

# ĐÁNH GIÁ HIỆU NĂNG CỦA HỆ THỐNG THÔNG TIN Đám Mây KHI TÍCH HỢP CÁC THUẬT TOÁN CÂN BẰNG TẢI

Nguyễn Tùng Lâm\*, Lê Thị Thu Hương\*

## ABSTRACT

*In the era of technology 4.0 today, enterprises, agencies or large organizations tend to build a complete cloud information system, integrating hardware infrastructure with software systems to minimize costs, build a long-term operation strategy. In order to build a perfect information system, we need to integrate the necessary systems, using optimal algorithms for the system.*

**Keywords:** Performance evaluation, cloud information system, load balancing algorithm

**Received:** 24/2/2022; **Accepted:** 28/22/2022; **Published:** 4/3/2022

## 1. Thiết kế và cài đặt hệ thống

### 1.1. Thiết kế hệ thống

Với hệ thống phần cứng ở đây ta sẽ mô phỏng một hệ thống phần cứng bằng việc sử dụng Openstack. Với Openstack là một hệ thống phân tán gồm nhiều service phối hợp với nhau để quản lý và phân phối tài nguyên đám mây.

Cấu hình thử nghiệm của máy tính cài đặt openstack:

- VCPUs: 4
- RAM: 12 Gb
- HDD: 200 GB

Như chúng ta đã biết, khi dịch vụ của người dùng có tình trạng đột biến về tải thì chất lượng dịch vụ sẽ bị giảm do tài nguyên được sử dụng cạn dần. Khi đó, auto-scaling sẽ thực hiện chức năng mở rộng (scale-up) để tăng lượng tài nguyên sao cho chất lượng dịch vụ không thay đổi và khi tải giảm thì hệ thống sẽ thực hiện chức năng thu lại (scale-down) nhằm giảm đi năng lượng tiêu thụ hệ thống và bớt đi chi phí cho người sử dụng dịch vụ. Nhưng tăng tải nguyên cũng như thu hồi tài nguyên làm sao cho kịp thời, đúng và hợp lý là một trong những mục tiêu của các kỹ thuật tự động mở rộng.

Một hệ thống sẽ có nhiều yếu tố tài nguyên (như CPU, bộ nhớ, mạng,...). Nó cần một hệ thống theo dõi, giám sát liên quan đến tài nguyên tiêu thụ trung bình cho tất cả các yếu tố xem xét.

Từ vấn đề mở rộng và hệ thống sẽ có thêm những máy ảo mới, việc cần tính đến là việc cân bằng tải sao cho hợp lý với khả năng của hệ thống.

Do đó một hệ thống sẽ gồm các thành phần giám sát, cân bằng tải để phục vụ cho việc scaling. Việc đánh giá hệ thống sẽ là việc đánh giá 3 thành phần giám sát, cân bằng tải và tự động mở rộng kết hợp, hoạt động ra sao trong hệ thống.

Với những yêu cầu như vậy ta xây dựng một hệ thống: Sử dụng hệ thống tự động mở rộng (Auto-scaling), Sử dụng khả năng mở rộng là theo chiều ngang, Có thành phần giám sát, quản lý tài nguyên (monitoring), Có thành phần cân bằng tải cho hệ thống (load blancing)

### 1.2. Xây dựng hệ thống thực nghiệm

Server từ openstack chạy hệ điều hành là Ubuntu server 14.04, cài đặt các thành phần LEMP trên server, ở đây ta không sử dụng Apache server mà cài đặt Nginx làm webserver

Xây dựng website du lịch để cài đặt thử nghiệm. Web được xây dựng với ngôn ngữ PHP, sử dụng data là Mysql server.

Web có các chức năng đặt các tour du lịch, đăng ký, đăng nhập với các mức quyền của từng loại thành viên.

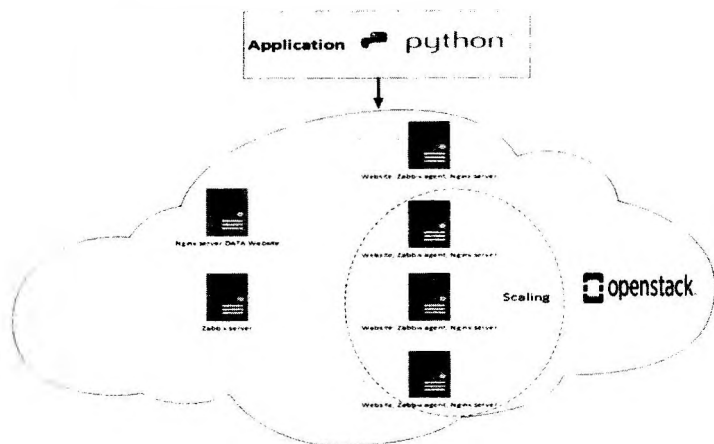
Mô hình của hệ thống

- Sử dụng hệ thống tự động mở rộng (Auto-scaling)
- Giám sát, quản lý tài nguyên (monitoring)

\* Trung tâm Tin học, Trường Đại học Nội vụ Hà Nội

sử dụng Zabbix

- Cân bằng tải cho hệ thống (load blancing) sử dụng Nginx



Hình 1.1. Mô hình hệ thống thực nghiệm khi mở rộng

## 2. Các thuật toán của cân bằng tải

- Roud robin: Là thuật toán luân chuyển vòng, các máy chủ sẽ được xem ngang hàng và sắp xếp theo một vòng quay. Các truy vấn dịch vụ sẽ lần lượt được gửi tới các máy chủ theo thứ tự sắp xếp

- Least conn: đơn giản là truy cập đến tại thời điểm đó sẽ được đẩy tới server tại thời điểm đó đang nhận ít truy cập đến nhất.

Với một hệ thống có các server gần giống nhau về cấu hình, LC có khả năng hoạt động tốt ngay cả khi tải của các kết nối biến thiên trong một khoảng lớn. Do đó nếu sử dụng LC sẽ khắc phục được nhược điểm của Roud robin

- Ip hash: là thuật toán dựa trên ip truy cập đến để xây dựng bảng băm cân bằng tải, nói dễ hiểu là thuật toán sẽ nhóm các cụm IP vào các server dựa trên thông số là IP truy cập đến nó.

Với ip-hash, địa chỉ IP của client được sử dụng như một chìa khóa để xác định những máy chủ trong một nhóm máy chủ nên được lựa chọn. Phương pháp này đảm bảo rằng các yêu cầu từ client sẽ luôn được hướng đến cùng một máy chủ trừ khi máy chủ này là không còn tồn tại.

## 3. Kiểm thử và đánh giá hiệu năng của hệ thống khi tích hợp các thuật toán cân bằng

tải

### 3.1. Kiểm thử hệ thống thực nghiệm

Để đánh giá khả năng kết hợp của kỹ thuật tự động mở rộng với các thuật toán cân bằng tải thì ta cần đánh giá hiệu năng sử dụng của website du lịch trong hệ thống. Để đánh giá hiệu năng Website ta sử dụng Apache Jmeter.

Apache JMeter là một phần mềm nguồn mở được viết bằng Java nhằm mục đích kiểm thử chức năng và hiệu suất. Mục đích ban đầu JMeter được thiết kế chỉ để kiểm thử các ứng dụng web nhưng hiện nay nó đã được mở rộng thêm nhiều chức năng khác.

Sử dụng chức năng Load testing trong JMeter Performance Testing

Load testing: Mô hình hóa dự kiến sử dụng bởi nhiều người dùng truy cập một dịch vụ website trong cùng thời điểm.

Bằng việc sử dụng jmeter ta sẽ kiểm tra hiệu năng của website. Đầu tiên ta sẽ kiểm tra hiệu năng của trang chủ website khi kết hợp kỹ thuật tự động mở rộng theo ngưỡng với từng giải thuật cân bằng tải. Tiếp theo ta sẽ xây dựng một kịch bản trên jmeter để kiểm tra khả năng xử lý của hệ thống với trang login.

#### • Kiểm thử hiệu năng trang chủ

Với việc kiểm tra hiệu năng của trang chủ ta sẽ đánh giá qua việc sử dụng: Lần lượt giả lập 250, 500 và 1000 lượt truy cập vào trang chủ; Sử dụng kỹ thuật tự động mở rộng là theo ngưỡng tài nguyên và theo thời gian; Sử dụng khả năng mở rộng theo chiều dọc; Và khả năng mở rộng theo chiều ngang gồm các kỹ thuật mở rộng theo ngưỡng và theo thời gian kết các thuật toán cân bằng tải Roud robin, least conn và Ip hash.

Kỹ thuật mở rộng theo ngưỡng thì hệ thống thực nghiệm có khả năng xử lý thấp do vậy ngưỡng đặt ra chỉ là 5% (với các hệ thống thực tế thì ngưỡng đặt ra với mỗi máy chủ ảo sẽ dao động từ 50-70%).

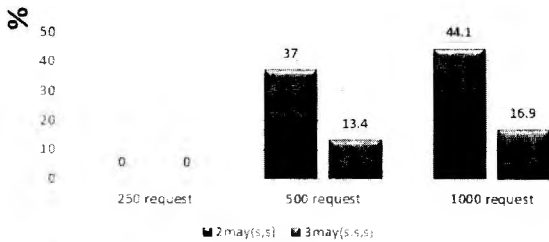
Kỹ thuật mở rộng theo thời gian thì thời gian để mở rộng là 9 giờ.

Ta sẽ đánh giá hệ thống qua thông số kiểm

thứ là: **Error %**: là phần trăm bị lỗi của các truy cập (lỗi kết nối hoặc lỗi cho đầu ra không mong muốn)

• **Sử dụng thuật toán cân bằng tải Roud robin**

Số phần trăm lỗi truy cập khi kết hợp ngưỡng với Roud robin



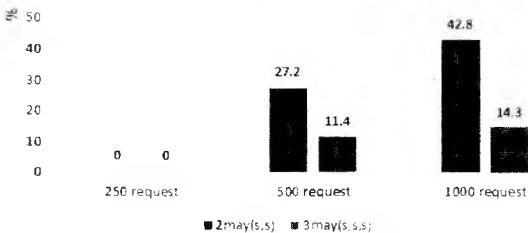
Biểu đồ 3.1: Biểu đồ số phần trăm lỗi truy cập khi kết hợp ngưỡng với Roud robin

Kiểm thử hiệu năng hệ thống bằng mở rộng theo chiều ngang, sử dụng Jmeter để kiểm thử hiệu năng của trang chủ khi kết hợp kỹ thuật tự động mở rộng theo ngưỡng được đặt ra là thời gian CPU hoạt động không được vượt quá 5% thời gian hoạt động với giải thuật cân bằng tải: Roud robin. Số lượng máy chủ ảo lần lượt là 2 máy và 3 máy có kích thước nhỏ.

• **Sử dụng thuật toán cân bằng tải Least conn**

Kết hợp kỹ thuật tự động mở rộng theo ngưỡng CPU không được vượt quá 5% thời gian hoạt động với giải thuật cân bằng tải Least conn. Sử dụng lần lượt 2 và 3 máy chủ ảo để kiểm thử hệ thống.

Số phần trăm lỗi truy cập khi kết hợp ngưỡng với least conn

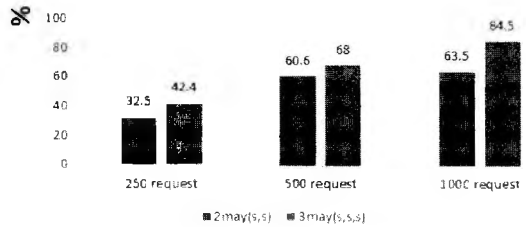


Biểu đồ 3.2: Biểu đồ số phần trăm lỗi truy cập khi kết hợp ngưỡng với Least conn

• **Sử dụng thuật toán cân bằng tải Ip hash**  
 Kết hợp kỹ thuật tự động mở rộng theo

ngưỡng CPU không được vượt quá 0,36% thời gian hoạt động với giải thuật cân bằng tải Ip hash

Biểu đồ số phần trăm lỗi truy cập khi kết hợp ngưỡng với Ip hash



Biểu đồ 3.3: Biểu đồ thời gian xử lý một truy cập thành công

3.2. **Đánh giá hiệu năng của hệ thống khi tích hợp các thuật toán cân bằng tải**

Qua những thực nghiệm kiểm thử trên hệ thống đám mây ta xây dựng thì có thể đánh giá như sau:

Số phần trăm lỗi truy cập của kỹ thuật tự động mở rộng theo ngưỡng với giải thuật Least conn là nhất với mọi số truy cập kiểm thử, hơn số phần trăm lỗi của ngưỡng với Roud robin không nhiều, hơn rất nhiều số phần trăm lỗi của ngưỡng với ip hash.

4. **Kết luận**

Kết hợp phương pháp ngưỡng với giải thuật Least conn là hiệu quả nhất so với việc kết hợp phương pháp theo ngưỡng với giải thuật Roud robin và Ip hash khi ta sử dụng kịch bản Jmeter đơn giản để kiểm thử.

**Tài liệu tham khảo**

[1] Kim Ngân, Thời điểm nào trong ngày người ta dùng facebook nhiều nhất ?, <http://giaoduc.net.vn/Ban-doc/Thoi-diem-nao-trong-ngay-nguoi-ta-dung-facebook-nhieu-nhat-post98885.gd>, 2012

[2] Tania Lorigo-Bohr'an, Jos'e Miguel-Alonso, Jos'e A. Lozano, Auto-scaling Techniques for Elastic Applications in Cloud Environments, 5, 2012, pp. 5-16

[3] Marc Cluet, Autoscaling Best Practices FOSDEM'14 Marc Cluet DevOps & Automation Advisory, pp. 1-45