in Chapter 7.

Special Considerations for Storage Area Networks

Deploying Storage Area Networks, or SANs, for current and new SAP system landscapes has kept many SAP professionals quite busy for the last two years. The latest iterations represent not only some of the fastest disk subsystems ever manufactured, but also represent everything that customers need in a highly available and scalable disk solution. Not only does a SAN give us the ability to expand quickly, but it also allows us to move data around with ease. Snapshots, disk clones, data replication, and more satisfy high-availability requirements, and help us to address disaster recovery, too.

- ▶ To learn more about the role SANs can fill when it comes to disaster recovery, see “Disk Subsystem Single Points of Failure,” p. 180 in Chapter 6.

With these new capabilities comes new complexity, of course, and certainly new paradigms. In some cases, we also have a new set of issues that must be addressed when deploying the latest in SAN technology—switched fabric designs, planning for different data access models within the same SAN, providing connectivity to new

shared resources like tape drives and tape libraries, and all of the complexity that comes with deploying new software solution sets that interoperate seamlessly (or nearly so!) within the SAN environment. Moving on, let's sneak a quick look at some basic best practices for implementing SAP in a SAN environment, with the understanding that the newer "virtual" SANs will be covered in detail later.

Deploying SAP on a SAN—General Best Practices and Observations

To approach this in an organized manner, we will start with some general SAN observations, and then cover the servers that play a role in the SAN, move to the SAN infrastructure, and finally wrap up with the disk subsystem itself. The following list has been assembled as a result of more than a hundred SAP-on-SAN design or deployment engagements:

- As the SAN is a special network encompassing all components from the HBA to the disk, it is incorrect to call each cabinet of controllers and disks a SAN. Rather, the entire connected solution is the SAN. If the development environment is off by itself and is not connected in any way to production, for example, we have in place a Development SAN and a Production SAN. After these SANs are connected (that is, via a fibre cable connecting one SAN's switch to the other SAN's switch), it then becomes a single SAN.
- Each server needs at least a single Host Bus Adapter, or HBA. Because this represents a glaring single point of failure, it is always recommended to install and configure two where high availability is important. Both HBAs are cabled to the same SAN, but to redundant switches (discussed later).
- A server should not connect to two different SANs—connecting to multiple SANs concurrently is not supported. In other words, a single server with two HBAs should not be connected to two different models or implementations of a SAN.
- If a server contains two HBAs, it actually now has two paths in which to access data on the SAN. To manage this condition, an OS or OS-supported software utility is typically employed on each server connected to the SAN. Hewlett-Packard's line of StorageWorks SANs uses a utility called SecurePath, for example. SecurePath require a license for each server, but ships with all StorageWorks cluster kits—SecurePath can help to ensure that the HBAs and data access paths are balanced, too, from a performance perspective.
- Each HBA requires what used to be called a GBIC, or gigabit interface connector, as might each port in your fibre switches. Today, GBICs are more commonly called transceivers.

- Fibre cables then run from each GBIC in each HBA to a GBIC (or otherwise pre-enabled port) in each redundant set of fibre switches.
- In regards to fibre switches, different switch vendors may have different rules regarding cascading their switches (to increase the port count available to the SAN, for example). I highly recommend that the specific vendor's documentation is referenced for details, with the understanding that many SAN fabrics in use today are limited to very few levels of cascading, or *hop counts*.
- Switched fabric switches should be configured with dual power supplies. Two switches are required at minimum to address high availability. A fully redundant SAN/server configuration consists of four fibre connections—two to the server, and two to the disk subsystem. These fibre connections should be carefully mapped to the switches so that each controller is connected to two switches, not one.
- Either something referred to as “zoning” or something referred to as “selective storage presentation” is used to ensure that a specific server can only access a specific set of disk drives or virtual drives. The implementation of this varies with the vendor as well as the product set. Regardless, though, be sure to implement one of these access protocols. Failure to do so risks corrupting data.
- Now that we have established connectivity from each server to the SAN, we need to carve up disk drives on the SAN. Until November of 2001, some of the highest-performing SANs on the market leveraged up to six separate SCSI controllers across six different SCSI busses, or drive shelves. Each drive shelf was capable of housing 14 drives, for 84 disk drives total. It is best to view the disks vertically, in that we stripe data “up and down” the disk subsystem. A maximum of six drives (when six shelves are deployed) can be configured per logical drive in this manner. By addressing our SAN disk configuration in this way, we mitigate risk inherent in a bus or shelf or shelf-power failure. This in turn prevents us from ever losing data in the event of one of these failures.
- Whenever possible, I promote the idea of using a graphical user interface to address day-to-day SAN management, and a scripted command-line approach for actually configuring a SAN (to allow for rapid cookie-cutter standard SAN installations).
- For SAP we need several sets of disks, depending on which database vendor has been selected. All data and log files (including redo logs, archive logs, SQL transaction logs, TempDB, and so on) must reside on the SAN. Further, as long as we are *not* using a clustering technology that requires local access to database executables, the Oracle or SQL Server (or Informix, DB2, SAPDB, and so on) executables can be located on the SAN as well.

- If you are clustering, you need to add a Quorum drive or similar such drive as well (actually, a mirrored disk pair distributed across two different busses is recommended). The quorum *must* be located on the SAN. Also, clustering SQL Server or Oracle means that these database executables must be installed on each node in the cluster, not out on the SAN.
- For the production database, I nearly always recommend that the production datafiles reside on LUNs (drive partitions) set up for hardware-based mirroring and striping (RAID 1+0 or 0+1, depending on the storage vendor's implementation). For simplicity, I like to create LUNs where the physical drives are vertically situated one on top of the other.
- SAP BW represents a potential exception to this general configuration rule of thumb for production systems. In the fast disk subsystems available today, the trade-off between performance and disk space between the different RAID levels is not as significant as it has been in the past. I discuss these RAID levels in greater detail in Chapter 7.
- If all of the preceding items are in place, you are well-prepared to address database growth. As the database grows you can add another RAID 1+0 set, for example. The OS will immediately recognize this additional disk space (if not, go to Disk Administrator and execute the option to “scan for new devices”), and allow us to format this space and create a new drive letter. Then, in order to increase your effective database size, the database administrator simply needs to create more table spaces or data files on the new disk partitions.

A good example of what a traditional SAN deployment might look like from a disk-layout perspective can be seen in Figure 10.9.

With your new understanding of how to take advantage of the features and capabilities a classic SAN environment can provide in regards to our SAP environment, let's move ahead and discuss the latest in SAN technology—the virtual SAN.

The Latest in SAN Technology—Leveraging Storage Virtualization

In review, *Storage Virtualization*, or SV, is defined as “the transparent abstraction of storage at the block level.” Regarding SV, the idea is to minimize the need for parameters that allow fine-tuning, low-level optimization, and “tweaking,” in favor of allowing the intelligent virtual disk subsystem to take care of everything. It's a paradigm shift in the biggest sense of the word—hands-off!

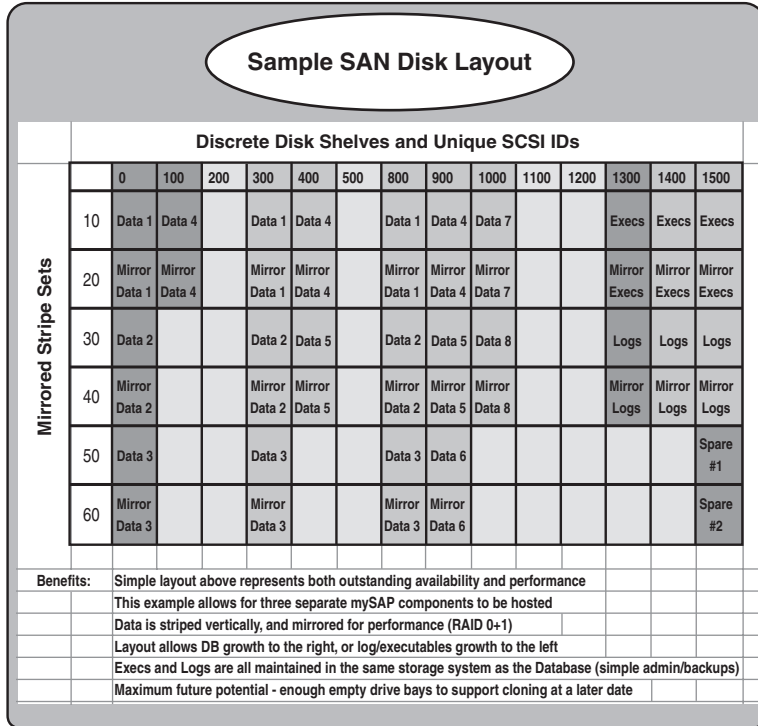


FIGURE 10.9 A well-performing and highly available disk subsystem requires that no two drives share the same storage shelf, power cable, or SCSI bus.

The ultimate in high-performance SV equates to striping and then mirroring all disk partitions across all 84–168 or more disk drives in a virtual storage disk subsystem. The ultimate in storage capacity equates to creating a RAID 5 partition across all of these drives. In either case, multiple partitions, or drives, can be created and presented to the operating system, or a single enormous drive can be created and presented instead. And many storage virtualization technologies automatically create optimally sized and configured RAID drives based on the space available in conjunction with historical data. It's that simple. The various block size parameters, read/write caching algorithms, read-ahead logic, and so on are all under the covers, so to speak. The best SV disk subsystems allow the few remaining configuration choices to be made via a browser-based graphical user interface, too.

- ▶ To read more about designing and sizing a Virtual Storage Array, see “Storage Virtualization—The Latest Paradigm in Enterprise Computing for SAP,” p. 246 in Chapter 7.

Virtual storage eliminates the physical one-to-one relationship between servers and storage devices, or databases and physical drives. Some important, albeit limited, configuration options still exist, however, and it is these options that concern us next.

Storage Virtualization Best Practices

When it comes to implementing and configuring a Virtual Array, you have control over the following options:

- As with a traditional SAN, you can still design and implement high-availability SAN switch architectures, including the ability to create zones for effectively segregating specific servers and their storage across multiple storage systems.
- You can create one or more groups within a physical Virtual Array storage system. Groups represent another way to segregate storage and servers, though this capability is limited to disks that are physically attached to the same disk controller subsystem (normally in one or more “cabinets” or “disk enclosures”). Thus, in an 84-drive Virtual Array, you might choose to create two groups of 42 physical drives each, three groups of varying sizing, or just a single group containing all 84 drives.
- You can manually create LUNs, or disk partitions, and select the group (and therefore disk drives) over which this LUN will be created. You can create many LUNs, or one LUN, whatever is optimal for the need at hand. For example, you might create three 100GB LUNs in a single group consisting of 42 drives, and place your SAPDATA files on these.
- Upon creating the LUNs, you can typically specify what RAID level each LUN is to use, and whether caching should be enabled or disabled (although in some virtualization product sets, this is simply not possible). In this way, if required of your solution, you can mix high-density RAID 5 LUNs (for database disk dumps, for example) with high-performance RAID 1+0 LUNs (for database data files and logs).

Some of the preceding points defy the preaching of many a database administrator or basis consultant. I can hear them now. “What, one giant virtual disk chopped up into pieces? You can’t do that, we’re supposed to keep our logs separate from our data! What happened to best practices?” Let them know that best practices differ now, based on the storage solution employed. And recommend that they read the PDF files found on the Planning CD that relate to configuring and optimizing, for example, HP’s Enterprise Virtual Array Storage System.

Finally, let the facts speak for themselves, as presented in Figure 10.10—the Virtual Array easily handles a variety of workloads, while other traditional high-end storage systems struggle.

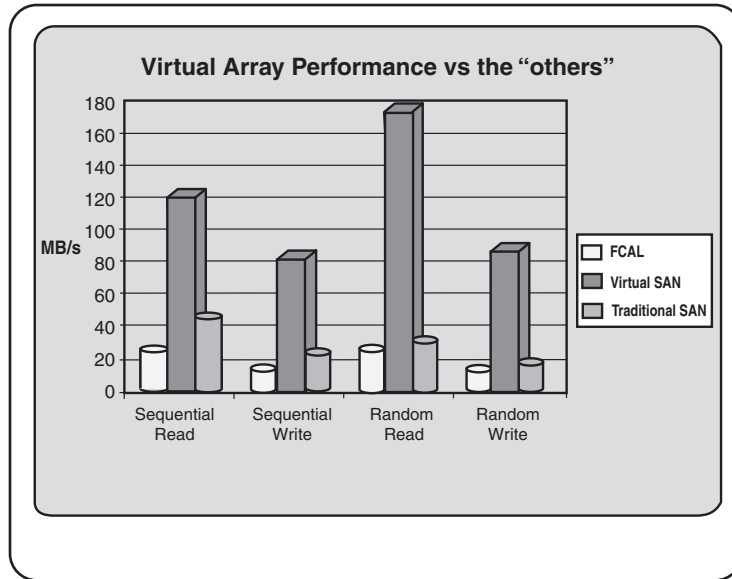


FIGURE 10.10 This chart shows the results of actual I/O throughput tests (measured in Megabytes per second), comparing a Virtual SAN Array to both a legacy FCAL-based SAP Production disk subsystem and a traditional high-performance SAN.

- For more hard data comparing the Virtual Array with legacy storage systems, see “Additional Stress-Testing Goals,” p. 613 in Chapter 16.

On the Road to Implementation

Before we close out discussions on designing and implementing the SAP Data Center, it’s important to look at how everything up to this point needs to be sustained. SAP deployments progress in phases or “waves”—each wave brings with it another collection of data center resources that must be planned for, installed, configured, and managed. For example, most new deployments start with bringing in the Technical Sandbox and Development systems. Later, the Test, QA, Integration, Staging, Training, and other environments are introduced to the Data Center. Eventually, the Production and Disaster Recovery systems are phased in as well.

Planning for SAP Data Center Operations

Before most of the SAP Technical Support Organization staff involved with setting up the SAP Data Center walk away, I need to cover some details related to managing the SAP Data Center. At this stage in the SAP project, we are more concerned with day-to-day and similar “tactical” operations than anything else. The strategic operational needs of the deployment will be addressed fully later.

I assume that an SAP-aware Enterprise Management Application has been deployed. That is, I assume that software agents are loaded on each SAP and SAP-related infrastructure server, and that a Management Console has been put in place to manage all SAP Data Center resources. I also assume that your Operations and SAP Basis teams have been trained on how to fundamentally use this all-important management tool, with the understanding that the learning curve is still pretty sharp over the next month or two.

In my experience, good SAP computer operations start with not only the use of an enterprise application, but also with *paper-based daily and weekly checklists*. Why the reliance on paper and pencil in this age of automated computing, and after spending tens of thousands of dollars on an SAP monitoring tool? Three reasons:

- These paper-based checklists will force SAP Operations to *learn how to effectively monitor an SAP environment*, using SAP-native tools like SAP CCMS—the same source of monitoring and management data leveraged by the popular SAP enterprise management suites.
- These checklists provide *backup*, should the Enterprise Management tool be unavailable (or the rolled-up or alert data unclear to the marginally experienced SAP support team).
- After a completed checklist becomes mandatory during shift-turnover, these checklists encourage accountability; they force the operations team to actually do the job of monitoring your mySAP solution.

Eventually, the paper-based checklists tend to lose their effectiveness from a training/knowledge perspective. At that point, the SAP Operations team often puts them back on the shelf, to be dusted off again later by new-hires and other new members of the support staff as they come on board. Some of my customers, though, continue to use their checklists long after their Enterprise Management tool is in place and functioning correctly, and the operations team is ramped up. They simply modify the checklists to better reflect the operational needs of the organization, or add the discrete tasks from the SAP checklists to the tasks and responsibilities of a larger-scope document maintained by the Data Center staff. And other accounts continue to use these checklists simply because they are effective; they still force a certain level of accountability, as long as the SAP Data Center Manager requests that they be used.

- ▶ To learn much more about operational and systems management activities in regards to SAP Data Center Operations, see “Back to SAP Operations,” p. 503 in Chapter 14.

Special Considerations When Deploying the Technical Sandbox

The first time SAP is installed in the Data Center will likely be in the Technical Sandbox (“sandbox” or “crash and burn system”) or similar testing environment. Oftentimes, the sandbox will look nothing like the Production system is expected to look. In the haste to get *something* running quickly, many SAP implementations introduce what can only be described as a stripped-down version of what will ultimately grow to be the actual Technical Sandbox.

The sandbox provides valuable hands-on experience to the SAP Basis, DBA team, and the SAP Operations team. It allows each of these technical groups the opportunity to see an SAP installation in action, and begin to realize the fruits of everyone’s labor up to this point. Further, it allows the entire SAP Technical Support Organization the chance to verify what it already should suspect as “holes” in the organization (these holes are addressed later in Chapter 12). Specific SAP components like Enterprise Buyer Pro and Enterprise Portal require knowledge and expertise in areas outside of simple SAP Basis installations, for example.

As the SAP Data Center grows and matures, the Technical Sandbox will grow with it. As a result, the following needs arise:

- Every SAP component needs to be installed and tested in the sandbox prior to installing it elsewhere. Thus, because it’s unlikely that anyone wants to maintain a sandbox with multiple SAP instances, robust tape backup/restore capabilities need to be introduced in the Technical Sandbox eventually.
- Every layer in the SAP Solution Stack must be represented within the sandbox. For instance, if a Web component like ITS (Internet Transaction Server) or Web Application Server will play a part in the Production system, it also needs to be represented in the Technical Sandbox.
- Finally, the SAP Operations team should begin to grow comfortable with managing and maintaining SAP, as I discussed earlier. This can be accomplished in a number of ways. For example, Operations should begin their “career” in SAP monitoring by starting with the Technical Sandbox. I’m talking about more than the SAP-aware Enterprise Management console. Other hardware/software-specific management consoles, like Insight Manager, HP Openview, Windows 2000 Performance Monitor, UNIX command-line performance utilities, and so on, provide must-have snapshot-in-time and historical trend-analysis value.

With the knowledge gained in performing the preceding tasks in support of the Technical Sandbox, the SAP TSO will be better positioned to address the needs of the next truly critical system within the SAP landscape—the SAP Development environment.

Special Considerations When Deploying the Development System

When the development system is installed and in place, for all intents and purposes it *becomes* a productive system—not for end users in the traditional sense, but for the expensive Functional experts and SAP ABAP programmers who will customize this system for many months to come. Therefore, the Development system needs to be treated like a Production system with respect to the following:

- **Downtime**—Bringing down the development system is unacceptable without plenty of warning. That is, a schedule for planned downtime must be developed, to address various operations and administrative tasks. These might include performing backups, executing client copies and refreshes, and so on. Why? First, because the SAP Functional experts and ABAP programmers are not cheap, and their time is therefore valuable—without a development environment, these folks will only sit idly by, still billing hourly fees that can easily add up to tens of thousands of dollars for the entire development team. Second, because downtime results in implementation schedule slips, and costs even more down the road.
- **Monitoring**—A proactive approach to monitoring the development system needs to be put in place immediately, both to minimize unplanned downtime and to quantify that the development system is performing well.
- **Backup/Restore**—It is absolutely critical that the work being performed by the development team can be safely backed up to tape, and preserved offsite in the case of a disaster. Why the seriousness? Because the work that goes into the development system to prepare for a productive system can easily equate to many millions of dollars in productivity.

By now, you should be well into your SAP implementation. In the next chapter, I will step back and discuss in detail the SAP Basis installations particular to most of the popular SAP components—R/3, BW, APO, CRM, EP, and more.

Tools and Techniques

For this chapter, the Planning CD contains the following files:

- A “Rack Best Practices” checklist, with tips and hints on how to best rack and stack your enterprise computing gear.
- Two documents in PDF format—these detail some of the configuration and performance options related to deploying HP’s Enterprise Virtual Array storage system.

- Daily and weekly SAP Operations checklists.
- Each of the figures found in this chapter, in Microsoft PowerPoint format, which include highly available network solutions, disk subsystem solutions, a SAN design template, and more.

Summary

Chapter 10 covered nearly everything that needs to take place or be addressed when it comes to developing your SAP Data Center. Here, we worked our way up the SAP Solution Stack, stopping along our implementation roadmap to discuss data center physical requirements, power and cooling considerations, the need for a robust network infrastructure, setting up racks and servers, configuring traditional disk subsystems and even the newer Virtual SAN Arrays, and more. And we also touched upon some of the basic operational activities that need to be addressed after all of these data center computing resources are in place.