

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
KHOA CÔNG NGHỆ THÔNG TIN

— * —

ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC
NGÀNH CÔNG NGHỆ THÔNG TIN

**CÂN BẰNG TẢI CHO CÁC HỆ
WEBSERVER LỚN VÀ ĐẢM BẢO
SCALABILITY**

Sinh viên thực hiện : **Võ Duy Pho**

Lớp CNPM - **K49**

Giáo viên hướng dẫn: **TS. Nguyễn Khanh Văn**

HÀ NỘI 6-2009

PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

1. Thông tin về sinh viên

Họ và tên sinh viên: Võ Duy Pho

Điện thoại liên lạc: 0944599926

Email: d.for.vo@gmail.com

Lớp: CNPM-K49

Hệ đào tạo: Chính qui

Đồ án tốt nghiệp được thực hiện tại: Trường đại học Bách Khoa Hà Nội

Thời gian làm ĐATN: Từ ngày 22/02/2009 đến ngày 29/05/2009

2. Mục đích nội dung của ĐATN

Tìm hiểu về lý thuyết cân bằng tải cho các hệ thống web server lớn bao gồm nhiều web server được cài đặt ở nhiều nơi khác nhau, tìm hiểu về lý thuyết xây dựng bộ cân bằng tải cho các web server, cài đặt các thuật toán cân bằng tải, khắc phục một số nhược điểm của các thuật toán này. Tìm hiểu các phương án cài đặt và cấu hình các bộ cân bằng tải vào hệ thống web server nhằm đảm bảo được tính mở rộng cao và khả năng chống lỗi.

3. Các nhiệm vụ cụ thể của ĐATN

- Tìm hiểu về kiến trúc website với khả năng mở rộng
- Tìm hiểu về cân bằng tải cho các hệ thống web-server lớn, các kỹ thuật cân bằng tải server
- Tìm hiểu về lý thuyết xây dựng bộ cân bằng tải, cân bằng tải cho server, cân bằng tải cho server toàn cầu và cân bằng tải cho cache
- Tìm hiểu và cài đặt một số thuật toán cân bằng tải bằng ngôn ngữ C dựa trên một sản phẩm cân bằng tải mã nguồn mở, đề xuất một số phương pháp cải tiến thuật toán.
- Cấu hình và cài đặt các bộ cân bằng tải vào hệ thống theo các kiến trúc khác nhau, đảm bảo được khả năng mở rộng và khả năng chống lỗi của hệ thống.

4. Lời cam đoan của sinh viên:

Tôi – *Võ Duy Pho* - cam kết ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *TS Nguyễn Khanh Văn*.

Các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

Hà Nội, ngày 25 tháng 5 năm 2009

Tác giả ĐATN

Võ Duy Pho

5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của ĐATN và cho phép bảo vệ:

Hà Nội, ngày tháng năm

Giáo viên hướng dẫn

TS. Nguyễn Khanh Văn

LỜI CẢM ƠN

Trong suốt 5 năm học tại trường đại học Bách Khoa Hà Nội vừa qua, em đã thu nhận được rất nhiều kiến thức bổ ích từ trong sách vở, và quý báu hơn là từ kinh nghiệm, tâm huyết của chính các thầy cô truyền lại. Em xin gửi lời cảm ơn chân thành tới các thầy cô đã tận tình chỉ bảo và giúp đỡ em trong thời gian qua.

Đặc biệt, em xin gửi lời cảm ơn sâu sắc nhất tới thầy TS. Nguyễn Khanh Văn đã trực tiếp hướng dẫn và giúp đỡ em hoàn thành đồ án tốt nghiệp này.

Em cũng xin gửi lời cảm ơn chân thành tới anh Nguyễn Vũ, các anh tại công ty ERAS Việt Nam và các bạn đã nhiệt tình chia sẻ kinh nghiệm cũng như cung cấp các tài liệu, công cụ giúp em hoàn thành đồ án tốt nghiệp này.

Em xin chân thành cảm ơn!

TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP

Sự bùng nổ của ngành dịch vụ web trong những năm gần đây khiến số lượng người truy cập vào mạng internet ngày càng tăng mạnh, đặc biệt là các website mạng xã hội và các website chia sẻ video trực tuyến. Với hàng trăm triệu lượt truy cập mỗi ngày, các website này đòi hỏi phải có một hệ thống server cực kỳ mạnh mẽ. Một hệ thống hoạt động tốt sẽ khiến người dùng hài lòng, sau một thời gian hoạt động số lượng người dùng sẽ tăng lên, máy chủ của hệ thống sẽ bị quá tải, dẫn đến yêu cầu nâng cấp. Sau khi nâng cấp lại tiếp tục như cũ. Vòng tuần hoàn này dẫn đến nhu cầu cần phải xây dựng một hệ thống website với khả năng đáp ứng người dùng trong một thời gian đủ dài và dễ dàng nâng cấp mở rộng khi cần. Đó chính là kiến trúc web với khả năng mở rộng.

Vấn đề quan trọng nhất trong kiến trúc web với khả năng mở rộng chính là cân bằng tải cho hệ thống web server. Một website toàn cầu sẽ có rất nhiều web-server đặt ở nhiều nơi trên thế giới, cân bằng tải cho hệ thống web-server này nghĩa là làm cách nào để các webserver không bị quá tải, luôn đáp ứng được nhu cầu của người dùng trong thời gian nhanh nhất. Trong sự cạnh tranh khốc liệt giữa các website dịch vụ, nhà quản trị nào có thể đáp ứng được nhu cầu của người dùng một cách tốt nhất, người đó sẽ thắng, vì vậy nhu cầu cân bằng tải là vô cùng cần thiết và là vấn đề sống còn đối với các nhà cung cấp dịch vụ web.

Có nhiều phương pháp để cân bằng tải, tuy nhiên lý thuyết chung là cần phải có một bộ cân bằng tải đứng giữa người dùng và hệ thống webserver. Bộ cân bằng tải này sẽ nhận yêu cầu từ phía người dùng, sau đó chuyển hướng các yêu cầu này đến các server phù hợp dựa trên các thuật toán phân tải, sau đó nhận dữ liệu từ server và trả về cho người dùng.

Trong đồ án tốt nghiệp này, người viết luận văn (NVLV) sẽ đưa tổng quan về các vấn đề cần phải giải quyết trong quá trình thiết kế hệ thống web với khả năng mở rộng. Sau đó sẽ đi sâu vào lý thuyết cân bằng tải cho các hệ thống webserver lớn, các kỹ thuật và phương pháp cân bằng tải hiện đang được sử dụng. Tiếp theo sẽ là kiến trúc và các thành phần cần xây dựng của một bộ cân bằng tải, các thuật toán phân tải đã và đang được dùng trên thế giới, các phương pháp cài đặt bộ cân bằng tải vào hệ thống để đạt được hiệu quả về khả năng đáp ứng người dùng cũng như khả năng mở rộng. Phần tiếp xin đưa ra cài đặt một số thuật toán cân bằng tải cụ thể dựa trên một số phần mềm cân bằng tải mã nguồn mở và một số cải tiến cho các thuật toán này. Cuối cùng là định hướng xây dựng một bộ cân bằng tải hoàn toàn độc lập thực thi các thuật toán phân tải cải tiến nhằm áp dụng cho các website đang hoạt động.

ABSTRACT OF THESIS

The boost of website services, especially the social network and online video-sharing websites, in recent years causes the number of clients using Internet rapidly increase. These websites are required to have a very powerful server system to match the demand of over hundred millions of clients. An interesting website certainly attracts a lot of users, then the clients go up, that makes servers overload and need to be improved. This process is repeated when there are more and more clients access to server again. This problem raise the question of how to build an website system which have ability to meet the accessing demand of people within long enough time and easily improved if there is necessary. That is scalable web architecture.

The most important factor in scalable web architecture is load balancing for web server system. A global website have web servers in many different places. Load balancing for the web server system is how to minimize clients response time without overloading. In the competition between web services, a manager who successfully match users' demands will become winner. Therefore, load balancing demand is very crucial for the survival of web service industry.

There many methods for load balancing but the general point is the necessity of load balancer between clients and web server system. This load balancer receives the requests from clients then directs them to suitable servers base on load distribution algorithms, next it gets the response and returns to clients.

We will present a general view on the problems which need to be solved in the process of webs ever designs with scalable ability. Then, we focus on the theory of load balancing for large server systems, techniques and methods of load balancer in use. We also mention the architecture and necessary factors for a load balancer, load distribution algorithms which have been used globally, methods of installing a load balancer into system for matching the demands as well as the scalable capability. The next part will introduce some specific load balancing algorithms base on an open source load balancer and improvements for these. The final part will provide a possible direction for building a completely independent load balancer which implements improved load distribution algorithms in order to apply in operating websites.

MỤC LỤC

TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP	1
ABSTRACT OF THESIS	2
DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ	6
ĐẶT VẤN ĐỀ.....	6
1. Tính cấp thiết của đề tài	7
2. Mục đích nghiên cứu	10
3. Kết cấu luận văn	11
CHƯƠNG 1: KIẾN TRÚC WEB VỚI KHẢ NĂNG MỞ RỘNG (SCALABLE WEB ARCHITECTURE).....	Error! Bookmark not defined.
1. Các khái niệm về kiến trúc với khả năng mở rộng	Error! Bookmark not defined.
2. Các vấn đề cần giải quyết trong quá trình xây dựng website theo kiến trúc với khả năng mở rộng	Error! Bookmark not defined.
2.1 <i>Cân bằng tải cho application servers</i>	Error! Bookmark not defined.
2.2 <i>Mở rộng Database server</i>	Error! Bookmark not defined.
2.2.1 Shared nothing Cluster.....	Error! Bookmark not defined.
2.2.2 Real Application Cluster.....	Error! Bookmark not defined.
2.2.3 Mô hình khuyên dùng	Error! Bookmark not defined.
2.3 <i>Tổ chức lưu trữ dữ liệu</i>	Error! Bookmark not defined.
2.4 <i>Cân bằng tải cho Cache</i>	Error! Bookmark not defined.
2.4.1 Định nghĩa	Error! Bookmark not defined.
2.4.2 Các loại Caches và cách cài đặt.....	Error! Bookmark not defined.
CHƯƠNG 2: KỸ THUẬT CÂN BẰNG TẢI WEB-SERVER.....	Error! Bookmark not defined.
1. Lý thuyết xây dựng bộ cân bằng tải cho web-servers	Error! Bookmark not defined.
1.1 <i>Kỹ thuật cân bằng tải server (Server Load Balancing – SLB)</i>	Error! Bookmark not defined.
1.1.1 Kiểm tra trạng thái server.....	Error! Bookmark not defined.
1.1.2 Lựa chọn server tốt nhất.....	Error! Bookmark not defined.
1.1.3 Kỹ thuật Session Persistence.....	Error! Bookmark not defined.
1.1.4 Cookie	Error! Bookmark not defined.

- 1.2 *Cân bằng tải cho server toàn cầu (GSLB)*. Error! Bookmark not defined.
 - 1.2.1 Domain Name System Error! Bookmark not defined.
 - 1.2.2 Cài đặt bộ cân bằng tải vào hệ thống mạng DNS...Error! Bookmark not defined.
 - 1.2.3 Lựa chọn site tốt nhất..... Error! Bookmark not defined.
- 1.3 *Chuyển mạch cache trong suốt*..... Error! Bookmark not defined.
 - 1.3.1 Các phương pháp cài đặt cache Error! Bookmark not defined.
 - 1.3.2 Các phương pháp cân bằng tải cho caches.....Error! Bookmark not defined.
 - 1.3.3 Nhận biết ngữ cảnh trong cache (Content-aware cache switching)Error! Bookmark not defined.
- 1.4 *Cân bằng tải sử dụng phần cứng và cân bằng tải phần mềm* Error! Bookmark not defined.
 - 1.4.1 Cân bằng tải sử dụng phần cứng Error! Bookmark not defined.
 - 1.4.2 Cân bằng tải sử dụng phần mềm Error! Bookmark not defined.
- 2. Các thuật toán cân bằng tải** Error! Bookmark not defined.
 - 2.1 *Thuật toán ngẫu nhiên (random)* Error! Bookmark not defined.
 - 2.2 *Thuật toán Round Robin (RR)* Error! Bookmark not defined.
 - 2.3 *Thuật toán Weighted Round Robin (Ratio)* Error! Bookmark not defined.
 - 2.4 *Thuật toán Dynamic Round Robin - DRR (Dynamic Ratio)* Error! Bookmark not defined.
 - 2.5 *Thuật toán Fastest*..... Error! Bookmark not defined.
 - 2.6 *Thuật toán Least Connections (LC)*..... Error! Bookmark not defined.
 - 2.7 *Thuật toán Observed*..... Error! Bookmark not defined.
 - 2.8 *Thuật toán Predictive*..... Error! Bookmark not defined.

CHƯƠNG 3: CÀI ĐẶT BỘ CÂN BẰNG TẢI TRONG MẠNG VÀ LẬP TRÌNH MÔ PHỎNG THUẬT TOÁN.....Error! Bookmark not defined.

- 1. Các phương án cài đặt bộ cân bằng tải vào hệ thống.**Error! Bookmark not defined.
 - 1.1 *Bộ cân bằng tải Haproxy*..... Error! Bookmark not defined.
 - 1.2 *Cài đặt đơn giản với phương pháp cookie-insert*.....Error! Bookmark not defined.
 - 1.3 *Cài đặt với khả năng mở rộng cao* Error! Bookmark not defined.

2. Cài đặt thuật toán cân bằng tải trên HAProxy	Error! Bookmark not defined.
2.1 <i>Thuật toán weighted round robin (WRR)</i> ...	Error! Bookmark not defined.
2.2 <i>Thuật toán least connections</i>	Error! Bookmark not defined.
2.3 <i>Một số cải tiến</i>	Error! Bookmark not defined.
3. Cấu hình và chạy chương trình.....	Error! Bookmark not defined.
KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN	Error! Bookmark not defined.
1. Tổng kết	Error! Bookmark not defined.
2. Định hướng phát triển	Error! Bookmark not defined.
TÀI LIỆU THAM KHẢO	Error! Bookmark not defined.

DANH MỤC CÁC HÌNH

STT	Tên hình	Trang
H 1.2-1	So sánh scale out và scale up	15
H 1.2-2	Bảng so sánh giá thành 2 phương pháp	15
H 1.2-3	Mô hình cân bằng tải web server	16
H 1.2-4	Mở rộng database server sử dụng kiến trúc SAN	17
H 1.2-5	Mở rộng database server theo chiều ngang	17
H 1.2-6	Real Application Cluster	19
H 1.2-7	Mô hình mở rộng database nên dùng	19
H 1.2-8	Ví dụ về mô hình CDN	21
H 1.2-9	Simple Cache	22
H 2.1-1	Cách làm việc của cookie user=1	27
H 2.1-2	Cookie read	27
H 2.1-3	Cookie-insert	28
H 2.1-4	Bộ cân bằng tải chèn một cookie	29
H 2.1-5	Bộ cân bằng tải ghi đè một cookie	29
H 2.1-6	Mô hình Global Server Load Balancing	31
H 2.1-7	Bộ cân bằng tải hoạt động như một authoritative DNS	32
H 2.1-8	Bộ cân bằng tải hoạt động như một forward DNS proxy	33
H 2.1-9	Sự kết hợp GSLB và SLB trong bộ cân bằng tải	34
H 2.1-10	Cài đặt cache ở trình duyệt của người dùng	38
H 2.1-11	Cài đặt cache như một transparent proxy	38
H 2.1-12	Cân bằng tải cho transparent-proxy cache	40
H 2.1-13	Cài đặt cache như một reverse-proxy	40
H 2.1-14	Bộ cân bằng tải cho Reverse proxy caches	41
H 2.1-15	Transparent-reverse proxy caches	42
H 2.1-16	Phương pháp hash buckets dùng trong caching	44
H 2.1-17	Ví dụ về luật giúp bỏ qua các ngữ cảnh động	46
H 2.1-18	Cân bằng tải sử dụng phần cứng	47
H 3.1-1	Mô hình cân bằng tải đơn giản với một bộ cân bằng tải	57
H 3.1-2	Luồng dữ liệu trong mô hình đơn giản	58
H 3.1-3	Mô hình phức tạp với 2 bộ cân bằng tải chia sẻ 1 địa chỉ IP ảo	59
H 3.1-4	Luồng dữ liệu trong mô hình 2 bộ cân bằng tải với keepalived	61
H 3.3-1	Cấu hình firewall cho CentOS	70
H 3.3-2	Cấu hình firewall cho CentOS (tiếp)	70
H 3.3-3	Kiểm tra thông số server trong HAProxy	74
H 3.3-4	Thông số các server của website www.tamtay.vn sử dụng HaProxy	75

Hình vẽ được đánh theo định dạng: H [Chương].[Mục lớn] [Số thứ tự]

DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ

Authoritative DNS	Authoritative Domain Name Server	Một website sẽ lưu thông tin về tên của nó trong nhiều server tên. Đây là server tên có thẩm quyền cao nhất
	Client	Khách hàng, người truy cập vào website nói chung
	Cluster	Dùng để chỉ một cụm máy chủ trong mạng LAN
DNS	Domain Name System	Hệ thống phân giải tên trong internet, thiết lập tương ứng giữa địa chỉ IP của một website và tên miền của nó.
	Forward-proxy	Proxy bên phía client, dùng để tăng tốc client, client phải chỉ đến proxy này để truy cập internet
	Gateway	Khả giống proxy, tuy nhiên chỉ có nhiệm vụ tạo kết nối giữa người dùng và server.
GSLB	Global Server Load Balancing	Cân bằng tải cho các máy server được cài đặt ở khắp nơi trên thế giới
HTTP	Hypertext Transfer Protocol	Giao thức truyền tải siêu văn bản, được sử dụng trong WWW
LB	Load balancing	Cân bằng tải: phân phối tải giữa các server nhằm đảm bảo sự cân bằng giữa chúng
LBer	Load Balancer	Bộ cân bằng tải: thiết bị phần cứng hoặc phần mềm dùng để thực thi hoạt động cân bằng tải
LC	Least connections	Thuật toán phân tải cho các server dựa trên số kết nối hiện tại của chúng
Local DNS	Local Domain Name Server	Server tên nằm trong mạng LAN của client, trình duyệt của client sử dụng local DNS để liên lạc với Domain Name System
	Proxy – Proxy Server	Server đứng giữa người dùng và webserver, có nhiệm vụ trao đổi thông tin.
	Reverse-proxy	Proxy bên phía server, tăng tốc server
RR	Round Robin	Thuật toán phân tải cho các server theo thứ tự xoay vòng
SAN	Storage Area Networks	Kiến trúc kết nối những thiết bị lưu trữ từ xa để xem chúng như là cục bộ với nhau
	Scalable - Scalability	Khả năng mở rộng: khả năng một website có thể đáp ứng được số lượng người dùng ngày một tăng
	Scale - Scaling	Mở rộng, tăng khả năng hoạt động
	Server	Máy phục vụ, chứa các thông tin, ứng dụng cho phép người dùng truy cập vào
SLB	Server Load Balancing	Cân bằng tải cho các máy server
	Socket	Sự kết hợp giữa địa chỉ IP và cổng nhằm xác định đích của một ứng dụng đang chạy trên server
SPOF	Single point of failure	Một node trong hệ thống mà nếu nó bị “chết” thì toàn bộ hệ thống sẽ bị tê liệt
TCP	Transmission Control Protocol	Giao thức điều khiển truyền vận, sử dụng tạo kết nối để trao đổi các gói tin
	Transparent-proxy	Proxy bên phía client, hoạt động “trong suốt” với người dùng
WWW	World Wide Web	Mạng máy tính toàn cầu

ĐẶT VẤN ĐỀ

1. Tính cấp thiết của đề tài

Sự phát triển của ngành dịch vụ web khiến cho số lượng người truy cập vào các website ngày càng tăng mạnh, đặc biệt là các mạng xã hội ảo hoặc website chia sẻ video trực tuyến. Các website này không chỉ cho phép người dùng trao đổi và chia sẻ thông tin, giao lưu và kết bạn mà còn cho phép người dùng lưu trữ tài liệu, tìm kiếm bạn bè, người thân, hay thậm chí cả đối tác kinh doanh. Mặc dù chỉ mới bắt đầu phát triển trong vòng 5 năm trở lại đây nhưng mạng xã hội và các trang chia sẻ video trực tuyến đã có số lượng người dùng lên đến con số hàng trăm triệu và vẫn đang tăng với tốc độ chóng mặt.

Với **mạng xã hội**, người dùng có thể chat, email, voice chat, chia sẻ file, viết nhật ký cá nhân (blog) và cùng nhau bàn luận về những vấn đề mà họ cùng quan tâm. Thêm nữa, người dùng có thể thông qua mạng tìm kiếm bạn bè hay đối tác theo tên (tên sử dụng hoặc email), sở thích cá nhân (như thể thao, phim ảnh, ca nhạc) hoặc lĩnh vực quan tâm (công nghệ, điện ảnh, kinh doanh...). Chính những tính năng đó làm cho người dùng tìm đến các mạng xã hội ngày càng nhiều hơn và thời gian hàng ngày họ dành cho nó cũng tăng lên. Có thể kể ra hàng loạt các mạng xã hội đang có số lượng người dùng lớn như MySpace, Facebook, Hi5, Friendster, LinkedIn... Theo số liệu cập nhật tháng 1/2009, myspace có 255 triệu tài khoản đăng ký và đứng thứ 9 trong bảng xếp hạng các website hàng đầu thế giới theo Alexa, Facebook có 177 triệu tài khoản và đứng thứ 4. Các website mạng xã hội khác như Hi5, Friendster... đều có số lượng tài khoản đã đăng ký trên 50 triệu.

Với các **website chia sẻ video trực tuyến** cũng vậy, sự phát triển của công nghệ kỹ thuật số giúp cho mọi người có thể dễ dàng tạo ra các đoạn clip vui nhộn, hay các video gia đình, họ muốn lưu trữ, muốn gửi cho bạn bè khắp nơi, vì vậy họ tìm đến các website chia sẻ video trực tuyến. www.youtube.com, website chia sẻ video lớn nhất hiện nay, có trên 6 tỉ video và 100 triệu video được xem mỗi ngày. Youtube đứng thứ 3 trong bảng xếp hạng thông lượng website của Alexa.

Sự cạnh tranh giữa các website mạng xã hội và các trang chia sẻ video trực tuyến là không thể tránh khỏi, bất cứ một nhà quản trị nào cũng muốn tăng số lượng người dùng cũng như muốn website của mình ngày càng nổi tiếng. Để làm được điều đó, ngoài nội dung hấp dẫn và phong phú, website cần phải đáp ứng được nhu cầu của người dùng một cách tốt nhất. Với 100 triệu video được xem mỗi ngày, tính trung bình mỗi giây có gần 1200 video được xem trên Youtube, con số này đòi hỏi Youtube phải có một hệ thống server mạnh mẽ và linh hoạt. Một website với số lượng truy cập ngày càng tăng sẽ khiến máy chủ của website đó không đáp ứng được hết nhu cầu của người dùng và bị quá tải.

Yêu cầu đặt ra đối với người quản trị website là cần phải giải quyết vấn đề quá tải này, làm sao để bất cứ người dùng nào truy cập vào website đều được phục vụ một cách nhanh nhất. Để làm được điều đó, người quản trị phải nâng cấp hệ thống server. Nghĩa là, cần phải tăng cấu hình máy server hoặc tăng số lượng server, đó chính là một trong hai phương án mở rộng server: **mở rộng theo chiều dọc (scaling up)** và **mở rộng theo chiều ngang (scaling out)**. Mở rộng theo chiều dọc nghĩa là nâng cấp một server có cấu hình ngày càng mạnh hơn tùy theo nhu cầu của người dùng, phương pháp này khá tốt trong trường hợp số lượng người dùng không nhiều tuy nhiên giá thành của nó đắt và khả năng mở rộng kém. Mở rộng theo chiều ngang nghĩa là sử dụng một hệ thống nhiều server nhỏ với chi phí thấp hơn và mỗi server phục vụ một lượng người dùng nhất định, phương pháp này có khả năng mở rộng tốt (lắp thêm server khi cần) và hoạt động khá hiệu quả. Với những trang có số lượng người truy cập nhiều và tỉ lệ truy cập cao, các nhà quản trị hệ thống thường lựa chọn phương án thứ hai – mở rộng theo chiều ngang.

Nâng cấp server cũng giống như là một hoạt động có chu kì trong vòng đời của hệ thống. Khi hệ thống server không đáp ứng được nhu cầu của người dùng, nó sẽ được nâng cấp. Sau khi nâng cấp, tốc độ cải thiện sẽ khiến người tin tưởng vào khả năng phục vụ của website, số lượng người dùng lại tăng lên, hệ thống server lại không đáp ứng được nhu cầu của người dùng và lại cần nâng cấp... Vòng luân chuyển đó khiến các nhà quản trị mong muốn tìm kiếm một giải pháp nào đó, mà website của họ có thể hoạt động ổn định trong một thời gian dài, và dễ dàng nâng cấp khi cần thiết. Đó chính là phương án mà hầu hết các mạng xã hội và các trang chia sẻ video trực tuyến như myspace, google video... đã chọn - **sử dụng kiến trúc web với khả năng mở rộng (scalable web architecture)**. Cách tiếp cận này cho phép nhà quản trị tăng hiệu năng của hệ thống, cho khả năng mở rộng và khả năng có sẵn cao, đáp ứng được yêu cầu trong suốt đời với người dùng (user-transparent), tuy nhiên có rất nhiều thách thức cần phải vượt qua để xây dựng được các chức năng của một hệ thống website trên nền trình duyệt web và HTTP.

Để xây dựng được một website theo kiến trúc này, nhà quản trị website phải thiết kế một cách “nhịp nhàng” các thành phần của hệ thống, các thành phần chính bao gồm xây dựng hệ thống web-server, xây dựng hệ thống database-server, quản lý session, thiết kế bộ cache cho hầu hết các thành phần của hệ thống. Một website xây dựng theo kiến trúc này khi muốn nâng cấp mở rộng sẽ theo chiều hướng scaling out, nghĩa là hệ thống sẽ có nhiều web-server và nhiều database-server kết nối với nhau, khi cần mở rộng sẽ lắp đặt thêm các server mới.

Khi hệ thống có nhiều server thì vấn đề đặt ra là khi một người dùng truy cập vào website, yêu cầu của người dùng đó sẽ được chuyển đến server nào xử lý, lấy dữ liệu ở database-server nào? Làm thế nào để các server trong hệ thống nhận được lượng tải phù hợp với server đó, đảm bảo cho không server nào bị quá tải? Làm sao

để hệ thống có thể đáp ứng được số lượng người dùng cao nhất? Đó chính là bài toán **cân bằng tải (Load Balancing – LB)** cho các server, cũng chính là phần quan trọng nhất trong việc giữ cho hệ thống hoạt động ổn định, là nhân tố quyết định đến khả năng đáp ứng người dùng của hệ thống. Thuật ngữ cân bằng tải không những chỉ dùng cho server, mà còn dùng trong hầu hết các thành phần còn lại của kiến trúc web với khả năng mở rộng, đó là cân bằng tải cho database, cân bằng tải cho cache, cân bằng tải cho firewall... Một website được cân bằng tải tốt sẽ tận dụng được cao nhất nguồn lực sẵn có. Chẳng hạn như website với 1 web-server có thể đáp ứng được cho 1000 người dùng cùng lúc, khi nâng cấp lên 4 servers, nếu cân bằng tải tốt sẽ có khả năng đáp ứng cho 4000 người cùng lúc, ngược lại sẽ chỉ có thể đáp ứng được 2000 người, thậm chí còn ít hơn số đó. Một trang web được xử lý tốt vấn đề cân bằng tải sẽ đáp ứng được người dùng một cách tốt nhất và có khả năng mở rộng cao.

Để cân bằng tải cho cụm server, cần phải xây dựng các **bộ cân bằng tải (Load Balancer)**. Các bộ cân bằng tải này được cài đặt ở giữa người dùng và hệ thống server, nhận yêu cầu từ phía người dùng, sau đó chuyển chúng đến web-server phù hợp. Bộ cân bằng tải lúc đó đóng vai trò là web-server duy nhất đối với tất cả mọi người dùng, chỉ cần truy cập đến địa chỉ chứa bộ cân bằng tải, người dùng sẽ nhận được thông tin mình cần, không cần biết phía sau đó hệ thống có bao nhiêu server, cũng không cần nhớ địa chỉ của các server này. Vậy **làm cách nào để xây dựng được một bộ cân bằng tải hoạt động hiệu quả**, nhằm đảm bảo tải được phân phối một cách cân đối giữa các web-server?

Một bộ cân bằng tải cần đáp ứng được hai yêu cầu chính. Thứ nhất, cũng là quan trọng nhất, đó là khả năng xử lý và chuyển yêu cầu của người dùng đến server phù hợp, để giải quyết vấn đề này, cần phải xây dựng một thuật toán phân tải hoạt động “mềm dẻo” trên nhiều yếu tố khác nhau. Thứ hai, nó cần phải kiểm tra và cập nhật được các thông số kỹ thuật cũng như trạng thái của các server, kiểm tra xem server nào đang hoạt động, server nào đang tạm nghỉ, server nào vừa được thêm vào cụm server, hay vừa được đưa ra (để sửa chữa, bảo trì...) trong hệ thống, kiểm tra sự thay đổi trong khả năng phục vụ (chẳng hạn số kết nối lớn nhất) hay trọng số của các server. Giải quyết tốt hai bài toán trên sẽ cho ra một bộ cân bằng tải hoạt động hiệu quả và ổn định.

Các bộ cân bằng tải thường được chia ra làm hai loại: loại hoạt động dựa trên phần cứng và hoạt động dựa trên phần mềm. Hoạt động dựa trên phần cứng nghĩa là nó sử dụng một thiết bị phần cứng hoạt động ở tầng mạng (network layer) và tầng giao vận (transport layer) của mô hình OSI, thiết bị này được thiết kế trên mô hình chuyển mạch và nó phân bố tải rất hiệu quả. Tuy nhiên giá của những thiết bị phân tải này thường rất đắt, vượt ngoài khả năng của những nhà phát triển hệ thống vừa và nhỏ. Bộ cân bằng tải phần mềm là một phần mềm được cài đặt trên một máy

riêng biệt hoặc trên máy server, lưu giữ địa chỉ của các máy server trong hệ thống để phân tải. Bộ cân bằng tải phần mềm có khả năng phân tải kém hơn các thiết bị phần cứng, tuy nhiên giá thành của nó rẻ hơn và phù hợp với các nhà phát triển hệ thống vừa và nhỏ, thậm chí còn có rất nhiều phần mềm mã nguồn mở hoạt động rất tốt. Một phần mềm hoạt động hiệu quả cũng có khả năng phân tải tương đương với các phần cứng có giá thành trung bình. Hơn nữa, dù có sử dụng phần cứng, các website vẫn cần phần mềm để hỗ trợ phân tải.

Điểm mấu chốt trong hoạt động của các bộ cân bằng tải là thuật toán phân phối tải. Không có một thuật toán nào thực sự phù hợp cho mọi môi trường. Với những điều kiện hệ thống nhất định, yêu cầu đặt ra là phải chọn thuật toán nào cho phù hợp. Các thuật toán phân tải đang được sử dụng thường được chia ra làm hai loại chính, đó là thuật toán phân tải tĩnh và thuật toán phân tải động. Các thuật toán phân tải tĩnh dễ thiết kế, hoạt động đơn giản nhưng hiệu suất không cao. Chúng được gọi là thuật toán tĩnh vì phương pháp chọn server để gửi yêu cầu tiếp theo được dựa trên các yếu tố tĩnh, không thay đổi. Ngược lại, các thuật toán phân tải động dựa trên các nhân tố động, thay đổi liên tục, do đó các thuật toán động này hoạt động linh hoạt hơn và đáp ứng yêu cầu phân tải một cách tốt hơn. Tuy vậy không có một thuật toán nào có thể đáp ứng được trong mọi trường hợp. Cần phải có một sự kết hợp mềm dẻo dựa các thuật toán này mới có thể xây dựng được một bộ cân bằng tải hoạt động tốt. Việc nghiên cứu các thuật toán phân tải mềm dẻo sẽ giúp cho các bộ cân bằng tải hoạt động hiệu quả hơn và đáp ứng được nhu cầu phân tải cao hơn.

Với sự bùng nổ của dịch vụ web, số lượng website ngày càng nhiều đem đến cho người dùng rất nhiều sự lựa chọn. Để tồn tại và phát triển, các website phải giữ được người dùng hiện tại của mình và lôi kéo thêm được nhiều người dùng mới. Để làm được điều này, các nhà quản trị hệ thống phải đảm bảo cho hệ thống của mình hoạt động tốt nhất, do đó yêu cầu lắp đặt thêm nhiều server mới cũng như yêu cầu cân bằng tải sẽ cho các server thực sự là nhu cầu cấp thiết. Đứng trước sự cạnh tranh về giá thành, việc phát triển và xây dựng các phần mềm cân bằng tải trở nên vô cùng hữu ích. Đứng trước yêu cầu đó, NVLV quyết định tìm hiểu về nguyên lý hoạt động cũng như các thuật toán cân bằng tải nhằm xây dựng nên một nền tảng vững chắc về chủ đề này, từ đó góp phần thúc đẩy quá trình xây dựng nên một bộ cân bằng tải hoạt động hiệu quả có khả năng đáp ứng nhu cầu cho các website đang hoạt động trong và ngoài nước.

2. Mục đích nghiên cứu

- Nghiên cứu tổng quan về kiến trúc web với khả năng mở rộng, tìm hiểu lý thuyết chung về cân bằng tải server, cân bằng tải database, cân bằng tải cho cache và quản lý session.

- Tìm hiểu các kỹ thuật cân bằng tải cho web server, lý thuyết xây dựng bộ cân bằng tải cho web server và điểm mạnh yếu của các thuật toán phân tải cho bộ cân bằng tải.
- Xây dựng và cài đặt thử nghiệm một số thuật toán cân bằng tải cơ bản, thực thi một số thuật toán cân bằng tải tĩnh và cân bằng tải động.
- Tìm hiểu cách cấu hình các bộ cân bằng tải cùng với web server theo các điều kiện khác nhau, nhằm đáp ứng được nhu cầu từng hệ thống web riêng biệt.

3. Kết cấu luận văn

Trong khuôn khổ báo cáo này, NVLV xin trình bày kết quả tìm hiểu được cũng như cài đặt thử nghiệm một số thuật toán cân bằng tải nhất định. Do nguồn lực và thời gian có hạn, nên NVLV chưa thực sự xây dựng được một phần mềm cân bằng tải hoạt động thực sự mà chỉ có thể cài đặt thử nghiệm một số thuật toán cân bằng tải dựa trên một bộ cân bằng tải có sẵn. Để giúp tiếp cận vấn đề một cách nhanh nhất, báo cáo này được trình bày với nhiều hình ảnh sinh động. Nội dung của báo cáo chia làm 4 chương như sau:

Chương 1: Tổng quan về kiến trúc web với khả năng mở rộng.

Chương 2: Kỹ thuật cân bằng tải ở web-server, lý thuyết xây dựng bộ cân bằng tải cho webserver và các thuật toán phân tải cho bộ cân bằng tải.

Chương 3: Trình bày về môi trường phát triển, các phương án cài đặt bộ cân bằng tải vào mạng, kết quả cài đặt thuật toán, một số cải tiến cho các thuật toán cân bằng tải. Cách cài đặt và cấu hình bộ cân bằng tải cho một hệ thống webserver cụ thể

Chương 4: Kết luận và định hướng phát triển.