Scalable Web Hosting Service

Ludmila Cherkasova
Computer Systems and Technology Laboratory
HP Laboratories Palo Alto
HPL-1999-52(R.1)
October, 1999

E-mail:{cherkasova}@hpl.hp.com

Web hosting service, web server farm, web server cluster, load balancing, scalability, performance analysis, SpecWeb

Web hosting is an infrastructure service that allows to design , integrate, operate and maintain all of the infrastructure components required to run web-based applications. It includes Web server farms, network access, data staging tools and security firewalls. Web server farms are used in a Web hosting infrastructure as a way to create scalable and highly available solutions. One of the main problems in web server farm management is content management and load balancing.

In this paper, we analyze several hardware/software solutions on the market and demonstrate their scalability problems. We outline a new scalable solution FLEX for design and management of an efficient Web hosting service. This solution can be applied to a Web hosting service implemented on different architecture platforms such as web server farms with replicated disk content, web server clusters having access to a shared file system or multi-computer systems using a global (shared) file system.

A preliminary performance analysis provides a comparison of the current solutions and FLEX using a synthetic workload generator based on SpecWeb'96 benchmark. FLEX outperforms current solutions 2-7 times.

Contents

1 Introduction

As the popularity of the Web grows, an increasing number of businesses are wishing to seize the potential market opportunities that it offers. The number of public web sites grows exponentially, and the business users account for the majority of that growth.

A successful business regard the Web not as a network service only, but as a market: buyers and sellers, competitors and partners, products and services - the core elements of the web market.

As the Web increasingly becomes a core element of business strategy, so the task of hosting web content has become mission critical. Few companies, however, have the resources, money and expertise to build their web site entirely in-house. Forrester Research Inc. suggests that the cost of setting up an internal web site can be $221,000 in the first year. For this reason, many businesses choose to outsource their Web hosting to Internet service providers and some equipment vendors, which according to Forrester can slash costs by 80%. Although such cost savings are an important factor, it is not the primary reason for outsourcing Web hosting. Forrester's research shows that the needs for knowledge, increased security, better service and support are all seen as more important in the decision to outsource.

Web server farms and clusters are used in a Web hosting infrastructure as a way to create scalable and highly available solutions.

One popular solution is a farm of web servers with replicated disk content (see Figure 1).

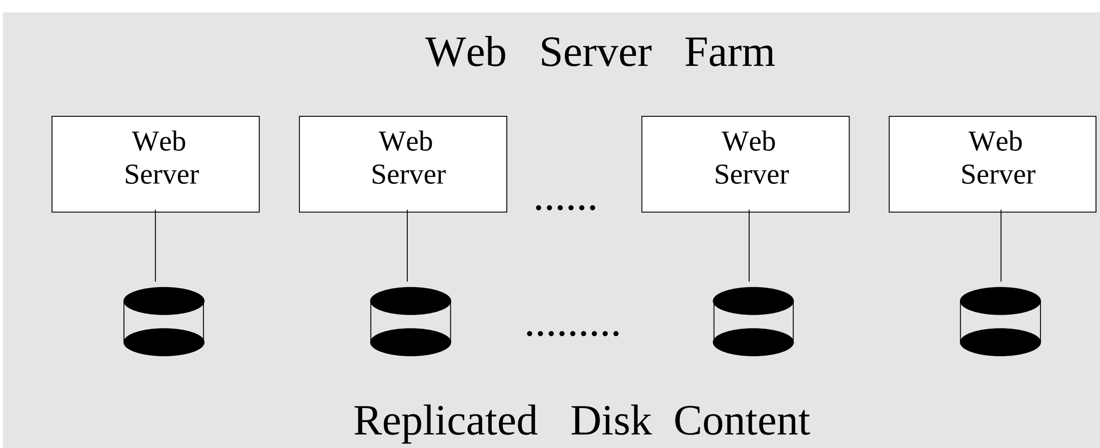

Figure 1: Web Server Farm with Replicated Disk Content.

Another popular solution is a clustered architecture, which consists of a group of nodes connected by a fast interconnection network, such as a switch. In a flat architecture, each node in a cluster has a local disk array attached to it. As shown in Figure 2, the nodes in a cluster are divided into two *logical* types: front end (delivery, HTTP servers) and back end (storage, disks) nodes. The (logical) front-end node gets the data from the back-end nodes using a shared file system. In a flat architecture, each physical node can serve as both the logical front-end and back-end, all nodes are identical, providing both delivery and storage functionality

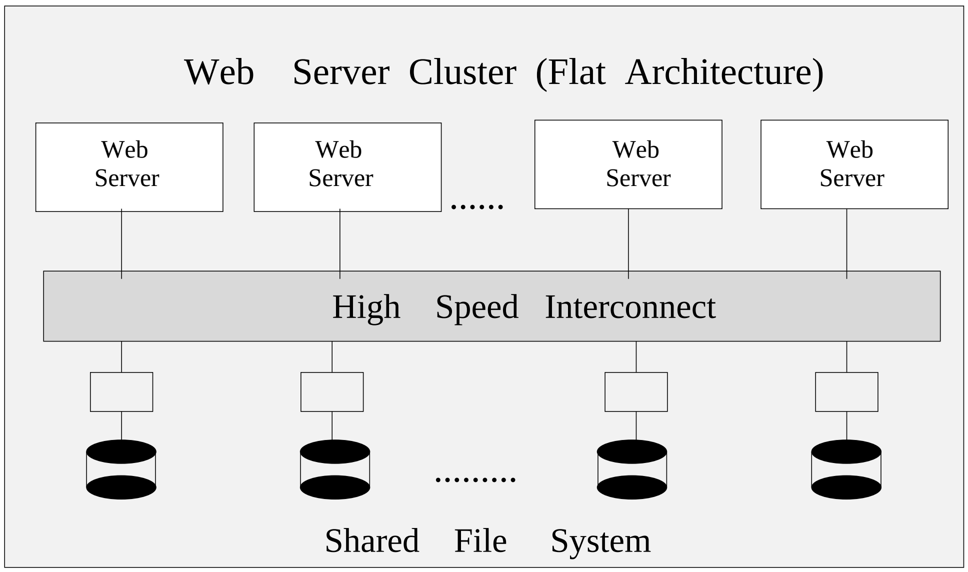

Figure 2: Web Server Cluster (Flat Architecture).

In a two-tiered architecture shown in Figure 3 the logical front-end back-end nodes are mapped to different physical nodes of the cluster and are distinct. It assumes some underlying software layer (e.g., virtual shared disk) which makes the interconnection architecture transparent to the The NSCA prototype of the scalable HTTP server based on two-tier architecture is described and studied in [NSCA94, NSCA95, NSCA96].

In the all solutions, each web server has the access to the whole web content. Therefore, any server can satisfy any client request.

One of the main problems in web server cluster (farm)[1] management is content management and load balancing. In this paper, we analyze several hardware/software solutions on the market and demonstrate their scalability problems.

Load balancing (of either kind) for a cluster (farm) of web servers pursues the goal to equally distribute the load across the nodes. This solution interferes with another goal of efficient RAM usage for the cluster (farm).[2] The popular files tends to occupy RAM space in all the

[1]We often use the terms of web server cluster and web server farm interchangeably, because the problems as well as the solutions are often very similar. Only in those cases when it matters, it is clearly specified.

[2]We consider only workloads with working set that does not fit in a single server nodes RAM.

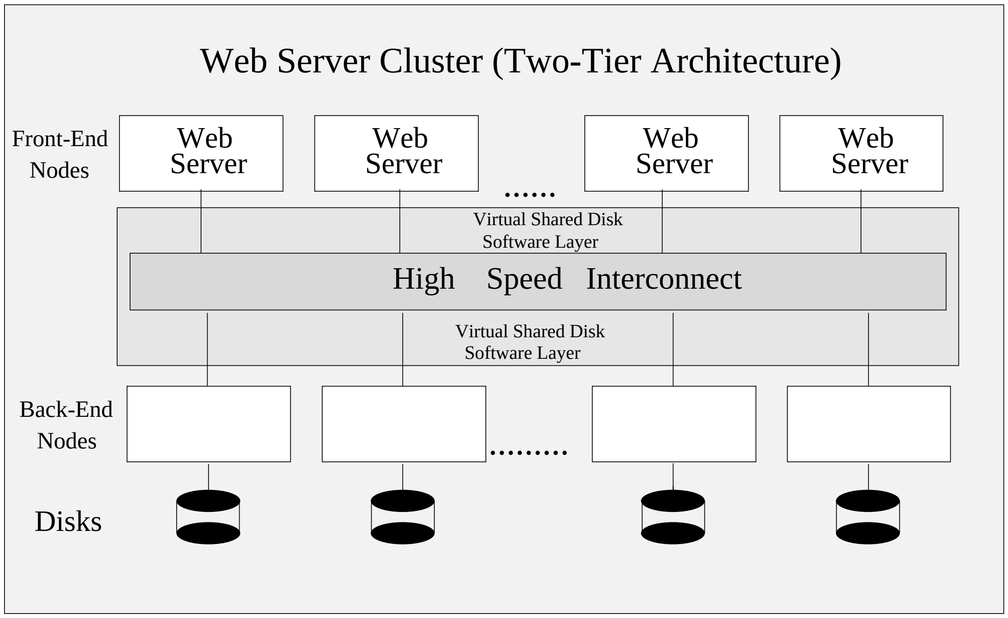

Figure 3: Web Server Cluster (Two Tier Server Architecture).

nodes. There is a redundant replication of “hot” content through the RAM of all the nodes which leaves much less of available RAM space for the rest of the content, leading to a worse overall system performance. This observation have led to a design of the new “locality aware” balancing strategies [VLN97, LARD98].

The cluster of web servers is considered in [VLN97]. They suggest the architecture similar to NCSA approach, but to store the data on disks attached to the servers in the cluster rather than in a centralized file server (flat architecture, see Figure 2). In order to exploit the aggregate memory capacity of the cluster, a new memory management strategy is proposed to control the data caching and data replacement achieving a good balance between an intra-cluster network traffic and disk I/O. Despite that the original load balancing on a cluster is done with Round-Robin DNS [RRDNS95](see more discussion in Section 4), the proposed memory management strategy allows to avoid unnecessary data replication and improve the overall performance of the cluster.

A new locality-aware request distribution strategy (LARD) is proposed for cluster-based network servers in [LARD98]. The cluster nodes are partitioned into two sets: front ends and back ends. Front ends act as the smart routers or switches. They implement LARD to route the incoming requests to the appropriate node in a cluster. LARD takes into account both a document locality and the current load. Authors show that on workloads with working sets that do not fit in a single server nodes RAM, the proposed strategy allows to improve throughput by a factor of two to four for 16 nodes cluster.

In this paper, we outline a new scalable solution FLEX for design and management of an efficient Web hosting service. The solution can be applied to a Web hosting service implemented on different architecture platforms such as web server farms, web server clusters or multi-computer systems [HP-MCS].

FLEX motivation is similar to the "locality aware" balancing strategies discussed above: we would like to avoid the unnecessary document replication to improve the overall performance of the system.

However, we achieve this goal via logical partition of the content on a different *granularity level.* Since the original goal is to design a scalable web hosting service, we have a number of customers and their sites as a starting point. Each of these sites might have different traffic patterns in terms of both the number and types of files accessed and the average access rates. By monitoring the traffic to each site and analyzing the combined traffic to a system in whole, FLEX proposes a balanced partitioning of the customers (web sites) by the number of nodes in the system.

The elegance and simplicity of the approach consists that the desirable routing can be done by submiting the correspondent configuration files to the DNS server, since each hosted web site has a unique domain name. The DNS server is going to route the incoming requests to a correspondent node in the system (accordingly to partition provided by FLEX). FLEX can be easily implemented on a top of the current infrastructure used by Web hosting service providers.

A preliminary performance analysis provides a comparison of the current solutions and FLEX using synthetic workload generator based on SpecWeb'96 benchmark. Depending on a size of the cluster, FLEX outperforms current solutions 2-7 times. In this report, we outline only the FLEX idea. The separate upcoming report is devoted to the detailed description of the algorithm, as well as proposes more advanced performance analysis.

The rest of the report is structured as follows. Section 2 provides a short analysis of Web hosting market and its future forecast, based on IDC numbers. Section 3 shows some tendencies in web sites size and rate growth, and motivates why static planning capacity methods do not work. Section 4 provides overview of typical hardware and software solutions used for Web hosting and load balancing. Section 5 analyzes performance pitfalls in scalability of the current Web hosting solutions. Section 6 outlines current Web hosting technology based on a notion of a virtual server. Sections 7, 8 outline the new solution FLEX and provide its preliminary performance analysis.

2 Web Hosting Market and Future Forecast

Due to explosive growth of the Web and an increasing demand on the servers, Web content hosting is an increasingly common practice. In Web content hosting, providers who have a large amount of resources (for example, bandwidth to the Internet, disks, processors, memory, etc.) offer to store and provide Web access to documents from institutions, companies and individuals who lack the resources, or the expertise to maintain a Web server, or are looking for a cost efficient, "no hassle" solution. The service is typically provided with a fee, though some servers do not charge fees for non-commercial accounts.

Demand for Web hosting and e-commerce services continues to grow at rapid pace. IDC [IDC98] forecast rapid growth for Web hosting over the next five years. In particular, businesses will invest heavily in Web hosting services following the resolution of the Y2K compliance issues.

Table 1 describes the IDC forecast for the Web hosting market, which is expected to reach nearly $12 billion by 2002.

	1997	1998	1999	2000	2001	2002
Revenues ($M)	414	770	1,479	3,018	6,095	11,825
Growth (%)	-	86	92	104	102	94

(1)

Figure 4 is a graphical presentation of the revenue growth: 30 times in 5 years.

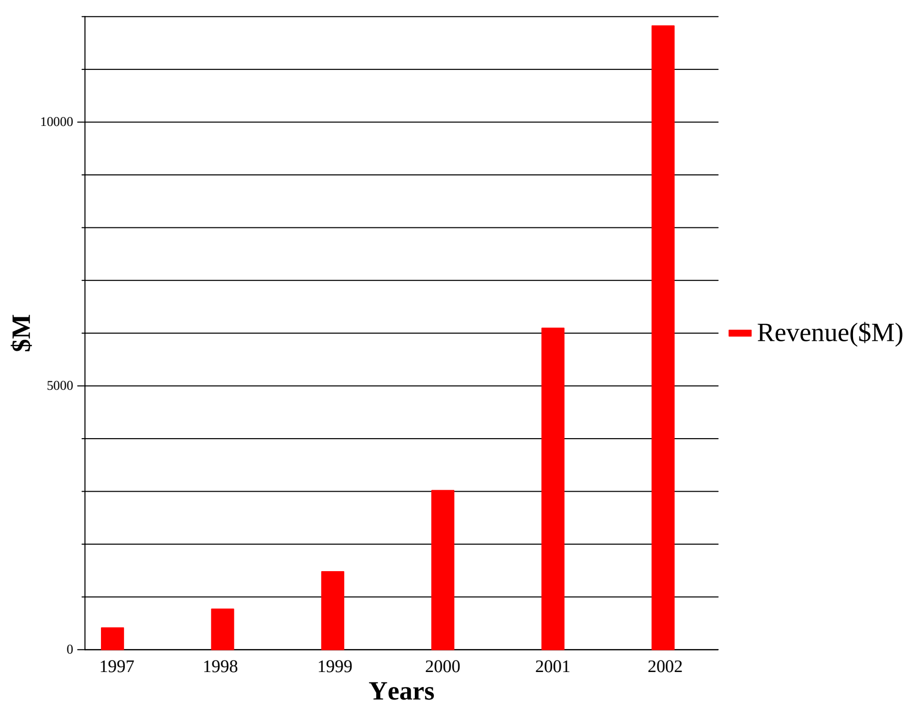

Figure 4: IDC Forecast of Revenue Growth for Web Hosting Services.

The web is being used for communication, research, marketing, customer support, selling and collaborative working. Many businesses achive payback within months owing to cost savings, attraction of new business, speed to market or better market intelligence.

Typical uses of a web site are wide and varied. Most people immediately think of advertising and selling. But companies are finding new applications to exploit the interactivity, availability and multimedia capabilities of the web all the time. Some of the most common activities include:

- *Customer services:* a web page can be used to offer 24-hour support, seven days a week. The information given can be far more detailed than is possible by phone.
- *News services:* they can offer instant up-to-date information around the clock about new product launches, partnership announcements, etc.
- *Complementary telemarketing:* businesses can use a web site to complement (800)-number reception desks.
- *Market research:* web sites can be used to obtain feedback from customers, suppliers, or employees.
- *On-line product catalogues:* retail, mail order, and telemarketing-based companies can use web sites for on-line product catalogues.
- *On-line purchasing:* to sell products and services over the web.
- *Product distribution:* this is popular with software companies which distribute free trial versions of their product.

Web hosting services are represented by the following market segments:

- shared web hosting
- dedicated web hosting, which includes complex dedicated hosting and custom hosting.

The shared hosting market targets small and medium size businesses. It is a robust, high volume, low-unit cost business. The most common purpose of a shared hosting Web site is marketing. In this case, many different sites are hosted on the same hardware.

The dedicated hosting market provides interactive capabilities, offers integration services with back end systems, and provides dedicated hardware for the site.

The following Table 2 from [IDC98] describes major distinctions among web hosting market segments:

Web Hosting Segment	Characteristics	Revenue per Month per Customer	Leading Vendors	Competitive Factors
Shared Hosting	Primarily Unix, low cost, product oriented, supports credit card transactions, consumer-to-business, small merchants, payment interfaces,collocation	Less than $1,500	netNet	Cost, e-commerce applications, customer service, management reports, human adjunct interface
Dedicated Hosting	NT and Unix, medium cost, product oriented, consumer-to-business e-commerce, payment interfaces, collocation	$1,500-20,000	Digex, Epoch, GTE, IBM, PSINet, UUNet, MCI	Rich set of specialized applications, multiple Web hosting centers, service guarantees, systems integration, multicurrency support
Complex Dedicated Hosting	NT and Unix, medium cost, product oriented, consumer-to-business business-to-business, payment interfaces, large merchants	$20,000-50,000	Digex, Exodus, GTE, IBM, UUNet	Quick-response application tuning, design and implementation of unique Web sites, site mirroring and, load balancing, hot sites support, data base support human interface support
Custom Hosting	NT and Unix, high cost, slight product emphasis, consumer-to-business business-to-business, solutions tailored to business opportunity, skills and tools driven	$40,000-250,000	Digex, Exodus, GTE, IBM, EDC MCI	Creative design, systems integration, custom Web site tools, business/market consulting,fulfillment human adjunct interface

(2)

More than 6,000 companies offer Web-hosting services, ranging from Web giants like Uunet Technologies and GTE Internetworking to smaller regional firms. Traditional business can save quite a bit of money – tens of thousands dollars, in some cases – by letting a Web hosting service store your Web data. Many companies – ranging from Fortune 100 firms to one-person storefronts – are outsourcing their Web sites. Larger firms, especially those seeing significant traffic, can save tremendously by using the providers' networking equipment and broadband connections to the Internet.

After an initial boom in the number of Internet service providers, the past 18 months have been witness to some industry consolidation, see Table 3.

ISP	Partner or Parent
AT&T WorldNet Services Inc.	TCG, CERFNet
Digex Inc.	Intermedia
GTE Internetworking	BBNet Planet, Genuity
MCI Web Services	MCI/WorldCom
Netcom	ICG
PSINet	IXC

(3)

IBM Global Network is the leading Web hosting provider with 21.7% market share. While AT&T WorldNet is the second largest Web hosting provider (11.1% market share), the combined services of WorldCom Advanced Networks represent the third largest Web hosting provider. The impact that MCI's merger with WorldCom will have on the Web hosting market is still difficult to gauge.

Hewlett-Packard has a part in Web hosting services as a hardware provider with specialized middleware like Web Quality of Service (Web QoS) designed to provide management for Website usage, availability and prioritization. PSINet provides three types of Web hosting services based in part on technology from a new agreement with Hewlett-Packard.

Another interesting angle of web hosting is the proportion and the amount of servers used to support it.

For example, Digex Inc. (its market share is estimated by IDC to 2.9%) provides dedicated servers exclusively. Digex has a large Sun server web farm that supports Unix-based Web hosting, and it owns and operates the world's largest dedicated Web site management facility for Windows NT, with more than 500 Windows NT servers. Overall, Digex manages 900 dedicated servers for 650 customers.

Frontier GlobalCenter is another large web hosting service provider (its market share is estimated by IDC to 4.8%) and it targets, so-called, complex web hosting. Frontier GlobalCenter supports dedicated servers on both the Unix and NT operating systems. A year ago, the mix was 90% Unix and 10% NT. Now, the mix has shifted to 70% Unix and 30% NT.

3 Variety of Web Sites and Their Growth

The recent paper [MS97] provides an excellent statistical analysis of web server logs obtained from different sites. It presents a taxonomy of these sites and characterizes their access patterns, and more importantly, their growth.

The sites in this survey [MS97] can be broadly described in three basic categories:

1. Academic sites:
   - Harvard University Electrical Engineering and Computer Sciences;
   - Harvard University Faculty Arts and Sciences;
   - Rice University Electrical and Computer Engineering;
2. Business:
   - Traditional Business;
   - ISP company page;
   - Adult-Entertainment;
   - Free Web Software Site;
   - Web Site Designer.
3. Informational:
   - Organization for Members of same Profession;
   - Government Agency.

The following Table 3 represents the size of the sites and the traffic handled during the recent month:

Site Name	Requests	Monthly Transfer Rate(MB)	Files on Site	MB on Site
Traditional Business	321,747	3,819	347	2.8
Harvard EECS	106,001	1,322	5,835	196.0
Harvard FAS	2,328,401	15,097	34,348	455.0
ISP	8,139	39	134	1.5
Rice ECE	85,763	854	4,655	115.0
Adult Content	69,906	857	223	5.5
Organization	42,301	251	95	0.8
Web Site Designer	43,523	104	119	0.7
Government Agency	26,049	214	185	1.2
Free Software	15,982,085	76,153	4,070	136.0

(4)

The disparity in levels of traffic and site size illustrate the fundamental difference in Web sites.

The study asserts that the three primary issues that characterize the site are:

- site composition and growth;

- growth in traffic;
- user access patterns.

The monthly growth of the requests rates for different sites differs significantly. As the growth for nearly all these sites is exponential, it takes different time to double. As we can see from the Table 3 there are sites like Free Software Site that nearly double each month, while other sites grow more slowly: for example, Web Site Designer doubled during a 12 month period. Some of these sites experience decrease of the traffic rates and actually demonstrate negative growth.

Site Name	Double (if positive growth) or Half(if negative growth) Interval
Traditional Business	2 months
Harvard EECS	3 months
Harvard FAS	3 months
ISP	3 years
Rice ECE	6 months
Adult Content	-3 months
Organization	-3 months
Web Site Designer	1 year
Government Agency	11 months
Free Software	1 month

(5)

User access patterns differ significantly too. For example, the Free Software Site has a singular, wildly popular product. The accesses on this site are heavily skewed: 2% of the documents account for 95% of the site's traffic. As more people learn about the software, more people visit the site to download the software.

For the Business site using the Web to market aggressively, the access rate growth occurs in bursts which correspond to major reorganization and renovation of the site.

The access rates to academic sites grow with the user population: as the site grows in size, so do the number of the requests to that site.

These types of large-scale distributed systems face new design and management problems and require new solutions. Large clusters can dwarf the power of the largest machines. The ability to grow clusters (farms) incrementally over time is a tremendous advantage in areas such as Internet service deployment: where capacity planning depends on a large number of unknown variables. Incremental scalability replaces inadequate static capacity planning with relatively fluid and dynamic reactionary scaling.

The results in [MS97] have a direct connection to Web hosting services. In order to design an efficient, high quality Web hosting solution the specifics of access rates and users' patterns should be taken into account. The traffic growth/decrease and the users' access patterns changes should be monitored in order to provision for those changes in time and in the most efficient way.

4 Typical Hardware and Software Solutions for Web Hosting

Web hosting is an infrastructure service that allows to design, integrate, operate and maintain all infrastructure components required to run web-based applications. It includes Server farms, network access, data staging tools and security firewalls. Some business require an environment supporting business critical processes, which must offer 24x7 availability, enterprise-wide scalability, worldwide coverage and trusted security. Many other businesses have simpler requirements, and are operating web environments that do not have the same critical requirements.

Web server farms and clusters are used in a Web hosting infrastructure as a way to create scalable and highly available solutions.

One popular solution is a farm of web servers with replicated disk content (see Figure 1). This architecture has certain drawbacks:

- replicated disks are expensive, and
- replicated content requires content synchronization, i.e. whenever some changes to content data are introduced – they have to be propagated to all of the nodes.

Another popular solution is a clustered architecture, which consists of a group of nodes connected by a fast interconnection network. In a flat architecture, each node in a cluster has a local disk array attached to it. As shown in Figure 2, the nodes in a cluster are divided into two *logical* types: front end (delivery) and back end (storage) nodes. The (logical) front-end node gets the data from the back-end nodes using a shared file system. In a flat architecture, each physical node can serve as both the logical front-end and back-end, all nodes are identical, providing both delivery and storage functionality

In a two-tiered architecture shown in Figure 3 the logical front-end back-end nodes are mapped to different physical nodes of the cluster and are distinct. It assumes some underlying software layer (e.g., virtual shared disk) which makes the interconnection architecture transparent to the nodes.

In all the solutions, each web server has the access to the whole web content. Therefore, any server can satisfy any client request.

What are the problems a service provider faces when trying to design a scalable and cost efficient solution using web server farms? One of the main problems is **content management and load balancing.** This problem we are going to address in this paper, and propose a new solution called FLEX.

The market now offers several hardware/software load-balancer solutions that can distribute incoming stream of requests among a group of Web servers.

Hardware load balancers are positioned between the Internet and Web server farm. It connects to the Internet router and the internal LAN using two separate network segments (typically Ethernet). It acts as a fast regulating valve between the Internet and the pool of servers. The load balancer uses a virtual IP address to communicate with the router, masking the IP addresses of the individual servers. Only the virtual address is advertised to the Internet community, so the load balancer also acts as a safety net. The load balancer's other network segment connects to a hub or switch with a pool of multiple physical servers attached. Typical configuration is shown in Figure 5.

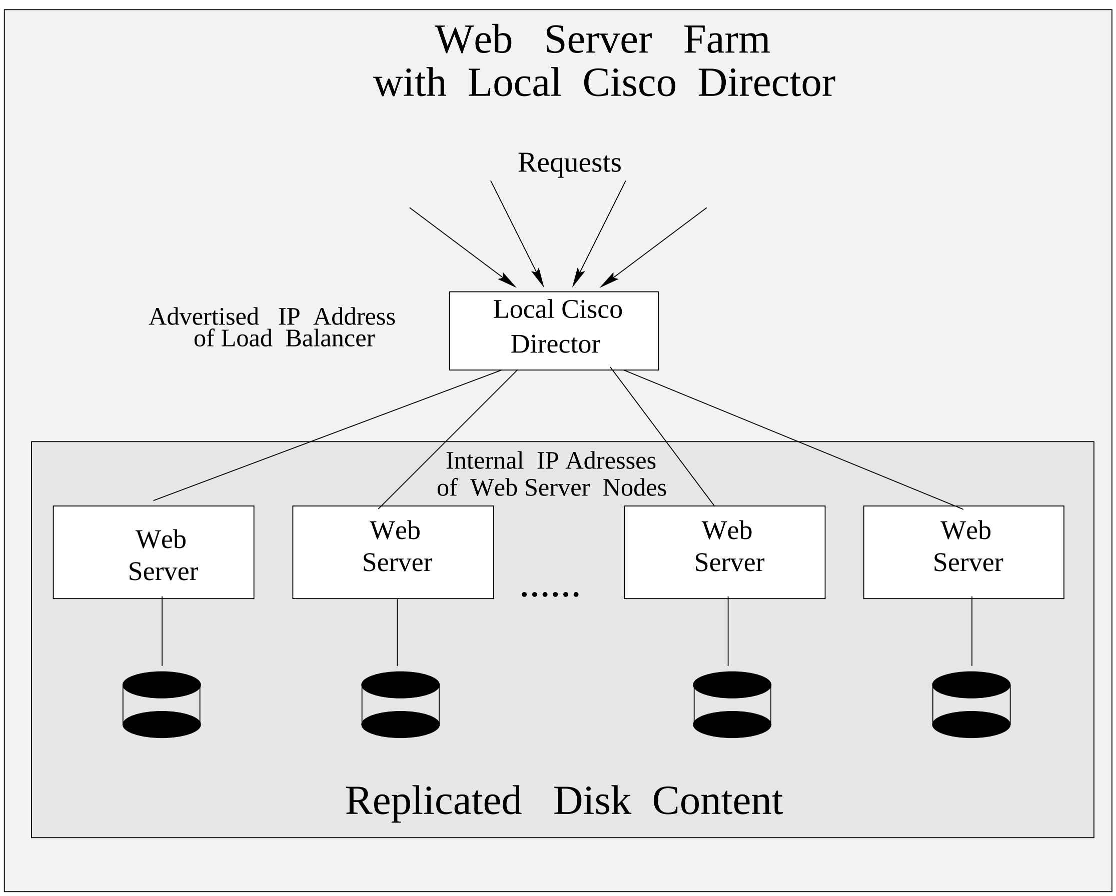

Figure 5: Web Server Farm with Hardware Load Balancer.

Load balancing hardware is provided by different vendors like Cisco Systems' Local Director

[Cisco], Alteon's ACE switch [Alteon], HydraWEB Technologies' HydraWeb [HydraWEB], F5Labs' BIG/ip [F5Labs], etc.

Hardware switches mentioned above are expensive (Cisco Local Director cost around $32,000). They might significantly increase a solution cost. All traffic to the content is directed through the switch. The balancing methods across different switches vary, but in general, the idea is to forward the request to the least loaded server in a cluster. If only one switch is used then it introduces a single point of failure. Minimal configuration of two switches increases the solution cost even further. Clearly, scalability might be a problem, since a switch could become a bottleneck.

Software load balancing on a cluster is a job traditionally assigned to a Domain Name System (DNS) server. Round-Robin DNS [RRDNS95] is built into the newer version of DNS. Round-Robin DNS distribute the access among the nodes in the cluster: for a name resolution it returns the IP address list (for example, list of nodes in a cluster which can serve this content, see Figure 6), placing the different address first in the list for each successive requests. Ideally, the different clients are mapped to different server nodes in a cluster.

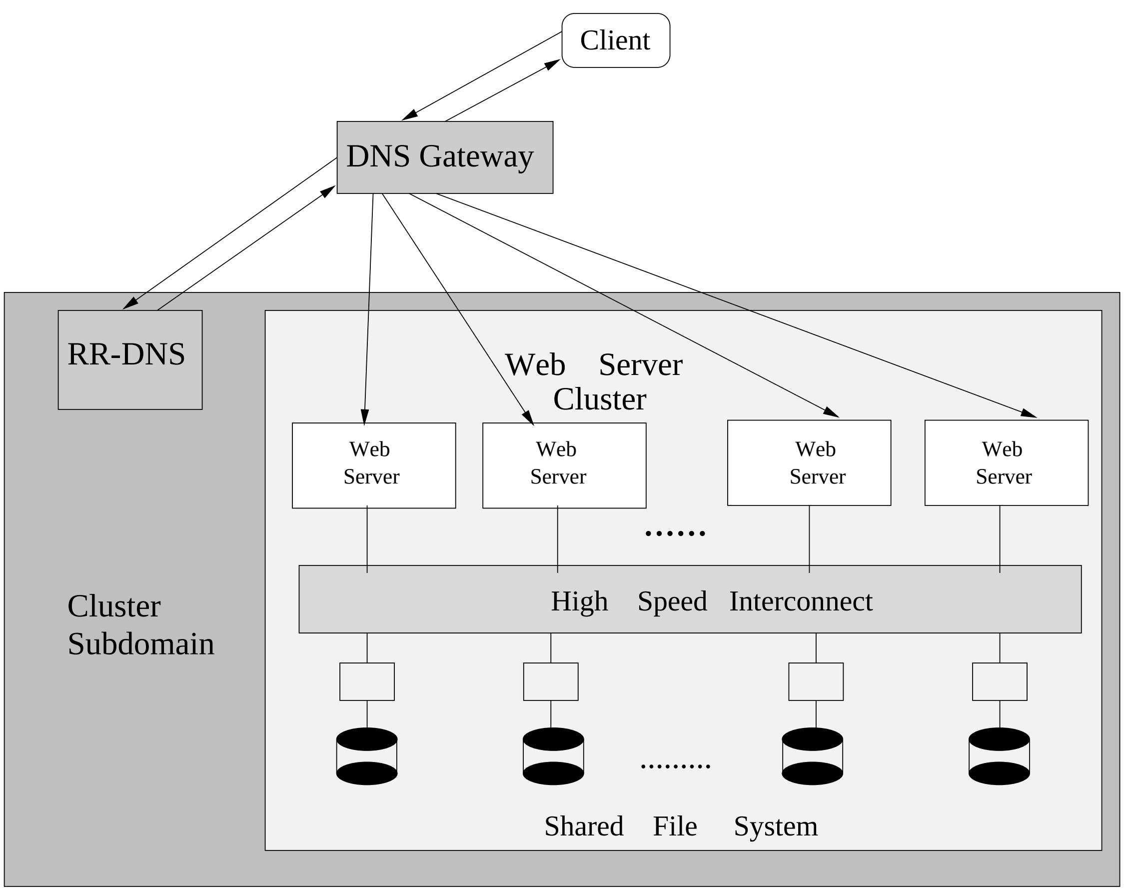

Figure 6: Web Server Farm with Hardware Load Balancer.

In most of the cases, Round-Robin DNS is widely used: it is easy to set up, it does provide reasonable load balancing and it is available as part of DNS which is already in use, i.e. there is no additional cost.

The new FLEX solution (which we are going to describe later) uses the DNS server as a part of its load balancing solution.

5 Performance Pitfalls in Scalability of the Current Solutions

A cluster (farm) of servers is used to increase the capacity and compute power of the solution.[1] Ideally, a cluster (farm) of N web servers should be N times more powerful than one web server.

However, to create a scalable solution one has to overcome number of problems in a design. We discussed some of them above: content management and load balancing.

Web server performance greatly depends on efficient RAM usage. A web server works faster when it pools pages from a cache in RAM. Moreover, its throughput is much higher too. We've measured web server throughput when it supplied files from the RAM (i.e. the files were already downloaded from disk and resided in the File Buffer Cache), comparing it against the web server throughput when it supplied files from the disk. Difference in throughput was more than 10 times.

One of the typical remedies to improve the web server performance is to increase RAM size and to configure a bigger File Buffer Cache. The significance of efficient RAM usage is difficult to underestimate.[2]

Load balancing (of either kind) for a cluster of web servers pursues the goal to equally distribute the load across the nodes. This solution interferes with another goal of efficient RAM usage for the cluster. The popular files tends to occupy RAM space in all the nodes. This redundant replication of "hot" content through the RAM of all the nodes leaves much less of available RAM space for the rest of the content, leading to a worse overall system performance. Under such an approach, a cluster having N times bigger RAM (which is a combined RAM of N nodes) might effectively have almost the same RAM as one node, because of the replicated popular content through the RAMs in the cluster.

An orthogonal approach is to partition the content and in such a way to use RAM space more efficiently. However, static partitioning will inevitably lead to an inefficient, suboptimal and inflexible solution, since the changes in access rates as well as access patterns tend to vary dramatically over time (see the tables above from [MS97]) while static partitioning does not accommodate for this.

[1] Cluster (farm) of computers is a common way of improving availability too. However, in this paper, we mostly concentrate on scalability and performance issues.

[2] The case of our interest is when the overall file set is greater than the RAM of one node. If the file set completely fits to the RAM, the Round-Robin DNS is the good solution to use.

6 Virtual Servers and Multiple Domains

Although the resent survey revealed over 1 milion web servers on the Internet, the number of web site exceeds this number several times. The illusion of more web sites existing than actual web servers is created through the use of *virtual servers (hosts)*.

Web hosting service is based on this technique. Web hosting service uses the possibility to create a set of virtual servers on the same server. There are different alternatives how it can be done.

Unix web servers (Netscape and Apache) have the most flexibility to address the web hosting problem. Multiple host (domain) names can be easily assigned to a single IP address. It allows to create an illusion that each host has its own web server, when in reality multiple "logical" hosts share one physical host.

There is also a possibility to create a separate IP address per each host. It could be used as well in Web hosting solution when the number of hosted sites is rather limited. It is less scalable when a web hosting service is dealing with a large number of relatively small sites.

For a while, in NT world, the second alternative was the only alternative. WWW Publishing Service can be configured to answer requests for more than one single domain name. To accomplish this, one should requests the IP addresses for the primary server and for each additional virtual server. Finally, these additional IP addresses have to be included in TCP/IP protocol configuration and DNS (or WINS) configuration files in order to resolve the IP addresses to the correspondent domain names. As noticed above, this solution is a good solution when the number of hosted sites is rather limited. It does not scale.

Last version of NT web server (IIS-4) introduces a new feature which allows to create multiple host names which can be assigned to the same IP address. This feature makes NT web server world to look similar to the Unix one.

As we will see further, we can use virtual servers and their IP adresses for load balancing on a web server farm (or a web server cluster) by making them visible in a special way to the outside world via DNS server.

One way to implement a Web Hosting Service will be to use a web server farm (or web server cluster) which has access to the whole content (whether it is achieved with replicated content or shared distributed file system). For example, there are total of 100 different sites (customers) which would like to publish their information on WWW. Each physical web server creates a virtual server per customer site, and announces prescribed IP addresses to correspondent domain names via DNS server configuration files. For load balancing in such a cluster, Round Robin DNS is a typical solution. The disadvantages of such an approach were described in Section 5.

The other simple way to approach the load balancing in such a cluster, is to statically partition and assign the customers to the servers. For example, 100 customers could be partitioned as 10 customers per server in the configuration of 10 web servers. However, any such static partition can not take into account changing traffic patterns as well as nature of changes in the content of the sites. So, it can not adjust the partition to accommodate and provision for the traffic and sites dynamics.

As we will see further, we can use virtual servers and their IP adresses for load balancing on a web server farm (or a web server cluster) by making them visible in a special way to the outside world via DNS server.

7 New Scalable Web Hosting Solution: FLEX

A new scalable solution, called FLEX, for shared Web Hosting consists in the following. By monitoring the access patterns and access rates to the customers content, the overall content can be **logically partitioned** in a number of "equally balanced" groups by the number of cluster (farm) nodes. Each customer group is serviced by some prescribed node in a cluster (farm).

For example, there is a total of C customers hosted on a cluster (farm) of N web servers. For each customer c, a "customer profile" CP_c is built. A customer profile CP_c consists of two following basic characteristics:

- AR_c - the access rates to a customer's content, i.e. *bytes/sec* requested of this customer content.
- WS_c - the total size of the most often requested files, so-called "working set".

The next step is to partition all the customers in N "equally balanced" groups: $C_1, ..., C_N$ in such a way, that cumulative access rates and cumulative "working sets" in each of those C_i groups are approximately the same.

The final step is to prescribe a web server N_i from a cluster (farm) to each group C_i.

REMARK1: The variation of the algorithm can be used for additional load (rate) balancing. For example, one site has a high bursty traffic. To smooth the high access rates this site can be assigned to be served by two or more servers (depending on the balancing goal and a threshold for desirable rate).

REMARK2: This algorithm can be used to provide a desirable degree of high availability, additionally to load balancing. For example, the algorithm can prescribe minimum 2 (or 3, if desired) nodes per site to increase the site availability in case of the other node failure.

Note, that the load balancing is easier in such configuration since the rate per customer site decreases correspondingly to the number of nodes prescribed.

The FLEX solution is supported by providing the correspondent information to a DNS server via configuration files. A resolution of the customer domain name is going to be a correspondent IP address on the prescribed node (nodes) in the cluster (farm). This solution is flexible and easy to manage. Tuning can be done on a daily, weekly or even hourly (if necessary) basis. Once the server logs analysis shows enough changes in the average traffic rates and patterns and finds a new, better partitioning of the customers for this cluster (farm), then new DNS configuration files are generated. Once a DNS server has updated its configuration tables,[2] new requests are routed accordingly to the new configuration files, which leads to more efficient traffic balancing on a cluster (farm).

Such a self-monitoring solution allows to observe changing users' access behaviour and to predict future scaling trends, and plan for it. This solution could also be used to provision some special advertisement or promotion campaigns when one could expect very high traffic rates for a certain content during some period of time. In those cases, for example, "hot" content can be prescribed to access via all the nodes in a cluster (farm).

8 Preliminary Performance Analysis of FLEX

In order to estimate potential performance benefits of a new solution as well as to illustrate the pitfalls of the current solutions, the high level simulation model of web cluster (farm) has been built using C++Sim [Schwetman95]. These results present the upper bound (the ideal case) of the performance benefits of FLEX, assuming the content and rate balancing can be done exactly. The specific cases are data and traffic (trace) dependent. We will present and analyze few of such cases in the future report. We feel that less specific and more general results outlined in this section do support the proposed approach in clear way.

SpecWeb96 [SpecWeb96] is the industry standard benchmark for measuring web server performance. Using a finite number of clients to generate HTTP requests they retrieve different length files according to a particular file size distribution.

SpecWeb96 file mix is defined by the files (requests) distribution from the following four classes:

- 0 Class: 100bytes - 900bytes (35%)
- 1 Class: 1Kbytes - 9Kbytes (50%)
- 2 Class: 10Kbytes - 90Kbytes (14%)

[2] The entries from the old configuration tables can be cached by some servers and used for request routing without going to DNS server. However, the cached entries are valid for a limited time only dictated by TTL (*time to live).* Once TTL is expired, the DNS server is requested for updated information.

- 3 Class: 100Kbytes - 900Kbytes (1%)

The web server performance is measured as a maximum achievable number of connection per second supported by a server when retrieving files in the required file mix. Current typical web servers running SpecWeb96 could achieve 500 - 4000 connections per second per processor (abbreviation: Ops/sec).

As a workload for our simulation model, we used SpecWeb96 like file mix.

The model uses the following basic assumptions about the web server capacity:

- web server throughput is 1000 Ops/sec to retrieve the files from the RAM (file buffer cache).
- web server throughput is 10 times lower (i.e. 100 Ops/sec) to retrieve the files from the disk than from the RAM.

Accordingly to the SpecWeb requirements: the total size of the file set for one web server running 1000 Ops/sec is 750MB. In such a way, if a web server under test has a RAM of 750MB or larger [3] then all the files can be eventually brought to the RAM, and all the consequent requests are going to be satisfied from the RAM.

When we use a cluster (farm) of web servers, we scale the size of serving content in the following way: a content served by the N nodes cluster (farm) is N times larger than a content of one node.

Load balancing (of either kind) for a cluster (farm) of web servers pursues the goal to equally distribute the load across the nodes. The described above software/hardware balancing solutions result in a simple Round-Robin balancing schema for this synthetic workload.

Fugure 7 shows a miss ratio (i.e. percentage of requests served from the disks) for different RAM sizes, different number of servers in a cluster (farm), and different balancing solutions.

First of all, the miss ratio per web server decreases when the server RAM size increases. It is natural and has been expected.

Second, Round-Robin balancing schema has a degrading performance as the number of nodes in a cluster (farm) increases. FLEX is a truly scalable solution: its miss ratio is defined by the RAM size and independent on a number of nodes in a cluster (farm).

Third , FLEX has much lower miss ratio for any RAM size. This difference between the Round-Robin balancing solution and the FLEX solution becomes especially dramatical for a

[3] Here and through the rest of the paper, we use RAM size to denote, in fact, a File Buffer Cache size.

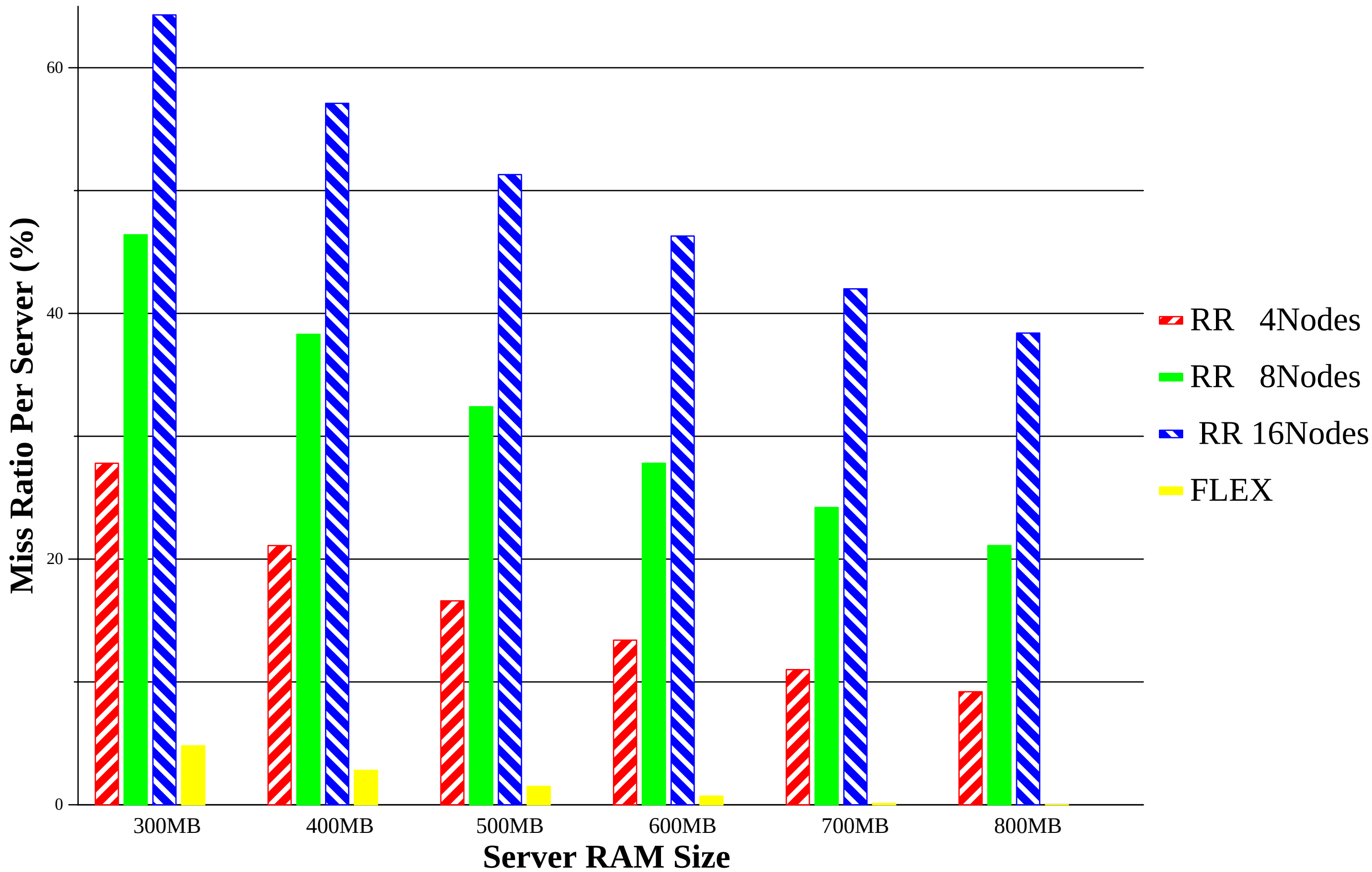

Figure 7: Miss Ratio per Server in the Cluster (Farm) of 4, 8, and 16 Nodes under Different Balancing Solutions.

larger RAM sizes.

Let us consider the results for the RAM size of 800MB. As it was mentioned before, a content served by one web server is total of 750MB. It means, that a whole content can be stored in RAM and any request can be satisfied from RAM, resulting in zero miss ratio. When the cluster (farm) size increases, the size of the total content increases correspondently. The total RAM size in a cluster (farm) (cumulative of all the RAM's in the cluster (farm)) increases the same way as well. Theoretically, the total RAM size allows to store a whole content. However, in reality each web servers in a Round-Robin balancing solution serves the files from the whole content which leads to replicated, “hot” files across the different RAMs. And as a result, the bigger is the cluster (farm) – the higher is the miss ratio per server. The new FLEX solution has a perfect (zero) miss ratio in this case, since we've partitioned the total content in the balanced groups, each one can be perfectly served by a correspondent web server.

Table 6 is a table representation of the miss ratio per web server in Fugure 7 for different RAM sizes, different number of servers in a cluster (farm), and different balancing solutions.

Web Hosting Solution Type	Server RAM Size						
	200MB	300MB	400MB	500MB	600MB	700MB	800MB
Round-Robin Balancing Schema Cluster of 4 Nodes	38.3%	27.8%	21.1%	16.6%	13.4%	11%	9.2%
Round-Robin Balancing Schema Cluster of 8 Nodes	57.2%	46.4%	38.3%	32.4%	27.8%	24.2%	21.1%
Round-Robin Balancing Schema Cluster of 16 Nodes	73.1%	64.3%	57.1%	51.3%	46.3%	42.0%	38.4%
FLEX Scalable Solution Cluster of N Nodes	9.1%	4.8%	2.8%	1.5%	0.7%	0.1%	0%

(6)

Table 6: Miss Ratio per Server in the Cluster (Farm) of 4, 8, and 16 Nodes under Different Balancing Solutions

The following Fugure 8 shows the server throughput for the different RAM sizes and the different number of servers in a cluster (farm). Web server throughput is directly inverse to a miss ratio: the higher is the miss ratio, the worse is the web server throughput.

Thus, the overall cluster (farm) performance drops for the bigger cluster (farm) sizes under Round-Robin balancing solution, while FLEX demonstrate a superior scaling performance.

For smaller RAM sizes FLEX outperforms Round-Robin balancing solution more than 2-4 times, while for larger RAM sizes this difference gets even higher: for the cluster (farm) of 16 nodes, balancing provided by FLEX leads to 7 times higher web server throughput. FLEX utilizes RAM's across the cluster (farm) in the most efficient way.

Table 7 is a table representation of the web server throughput in Fugure 8 for different RAM sizes, different number of servers in a cluster (farm), and different balancing solutions.

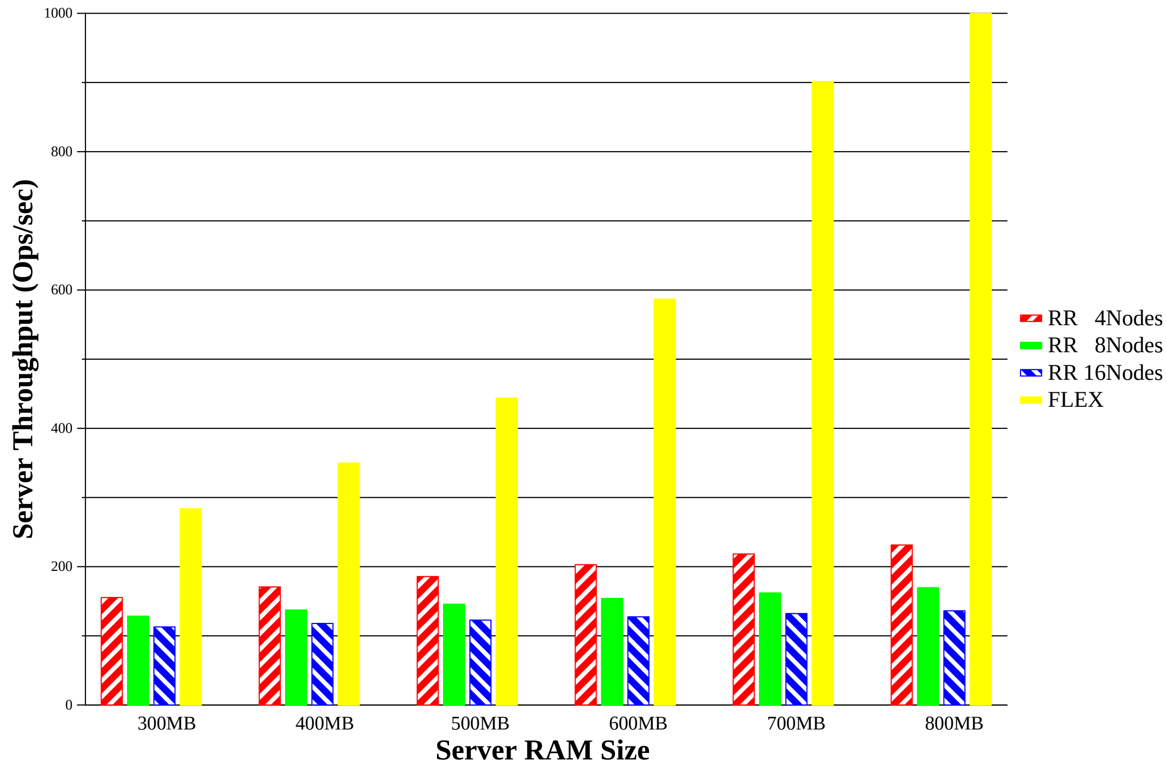

Figure 8: Server Throughput in the Cluster (Farm) of 4, 8, and 16 Nodes under Different Balancing Solutions.

Web Hosting Solution Type	Server RAM Size						
	200MB	300MB	400MB	500MB	600MB	700MB	800MB
Round-Robin Balancing Schema Cluster of 4 Nodes	138.7	155.4	170.7	185.7	202.9	218.3	231.3
Round-Robin Balancing Schema Cluster of 8 Nodes	118.9	128.4	137.4	145.7	154.0	162.0	169.6
Round-Robin Balancing Schema Cluster of 16 Nodes	108.1	112.9	117.9	122.7	127.5	132.2	136.2
FLEX Scalable Solution Cluster of N Nodes	224.3	283.7	349.8	443.8	586.8	901.4	1000

(7)

Table 7: Web Server Throughput (Ops/sec) in the Cluster (Farm) of 4, 8, and 16 Nodes under Different Balancing Solutions

The next question to address is the generality of our assumptions.

How much the assumptions of the web server capacity and the relative slowdown of the disk accesses do influence the performance results?

In the following reasonings, we use the following notations:

- OPS - a web server capacity when the requests are served from the RAM;
- $SlowDown$ - a coefficient of slowdown when the requests are served from the disk;
- $Miss_{rr}$ - a miss ratio under round robin balancing strategy, i.e. ratio of the requests missed in the RAM and satisfied from disk;
- $Miss_{flex}$ - a miss ratio under new FLEX balancing strategy, i.e. ratio of the requests missed in the RAM and satisfied from disk.

The server throughput (the upper bound) under the Round Robin balancing strategy could be approximated as following:

$$(1 - Miss_{rr}) * OPS + Miss_{rr} * \frac{OPS}{SlowDown}$$

Correspondingly, the server throughput (the upper bound) under the FLEX balancing strategy could be approximated as:

$$(1 - Miss_{flex}) * OPS + Miss_{flex} * \frac{OPS}{SlowDown}$$

Both equations can be rewritten in the following way:

$$(1 - Miss_{rr} + \frac{Miss_{rr}}{SlowDown}) * OPS$$

$$(1 - Miss_{flex} + \frac{Miss_{flex}}{SlowDown}) * OPS$$

The performance improvements FLEX vs Round Robin strategy can be approximated asthe ratio:

$$\frac{(1 - Miss_{flex} + \frac{Miss_{flex}}{SlowDown}) * OPS}{(1 - Miss_{rr} + \frac{Miss_{rr}}{SlowDown}) * OPS}$$

It could be simplified in the following way:

$$\frac{1 - Miss_{flex} + \frac{Miss_{flex}}{SlowDown}}{1 - Miss_{rr} + \frac{Miss_{rr}}{SlowDown}}$$

As we can see, the values of $Miss_{flex}$ and $Miss_{rr}$ play the crucial role in this equation. The value of *SlowDown* has less influence, whether its value is 7, 10 or 15 times, the impact of it on the performance improvements FLEX vs Round Robin strategy is less signigicant, it is a "correction" factor.

The value of server capacity *OPS* has been reduced from the equation. The slight impact of *OPS* is implicitly hidden in the definition of $Miss_{flex}$ and $Miss_{rr}$ (via the scaling factor of the file set size). The values of $Miss_{flex}$ and $Miss_{rr}$ are mostly defined by the RAM size (relatively to the file set size) and the efficiency of the balancing strategy.

Therefore, the performance results provided in this Section outline rather general performance benefits achievable with the new balancing strategy FLEX.

9 Conclusion

Web hosting is an infrastructure service that allows to design, integrate, operate and maintain all infrastructure components required to run web-based applications. Web server farms are used in a Web hosting infrastructure as a way to create scalable and highly available solutions. One of the main problems in web server farm management is content management and load balancing.

In this paper, we analyzed several hardware/software solutions on the market and demonstrated their scalability problems. We outlined a new solution FLEX which provides a truly scalable performance. A preliminary performance analysis provides a comparison of the current solutions and FLEX using synthetic workload generator based on SpecWeb'96 benchmark. FLEX outperforms current solutions 2-7 times.

The benefits of the FLEX can be summarized as follows:

- FLEX is a cost-efficient balancing solution. It does not require installation of any additional software: by analyzing the server logs, FLEX generates a favorable partitioning, and forms updated information for a DNS server. Shortly, after new configuration files are submitted to a DNS server, a new balancing schema takes place.
- FLEX is a self-monitoring solution. It allows to observe changing users' access behaviour and to predict future scaling trends, and plan for it.
- FLEX is truly scalable solution. The performance of FLEX is mainly defined by the server RAM size rather than a number of servers in a cluster (farm).
- FLEX allows to save an additional hardware by more efficient usage of available resources. It could outperform current market solutions up to 2-7 times. Additionally, as a useful side effect: it allows to use less number of file servers and disks at the system back end.

Through the paper, we mostly proposed FLEX as a balancing solution for shared Web hosting using web server farms. In fact, it can be used to design and manage the dedicated and custom sites, having rich, but well-structured content. By introducing additional domain names for the parts of the content, it is possible to used FLEX approach to balance the access to these parts of the content in the cluster (farm).

10 References

[Alteon] URL: http://www.alteon.com/products/acelerate-data.html

[Cisco] URL: http://www.cisco.com/warp/public/751/lodir/

[F5Labs] URL: http://www.f5labs.com/

[HP-MCS] The HPL Multi-Computer Systems (MCS) Program. URL: http://www.hpl.hp.com/research/itc/csl/x86/

[HydraWEB] URL: http://www.hydraweb.com/z2_index.html

[IDC98] Web Hosting and E-Commerce Service Providers. Analysts: P. Burstyn, P. Johnson. International Data Corporation, Doc.No 16894, October 1998.

[LARD98] V.Pai, M.Aron, G.Banga, M.Svendsen, P.Drushel, W. Zwaenepoel, E.Nahum: Locality-Aware Request Distribution in Cluster-Based Network Servers. In Proceedings of the 8th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS VIII), ACM SIGPLAN,1998, pp.205-216.

[MS97] S.Manley and M.Seltzer: Web Facts and Fantasy. Proceedings of USENIX Symposium on Internet Technologies and Systems, 1997, pp.125-133.

[NSCA94] E. Katz, M. Butler, R.McGrath: A Scalable HTTP Server: The NSCA Prototype. Computer Networks and ISDN Systems, v.27, pp.155-163, 1994.

[NSCA95] T. Kwan, R. McGrath, D.Reed: NSCA's World Wide Web Server: Design and Performance. IEEE Computer, November 1995, pp.68-74.

[NSCA96] D. Dias, W. Kish, R. Mukherjee, R. Tewari: A Scalable and Highly Available Web Server. Proceedings of COMPCON'96, Santa Clara, 1996, pp.85-92.

[Schwetman95] Schwetman, H. Object-oriented simulation modeling with C++/CSIM. In Proceedings of 1995 Winter Simulation Conference, Washington, D.C., pp.529-533, 1995.

[SpecWeb96] The Workload for the SPECweb96 Benchmark. URL http://www.specbench.org/osg/web96/workload.html

[RRDNS95] T. Brisco: DNS Support for Load Balancing. RFC 1794, Rutgers University, April 1995.

[VLN97] S.Venkataraman, M.Livny, J.Naughton: Memory Management for Scalable Web Data Servers. Proceedings 13th International Conference on Data Engineering, IEEE Comput. Soc. Press, April 1997, pp. 510-519.