IMPERVA®

How to Select a Load Balancing Solution

GUIDE

What's inside

Introduction

Targeted to IT managers and staff seeking efficient and reliable networking operations, this document provides information on load balancing solutions and defines how load balancing benefits small, medium and large enterprises. It also defines best practices regarding the selection of a load balancing solution–one that can improve network performance and reduce costs while enhancing security and improving business continuity.

Glossary

ADC

Application delivery controller: A device or service that is application aware and can offer enhanced content delivery based upon specific application requests.

CDN

A content delivery network is comprised of interconnected servers located in different parts of a country (or the globe) that store files to accelerate website visitor access.

Cluster

A grouping (or pool) of servers that offer similar services and can cooperate in delivering content to client systems based upon algorithms executed via a load balancer or CDN.

Continuity

A concept that uses persistence and/or failover technologies to guarantee network resource access without interruption or data loss

DDoS

Distributed denial of service: An attack using multiple compromised networks to overload a website's ability to respond to legitimate requests

Failover

A capability that provides automatic switching between systems to provide redundancy in the event of a primary system failure

Hypervisor

A hypervisor is a piece of computer software, firmware or hardware that creates and runs virtual machines.

Load Balancer

A device (or service) that spreads traffic loads over multiple servers

OSI Model

OSI (Open Systems Interconnection) is a reference model for how applications can communicate over a network. It defines the communication process between two endpoints in a network, divided into seven distinct groups of related functions, or layers. Please reference the chart at the end of this document for more information.

Persistence

Functionality provided by a solution that guaranties connectivity, even if a member system fails

SaaS

Software as a service: Applications delivered using a service/utilityorientated model

Executive Summary

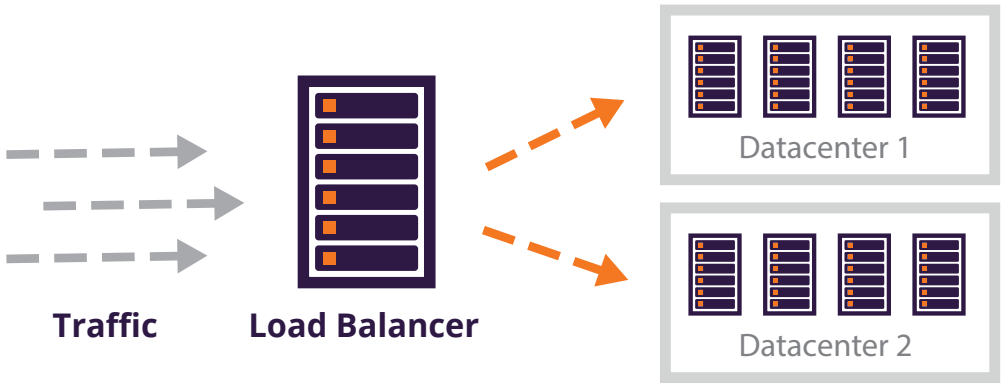
The demands placed on enterprise networks are growing at a rapid pace. Consider the distributed nature of cloud services with a growing mobile workforce and it becomes easy to see why network performance has become more important than ever.

Preventing performance issues while maintaining network reliability is a common need of any size organization. One of the best ways to handle growing performance demands is to implement load balancing—a technology that distributes traffic and application loads across multiple servers while also ensuring business continuity via automatic failover processes.

Yet choosing an appropriate load balancing solution is not easy. There are numerous methods for deploying a load balancing solution, ranging from cloud-based services to physical appliances. Regrettably, there is no "one size fits all" solution for enterprises so careful consideration should go into the selection and vetting of solutions for your environment.

This whitepaper serves to distinguish technological load balancing differences and separate the buzzwords from the facts. Understanding terms such as layer 7, round robin, server pools, clusters, and persistence is an important first step.

Ultimately, selecting an appropriate load balancing solution depends upon conducting research, determining the type of solution that best fits a given set of circumstances, and evaluating the return on investment (ROI) offered by that solution.



A load balancer shares a network load across multiple servers.

What is Load Balancing?

Load balancing is not a new concept for managing server and network performance. It has its roots in the early days of distributed computing, where a physical "load balancer" was located between endpoint devices and application servers providing data to those client systems. In its simplest form, a load balancer would determine which server was least busy and direct network traffic to that device.

As networking evolved and distributed systems were introduced into the enterprise, the task of load balancing took on a new meaning. How loads are balanced across enterprise resources was redefined. The market responded to those changes and a multitude of vendors began offering a wide variety of technologies to handle this task.

Several products on the market today represent different approaches to load balancing. For example, modern enterprise routers can distribute traffic across multiple paths to the same destination. While they balance loads across different network resources, they remain unaware of application demands and other elements that can drive traffic.

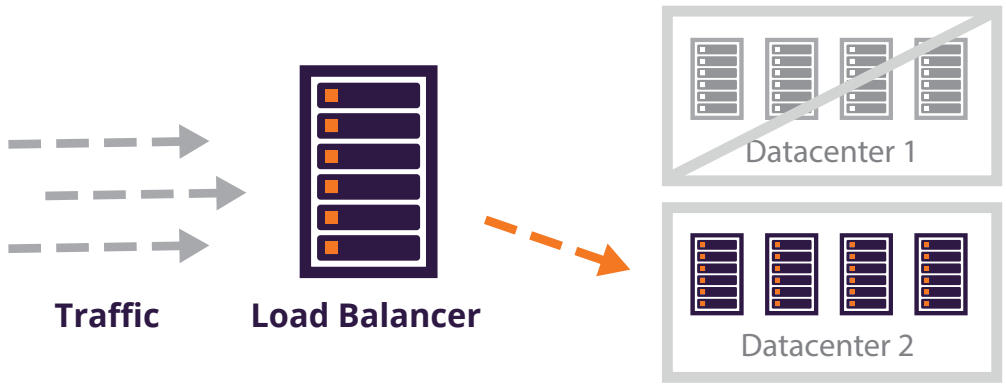
Contrast these with dedicated server load balancers, designed to distribute traffic among server resources (rather than network resources). These focus only on the demands placed upon server clusters—not the general network traffic.

And then there are application load balancers. These attend to the demands placed upon a single application, distributing the load across servers hosting that application.

Simply put, the load balancing market has become fragmented based upon a narrow set of mutually-exclusive parameters.

The earliest solutions in this space were powered by very simple load balancing algorithms. They quickly evolved to include a superset of capabilities which centralize a multitude of functions. Today's advanced solutions perform much more than load balancing—they also provide traffic engineering (shaping) and intelligent traffic switching. Additionally, modern load balancers can perform sophisticated health checks on servers, applications, and content to improve availability and manageability.

Load balancers deployed as a server farm front-end can protect the hardware from malicious users, enhancing overall security and uptime. Load balancers use IP packet information, or application request content, to make intelligent decisions in directing traffic to the right data center, server, firewall, cache, or application.



Failover is one of the most important capabilities offered by a load balancing solution.

The Need for Load Balancing

While not all organizations have a specific need for load balancing, most can benefit from what the technology provides. Determining the need for load balancing is predicated on the demands placed on an organization's servers and networks. Enterprise networks are used to connect servers with employees, customers or suppliers, and this connectivity has become mission-critical. It is deemed unacceptable for a network to fail or experience poor performance; either issue compromises the ability to conduct business. Nowhere is this more apparent than with organizations which rely on 24 x 7 x 365 Internet connectivity for their livelihood.

Using an e-commerce business as an example, its infrastructure typically consists of several interconnected components, such as edge routers, switches, firewalls, caches, web servers, and database servers. A failure of any individual component causes downtime. With the proliferation of servers for various applications and their intricacies, the impact of failures grows exponentially.

Today's data centers—bulging with server farms—are the culmination of extending multiple applications and services to ever-growing audiences. The complexity and challenge in managing and scaling server farms is a driving factor behind the need for intelligent switching. With that in mind, IT managers must ensure scalability and high availability for all components—beginning with Internet-connected edge routers and culminating at the back-end database and application servers. To address the needs of these complex environments, modern load balancing technologies have been interwoven to bolster performance and assure continuity—while also providing a security fortification to thwart attacks and malfeasance.

Many enterprises have turned to cloud based services, SaaS (software as a service), virtualized data centers, and hybrid cloud environments to deliver their applications to end users. This adds another layer of complexity to the load balancing equation: certain IT resources and servers may actually be running on leased, virtualized cloud-based systems, while companion components may be running in an onsite data center. Such hybrid environments dictate that load balancing tasks also be moved into the cloud. Considering the fastest path to data routing rules, this solution provides distributed access and regionally-isolated continuity for applications, servers, and databases.

If your organization offers access to computing solutions from multiple locations via the web, load balancing is a must-have capability to ensure maximum performance, failover, and security—all of which lead to business continuity and reduces or eliminates downtime.

Load Balancing – A Variety of Choices

Several load balancing solution options are available, each having its own pros and cons. Understanding your options and what each offers is critical in deciding which type of technology to deploy.

In their simplest form, load balancers distribute traffic to one or more servers (sometimes called nodes). At a finer level of granularity, a load balancer also relies on a member element that includes the TCP port of the application that will be receiving traffic.

Ideally, a load balancing solution lets organizations distribute inbound traffic across multiple back-end destinations. These destinations are usually organized into a cluster, pool, or server farm. In its simplest iteration, a cluster is a collection of similar services available on any number of hosts.

Further complicating basic load balancing concepts is the fact that some vendors offer a variety of options, features, and settings that can become a determining selection factor. This is particularly relevant when an organization looks to unify load balancing with availability and security services.

Basic server load balancing can be performed using the following methods:

Static Methods

- **Round-robin**
  One of the simplest load balancing methods, round-robin works by passing each new connection request to the next server in line, eventually distributing connections evenly across the array of load balanced hardware. Round-robin mode works well in most configurations, especially if the equipment you are load balancing is roughly equal in processing speed and installed memory. Using roundrobin, the load is typically distributed more evenly for applications handling high traffic volumes (with many simultaneous requests).
- **Static Content Relevant to Website or Application**
  Load balancing at OSI layer 2, or bridged load balancing, is a very simple model. A virtual IP address is created in the same subnet as the physical servers; packets destined for the virtual address are forwarded to the physical servers. This method is very easy to deploy and can be adapted to existing infrastructure without introducing additional network changes. Usually limited to single networks, it can be hampered by layer 2 issues, such as loops and spanning tree problems.
- **DNS load balancing**
  As the name implies, balancing is done with DNS. A single name resolves to multiple names or IP addresses. These real names, or IP addresses, are then targeted in a round-robin manner. This is also very easy to configure, but lacks any intelligence.
- **Ratio (member) to ratio (node)**
  According to pre-defined ratio weights, this solution uses a traffic management system to distribute connections among servers in a static rotation. Over time, the number of connections received by each system is proportionate to the ratio weight defined for each member (port) or node (server) in the pool. Each ratio weight is initially set when a pool member or node is created. This method is best suited for defining static load balancing.

- **Routed load balancing**
  This load balancer operates at the network level (OSI layer 3). It's housed on a separate network, uses a virtual IP address, and its target servers reside on different networks. This method offers ease of expandability and works well in environments where server pools are geographically diverse. Network design and additional IP address space is often required, however.

Dynamic Network Level (Layer 3/4) Methods

- **Dynamic ratio (member) to dynamic ratio (node)**
  The dynamic ratio method selects a server based on various, real-time performance analysis aspects. Similar to the ratio method, it differs in that ratio weights are system-generated and the ratio weight values are not static. This method is based on continuous server monitoring, with server ratio weights continually changing.
- **Least connections (member) to least connections (node)**
  The least connections method is relatively simple, in that the system passes a new connection to the pool member or node having the least number of active connections. This method functions best in environments where all servers have similar capabilities; otherwise, some latency can occur.
- **Weighted least connections (member) to weighted least connections (node)**
  This method specifies that the system use an administrator-specified value for connection limits in order to establish a proportional algorithm for each pool member. The system bases its load balancing decision on that proportion and the number of current connections to that pool member. It works best in environments having unequal server capacities.
- **Observed (member) to observed (node)**
  With this method, nodes are ranked based on the number of connections. It tracks the number of layer 4 connections to each node over time and creates a load balancing ratio. The need for this method is rare and is not recommended for large pools.

Dynamic Application Level (Layer 7) Methods

- **Fastest (node) to fastest (application)**
  This method chooses a server based on the least number of current sessions. It requires assignment of both a layer 7 and a TCP-type profile to a virtual server. If a layer 7 profile is not configured, the virtual server falls back to the least connections load balancing mode. Least connections mode only uses active connections in its calculations. The Fastest methods are useful in environments where nodes are distributed across separate logical networks.
- **Least sessions**
  This method selects the server currently having the least number of entries in its persistence table. Use of this method requires that the virtual server reference a type of profile that tracks persistence connections, e.g., Universal or Source Address Affinity. This method works best in environments where load balanced servers have similar capabilities.
- **Least pending requests**
  This is the best distribution method for accurately balancing a load across a web server array. Measuring how many HTTP requests it is currently processing is the most accurate way of ascertaining the load on a server. The least pending requests method enables optimal traffic distribution according to the real-time load on the server. This method does not support "session stickiness;" requests from a single source can be forwarded to different web servers.

More than One Flavor of Load Balancers

Today, load balancers come in three basic flavors: physical appliance, virtual appliance, or cloud-based service. Each has its own advantages, disadvantages, and technology subsets:

Physical Appliance

Traditionally, load balancers were placed at the edge of a network, in between the router/firewall and a server cluster. Traffic would enter through the firewall, with the load balancer directing it to the least busy application server. Over time, physical load balancing appliances morphed into multi-function devices, combining firewall and other services. In essence, this made the load balancer the network gatekeeper—an all-in-one, unified solution for controlling traffic destined for services and applications.

Not all load balancers evolved into firewalls and multi-faceted appliances. Some became application delivery controllers (ADCs), while others became enterprise caching devices and SSL accelerators—shifting only some of the firewall and clustered server burden to the load balancers.

Regardless of the incorporated services, these devices all shared something in common: they were proprietary hardware, physically located in the same data center as application servers and other infrastructure components. The supposed advantages offered were centralized control, ease of deployment, and ease of scalability (where multiple devices could be chained together to provide scale, failover, or additional capabilities). The downside of the physical appliance model was high hardware costs, high software costs, frequent maintenance, patching, and a lack of geographical distribution offerings. In other words, appliances worked fine for a single site. Meanwhile, the ability to support branch offices, multiple global locations, and geographically-dispersed server farms became very difficult, very expensive, and—in some cases—impossible to deploy.

Physical appliances are a good fit when using in an on-premises data center, or managed hosting solution—such as Rackspace or other noncloud- based hosting provider. When using a load balancing appliance in this scenario, it has direct access to application servers with which it is interfacing. For mission-critical business applications, some organizations prefer the physical appliance hosting model because of its rock solid nature.

Virtual Appliance

The real power driving any load balancer is its software. In physical appliances, the software usually runs on a proprietary operating system, which in turn runs on proprietary hardware—normally tuned for high performance. In many cases the hardware relies on advanced ASIC processors, as well as large amounts of high-speed RAM and local storage.
The virtual appliance shifts the software away from proprietary hardware by using a hypervisor, which is the term assigned to a virtual machine (VM). The VM creates a compatible environment in which the load balancing software functions. In other words, load balancers can be configured to run on off-the-shelf hardware equipped with appropriate network interface cards and enough processing power, RAM, and storage to meet the virtual appliance needs.

A virtual machine offers several advantages over a physical appliance:

- There is no need for proprietary hardware; organizations can use less expensive, commonly-available servers to run load balancing services.
- Expensive hardware service contracts, proprietary hardware options, et al., don't inflate the capital outlay.
- Virtual appliances are easier to back up and move to other servers, minimizing problems associated with hardware failures.
- Vendors can focus on advanced software development rather than hardware support.

Given these advantages, many vendors offer VM versions of their load balancers as replacements for their physical products. Because they can use less expensive hardware having a lower TCO, this model is ideal for organizations whose applications and data centers already use some form of hypervisor or virtualization. This is

While virtual appliances may offer a better solution that physical devices, there are still limitations. Since the software is running on a VM, virtual appliances may lack the throughput capabilities of a sophisticated physical device, since specialized ASICs, enhanced networking cards, and other performance-orientated hardware are removed from the picture can impact performance.

Other disadvantages can include issues with the chosen virtualizationplatform, where patches and/or upgrades can lead to incompatibilities. Further, virtual appliances introduce an additional layer, VM management usually being performed via a separate console.

Load Balancing as a Service

Often referred to as cloud load balancers, load balancing as a service offers a very robust solution for distributed networks, multi-branch offices, and enterprise networks. A cloud load balancer can take many different forms: the basic concept being its location in a cloud data center where it handles load balancing, failover, and other chores using a service-based methodology.

Cloud based load balancers have some inherent advantages over other load balancing technologies. First and foremost is their ability to unify load balancing across external cloud services. For example, businesses leasing SaaS offerings can use a cloud-based load balancing service to act as an ADC, thereby eliminating single points of failure by dynamically rerouting traffic around outages.

For optimal distribution and capacity utilization, cloud-based load balancing can manage back-end servers, while predictively and dynamically scaling at the traffic management layer. And when functioning as an ADC, a cloud load balancer can leverage a decentralized cluster of brokers to maintain continuity, maximize throughput, and eliminate application downtime.

Other advantages include:

- **High availability**
  A cloud-based service can leverage prearranged and automated controls to failover to distributed service servers, while automatically rerouting traffic around outages, interruptions, and failed network connections.

- **Centralized control**
  The load balancer's management console is often integrated with other selected services, offering a unified view into the CDN. This makes it much easier to institute changes and determine how they impact the complete CDN.
- **Caching web data/sites**
  Commonly-requested data can be stored in an ADC cluster, reducing the number of requests to web servers. In turn, this reduces response times, increases throughput, and reduces loads on application/web servers.
- **Security**
  Attacks such as DDoS, SQL injection, and other malicious traffic can be detected and blocked at the load balancer level, thereby preventing harmful traffic from arriving at an application/web server.
- **Flexibility**
  Cloud-based offerings provide elasticity and automated scale, where traffic loads can be monitored and spin up additional resources as needed. Optional features (e.g., caching, analytics, and SSL acceleration) can be added or removed as your organization's environment changes.
- **Reduced startup costs**
  By eliminating the need to purchase hardware, software, and other consumables, costs are contained. No capital expenditures are usually needed to leverage a cloud-based service; fees are based upon services consumed and are easily budgeted.

Turning to the cloud for load balancing makes a great deal of sense for organizations seeking to operate hybrid networks—where a combination of SaaS applications, internal/local applications, and multiple access sites are involved. In fact, it has become a very common practice among businesses adopting the cloud as a distribution medium. This strategy also makes sense for enterprises running systems in cloud computing environments such as Amazon Web Services (AWS), EC2, and VMware.

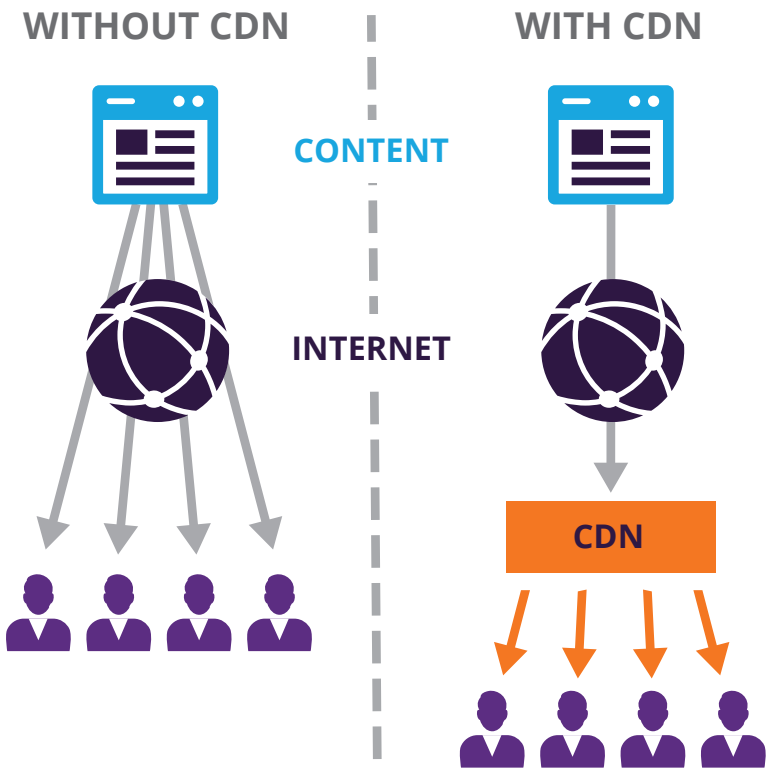
More to Load Balancing Than Meets the Eye

Beyond routing traffic, load balancers can be used to ensure session persistence, as well as redundancy for highly-active systems. Regarding redundancy, load balancers offer several different methods to enable failover:

- **Failover cable based**
  One of the easiest methodologies for enabling failover comes in the form of a cable connection between devices. It's used when a pair of load balancers operate together to deliver traffic to a cluster (or multiple clusters). Using a "heartbeat" signal transmitted over the failover cable, the load balancers are aware of each other's operation. If one signal fails, the other load balancer assumes the role of master and handles the traffic. However, this failover method can introduce other problems, such as the inability to ascertain the reason for failureor the possibility of creating multiple, contradictory paths to the cluster (if spanning tree protocol is enabled on the network). What's more, cable-based failover systems require the member devices to be located within physical proximity.
- **Stateful failover**
  This method is used to preserve session information in the event of a load balancer failure. With stateful failover, both load balancers share information about the traffic. If one unit fails, the other one has all of the traffic information necessary to maintain a connection, without loss of data. Simply put, if the active unit fails, the standby unit maintains a persistent connection without impacting the end user. Stateful failover normally requires that the load balancing devices be located within physical proximity.
- **Distributed failover**
  This method keeps sessions "sticky," where a user's traffic is always routed to the same server to maintain persistence. It is especially important for those visitors accessing e-commerce applications. IP traffic monitoring frequently checks to determine if a given service remains responsive. If the service is down, the affected device is removed from load balancing rotation; traffic is forwarded to an alternate device or the next member of the cluster.

Load balancers take on additional roles in the enterprise, providing services that were once only in the domain of standalone appliances. These include caching, content delivery, SSL acceleration, and security services. Examples include:

- **Integrated caching**
  Within their cache, load balancers can store frequently-accessed, relatively static data, which can help accelerate content delivery. With their providing cached content, client devices don't have to wait for a response from a remote server and data is received more quickly. Ideally, the cache dynamically adjusts to the type of content being requested, and then optimizes the content in the delivery of static pages.
- **Content distribution network (CDN)**
  A content-distribution network (CDN) is a network of data centers which work together as a single service that provides performance benefits to both application users and operations alike. These benefits including faster and more consistent response times. This is typically accomplished by concurrently serving content out of local data centers with a closer physical proximity to end users than the server of origin, thus cutting down on transit time and distance.



- **SSL acceleration**
  SSL can present a problem for high traffic environments. Encryption and decryption services force many organizations to deploy appliances designed to handle the additional load placed on network resources. In the past, load balancers were unable to act on SSL-based traffic; because packet payloads are encrypted, the load balancer was prevented from ascertaining little more than destination information. However modern load balancers can decrypt SSL-based traffic and being strategically placed in front of server pools and clusters, makes them ideal to take on an SSL acceleration role. Enterprises should demand much from any load balancing solution, especially given the unique position it has within each network. Regardless of whether it is deployed as a cloud-based service, virtual appliance, or physical device, a load balancer can offer much more than traffic prioritization and server pool delegation. Instead, modern load balancers can act as the foundation of a CDN, providing power behind SSL connectivity, application accelerators, and security devices.

Making the Smart Choice

No longer does choosing a load balancing solution have to be an exercise in guesswork. There is ample evidence for how a load balancing solution fits into corporate ROI equations, and TCO calculations are relatively easy to find. In addition to economic considerations, the following best practices are recommended to 1) ease the vetting process, and 2) lead IT departments in making the best choice within the shortest timespan.

- **Identify immediate and long term needs**
  For many, picking a load balancer is driven by traffic problems, where growth has exceeded the capabilities of a given infrastructure. Usually, it is high server utilization driving the selection process, where an organization concludes that scale is necessary and additional servers must be added. However, simply choosing a load balancer as a way to equally distribute traffic across a server farm is very short-sighted. You should also consider how the network should be distributed, what level of failover is required, the demands placed on infrastructure by encryption, in addition to security and continuity implications of building a server cluster.

- **Calculate anticipated loads**
  In many cases, an organization selects a given load balancing solution to meet its current traffic growth. However, many either overspend on hardware (and services) that isn't needed, or under-buy, creating a situation where they quickly outgrow their chosen solution. To avoid these common pitfalls, IT needs to identify the underlying factor behind traffic growth.

  For example, some businesses are seasonal in nature, experiencing growth only during certain peak shopping periods. Another business may experience sudden growth due to a new product launch, or a new service added to its portfolio, while yet another experiences growth due to a mergers or acquisition. Regardless of cause, decision makers need to calculate growth and then select the appropriate solution offering the requisite level of elasticity.

- **Plan for growth**
  Whether or not your organization needs features such as SSL acceleration, caching, or content delivery is a critical factor in choosing the best solution. However, in many cases those decisions can be deferred until a later date, provided the selected solution offers future support for these features.

- **High availability (HA)**
  Although a load balancing solution may be chosen to reduce server utilization and maximize traffic flow, IT departments should not ignore the importance of high availability and business continuity. An HA system is designed to transparently maintain availability should some or all of the production environment fail or suffer from an outage. This option can be the difference between a profitable business and afailure to conduct business. Organizations should always make HA a priority when choosing a load balancing solution.

- **Security concerns**
  Introducing load balancing into your network environment means more traffic can be handled and—most likely—more application servers are involved in the data center. Moreover, load balancing also enables the geographical distribution of server clusters/pools. From a security perspective, this equates to attackers having more targets. Security should be therefore be incorporated into your load balancing solution.

- **Support**
  One of the most critical, yet often overlooked issues in choosing a load balancer comes in the form of support. You need to make sure you can get replacement parts, service, and be able to reach tech support at a moment's notice in the event of a problem.

- **Create baselines**
  Determine what your basic requirements are for deploying a load balancer. Which applications need high availability and application delivery. Remember to calculate peak and idle loads to determine scale levels and whether caching, compression, or encryption is needed.

While IT managers should consider these best practices, executives need to weigh the financial aspects in selecting the load balancing solution. These include identifying deployment and support costs, keeping in mind elements such as:

- **ROI**
  Calculating the ROI for any piece of IT hardware, software, andancillary services) can be a complex accounting exercise. With load balancers, however, the process is simplified by comparing the cost of lost business against what a load balancing solution brings to the table. For example, downtime can be calculated and used as a metric to compare against load balancing deployment costs. Most vendors offer calculations and worksheets to assist IT managers in determining the ROI; use these materials to generate baseline numbers.
- **TCO (Total cost of ownership)**
  Calculating an accurate TCO estimate facilitates the budgeting process, especially when evaluating the perceived value offered by any one solution. Decision makers need to calculate not only the initial deployment costs, but also ongoing expenditures such as service contracts and software upgrades.

Conclusion

Choosing a load balancing platform can be a complex endeavor requiring an in-depth understanding of your enterprise infrastructure for content and application delivery. The process is further complicated by the wide variety of available options. However, the cloud shows great promise as a technology equalizer—letting businesses experience all that load balancing has to offer without incurring huge capital outlays and encountering limits of scale.

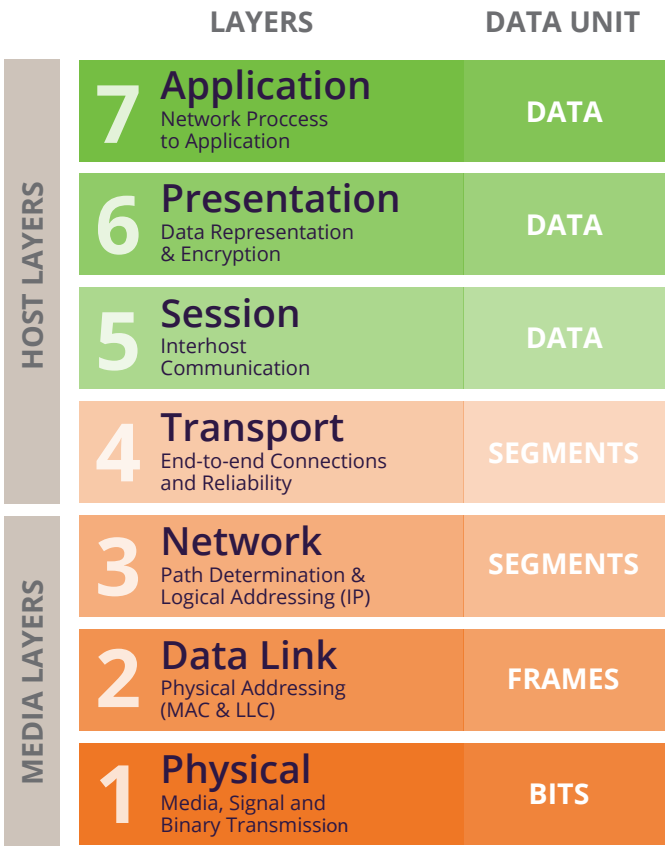
It can even be argued that a single-site operation can benefit from cloudbased load balancers, since many of these offerings include security features. These include DDoS protection, SQL injection attack prevention, and many other security-related features that can help to secure any website.

The key is to pick a solution after considering the implications of future growth, scale, and elasticity. Once these elements are quantified, your decision process becomes much easier—letting stakeholders focus on the financial aspects, as well as offered feature sets.

Appendix

Understanding the OSI Model

The OSI model, referred to in many instances by specific layers, can be a complex concept for many IT decision makers. However, it is little more than a way to categorize the type of traffic that travels over the network, e.g., the payload contained in a packet and how it interacts with networking components. The best way to understand how the OSI model impacts network operations is by defining its seven layers:



The OSI model

To add intelligence in moving traffic across a network, load balancing solutions, CDNs, routers, ADCs, firewalls, and other networking equipment/services can use OSI model layers. For example, a routing device that uses application layer data is often referred to as application aware and can make routing decisions based upon information provided by an application.

IMPERVA INCAPSULA

About Imperva Incapsula

Imperva Incapsula is a cloud-based application delivery service that protects websites and increases their performance, improving end user experiences and safeguarding web applications and their data from attack. Incapsula includes a web application firewall to thwart hacking attempts, DDoS mitigation to ensure DDoS attacks don't impact online business assets, a content delivery network to optimize web traffic, and a load balancer to maximize the potential of web environments.

Only Incapsula provides enterprise-grade website security and performance without the need for hardware, software, or specialized expertise. Unlike competitive solutions, Incapsula uses proprietary technologies such as client classification to identify bad bots, and big data analysis of security events to increase accuracy without creating false positives.