SWG – HỆ THỐNG CLUSTER CHO CÁC ỨNG DỤNG WEB VIỆT CÓ NGỮ NGHĨA

SWG – A CLUSTERED SYSTEM FOR VIETNAMESE SEMANTIC WEB APPLICATIONS

Nguyễn Quang Hùng, Nguyễn Thanh Sơn và Thoại Nam

Khoa Công Nghệ Thông Tin, trường Đại học Bách Khoa TP.HCM, Việt Nam
hungnq2@dit.hcmut.edu.vn, sonsys@hcmut.edu.vn, nam@dit.hcmut.edu.vn

BẢN TÓM TẮT

Sự phát triển nhanh chóng của các công nghệ mạng tốc độ cao, công nghệ chế tạo bộ vi xử lý và các phiên bản hệ điều hành Unix/Linux khác nhau, tất cả đã làm cho các hệ thống cluster trở thành nền tảng chủ đạo đối với các hệ thống song song và phân bố. Hệ thống cluster mang lại khả năng tính toán hiệu năng cao (high performance), và nó rất hữu dụng để chạy các ứng dụng khoa học lớn. Trong đó, ứng dụng Web có ngữ nghĩa tiếng Việt cũng là một dạng ứng dụng đòi hỏi sự tính toán lớn. Bởi vì ứng dụng Web có nghĩa này cần lưu trữ và truy xuất hàng trăm ngàn thực thể có tên bởi sự phối hợp xử lý của nhiều quá trình phức tạp. Bài báo này đề nghị một giải pháp – một hệ thống tính toán hiệu năng cao (tên SWG) được xây dựng nhằm mục đích chạy các quá trình phức tạp gồm: (1) lưu trữ và truy vấn cơ sở tri thức tiếng Việt; (2) chú giải các trang web tiếng Việt; và (3) truy xuất các tài liệu đã được lập chỉ mục.

ABSTRACT

Enabling technologies in high-speed communication, CPU-production, Unix/Linux operating systems today have made cluster systems become mainstream of parallel and distributed platforms for high-performance, high throughput and high-availability computing. The Unix/Linux cluster systems, were connected by thousand of similar workstations, are popular used to run large e-science applications. Such a kind of large applications is the kind of Vietnamese semantic web applications, in which hundred thousands of entities in a knowledgebase are accessed and many coordinated complex processes will be done. In this paper, a high performance computing (HPC) clustered system for semantic web application– named SWG, used to run Vietnamese semantic web applications are proposed. The SWG system is used to (1) store and query Vietnamese knowledgebase by Sesame API, (2) annotate Vietnamese web pages, and (3) access indexed documents by Lucene.

1. GIỚI THIỆU

Các hệ thống Unix/Linux cluster là sự ghép nối của nhiều máy tính trạm và máy chủ lại với nhau thông qua mạng truyền tốc độ cao (hàng Gbits/sec) [3]. Trong đó mỗi nút (máy tính trạm/máy chủ) chạy hệ điều hành Unix/Linux, và các dịch vụ cần thiết khác như Network File System (NFS)… Hiện nay, các hệ thống cluster dùng Unix/Linux này mang lại hiệu quả đầu tư cao hơn so với các máy tính cỡ lớn (mainframe).

Ứng dụng web có ngữ nghĩa tiếng Việt [4] được xác định là một dạng ứng dụng có dữ liệu xử lý lớn và bao hàm nhiều quá trình phức tạp. Dữ liệu (của ứng dụng Web có ngữ nghĩa) là số lượng các thực thể có tên như là con người, sông, địa danh, công ty… tồn tại trên nước Việt Nam là rất lớn (hàng trăm ngàn thực thể có tên như vậy). Tất cả các thực thể có tên này được biểu diễn dưới dạng Resource Description Framework (RDF) của tổ chức W3C và được lưu trử/truy vấn trên hệ lưu trữ như là Sesame

(www.openrdf.org). Sesame cho phép dùng Sesame Application Programming Interface (API) để lưu trữ và xử lý các truy vấn dạng SeSQL trên các đồ thị RDF. Bởi vì số lượng các thực thể có tên nhiều và số lượng client sẽ gởi các câu truy vấn đến Sesame là rất nhiều (tương lai có thể phục vụ người dùng trên Internet). Do đó dẫn đến sự ra đời của hệ thống cluster (tên SWG), nhằm để đảm bảo các ứng dụng Web có ngữ nghĩa phục vụ số lượng hàng ngàn người dùng đồng thời, tạo ra kết quả trong thời gian trung bình thấp nhất, và tính sẵn sàng cao.

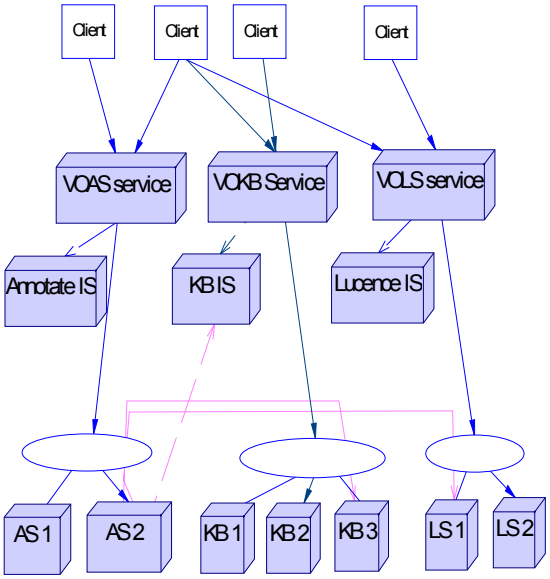
Trên hệ thống SWG, các nút tính toán là các máy chủ mạnh, phân ra làm nhiều nhóm, mỗi nhóm sẽ thực thi một loại quá trình chuyên biệt như là các quá trình xử lý câu truy vấn của Sesame, hoặc là các quá trình chú giải các trang web tiếng Việt, hoặc là các quá trình xử lý yêu cầu truy xuất các tài liệu đã được đánh chỉ mục bởi Lucene… Hơn nữa, kiến trúc hệ thống SWG có sự hướng đến mô hình dịch vụ của Grid computing [1, 2].

Để dễ dàng tiếp cận, bài báo được bố cục như sau: phần kế tiếp bài báo đề cập đến kiến trúc phần mềm của hệ thống SWG và hai quy trình xử lý yêu cầu chú giải và yêu cầu truy xuất cơ sở tri thức của client bên ngoài hệ thống SWG. Phần 3 bài báo đề cập đến các thử nghiệm đang tiến hành và phân tích số liệu, và phần cuối cùng bài báo đi đến kết luận và hướng phát triển.

2. KIẾN TRÚC PHẦN MỀM CỦA HỆ THỐNG SWG

2.1. Kiến trúc hệ thống SWG

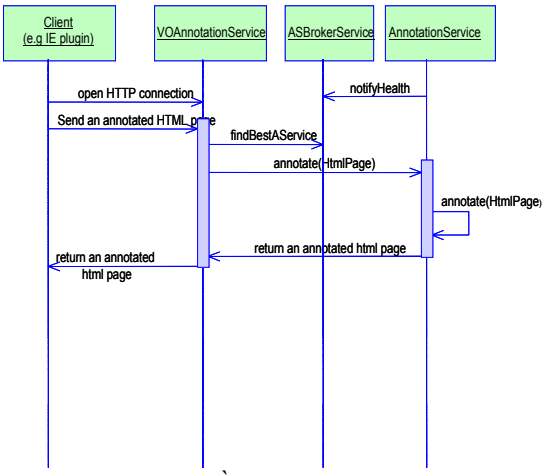
Phần này đề cập đến kiến trúc phần mềm của hệ thống SWG. Hình 2.1 bên dưới mô tả tổng quan kiến trúc của hệ thống SWG.



Hình 2.1 – Kiến trúc của hệ thống SWG

2.2. Quy trình xử lý yêu cầu chú giải của client bên ngoài hệ thống SWG

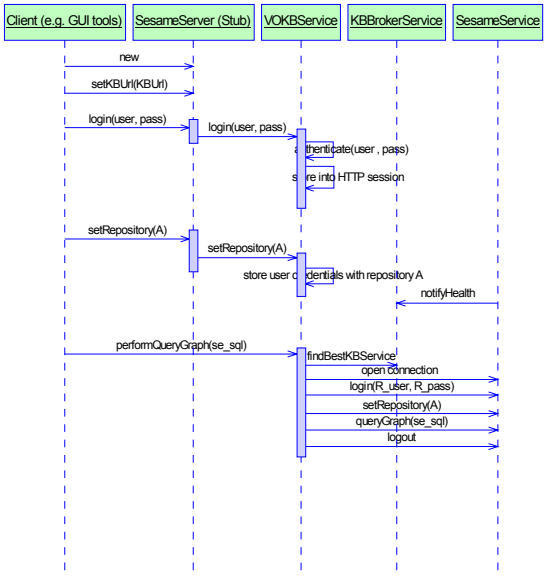
Một client bên ngoài hệ thống là những ứng dụng (không phải là các quá trình xử lý yêu cầu chú giải, truy vấn cơ sở tri thức, Lucene... ở phía server) như là các plugin vào các trình duyệt. Các client này sử dụng các hàm API để kết nối và gởi yêu cầu chú giải một trang web tiếng Việt đến dịch vụ VOAnnotationService. Hình 2.2 bên dưới mô tả lược đồ trình tự cho quy trình xử lý yêu cầu chú giải.



Hình 2.2 – Lược đồ trình tự mô tả quy trình xử lý yêu cầu chú giải của client

2.3. Quy trình xử lý yêu cầu truy xuất cơ sở tri thức của client bên ngoài hệ thống SWG

Hình 2.3 bên dưới là lược đồ trình tự cho quy trình xử lý một yêu cầu truy xuất cơ sở tri thức của client. Để truy vấn được cơ sở tri thức, các client phải dùng thư viện Sesame API.



Hình 2.3 – Lược đồ trình tự mô tả quy trình xử lý yêu cầu truy xuất cơ sở tri thức của client

3. THỬ NGHIỆM

3.1. Thử nghiệm 1: Truy vấn bằng SeSQL trực tiếp đến Sesame server

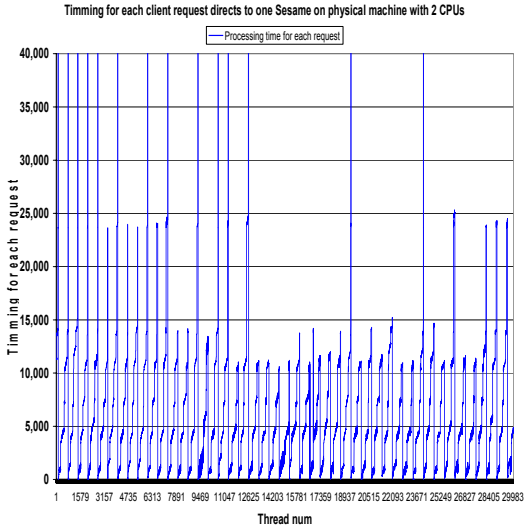
Để thử nghiệm, một chương trình (kỹ thuật Java Multithread) giả lập nhiều client đồng thời truy vấn tới cùng một Sesame server. Chương trình java multithread này trong quá trình chạy sẽ lần lượt tạo ra nhiều thread, mỗi thread đại diện cho một Sesame client dùng Sesame API để truy gởi câu truy vấn dạng SeSQL ("select * from {x} rdf:type {vnkimo_rdfs:Quốc_gia}"). Khi truy vấn trực tiếp thì địa chỉ URL của Sesame server là tồn tại thực. Giải thuật giả lập như sau:

i) Tại bước lập thứ k, thread chính tạo ra k thread và gọi phương thức Thread.start(). Rồi thread chính dừng lại một khoảng t_sleep mili giây.

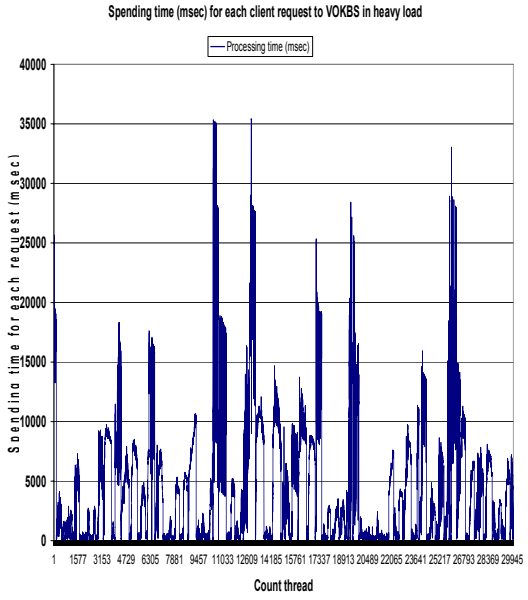
ii) lặp lại cho đến khi k = MAX_THREAD.

Trong thử nghiệm 1, chúng tôi gán MAX_THREAD = 1000, t_sleep = 5000 mili-giây, Sesame URL là http://172.28.10.27:8080/sesame/ và chỉ dùng một máy tính chạy Sesame server có 2 CPU.

Hình 3.1 là biểu đồ cho thấy thời gian $T_{spending}$ mà mỗi client tiêu tốn khi truy vấn cơ sở tri thức trong trường hợp tải nặng (tần suất có hơn 200 yêu cầu/giây). Thời gian $T_{spending}$ này được tính từ lúc client bắt đầu mở kết nối đến Sesame, rồi truyền câu truy vấn SeSQL, rồi chờ cho đến khi client nhận đầy đủ dữ liệu trả về và gọi hàm để tính số dòng và số cột của bảng dữ liệu.



Hình 3.1- Truy vấn trực tiếp Sesame. Biểu đồ cho thấy thời gian $T_{spending}$ mà mỗi client tiêu tốn khi truy vấn cơ sở tri thức trong trường hợp có hơn 200 yêu cầu/giây.



Hình 3.2 – Truy vấn cơ sở tri thức thông qua dịch vụ VOKBService. Biểu đồ cho thấy thời gian $T_{spending}$ mà mỗi client tiêu tốn khi truy vấn cơ sở tri thức trong trường hợp có hơn 200 yêu cầu/giây.

3.2. Thử nghiệm 2:

Trong thử nghiệm 2, chúng tôi vẫn tiến hành như trong thử nghiệm 1. Tuy nhiên, có sự thay đổi về số lượng các máy tính: i) có 1 máy tính chạy chương trình giả lập java multithread; ii) có 1 máy tính được cấu hình để chạy dịch vụ VOKBService (địa chỉ URL là http://172.28.10.26:8080/vokbs/) và VOInfoService (có địa chỉ URL là http://172.28.10.26:8080/is/); iii) có 2 máy tính chạy hai phiên bản Sesame server (có các địa chỉ URL là http://172.28.10.27:8080/sesame/ và http://172.28.10.28:8080/sesame/).

3.3 Đánh giá sơ bộ về 2 cuộc thử nghiệm

Qua số liệu đo đạc (minh họa trên hình 3.2 và 3.3 bên trên), chúng tôi nhận thấy trường hợp tải nặng – nghĩa là khi có nhiều yêu cầu truy vấn SeSQL gởi đến đồng thời thì sự phân chia yêu cầu này ra nhiều nút SeSame tính tóan là hợp lý nhất. Hệ thống 1 máy đơn Sesame (testcase 1) sẽ không thể chịu đựng được khi trung bình có hơn 1000 yêu cầu đồng thời trong một giây, còn hệ thống SWG trong trường hợp testcase 2 có thể chạy thành công thử nghiệm lên đến hơn 1500 yêu cầu đồng thời trong một giây. Hơn nữa, thời gian tiêu tốn trung bình trong trường hợp testcase 1 là: *6129,5 mili-giây*. Còn thời gian tiêu tốn trung bình của hệ thống SWG trong trường hợp testcase 2 dùng dịch vụ VOKBService để phân chia tải là: *4548,7 mili-giây*. Như vậy chúng ta có speedup của hệ thống là:

$$S = 6129,5 / 4548,7 = 1,35$$

4. KẾT LUẬN

Tóm lại qua hai phần thử nghiệm trên, chúng tôi kết luận kiến trúc của hệ thống SWG hoàn toàn phù hợp các ứng dụng Web có ngữ nghĩa tiếng Việt. Hơn nữa kiến trúc hệ thống SWG có thể được áp dụng cho nhiều bài toán thực tế phức tạp khác. Sau bài viết này, hệ thống SWG sẽ được tinh chỉnh nhằm đem lại hiệu suất cao hơn.

TÀI LIỆU THAM KHẢO

1. I. Foster, C. Kesselman: The Grid 2 – Blueprint for a new computing infrastructure, 2nd ed., Morgan Kaufmann, 2004.
2. N.T. Sơn, N.Q. Hùng: A practical service-oriented architecture for semantic search application, trong Hội nghị quốc tế COSCI 2005, Vietnam, 2005.
3. R. W. Lucke: Building Clustered Linux Systems, Prentice Hall PTR, 2005.
4. C. H. Trụ, H. N. Tuyên và V. Q. Duy: Hướng đến Web tiếng Việt có ngữ nghĩa, trong Kỷ yếu Hội thảo ICT.rds'04, Hà Nội, Việt Nam.