

### More On Lifecycle Aspects

This page contains supplementary material for part one of *Performance Assurance for IT Systems*, which roughly follows the system lifecycle. It is divided into:

- **downloads** are either spreadsheets or longer pieces of text that are contained in pdf files
- **observations** are typically short, in-line pieces that are considered to be useful additions to the discussions in the various chapters. It is obviously assumed that you have access to a copy of the book to understand the context of the additional material. You can [download all observations](#) (pdf)

Comments are welcome.

### Downloads

You need to register to be able to access any downloads that are flagged as password-protected. Note that registered users will be automatically emailed when new material is posted on the site.

- [A sample hardware sizing spreadsheet](#) (Excel format - Office 2000) that is based on the worked example in Chapter 5 (password-protected).
- [Sample checklist for evaluation of stress testing tools](#) (pdf)

### Observations

The specific section in the book to which an observation relates is given in the heading unless it is a more general topic.

#### **Contents:**

[Equal Performance for a Fraction of the Price](#)

[Technology Associations and Forums](#)

[SPC – Industry Benchmarks for Disk Subsystems](#)

[Benchmarking Tricks Of The Trade – Vertical Scaling](#)

[More Benchmark Tricks Of The Trade](#)

[Volumetric Requirements “By Numbers”](#)

[More On Sizing Information](#)

[Working With Limited User Volumetrics](#)

[More on Hardware Sizing During Procurement](#)

[Sizing for a Technology Refresh](#)

[More on Hardware Sizing Reviews](#)

[Procurements: Five Minutes to Midnight and Time for a Different Solution](#)

[Bid Consortia](#)

[Logons](#)

### Production Planning

#### **Equal Performance for a Fraction of the Price (Chapter 2)**

Here is an example of an enterprise level server that was attractive to an organisation because the price was less than half that of the competition. It shows some of the factors, some of which may be non-technical *per se*, which can be used as part of a technical evaluation. The question to me was did it actually provide equal performance. The factors which made me dubious were:

- The marketing literature did not appear to be going “head to head” against comparable high-end servers from other vendors
- The technical literature that was available for this server and its related family was fairly scant in comparison to (a) the competition and (b) this vendor’s other range of servers
- There were no published benchmark figures for the high-end server under consideration
- The fact that it was “new” was used as a reason for the above omissions. In my experience, such items are invariably in place as part of the (pre)announcement of the product. What tends to slip is the shipping date of a workable system.

#### **Technology Associations and Forums (Chapter 2)**

It has become *de rigueur* for vendors that are involved in a new technology to join forces. These technology associations, occasionally called forums, invariably have web sites to promote the technology. These web sites often appear on the first page of hits when a web search for the technology is performed, as researchers (including myself) are naturally drawn to them. I have to say that the majority of such sites are disappointing to individuals who are looking for in-depth information. They seldom provide any detailed technical background or papers on the technology, often limiting themselves to relatively insipid overviews. They appear to be rudimentary marketing vehicles. I am certain that they could do much to improve the value of such sites.

#### **SPC - Industry Benchmarks For Disk Subsystems (3.3)**

[SPEC](#) and [TPC](#) are industry benchmarks that are discussed in the book. A notable addition is [SPC](#) (Storage Performance Council). Its members include the major server vendors plus hardware and software vendors in the storage arena. The current benchmark is called SPC-1 whose primary objective is to measure random IO. SPC-2 and SPC-3 are planned: SPC-2 will focus on sequential IO (a draft is available on the SPC web site); while SPC-3 will cover protocol independent, application level storage performance.

SPC-1 mandates three discrete areas of disk that will be accessed during the tests. They are called Application Storage Units (ASUs) and they reflect different types of usage: data store, user store and log sequential write. Each ASU is accessed by a separate workload, i.e. there are three workloads. A workload can be subdivided into a number of parallel streams. In SPC-1, the data store workload is composed of 4 streams; the user store of 3 streams; while the log sequential write has just one stream. The data store streams, which constitute 59.6% of the total IO, consist of: read/writes randomly distributed across the whole ASU; highly localised IO in specified areas of the ASU; and sequential reads. The user store streams (12.3% of the total IO) consist of: read/writes randomly distributed across the whole ASU; and highly localised IO in specified areas of the ASU. The final workload (28.1% of the total IO) consists of a single stream of logging and other sequential write activity.

The summary of results in the disclosed executive summary contains: IOs per second; price-performance; total ASU capacity; data protection level (e.g. mirroring); and price. The full disclosure report includes: the distribution of IO throughput over time; and the distribution of response times.

### **Benchmarking Tricks of the Trade – Vertical Scaling**

One of the primary objectives of any vendor is to demonstrate that their product scales, ideally both vertically and horizontally. Reasonable vertical scalability (say a factor of 1.8 if the hardware is doubled) can be difficult due to the basic overheads that are inherent in any SMP-based system. One trick is to make use of new minor releases of operating system or other software, possibly a DBMS, where they include tuning enhancements that may improve performance by 15-20%. The smaller system (say a 4-way SMP) is benchmarked using the previous version(s), while the larger system (say an 8-way) is benchmarked using the improved version(s). In favourable conditions, it may appear that the 8-way SMP provides twice the performance of the 4-way, which of course it does not, at least not on an apples versus apples basis. A usual giveaway is when the larger system (the 8-way in our example) magically produces more than twice the performance!

### **More Benchmark Tricks Of The Trade (3.7)**

I recently came across a variation of the "comparing apples and oranges" theme. On this occasion it was not a vendor; it was in fact an academic team. They were attempting to compare their software against a range of similar commercial and non-commercial offerings. The benchmarking and the discussion of the results were fairly comprehensive and impressive. However, they then took the additional step of performing a comparison with a mainframe-based solution. They proceeded to claim that their software performed better than the mainframe on a server that was a fraction of the price. They failed to point out, presumably in their enthusiasm, that while their claim was probably valid on a uniprocessor, their software could only run on a uniprocessor, the mainframe solution was not constrained in this one important respect. This selective and potentially misleading comparison was a blemish on an otherwise excellent piece of work.

### Volumetric Requirements “By Numbers” (4.3.4)

In the chapter on Non-Functional Requirements and Solutions I mentioned the tendency of organisations that have provided incomplete (or no) volumetrics to bidders to invent figures, well into the procurement process, usually when they are struggling to compare solutions from the various bidders. It is common to find a belated request to the bidders to provide solutions and costings for “1,000 users, 2,000 users *et cetera*”. This approach is perfectly valid, particularly for a new business venture where ultimate success or failure cannot be determined. However, it would be infinitely preferable if the original ITT documents outlined the requirement in these terms.

Notwithstanding the timing of the requirement, my simple wish is that such volumetrics should bear some resemblance to reality. In one particular procurement, the lack of volumetrics led to detailed investigations into the volume of business, which was reasonably finite and hence predictable, and into the number of staff that would be required to support the total workload. Ultimately, it was possible to arrive at reasonably plausible levels of user concurrency on the proposed system (+/- 30%). The belated figures from the customer indicated that they perceived the minimum concurrent user population to be 6-7 times greater than the figures that we had previously derived. It was difficult to understand the rationale behind the customer’s figures, other than the customary tendency to err on the conservative side. It followed that the hardware costs and software licence costs increased significantly. I am not sure quite how this fitted in with the customer’s request of “value for money”.

Returning to a more coherent use of this approach, many organisations use the technique as part of a commercial ploy to obtain a cost per  $x$  users, where  $x$  may be anywhere between 100-1000. The expectation is that 2,000 users will cost twice as much as 1,000 users, i.e. the cost increase will be incremental in line with the increase in the number of users, as opposed to any step changes. While there are applications where this expectation can be satisfied, mobile communications is arguably the most obvious example, equally there are applications where the requirement can be extremely difficult, if not impossible, to meet. The primary obstacle on many systems that grow significantly is often the scaling of the central database server. Artificial partitioning over multiple, modestly sized servers can easily result in maintenance, management and performance problems. Therefore, be wary of trying too hard to find a technical solution for what is, purely and simply, a commercial problem.

### More On Sizing Information (4.4)

A recent procurement highlighted the need to push hard for volumetric information. The procurement was one of the lengthier varieties, so lack of time was not a valid excuse for failing to produce a set of coherent volumetrics. The number of bidders had been gradually whittled down to two. The available information was limited to the number of registered users by area. The client promised to carry out a detailed volumetrics study and to make the results available to the bidders. Initially, it was stated that the study had been

completed but that the results were considered to be unsatisfactory. The figures were promised soon but they never appeared and the bidders, who could eventually wait no longer, were forced to proceed without this necessary information.

Let us assume for the moment that the client was telling the truth. Why did they not share the data with the bidders and ask them what they thought; perhaps the data may have been useful, either in its basic form or in a revised form after some agreed remedial work. Nobody will ever know; the client did not choose to be open and seek help; and the bidders failed to press the client sufficiently. If we take the cynical view that there was no information, the bidders wasted time waiting for data when they could have been getting on with the sizing. As stated in the book, the quality of the sizing in this sort of situation will inevitably suffer, and there may well be adverse price implications.

Continuing on the volumetric theme, another organisation had no clear idea how web-based training might be used in their organisation but they wanted the ability to support it, whatever it was! It also followed that they had no volumetrics to apply to whatever functionality they might use. Their approach was to get each bidder to tell them what facilities they would use and how many users there would be. While there is undoubtedly a case to be made for using the bidders' experience on any previous implementations, it should be done before any ITT is issued so that any combined "wisdom" is expressed in the requirements. Attempts by the procurement team to do this stuff on the fly will simply make any comparison of bidders' solutions even more difficult, particularly as the information was not used to provide a level playing field for the bidders.

### Working With Limited User Volumetrics (4.4 and 5.4)

There is a growing tendency for companies to simply specify the number of staff within the organization, hardly a useful metric to aid the hardware sizing process. Occasionally, it may be accompanied by a terse "assume that  $x\%$  will be active during peak periods" where  $x$  is usually a largish and unrealistic off-the-cuff figure in excess of 60%. The first step is to verify the likely user population at peak times. For example, take due account of the following factors which can reduce the concurrent user population significantly:

- Shift or extended hours working patterns can obviously reduce the number of staff that are working at any given time
- Members of the workforce who may be out of the office for a large part of the day and hence may not be able to use the system at these times
- Part-time staff
- People who are not permanent members of staff and may only fill in on an ad hoc basis, when required
- Other reasons, e.g. make allowances for sickness, training, courses, and meetings
- Any physical limitations, e.g. 50 seats in a Call Centre
- Supervisory staff / management who may make little or no use of an operational system.

## Performance Assurance for IT Systems

---

Use the guidance in the book to estimate the percentage of the resultant number of users who will be concurrently active; where short, sharp sessions are the norm the level of concurrency is likely to be on the low side, whereas long sessions (several hours) will typically see a high rate of concurrency.

Where information on the potential workload volume can be obtained, it should be used as a means of checking the validity of the concurrent user estimates. For example, regional or national statistics on the number of patient visits to the doctor (GP) per year and the number of prescriptions dispensed may provide useful insight into the number of doctor sessions that are required to support a given patient population, (say) assuming 10-15 minutes per consultation.

As emphasised in the book, it is important to note that hardware sizing is mainly concerned with active users; an active user is one who is executing application tasks, examining / using the output, thinking, or typing, activities which occur at varying but regular rates of frequency. An inactive user is a person who is simply logged on and does a negligible amount of work that involves the application. Inactive users can usually be ignored on stateless servers but they will impose some form of memory overhead on stateful servers, although the precise impact will vary, depending on the software and the implementation of the virtual memory management system within the operating system, e.g. it is more of an issue on a product such as Microsoft's Terminal Server where each logged-on user may have a sizeable virtual memory footprint.

### More On Hardware Sizing During Procurement (4.6)

The web-based training procurement that is mentioned above in "More on Sizing Information" went one step further by asking the bidders, although not in any formal documents, to provide "auditable evidence" of the accuracy of their sizing calculations. As discussed in the book, hardware sizing, particularly early in the lifecycle, is frequently more judgement than science. This request smacked of individuals trying to protect themselves at all costs. It goes without saying that this is not a productive approach.

### Sizing for a Technology Refresh (Chapters 4, 5 and 6)

It is now *de rigueur* in many procurements to address the need for a "Technology Refresh" in year 4 or 5 of the production system. Interestingly, the seemingly accepted definition of "Technology Refresh" is limited to the replacement of hardware components. The most common mistake is to assume that the system will remain relatively static in terms of both its usage and its technical / functional complexity. A second assumption, that hardware speeds will increase over time (e.g. Moore's Law), all too often results in conclusions that the hardware costs will be significantly reduced in 4 or five years time. In the real world very few things are static; there is constant change and evolution both in business and technology terms. Here are some points to consider before jumping to any optimistic conclusions such as replacing today's enterprise server with a workgroup server in four years time!

- Assume a minimum 10-15% increase in server usage per year just to keep pace, viz. moderate functional changes, new versions of underlying software, e.g. DBMS, performance tuning (using more grunt to get better response times through parallelisation), *et cetera*. Similarly, allow 20-25% increase in disk space requirements unless such growth has already been included in the basic sizing calculations. Any significant functional changes are obviously not included above. However, they should come under change control so that additional hardware requirements can be justified
- "Software gets slower quicker than the hardware gets faster" (otherwise known as Wirth's Law). This phrase mainly alludes to the inexorable rise in the complexity of the underlying software. If a system has a projected lifespan of 10 years with room for a single technology refresh around the halfway point, what chance is there that new or revised technologies will be introduced at some juncture, items that you are not even aware of today? Just think back to 1995 (if you are old enough!) and some of the changes that have occurred since that time: object-oriented systems; web-based systems; Java and Com / .Net; XML; messaging *et al*. The increase in complexity inevitably brings with it the requirement for more hardware
- Beware of possible side-effects of any technology change. For example, a move to a thin client may have a significant impact on network bandwidth requirements and central infrastructure-related components (firewalls, load balancers, web servers, LAN design)
- Additional hardware that may be required to rectify any original design, under-sizing or performance problems (see Chapter 12 of the book, *So What*)
- Do not limit the costing to hardware replacement. Ideally, make some allowance for the costs of new or revised technology software products plus implementation costs.

If there is uncertainty surrounding the likely degree of change I would recommend that it is assumed that the future hardware replacement costs will be similar to the current hardware costs, i.e. assume that the various effects will be broadly neutral in cost terms.

### [More On Hardware Sizing Reviews \(5.7\)](#)

#### *5.7.3 Unproductive Approaches*

This is another example of a client trying valiantly to stick as close as possible to the budget that resulted from the original business case. In this instance provision had been made for resilience by configuring two production nodes. The original sizing made no mention of running the nodes in "active-active" mode, and indeed the figures used seemed to make no allowance for the overheads that would be expected from running a multi-node database. When faced with a recommendation for more hardware based on

the assumption of an “active-passive” cluster when the sizing was revisited, the line taken was that it had always been assumed that an “active-active” cluster had been assumed all along. It is possible that they may just get away with this assertion, depending on how the system is designed and implemented but the likelihood is that they will be caught short.

### Procurements: Five Minutes to Midnight and Time for A Different Solution (Chapter 6 mainly)

I witter on in various places in the book about the propensity of bidders to alter their solution radically just prior to the proposal going out of the door. The driver is invariably a desire to reduce the cost of the solution. From a hardware perspective it involves switching to different vendors who can magically offer the same power for a fraction of the price. This is quite a common phenomenon on proposals that are produced within (say) three months of the RFP (Request for Proposal) being issued. Read the book for “chapter and verse” on the issues that are encountered and how to avoid them. I mention it again here because you would think that the longer procurements, particularly those of one year and more, would be less likely to suffer from this malaise. Unfortunately, this is frequently not the case. Last minute change is infinitely more painful for longer procurements for the simple reason that the proposed solutions will be significantly more detailed than for the shorter proposals, making changes more far-reaching and, with midnight approaching, more prone to error. Here are some observations (apologies to readers for any repetition):

- **Change of Hardware Vendor(s).** As stated in the book, decisions on any requirement from management to obtain costs from multiple vendors should be taken early in the procurement process. This is really a “no brainer”. However, late changes to small commodity boxes, e.g. single or dual processor systems, particularly where the same chip (Intel) and operating software (Windows or Linux) can be used, can usually be accommodated without too much trouble. The pain usually centres round competing claims for enterprise level servers and storage systems and the changes that may be required in the operating software and middleware. Unless you know better, stick to the maxim that you tend to get what you pay for, i.e. be very wary of extravagant claims and low prices.
- **Timing of Reviews.** As I have mentioned in Chapter 7 on Design I am a strong adherent of review activities that are carried as early in the process as is practical. A review that is performed in the last month of a one year bid process, which recommends significant change, is likely to generate bad feeling, not to mention the likelihood of introducing errors in the ensuing scramble to implement agreed recommendations. Comments that are made in early reviews are more likely to be seen in a constructive light and therefore more favourably received, and hopefully the amount of any rework will be minimal. In a one year bid early review activities should be performed no later than the halfway mark. I accept that a final review is necessary towards the end of the bid process but it should be more of a rubber-stamping exercise.

- **Objectives of Reviews.** Many reviews have no concrete objectives. This is coupled with the tendency of some reviewers to demonstrate their knowledge and experience by making recommendations that take no account of the time left available for corrective action. Pragmatism is the key here, particularly for late reviews, both in the setting of objectives and in the scope of recommendations. If the solution is truly woeful, let us say that from a performance perspective that “it will never fly”, it may be too late to rescue the situation; there is nothing to stop the reviewer from pointing this out as a general observation, but promoting wholesale change is unlikely to be productive.

### Bid Consortia (Chapter 6 – Bid Logistics)

In the book I covered some of the problems that surround multiple organizations who are working together as part of a consortium. The discussion was focused mainly, though not exclusively, on projects. This piece concentrates on bids.

From a hardware sizing and performance perspective, many bids suffer from a lack of solid sizing information until quite late in the process, making it difficult for the sizer and those who are responsible for the design of the technical infrastructure to meet the extremely tight timescales that are often imposed on bids, many initial responses now being required within a month. The fundamental problem is that too many ITTs (Invitations to Tender) are of poor quality, suffering from: ambiguity, a lack of clarity with respect to the requirements, and frequently no basic volumetric data. This inevitably delays the formation of a functional solution and invariably makes the designers nervous about providing necessary information to those individuals further down the solution chain who are looking at the non-functional aspects. In the book I have advocated the approach of iterative sizing over the bid cycle, almost irrespective of the state of the solution at a given time in the bid, as a means of avoiding eleventh hour panics. While this is achievable for an “in-house” bid team, it is more difficult for a bid consortium. In recent years there has been a strong likelihood, particularly on larger procurements, that a bid team will be made up of groups / individuals from multiple organisations. Apart from the prime contractor, there may be: one or more other companies that are contributing to the overall functional solution, particularly where products form part or all of the solution; a company that specialises in technical architects for the chosen technical platform; somebody who is looking at sizing and performance; a company who will provide a hosting service to run the eventual solution; and one or more hardware vendors or resellers. The presence of these disparate participants, each replete with their own baggage (each outfit almost always has its own in-house standards, culture, and *modus operandi* which, surprise surprise, differ slightly or sometimes even drastically from everybody else’s), naturally tends to make life more difficult. Issues include:

- The various third-party outfits are often expected to provide their parts of the solution almost in a vacuum, i.e. without a reasonably coherent understanding of the overall solution. Some third-parties, somewhat bizarrely as far as I am concerned, are happy to do this

## Performance Assurance for IT Systems

---

- Some of the third-parties may only be available on a part-time basis. This may be because that is all that they can manage (due to other commitments) or because the prime contractor dictates it. In either case, this may increase their lack of understanding of the solution
- the prime contractor limits the amount of information that is given to other members of the consortium. The only satisfactory reason for doing this that readily springs to mind is when a member is also part of one or more other consortiums that are also bidding for the same contract (this is most likely to happen with software product and hardware vendors). All things being equal, it is better to give too much information rather than too little. In particular, avoid making judgements on what you think that a given member needs
- The prime contractor does not allow essential documents that are produced within the consortium to be freely circulated, e.g. draft functional overviews of the solution. In one bid that I came across it belatedly transpired that there was a fairly detailed, and hence very useful, design document that was not made available to relevant third-parties. While there are frequently good reasons to withhold some documents, e.g. anything that is commercially sensitive, it is counterproductive if there is a blanket ban
- Consortium meetings may be restricted to discussing progress, i.e. there may be few, if any, meetings to progress the technical solution
- Meetings, and even conference calls, are not allowed between the various third-parties unless the prime contractor is present. As the prime contractor probably has insufficient resources for the bid, otherwise some of the third-parties would probably not be participating, this imposition is likely to impede progress
- Having said this, I have come across some third-parties who are reticent to deal with other third-parties, and who occasionally insist on dealing only with the prime contractor. This situation can arise where third-party A considers that they should have third-party B's piece of the action. Once again, this is unhelpful and delays progress
- Feedback from the prime contractor to third-parties on their deliverables is slow, or even non-existent, during the course of the bid. In the worst case, this can preclude an iterative approach to sizing and technical infrastructure. In some instances this can be caused by the prime contractor's design authority, a person who frequently has too much to do and who quickly becomes a bottleneck, particularly if he himself is only part-time on the bid!

These issues are not insurmountable. They can be addressed. The key is to pre-empt them by establishing the necessary ground rules and general *modus operandi* at the

consortium's kick-off meeting, rather than stumbling along in blind faith (or should that be forlorn hope?).

### Logons

Non-functional requirements to support logons are invariably limited to response time targets that tend to be unrealistically short, bearing little relation to the actions that may have to be performed as part of the process. If ever there was a case where it was justified to wait until the design stage to agree on response times then logon is it.

The rate at which logons can be handled may be extremely critical to the business. A large office-based system with many hundreds or even thousands of users may have a relatively short window at the start of the working day in which to complete the majority of logons, possibly as little as 20-30 minutes. In fact, the requirement can be even more stringent if the entire system fails during the period and users are forced to log back in. Therefore, it is arguably preferable to express the requirement in terms of logon throughput, e.g. logons per second rather than response time.

Logon processing can be lengthy and potentially resource intensive and pragmatic design decisions may be necessary on larger systems to avoid unnecessary delays. An obvious example is the use of Windows-based thin client technology where multiple Terminal Servers are used. One feature that is supported is the use of roaming user profiles. Here, a master set of profiles is maintained. When a user logs on his profile is copied to the Terminal Server that is allocated for this session. The profile includes the My Documents folder, which, if left to grow unchecked, will result in copious copying. This copying is repeated at logoff to return the updated profile to the "master" set. See the taster *Thin Client Technology for Windows-based Systems* for further information on this particular topic.

Stress testing for systems with large user populations should consider the need to run specific logon tests to understand what level of logon concurrency can be supported and where the bottlenecks lie.

### Production Planning

We are encouraged to view the initial roll-out of a system as being relatively straightforward, particularly as there are now features which can seemingly minimise the amount of manual effort that was required in the past. Unfortunately, planning often tends to focus on the initial implementation with little thought given to potential future problems that may be attributable to growth, issues that surround the management of large amounts of data, or short-term thinking. The following points are limited to performance implications although the subject is naturally much wider:

- **Data.** Miscalculations in this area can result in significant, not to mention tedious, remedial work. The recent tendency for some vendors to recommend a single RAID 5 array for an entire system should be treated with caution. The

approach may be perfectly satisfactory for small systems but it may cause performance problems on systems with any reasonable sized workload. The first question revolves around the wholesale use of RAID 5. It is attractive because it is the cheapest option. RAID 5 (rolling parity) is fine for data that exhibits low volatility, e.g. documents that are written once and read many times or low volume updates. It is not suitable for high volume, write intensive tasks. One of the most obvious examples of high volume, write intensive activity is a database log (containing after images and in some products before images as well). RAID 1 (mirroring) is preferable to RAID 5 for this type of data on any reasonable-sized system. Where appropriate, mix RAID 1 and RAID 5 to obtain the best performance at the cheapest price. There must be sufficient paths to the data at peak loads to allow satisfactory disk IO performance. Remember that the projected peak load may be several years into the future, not in the first couple of months. To this end, be wary of simply allowing data to be added in a haphazard way, consider the even distribution of data across the disk subsystem, making full use of the available host connections, disk controllers, and RAID arrays or individual disk devices in a non-RAIDed system. This may involve partitioning data over multiple table spaces or filesystems. Some products may ease the task, e.g. allowing partitioning over time. The question that you should ask yourself is “are the inbuilt features sufficient for the task in hand, or do I need to put in manual effort to achieve the necessary effect”. Where possible, isolate high volume elements such as the database log that was mentioned above so that their performance is not affected by other elements. Allow sufficient capacity and paths for the temporary areas that will be required for the inevitable data reorganisations.

- **Infrastructure.** If the infrastructure that will be required to support the projected peak load is not implemented for day one, it follows that it should be capable of being upgraded in a fairly straightforward manner. It is preferable to configure the appropriate core LAN switch at the start. An obvious extension to the LAN infrastructure that may be required in the future is the scaling of the front-end of a multi-tiered system, e.g. increased numbers of terminal servers (thin-client technology), web, or application servers along with the associated LAN capacity. In very large systems this may lead to a requirement for additional load balancing devices. Some firewall implementations have used a single physical firewall to provide protection at both the front and back-end of a system. Growth or under-sizing, particularly of the back-end traffic, can lead to the need to deploy multiple physical firewalls.
- **Server configuration.** Ensure that servers can house sufficient network connections to fit into the infrastructure that will be required to support the eventual peak load. Where appropriate, servers should be able to be upgraded, e.g. additional CPUs and memory, unless wholesale replacement is an integral element of the plan, possibly as part of a technical refresh.

