Architecting a Service Provider Infrastructure for Maximum Growth

By Stan Stringfellow - Special to Sun BluePrints™ OnLine

Sun BluePrints™ OnLine - June 2000

`http://www.sun.com/blueprints`

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303 USA
650 960-1300 fax 650 969-9131

Part No.: 806-5836-10
Revision 01, June 2000

Architecting a Service Provider Infrastructure for Maximum Growth

Scenario

This is the first in a series of Sun BluePrints™ Online articles that will examine the issues involved in building scalable and highly available service provider infrastructures. ISPs, ASPs, NSPs, corporate Web services, Telco services, and digital wireless network services all benefit from the principles that will be discussed in this series of articles.

The service provider (SP) market defies prediction. There are currently more than 8,800 SPs and growing according to theList.internet.com, despite the fact that analysts have long been predicting a major consolidation in the market. Why is this? Because existing SPs have been focusing on maintaining close, service-based relationships with customers. And those customers are making extraordinary demands for services.

There are at least four types of service providers today:

- Internet Service Providers (ISPs) offer internet access for consumers and small-to-medium sized businesses; Web hosting for consumers and businesses; and outsourced mail, security, and remote access.
- Application Service Providers (ASPs) offer complex value-added application hosting and access; consumer-based services such as Web-based income tax preparation; and business-based services such as enterprise resource planning (ERP) applications by subscription.
- Network Service Providers (NSPs) offer Internet access for ISPs and large businesses; Internet data centers—the future home office; everything from rack, stack, and ping to full application design and support; and more than just data since NSPs carry wireless and voice over IP traffic.
- Full Service Providers (FSPs) offer more than one of the above services.

The foremost concern of mostl service providers is the need to accommodate growth. Growth comes from several areas: expanding the number of customers, growth of the customers themselves (for example more traffic to their e-commerce sites), and growth that results from customers adopting additional services. Good publicity can result in unexpected spikes in demand. Even bad publicity can raise Web site traffic as much as 25% due to the “rubbernecker” effect!

There are several issues that must be considered when attempting to manage service provider growth:

- Cost-effective scalability—Does an increment in cost result in an increment in capacity?
- Containing management costs—Can you increase capacity without increasing management costs?
- Rapidly-expandable architectures—Can network architecture quickly scale to accommodate growth or changes in business models?

In this article, we will examine the issues surrounding horizontal scaling and vertical scaling—a critical area that you must consider when designing scalable architectures. We will then take a look at an example SP messaging architecture, as well as the sizing methodology that should be applied when planning such an architecture.

Horizontal and Vertical Scaling

Sun’s viewpoint is that growth is best managed by a combination of two types of scaling architectures:

- *Horizontal scaling* increases capacity by adding more servers.
- *Vertical scaling* increases the processing power of existing servers by adding more resources to each server.

Horizontal scaling is most effective when an application can be distributed across many servers. This architecture is often a good choice when an application supports distributed state or when load balancing can be used, and when software licensing costs are low.

You might use horizontal scaling to partition your architecture by service, allocating multiple servers to each service and load balancing across them:

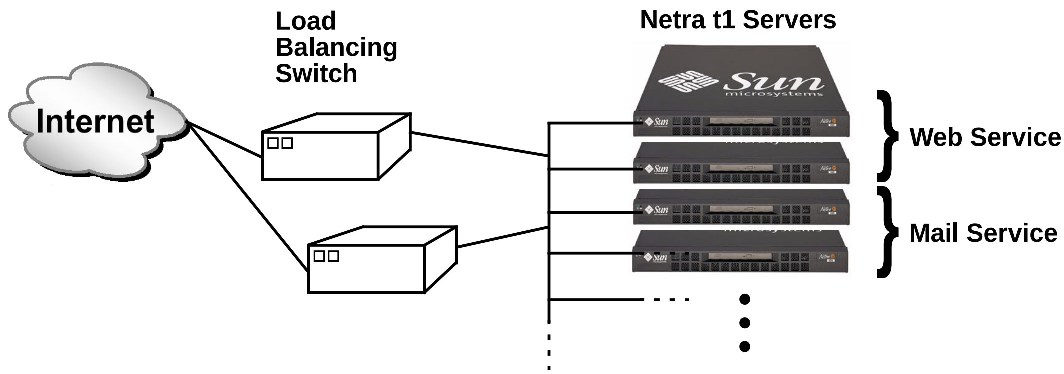


FIGURE 1 Partitioning by Service Using Horizontal Scaling

Note that several different algorithms can be applied to support load balancing, including round-robin, least response time, resource utilization-based, maximum number of sessions, and many others.

As your business grows, horizontal scaling can be used easily and effectively to further partition each service by task layer, allocating one or more servers to each layer:

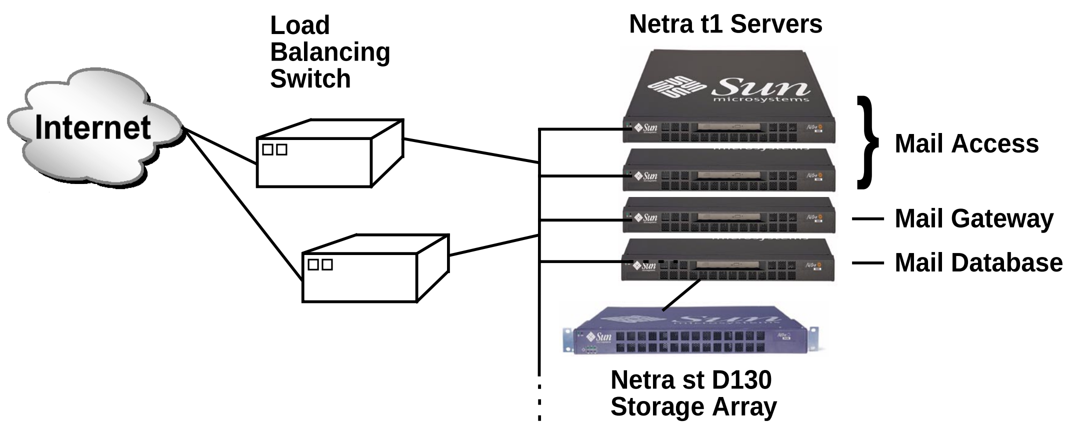


FIGURE 2 Partitioning A Service by Task Layer Using Horizontal Scaling

The advantages of horizontal scaling are:

- It is easy to deploy with small to medium levels of traffic.

- It is easy to support and maintain without bringing down any component of the architecture. You can simply add another server to the farm or configuration.
- There are predictable, linear costs across a broad range of capacity requirements.
- Higher availability results from load balancing.
- Serviceability is increased since small servers are field-replaceable (if those servers do not maintain unique persistent state that cannot be regenerated easily).
- Security is improved because penetration of one host does not necessarily yield access to the entire SP network.

The disadvantages of horizontal scaling are:

- The cost per unit of scalability may be higher.
- Power supply, chassis, and so forth are needed for each increment.
- Software licensing costs are higher in cases where licenses are sold on a per server basis.
- Management costs may increase proportionally with the number of servers, although best practices can help to alleviate this, and in some cases virtually negate it.

Vertical scalability is effective when multiple processes can co-exist or when multiple threads are used. Examples include Web, application, and database servers. Vertical scalability is sometimes required. For example, large database services can only be implemented using vertical scalability. This is in part because of the high I/O bandwidth that can be sustained on a big box such as the Sun Enterprise™ 10000 server.

The advantages of vertical scaling are:

- Capacity can be easily expanded if you have a server with sufficient headroom.
- Additional capacity is used efficiently by the SMP architecture.
- The management costs may be lower.
- The software licensing costs may be lower.
- The costs of adding capacity incrementally may be lower.
- Resource management techniques can be used to manage applications.

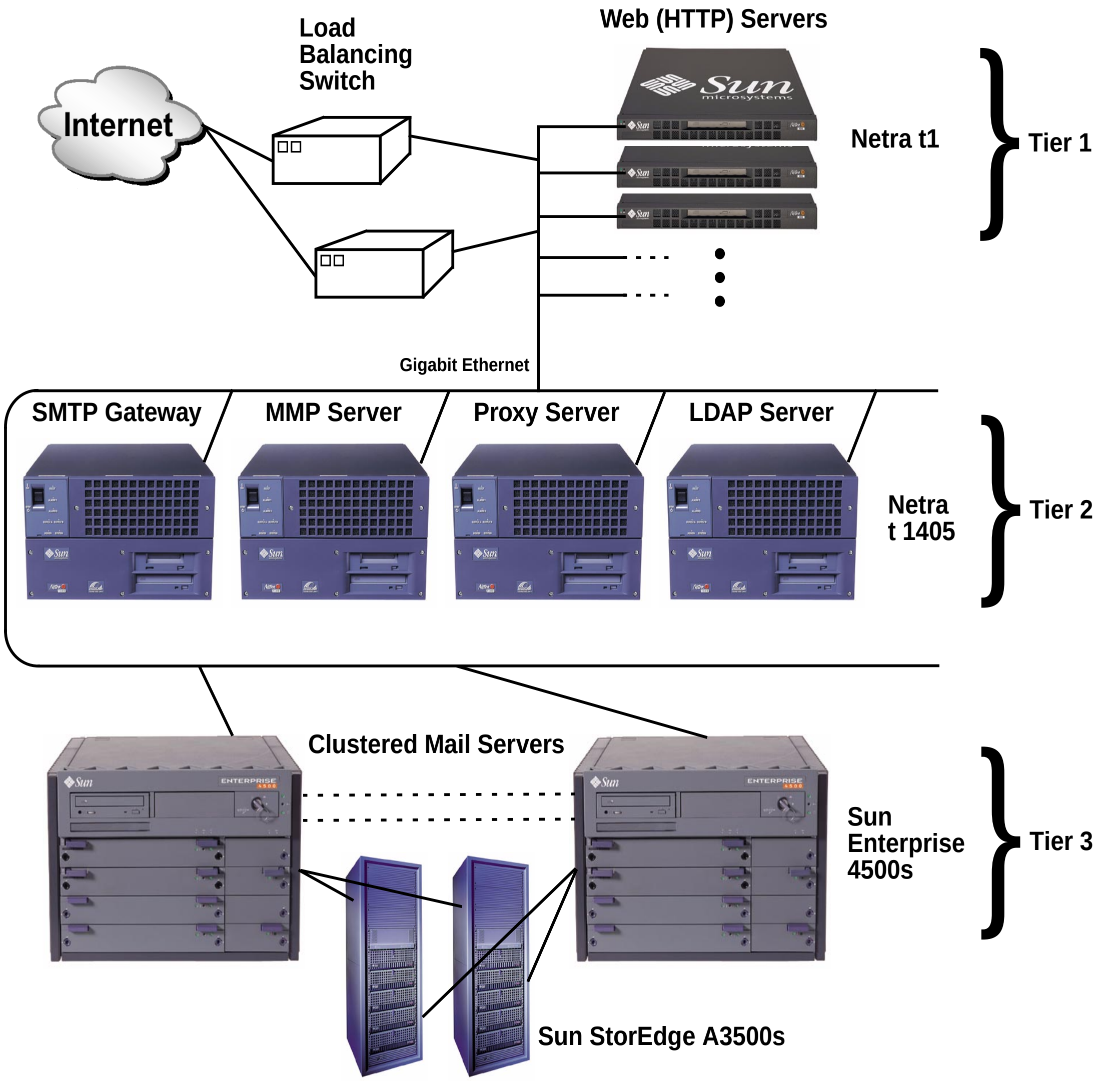
The disadvantages of vertical scalability are:

- When you run out of capacity, you must upgrade to a new server.
- Not all applications utilize threads or multiple processes well.

You should design your architecture—hardware and software—to exploit *both* horizontal and vertical scalability. Without both forms of scalability, your choice of how to respond to increasing workloads is limited. You might be forced into expensive infrastructure choices, when a good architecture could have prevented the problem.

SP Messaging Architecture

Here is a simple view of a highly scalable SP messaging architecture that is based on Netscape Messaging Server software running on Sun™ servers:



This example shows a flexible three-tier architecture that is scalable both horizontally and vertically. For example, the Netra™ t 1405 servers in the second tier can each host from one to four CPUs. Thus, if you needed to add capacity to the

SMTP service, you could either add more processors (vertically scale) or add another Netra t 1405 server (horizontally scale). It is often advantageous to use the same type of server, if possible, across the services network. This makes it easier to reallocate servers among the different services and task layers in order to tune the configuration and to respond to dynamic business needs and growth. In this example architecture, the Web servers are single-processor Netra t1 servers, while the rest of the service network is made up of four-processor Netra t 1405 servers. Although these are different types of servers, they all run the same Solaris™ Operating Environment which makes it easier to re-allocate them as necessary.

The SMTP server communicates with remote mail servers over the Internet. When a customer dials into the SP network (via a part of the architecture that is not shown above) and sends an email message, the message is placed into the message store on the mail server. The message is also queued for delivery. For remote deliveries, the message may pass through several mail hops before it even gets to the Internet. It is possible to use two SMTP gateway servers, one geared to send and one geared to receive, but this may be overkill since you are likely to run out of network bandwidth before you reach the capacity of the four-processor Netra t 1405 SMTP gateway server shown in this architecture.

All messages are managed in the message store on the mail server, in this case a Sun Enterprise 4500. This is essentially a very large database application, and can only be scaled vertically. The Sun Enterprise 4500 can host up to 14 CPUs, and has fully interleaved, low-latency memory with a capacity ranging from 64 MBytes to 28 GBytes. It also uses a Gigaplane Bus, a split transaction system bus that is designed for high capacity. This substantially improves I/O performance which is necessary for a large mail server database application.

The LDAP server, maintains information about where everything is located. For example, in a large SP site, there might be multiple message stores. Perhaps there are 3,000,000 users allocated to three message stores (with 1,000,000 users per message store). The LDAP server maps users to the proper message store, so that when a user logs in it is possible to associate that user with the proper mail account. Similarly, when Internet mail arrives from joe@some_isp.com, the LDAP server maps the message to the proper recipient. With LDAP, you can easily add, delete, or migrate users by simply changing their LDAP directory entry, and deleting or relocating their message store.

The message multiplexor (MPP) server is specific to the Netscape Messaging Server product. The MPP server provides a single point of contact for users. Thus, users connect to one host, the MMP, which decides where they will actually be directed to for their work. The MMP is like a traffic cop who knows what machines are available. For example, if a user is accessing the SP site from the Web, they are directed to an available HTTP server with the assistance of the MMP server.

The Proxy server in this example is a Web proxy server (as opposed to a mail proxy server). A Web proxy server handles things such as Web page caching. Proxy servers are often used in corporate accounts. These servers push frequently accessed Web pages to the edges of the network, and thereby speed delivery of content to the users.

Why are Netra Servers Used In This Architecture?

Why are Netra servers used in this architecture, as opposed to other possible candidates such as Sun Enterprise 250 or 450 servers? The reason is that Netra servers have special qualities that are becoming more and more popular among all types of service providers. The Netra products are designed for easy serviceability and can function in lights-out management (LOM) environments which are used by many service providers today. (LOM installations are often deployed in remote locations, and they must be up and running continuously without the benefit of an onsite staff.) The Netra products support:

- Remote power on/off control
- Remote console access
- Runs standard Solaris Operating Environment
- Local and remote alarms
- Status LEDs
- Front-accessible components
- Rear cabling
- DC power option

The DC power option, which is used in many Telco environments, is gaining popularity among service providers because the clean signal produced by DC power supplies means the servers are more reliable and easier to manage during power failures.

The Netra products are also notable because they incorporate Telecom carrier class features, including ruggedized packaging. All Netra products are Telcordia Network Equipment Building System (NEBS) Level 3 certified (see `http://www.telcordia.com`), as well as European Telecommunications Standards Institute (ETSI) compliant (see `http://www.etsi.org`). Compliance with the rigorous NEBS Level 3 criteria has long been a requirement for equipment used in the Central Offices of the North American Public Switched Network. NEBS Level 3 certification (the highest NEBS certification level) means that Netra servers have been shown to:

- Operate in wide temperature extremes of 0 to 50 degrees Centigrade
- Withstand very humid conditions with no water condensation inside of a system

- Tolerate 8.0 level earthquakes (zone 4)
- Remain impervious to dust and particle invasion
- Operate without being affected by environmental ESD or EMI
- Adhere to an optimized, standard Telecom rack mount form factor, including tight depth specifications which allow more units to be housed per square foot of floor space

The Netra t1 server fits into a one rack unit height, allowing 36 Netra t1 server to be deployed in a single standard Telco rack. Because of their form factor and rugged packaging, the Netra products can be deployed anywhere from a Telco Central Office to the back of a truck—as is actually the case in some instances!

The Netra products are specially designed and manufactured to meet the NEBS Level 3 requirements:

- Heavier gauge metal is used, with less plastic
- Internal nickel plating is applied to components
- Heavy duty air filtration is designed into the cooling systems
- Variable speed fans are used and are monitored by the systems
- Fire and flame blocking materials are employed

For more information about Netra servers, see `http://www.sun.com/netra`.

Sizing Methodology

In order to determine how best to configure and utilize servers running your SP messaging solution, and to plan for future growth, you need to analyze the computing environment to help ensure that it is sized appropriately. In particular, you should:

- Determine the usage profile
- Define the peak time
- Define the system load
- Measure system throughput
- Select an appropriate configuration
- Incorporate other server functionality sizing and make tradeoffs with other non-performance related requirements such as RAS and cost issues
- Prototype to check assumptions

We will look briefly at the first four of these tasks.

Determine the Usage Profile

Usage profile attributes, such as the number of users logged on to the system and the amount of work being performed combine to specify the amount of stress placed on a messaging solution system. As the number of users increases, so does the traffic generated. The solution must support this load at a minimum.

Table 1: Typical Usage Profile

Item Measured	Typical Value
Percentage of Configured Users Logged On	5%
Percentage of Configured Mailboxes Containing Mail	20%
Average Number of Messages per Populated Mailbox	5
Size of New Mail Messages	1 KB to 1 MB
Average Size of New Mail Messages	5 KB
Average Number of Messages Received per Populated Mailbox per Day	5
Average Number of Messages Sent per Logged On User per Day	2

Determine Peak Time

You need to determine the peak load time and ensure that your system can support it. The peak time—the most critical time to support users—varies from site to site. While it is not cost-effective for a server to handle the simultaneous use of all of its services by the entire user community, steps can be taken to ensure it handles peak loads well. To accomplish this, you must determine when the peak load will occur and for how long. For example, the peak time might occur for a half hour each morning and afternoon, or for one hour in the middle of the afternoon. The system should be sized assuming the user load for the entire day will occur during this peak time. Once such patterns are analyzed and understood, choices can be made that help the system handle the load and provide the services users demand. Experience at Sun and iPlanet, E-Commerce Solutions, a Sun-Netscape Alliance, indicates that a typical SP peak period lasts two hours.

Define System Load

You need to determine your user load in order for any sizing analyses to be useful. The user load is defined as a mix of operations, as shown in the following table.

Table 2: Typical SP Peak System Loads

Item Measured	Total Size of User Population				
	50,000	100,000	250,000	500,000	1M
Average Number of Users Logged on During Peak	2500	5000	12,500	25,000	50,000
Average POP Checks per Second During Peak	4.17	8.33	20.83	41.67	83.33
Average Messages Downloaded per Second During Peak	1.74	3.47	8.68	17.36	34.72
Average Number of Messages Sent (via SMTP) per Second During Peak	0.69	1.39	3.47	6.94	13.89

Measure System Throughput

To evaluate your SP messaging solution, you can use the Mailstone utility, a widely used and openly available stress-testing tool that measures solution capacity by testing how a system performs under load conditions. Designed to simulate tasks that users perform, Mailstone tests the load-carrying capabilities of a system configuration. During each test, clients make mail check requests, and send and retrieve mail messages. As the operations are performed, Mailstone determines how much data is moved with each request, as well as how quickly the task is performed. The throughput and response time calculations for each client are combined to determine the overall server throughput.

For information about Mailstone, see:

`http://docs.iplanet.com/docs/manuals/messaging/nms415/mailstone/stone.htm`

Conclusion

You should design your SP architecture for maximum scale in order to support rapid growth. Use a multi-tier architecture so that layers can be scaled individually. Incorporate both horizontal and vertical scaling to support rapid growth. And, use an appropriate sizing methodology to analyze your current and future capacity requirements.

Acknowledgements

Thanks to Judy Depuy who contributed to this article. Also thanks to Cari Yuen and Richard Elling for their guidance in writing this article.

Author's Bio: Stan Stringfellow

Stan Stringfellow is an independent author, technical writer and software developer who is currently doing work for the Sun BluePrints program. He holds a B.A.C.S. from UC San Diego and has written many manuals including the original softwre manuals for the Sun Enterprise 10000 (Starfire™). He may be contacted at stan@stringfellow.com.