White Paper

metagroup.com • 800-945-META [6382]

February 2005

The Role of the Adaptive Network in Service-Oriented Architectures

A META Group White Paper

Prepared by META Group for F5 Networks, Inc.

"For networkers to successfully deliver applications, it is not just a matter of adding more capacity or connectivity…. A higher degree of automation, integration, and architectural design is required, with network-based intelligence as the foundation."

Contents

Copyright © 2005 META Group, Inc. All rights reserved.

The Critical Role of Networks

IT infrastructure teams are struggling to keep up with the demand of legacy and new applications in the data center. The business that the IT organization (ITO) serves is itself under pressure to deliver more with the same or even fewer resources — new security regulations are placing demands on corporate governance and information protection, and issues of compliance, cost reduction, business performance management, and cross-business function processes are top-of-mind to business leaders. Increasingly, the business is leaning on IT to improve its own efficiencies.

Within the walls of the ITO, the pace of new applications is accelerating. Driven by ever-increasing demands, developers are adopting new rapid development methodologies such as RAD (rapid application development) or extreme programming techniques, and programming teams are unleashing a flood of new programs and services that run on the corporate network as well as extend outward to customers and business partners. One international pharmaceutical company has more than 8,000 applications, with the number increasing each month. In support of these applications — both legacy and new — the question becomes one of how the IT organization will achieve the objectives of flexibly securing access to these applications and ensuring their constant availability, scale, and performance. It is either cost-prohibitive or in many cases impossible to design these functions into the applications themselves — thus, the essential role that the network plays in unifying the application and its dependent infrastructure.

Without its network, the business has no applications, and without its applications, the business cannot function.

This is a big break from the past, where the network was frequently given secondary consideration and viewed simply as a means of connectivity. Yet without its network, the business has no applications, and without its applications, the business cannot function. For many, the loss of the network leads directly to a loss of revenue (see Figure 1). Yet the network traditionally has been inflexible and unable to respond to the application challenges that are ever present, whether they are security holes, lack of scale, performance issues across the WAN, or downtime. This is because the network has lacked the necessary intelligence to adapt and provide cost-effective services that can be used on behalf of the application.

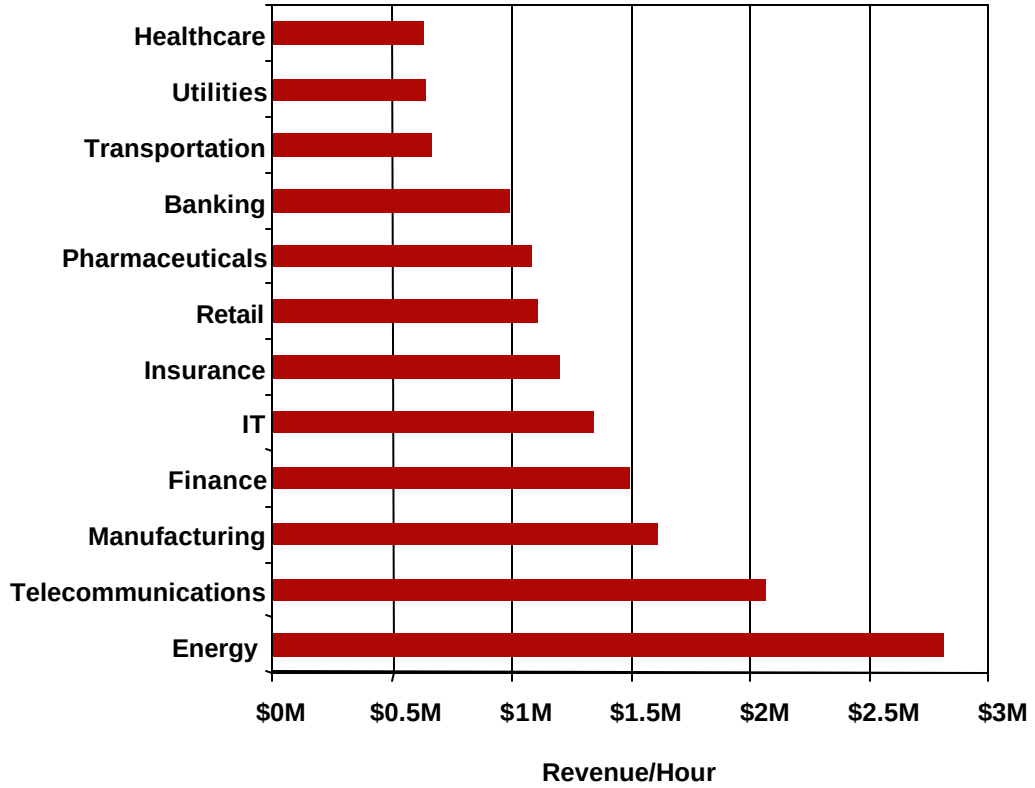
For networkers to successfully deliver applications, it is not just a matter of adding more capacity or connectivity. New approaches to application delivery — such as service-oriented architectures, peer-to-peer, and grid computing — radically change the flow of traffic between points on the network. Network technicians are

Copyright © 2005 META Group, Inc. All rights reserved.

in constant catch-up mode, with barely enough time to build what is needed, let alone understand application flows and optimize their infrastructure.

The typical reactive approach to addressing application challenges in the network is to apply point solutions, such as server load-balancers, acceleration devices, intrusion detection/prevention devices, or content transformation solutions. This results in high cost of infrastructure support, where the problem domain (the application) is transferred to the network in the form of many disparate devices performing singular functions from different vendors — which is exactly the sort of architectural design error that best-practice companies avoid. A higher degree of automation, integration, and architectural design is required, with network-based intelligence as the foundation.

Figure 1 — Potential Loss of Revenue by Industry Sector



Source: META Group

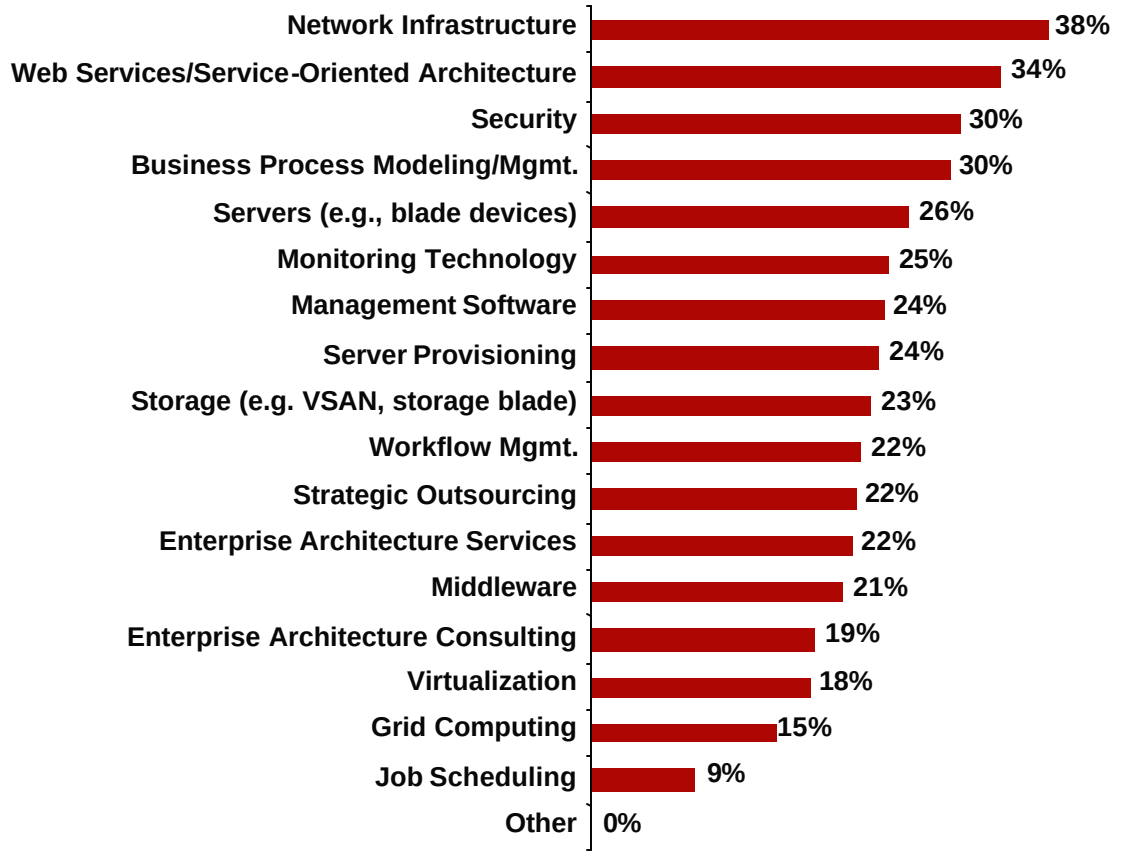
Infrastructure-related delays frequently are a factor in project overruns. Although the emphasis is rightly on the benefits of a new application to the business, the

Copyright © 2005 META Group, Inc. All rights reserved.

impact on the infrastructure is not assessed until later — often *too much* later to do anything about it. Of new application projects, 65% have a predetermined deadline, with little time allowed for infrastructure enhancements. As a result, applications are rolled out that deliver inadequate performance for users or require rearchitecting — at a much higher cost than doing it right the first time.

Going forward, the critical step of developing consistent, reliable, and repeatable infrastructure services will be increasingly valuable to IT organizations and the business at large. Therefore, the new goal for the progressive ITO is to become "adaptive." Adaptive efforts go by many names — from "utility," to "grid," to "on demand" — but they all have the same key emphasis of aligning IT capabilities, costs, and goals directly with those of the business. Intelligent network services play a crucial role: In a 2004 META Group survey of 308 Global 2000 decision-makers, network infrastructure was cited as the number-one area of investment required to become adaptive (see Figure 2).

Figure 2 — Areas of Investment to Become Adaptive



Source: META Group Adaptive Organization Study 2004

Copyright © 2005 META Group, Inc. All rights reserved.

Embracing Change: Agility as a Cornerstone of Data Center Design

Change is a constant within the ITO, and leading services companies are actively embracing it. For example, eBay adds up to 2 million new items a day, registers up to 40,000 new users a day, and changes up to 30,000 lines of code per week — all while operating continuously. Chizumaru, one of the largest Japanese content providers, automatically deploys software stacks to hundreds of servers using a rolling-upgrade model. A leading worldwide entertainment company has a service-oriented architecture (SOA) that is designed to bring an application online from being a bare metal server to production-ready in a matter of minutes — a process that previously would have taken weeks. For these companies, the importance of agility cannot be overemphasized.

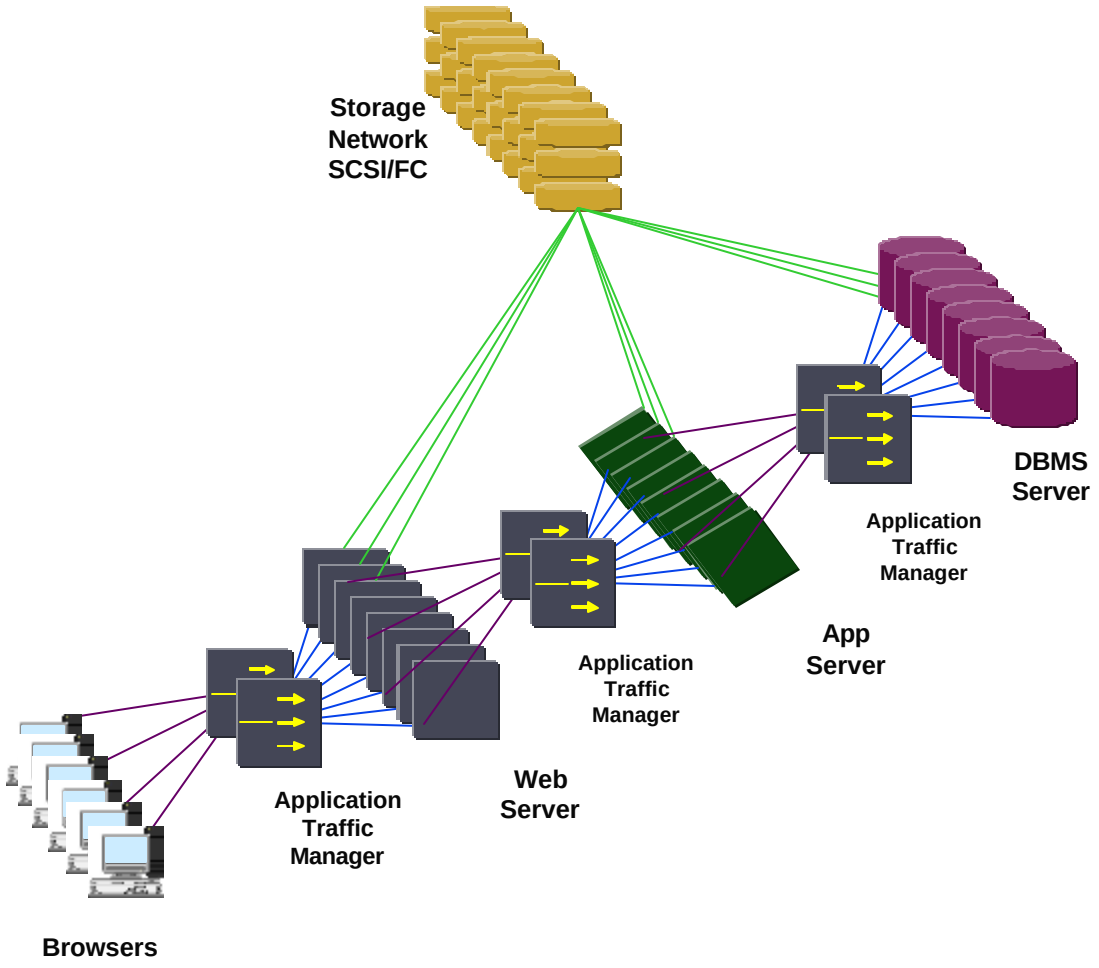
Scaling Out Infrastructure: Building a Network of Networks

Twenty years ago, in the mainframe/minicomputer era, users were essentially required to buy their servers, terminals, storage, and networks from one vendor. Traffic flows were deterministic and easily modeled. But then distributed computing introduced new complexity into the data center. Users now have the flexibility to mix and match networks from one vendor, with servers from another, storage from a third vendor, and clients from yet another vendor. This new approach allows for dynamic growth in a "scale-out" fashion, enabling each tier to be sized to the appropriate workload. In this model, components in each domain form their own communication networks, even though the system must work together as a cohesive whole (see Figure 3).

This scale-out approach introduces complexity and creates the need for an organization-specific IT architecture, or technology blueprint. Successful IT organizations link the architecture development process to the enterprise planning process, building a common set of descriptions and artifacts (e.g., architectural models and diagrams, documents, governance rules, technical standards, policies, process models). Managed appropriately, these common architectures solve application and infrastructure problems and thereby deliver demonstrable return on investment and improved agility and cost control as well as better integration with security policies.

Copyright © 2005 META Group, Inc. All rights reserved.

Figure 3 — Technical Architecture Is Essential



Source: META Group

Yet these architectural descriptions are not just abstract documents. An enterprisewide technical architecture creates a business-driven blueprint for the application of technology toward solving business problems. Effective architectures are prescriptive, and technical architecture must guide and direct infrastructure and development engineering decisions. Overarching architectural principles are a critical part of the infrastructure development process. And to be truly adaptive, an architecture embraces virtualization — allowing physical infrastructure to be logically created and managed in software. The more components of an architecture that can

Copyright © 2005 META Group, Inc. All rights reserved.

be virtualized with a common set of tools or technologies, the greater the flexibility the organization has in being able to minimize the impact of change.

Adaptive Infrastructure Design Principles

The following descriptions and technical assessment guides for the adaptive infrastructure design principles are presented to serve as a guide in evaluating application traffic management solutions and determining how they map to these specific principles.

Change or Adaptability Principle
Description
The only constant is change. Therefore, infrastructure must be designed so that the impact of change is understood, enabled, and efficient.
Technical Assessment Guide
Look for solutions that can virtualize nearly any IP-based resource from both a WAN as well as a LAN perspective. If a model can be created where components of the architecture are abstracted behind a virtual entity, this allows for uninterrupted change, enabling additions, removals, and updates.
Reuse Principle
Description
Infrastructure should be designed to be reused as much as possible.
Technical Assessment Guide
When considering solutions that can virtualize resources, determine whether they support a workflow model and a component- or object-based reuse principle. If, for example, we define a common scale-out design and policy, are we able to define that policy once and reuse it across multiple similar virtual entities? The same would apply for things such as application security or application compression policies. Can they be defined once and applied across similar virtualized entities, or must they be created each time?
Industry Standard Principle
Description
Patterns, technical services, components, and product standards for infrastructure should reflect industry standards, where applicable (especially Internet and Web services standards).
Technical Assessment Guide
Look closely at strict adherence to standards, including RFCs (requests for comments) and emerging standards coming from recognized bodies such as IEEE, WS-I, and OASIS. It is not uncommon for vendors to take "shortcuts" regarding standards to increase their performance or get to market much more quickly. This can potentially wreak havoc within your environment. No network is entirely "clean," and situations always arise where we encounter environments with out-of-order packets and fragments. Without strict adherence to standards, your investment could be lost.

Copyright © 2005 META Group, Inc. All rights reserved.

Service Principle

Description

Patterns and technical service designs should emphasize the use of shared services — that is, specifically packaged reusable solutions made up of virtualized shared infrastructure components accessed via standardized APIs (application programming interfaces) and protocols in a loosely coupled way (e.g., Web services), with some emphasizing a complete hosted solution.

Technical Assessment Guide

The service principle maps to the concept of being able to virtualize any data center, WAN connectivity, IP-based application, server, appliance, or network device. The concept of a shared service within this paradigm means applying the policies that make that virtualized resource or application perform better by offloading tasks that make sense (e.g., if you have hundreds of applications all implementing and maintaining a different authentication model, look for a solution that can centralize that function for those applications). Think of authentication as a service that the applications can subscribe to instead of it having to be implemented within each application itself (the directory services still reside elsewhere and can be heterogeneous like the applications — it is the enforcement of authentication that is being centralized). The authentication is then abstracted up a layer and can be implemented much more quickly, maintained much more simply, and audited to ensure that adherence to corporate policies becomes physically practical. The same principles would hold true for tasks such as load balancing, SSL encryption/decryption, TCP optimization and compression services, and application security policies.

Reality Principle

Description

The design can be implemented and/or is already well supported.

Technical Assessment Guide

The best guide to determining this from a vendor is to analyze its partnerships with other vendors, evaluate technical guides that document standard implementations, and assess how far the vendor goes in ensuring that designs work and have been tested. Customer case studies are useful, but they rarely cover the depth needed to assess the reality principle.

Omission Principle

Description

Although a design is inclusive of all components, components may be omitted if not required for a given implementation.

Technical Assessment Guide

This is critical. Revisiting the service principle, if a given design and some of its components do not require a shared service, can a solution be configured to turn that service or a part of that service off? For example, some components of a virtualized entity may not benefit from compression services. In fact, overall performance, whether application or client performance, may be hurt by compression — it is not a "one size fits all" technology. Therefore, it becomes important to look for solutions that have the granularity to omit certain components from a service and are able to target that shared service to only those components that require it or will benefit from it.

Copyright © 2005 META Group, Inc. All rights reserved.

Life-Cycle Principle

Description

A design should include solutions for each relevant stage of the overall application delivery life cycle (e.g., planning, testing, staging, and production services).

Technical Assessment Guide

When applying this principle as a technology guide, it also relates to the principle of reuse. To support the life-cycle principle, a solution should be able to map into workflow. Therefore, solutions should be sought that allow the organization to plan and define a virtualized scenario, as well as define policies and stage and test them. This type of solution should also be able to be shared by multiple parties as a staging platform to maximize the investment. In addition, a solution provided by a vendor should have a fairly deep and broad support network that can assist in the planning cycle.

Assistance would be in the form of documented solutions, preferably specific to applications and network topologies, configuration guides, technology tips, consulting, and integration scenarios as well as sample code and sample policies or profiles that have been tested by the vendor. These can then be leveraged by an organization as a starting point in the planning and design stage. Look for solutions that also have very good logging — that is, descriptive and meaningful error codes that can be invoked based on events for debugging purposes during the staging and testing phase.

Total Cost of Ownership Principle

Description

Infrastructure designs should include full cost or total-cost-of-ownership (TCO) models (e.g., life cycle, exit).

Technical Assessment Guide

When applying the TCO principle, it is important to assess a vendor's formal end-of-life definition and support policies associated with both hardware and software, to ensure that the investment made today will, at a minimum, safely cross the normal three-year depreciation cycle. A solution's local as well as distributed management framework must also be closely examined. This will have a significant impact on TCO. For example, if you want to enable reuse, that objective is focused on minimizing costly errors and duplication of effort. Does the management or user interface support that objective? In addition, ongoing maintenance and troubleshooting of a solution should also be enabled through granular metrics, searchable and descriptive logging, and an integrated interface to ensure rapid diagnosis with a focus on lowering maintenance costs.

Some of the most talked about initiatives in recent years, which are all related, are dynamic provisioning, autonomic computing, grid, and on-demand computing. Despite the hype surrounding these phrases, there is also a dose of reality associated with them. To realize the often-discussed potential of these initiatives, a solution must have a tested and standards-based API and software development kit (SDK) that allows it to integrate with other components of the architecture or ecosystem.

(cont.)

Copyright © 2005 META Group, Inc. All rights reserved.

TCO Principle (cont.)
Technical Assessment Guide (cont.)
As previously noted, how do you automatically perform graceful updates to existing services in a production environment? This is accomplished by enabling each component to automatically remove itself from a virtualized resource and be placed back in again once the update is complete. You should not have to physically touch the solution that is doing the virtualization. This typically labor-intensive task should be able to be accomplished programmatically. To do so, there must be a standards-based interface to the solution that components can securely manipulate, whether for automating updates or for proactively reacting to a security intrusion by automatically telling the virtualization device to block an offending IP address.
TCO Funding Principle
Description
Design costs should indicate how funding should be provided to keep infrastructure maintained and up-to-date in future years.
Technical Assessment Guide
Vendors' pricing and packaging models must be closely examined. What may look attractive on the surface could be a "red herring." Does the vendor offer a "pay as you go" model, or are you forced to buy everything upfront, even functionality that may not be needed today or tomorrow? When specifying a design, attempt to anticipate functionality needed now, in six months, and even in 18 months, ensuring that budget allowance is built into the solution or architectural life cycle.
Exception Principle
Description
There will be an exception process for handling deviations from the design.
Technical Assessment Guide
The one thing that is constant is change. Look closely at how flexible or adaptable a solution may be. Too frequently, we have seen customers lock themselves into a design simply because the underlying architecture of a solution could not adapt. For example, to be prepared for the exception principle, ask yourself, can the vendor's solution support more than just one application type? Is it focused exclusively on Web applications? What about voice over IP (VoIP) or remote desktop protocol (RDP) for thin clients, or some vertical industry-specific applications such as FIX for financial services companies? Keep in mind that a key goal is being able to create a common and singular model that can be leveraged by many different application types, not just one. To be prepared for the exception principle, and to evolve from being a reactive IT organization to being a services enabler, the underlying solutions within the architecture must be adaptable.
Change Management Principle
Description
The design will support regular, reasonably paced, and coordinated changes over time.
Technical Assessment Guide
Applying the technology guide to this principle highlights the importance of functions such as auditing. Does the solution have robust auditing of changes? Is the log that stores the information non-modifiable? *(cont.)*

Copyright © 2005 META Group, Inc. All rights reserved.

Change Management Principle (cont.)

Technical Assessment Guide (cont.)

Optimally, a solution would have an events-based interface or even an API that could automatically notify an external change management system when there are status or state changes. Key enablers of the change management principle are reuse and shared services. For example, if a virtualization solution is being used to perform client-session persistence for stateful applications, can that service be written once and shared among many virtualized applications? If so, modification becomes much simpler, since the change is made once and all other subscribers to that service inherit that change automatically.

Use Principle

Description

Processes should be updated to use the design throughout the solution delivery life cycle.

Technical Assessment Guide

For life-cycle processes to work and the process update to be as simple as possible, the user interface should be logical enough to support the process documentation effort. If a solution's user and management interface is not logical, it will encumber this effort. We have seen what should have been simple operations, such as creation of a virtual entity and its associated nodes and monitors, range from three steps to up to 30 steps, depending on the solution. Clearly, the more complex the process, the more costly and the more prone to error it will be.

Package Fit Principle

Description

Purchased packages will be evaluated and measured for compliance with defined patterns and technical services.

Technical Assessment Guide

To make this technical assessment, the onus is on the vendor to provide as much technical documentation as possible so that compliance can be best evaluated. It is important to assess whether a vendor actually provides this material and to be wary of vendor performance claims. What is frequently published on Web sites and in marketing literature is not what you will find in real-life, use-case scenarios.

For example, when SSL or compression performance numbers are published by a vendor, these numbers have frequently been generated in isolation, as if the solution was only processing SSL or only performing compression. However, in the real world, we know that these solutions are going to be processing multiple functions and different types of traffic concurrently — not as singular isolated events. To get to the meat of such claims and be able to assess performance compliance, ask for performance numbers that show how a solution performs with multiple functions enabled and running concurrently. Preferably, you should receive information that maps as closely as possible to what you anticipate your IP traffic will be.

Copyright © 2005 META Group, Inc. All rights reserved.

Simplicity Principle
Description
There will be a small number of designs intended to satisfy most needs. Any given service, despite its internal technical complexity, should define an elegantly simple interface to enable reuse. Complex interfaces will slow reuse.
Technical Assessment Guide
If simplicity is a goal or objective, do not forget to evaluate the management framework of the solution. Can elements be edited, added, or removed easily? Is creation of new elements cumbersome or logical? Can you give defined objects real names instead of non-descript IP addresses?
Service-Level Design Principle
Description
An appropriately varied set of service levels should be designed into each infrastructure design or configuration.
Technical Assessment Guide
In evaluating application networking solutions that fit into your SOA, the ability to track and report on specific metrics related to the devices themselves as well as the virtualized components is essential. Solutions must have not only robust SNMP support and well-defined MIBs (management information bases) to collect this information, but also an open API that allows you to collect, customize, and map data to your specific SLA needs. Regarding virtualized components, solutions should be sought that report statistics not only at a high level (e.g., total throughput of traffic and uptime of a virtual service), but also at a more granular level as well as down to the individual components of that virtual service. This could be specific nodes or collective pools of nodes, for example. This data is not typically collected via SNMP and must be able to be collected easily — hence the need for an open API. If the API is event-driven, when specific defined actions occur, an automatic action can be taken. For example, when capacity for a service exceeds a threshold (e.g., 5 Mbps), data may automatically be published to a repository.

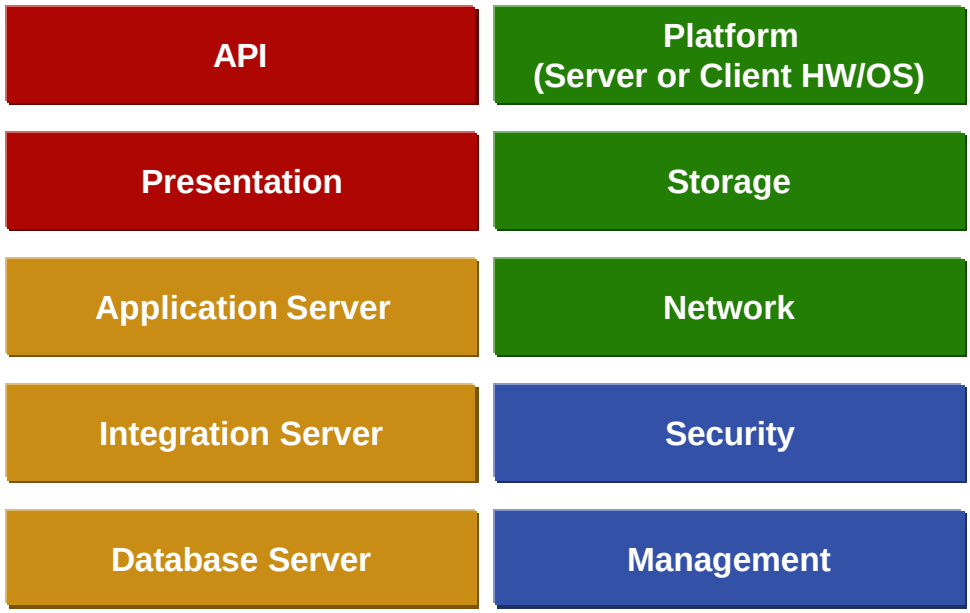
Reducing Complexity: A Key Goal

One key goal of both architectural and infrastructure planning and the engineering technology models is to reduce complexity. Reducing complexity brings with it many other benefits, including lowering cost and improving the speed at which changes may be made. When designing more distributed and client/server (and now n-tier and service-oriented) architectures, many disparate components (individual product and service categories) must be organized properly. In META Group's Adaptive Infrastructure catalog, there are literally dozens of components that comprise a total solution.

Copyright © 2005 META Group, Inc. All rights reserved.

However, both architecture and infrastructure planning approaches recognize that a large undifferentiated component list is not organized. Consequently, most organizations have suggested use of higher-level aggregate models to better organize this morass of components. These components are organized into domains, as illustrated in Figure 4.

Figure 4 — Common Technology Domains



Source: META Group

The Implications of Service-Oriented Architecture for Data Centers

SOA models, including Web services, present a significant shift in the way organizations will design business systems and the applications that support them. This shift will affect not only the applications, but also the roles and the responsibilities within the firm and the various tools and techniques that are used to build them. SOA promises highly adaptable services, which are connected via an easily changed model that parallels the business process and that can be related to it in a meaningful manner for a business user. SOA emphasizes coarse-grained design and standardized interfaces that help enable the creation of composite applications.

Copyright © 2005 META Group, Inc. All rights reserved.

Another unmistakable future architectural trend will be systems being composed of increasing numbers of functionally identical subsystems of decreasing size (a.k.a. resource pooling). The design goal is to build a highly reliable and highly scalable virtual system out of large numbers of moderately reliable and moderately scalable subsystems. Generally speaking, to virtualize a set of diverse concrete resources is to access them through a single uniform interface that, from the users' perspective, enables them to behave as one unified resource that can be shared with varying degrees of dynamic behavior. Hence the strategic role of application traffic management solutions and how they serve as a foundation for these architectures.

Virtualization creates a single-system illusion delivered by two complementary forces:

- ***Miniaturization:*** The system's size and complexity, relative to its performance, decreases over time — whether the system is at the chip, disk, or port level; at the board level; or even at the server level.

- ***"Massification":*** This is the combination of multiplexing and multiprocessing — multiplexing of ever-larger numbers of users (whether those users are people or systems) onto ever-larger numbers of uniform processors (whether those processors are transistors, CPUs, disk drives, or application/database server engines). The number of subsystems composing a system's instances grows over time into thousands of processors, servers, and spindles.

The exploding complexity of the future data center is not only driven by the degree of integration among all systems, but also compounded by the fact that the rate of change of those integrated systems and their interconnections is far faster — from network topology changes, to software configuration changes, to application integration changes. The future data center will be populated by storage, server, and network elements from diverse vendors, and be defined by increasingly standardized, interoperable interfaces. This will benefit the IT buyer, due to the competitive landscape of hardware and software vendors.

The Security Challenge: "Port 80 Overload"

New application architectures using XML and SOAP commonly use TCP port 80 (the HTTP port) to transfer application data — often the only open, trusted port through the corporate firewall. Although this serves as a reliable way to ensure communication without being blocked, unfortunately it also has the side effect of eliminating the firewall's intended purpose — that of securing applications. As

Copyright © 2005 META Group, Inc. All rights reserved.

more traffic gets routed through port 80, it becomes essential that intermediate devices look deeper into the SOAP envelope to inspect applications.

Taking the next logical step, to be truly effective and in the long run more manageable, intermediate devices must understand the behavior of the application. Understanding what is correct or positive behavior then becomes the model for application security. A solution should only allow behaviors that it knows to be correct. It is much easier to maintain a system that knows that an application has only 20 allowed inputs of specified length and values and should prevent any other type of input than it is to constantly scan for the thousands of negative signatures and potential threats.

Therefore, these new devices must not simply operate at the network, transport, and session layers as with common network firewalls, but must also be application, XML, and SOAP aware. These devices are frequently known in the market as XML or Web application firewalls, gateways, or accelerators.

The benefit of better application and protocol awareness is similar to virtualization, in that it abstracts or offloads security and management functions from the application itself and centralizes enforcement. For application developers, this is a huge relief, since they have some assistance in ensuring the security for their applications, and for security auditors, full auditing becomes practical. It is also much easier for operators to analyze the policies of a few devices supporting hundreds of applications than it is to attempt to directly audit the hundreds of applications themselves.

Supporting Variation But Managing Complexity: The New Role of the "Load Balancer" as the Application Traffic Manager in Dynamic Resource Allocation

The application traffic manager … can make intelligent decisions based on various inputs in each layer of the architecture.

The scale-out approach has proven to be massively scalable, as demonstrated in real-world situations with public Internet services such as eBay and Expedia. This approach delivers substantial advantages in product selection and in aligning capital to current needs without the need for high fixed-cost investments. Yet organizations still struggle with unifying the total set of domains and the myriad components into a single unified system. Databases and application servers tend to only manage themselves and are blind to the network. Simple network devices have no understanding of applications. Thus the need for the application-fluent network device — the application traffic manager — that can

Copyright © 2005 META Group, Inc. All rights reserved.

make intelligent decisions based on various inputs in each layer of the architecture — from application and database servers, to server and storage hardware, to network capacity, and even to the individual user level.

As a term, "load balancing" describes only a small percentage of the value that application intelligent networking devices bring to the data center. The Layer 7 (i.e., application layer in the OSI stack) intelligence of these products makes them flexible network devices, enabling load balancing of multiple network resources (e.g., firewalls, ISP links, mail, voice over IP [VoIP] servers), both within the data center and across multiple data centers, while taking on many more roles beyond simply load balancing. In fact, the versatility of these devices creates somewhat of a quandary for vendors and architects alike, when asked to describe the specific utility that they bring. Although examples of typical use cases abound, many organizations are applying these devices in unforeseen ways — to both Web-based and traditional applications, and even to real-time applications such as VoIP.

As the core functionality of these devices matures, vendors and IT organizations are considering the future direction of these network jacks-of-all-trades. Essentially, the role of these devices has become to improve performance, efficiency, and security through the use of multiple techniques that are intelligently applied to the application flow.

The role of the intelligent application traffic manager is to:

- ***Offload processing:*** From other devices, such as SSL processing, connection management, compression, content caching, application spooling, and session cookie management.
- ***Ensure fair access to resources:*** Using quality-of-service policies.
- ***Improve reliability:*** By allowing devices behind the application traffic manager to be placed in and out of operation without disrupting service.
- ***Improve performance:*** Through various techniques such as compression, TCP optimization, caching, and application spooling.
- ***Secure resources in the data center:*** By using a policy engine and having an awareness of users and applications, and by becoming an authentication enforcement point.
- ***Centralize management of these functions:*** To improve overall service reliability and contain cost. In fact, management features are more important than capital equipment cost. The IT architect for a leading online services firm

Copyright © 2005 META Group, Inc. All rights reserved.

estimates that more than 90% of the TCO for network devices lies in operational cost, principally in the cost of people and secondary support services.

Case Study: SSL Offload

A multinational company created a Web-based portal to serve the needs of its 90,000 employees. It must maintain privacy and security for all online employee activities, so it uses SSL on all of its sessions. By offloading SSL to an appliance, the company reduced the total server processing load by 15%, dramatically cutting its need for hardware, software, and personnel to support the application.

In actual operation, many customers report they continually discover new uses for these appliances. One company will use its load-balancing devices to maintain secure cookies issued by the Netegrity SiteMinder application. Another is using the devices to 'fix' poorly architected applications. This company uses a critical application that has an unusual (and unfortunately, poorly designed) licensing scheme: After authenticating a valid user, the application would send a static host IP address back to the client for the subsequent session. This would result in a highly fragile application — this single client-to-server dialog had to be maintained throughout the user's session. But by allowing an application traffic manager to act as the full proxy, the session could be maintained across a pool of servers, thus dramatically improving reliability.

Case Study: Reducing Support Cost by Simplifying Management

For a large transportation services firm, the proliferation of Web applications was driving its support costs up dramatically. It chose to reduce infrastructure complexity by introducing an application traffic management solution into its data center. After considering other "simple" load balancers with only command-line interfaces, it went with a vendor with a fully graphical user interface. The company estimates that this choice reduced the required headcount for support personnel by between 1.5 and 2.

Reducing TCP Connections

When a browser connects to a Web site, it may open many TCP connections with the site, increasing back-end server load and bandwidth requirements while introducing excessive latency. By offloading this capability to an intelligent network device using TCP multiplexing, server load is reduced and end-user performance is increased. This capability is further enhanced through the use of HTTP v1.1 Keep-Alives, allowing the device to maintain a single TCP connection for the user. This can more than double the number of connections that a Web farm can support.

Copyright © 2005 META Group, Inc. All rights reserved.

Case Study: Building Overflow Capabilities and Tuning Performance

Web sites often are subject to usage spikes, such as those caused by flash crowds. By using the system monitoring capabilities of its application traffic management solutions, an online services firm automatically sends traffic to a set of standby servers when the active set of servers hits a given usage threshold. The solution monitors Windows servers by periodically issuing Windows Management Interface calls requesting information on each server's CPU utilization. By monitoring database-level requests, the company can also split database reads from writes, redirecting clients to a server specifically tuned for that function.

Case Study: Automating Application Updates and Rollouts

A large content provider publishing advanced wireless information and applications faced the challenge of updating hundreds of application servers each day. It was not financially scalable or reliable to use technicians to manage the painstaking process of updating each server individually. By integrating its application code management and deployment processes with its network through a SOAP/XML API — a standard Web service interface to remotely manipulate the network device configuration — the company was able to streamline control and update managed Web servers to completely automate application updates. Activities that previously required two to three people per day are now fully automated in an error-free manner. The network device is accessible via a software API and now offers a powerful control point to support around-the clock deployment efforts.

All of these case examples illustrate the creation of a service oriented architecture. These companies are using the same solution, but applying it in different ways, depending on the application type and the application infrastructure design principles used. Therein lies the power of these solutions and why we consider them foundational; this is representative of what we mean by being adaptive.

Features of Application Traffic Management Devices

Essential Features

Application traffic management devices have evolved well beyond simple load balancers. These devices:

- Accelerate application performance and minimize the cost of infrastructure
- Use various real-time inputs for traffic management decisions, including server hardware, operating system, database, and application server characteristics (e.g., current CPU utilization)

Copyright © 2005 META Group, Inc. All rights reserved.

- Implicitly understand the most popular IT applications (i.e., no explicit signaling by the application is required)
- Can inspect payload such as SOAP envelopes and parse XML messages, performing deep content inspection and recognizing attack signatures and activity
- Provide packet-filtering firewall and denial-of-service isolation
- Take user characteristics into account in traffic management decisions, such as user type or connection speeds
- Are completely transparent to the client or application (i.e., do not require application code changes or client software)
- Are reusable across different user constituencies such as branch-office users, remote access employees, business partners, and customers
- Are reusable across multiple applications and application types
- Are remotely accessible as an application service through a standards-based API for advanced forms of external management and manipulation to adapt to ever-changing IT processes and real-time automated network configuration changes (the API should be well-documented and have a supported software development kit for integration within the SOA)
- Accelerate both clear text and encrypted traffic
- Do not shift the bottleneck to another element
- Solve more than one specific performance limitation
- Can be implemented in a variety of form factors, from small fixed-configuration appliances to large modular systems

Manageability Attributes

Application traffic management devices have the following manageability attributes:

- Are easy to install, configure, and test
- Provide reporting that contributes to troubleshooting/problem isolation
- Are fault-tolerant in operation and "fail-to-wire" such that the application will continue to function if the acceleration device is removed or offline (optionally, can be configured with no single point of failure)
- Can provide high availability and scalability (e.g., by optionally using clustering or other forms of failover)

Copyright © 2005 META Group, Inc. All rights reserved.

- Provide operational visibility through internal software health monitoring and integration with network monitoring tools
- Support flexible management options, including distributed, hierarchical, and remote management (i.e., a secure Web interface)

In general, as application functionality matures, it becomes possible to deliver an infrastructure service hosted on intelligent switches, which can be reused across numerous applications. The goal then for new application-aware network devices is to provide a common set of infrastructure services extending well beyond basic load-balancing service — into SSL offload, bulk encryption, and other functions — with the capability of distributing policy management roles into application groups through the use of simple APIs. However, before an organization commits development resources to a specific API set, it must be certain that the API vendor is a long-term player in the intelligent network device market.

Bottom Line

- Intelligent network devices play a pivotal role in delivering applications in the new data center, unifying disparate infrastructure into a cohesive system and directing user traffic to the most appropriate resource.
- The functionality of these devices is moving far beyond basic load balancing and evolving into a future platform for common infrastructure services such as encryption, compression, caching, and application security.
- Due to the variety of applications and frequency of change in the data center, users should examine general-purpose devices with a wide range of functionality and form factors rather than specific-purpose appliances.
- Over the life cycle of the system, the operational cost of intelligent network devices is much higher than the initial capital outlay. Buyers should look for management features that reduce complexity and ease operations.
- If the switching product is a strategic element of the data center infrastructure, users should engage developers to make use of the policy management features (including APIs) available in the platform.

David Willis is a vice president and research lead with Infrastructure Strategies, a META Group advisory service. For additional information on this topic or other META Group offerings, contact info@metagroup.com.

Copyright © 2005 META Group, Inc. All rights reserved.

About META Group

Return On Intelligence[SM]

META Group is a leading provider of information technology research, advisory services, and strategic consulting. Delivering objective and actionable guidance, META Group's experienced analysts and consultants are trusted advisors to IT and business executives around the world. Our unique collaborative models and dedicated customer service help clients be more efficient, effective, and timely in their use of IT to achieve their business goals. Visit metagroup.com for more details on our high-value approach.