

KIỂM THỬ VÀ CẢI TIẾN HIỆU NĂNG CỦA HỆ THỐNG ĐĂNG KÝ TÍN CHỈ TẠI TRƯỜNG ĐẠI HỌC KINH TẾ - ĐẠI HỌC ĐÀ NẴNG

TESTING AND IMPROVING THE PERFORMANCE OF THE CREDIT REGISTRATION SYSTEM AT UNIVERSITY OF ECONOMICS – THE UNIVERSITY OF DANANG

Đặng Ngọc Châu¹, Đỗ Văn Nhỏ², Nguyễn Quang Vũ³, Nguyễn Thanh Bình⁴

¹Trường Đại học Kinh tế, Đại học Đà Nẵng; chau.dn@due.edu.vn

²Trường THPT Chuyên Lê Quý Đôn, Đà Nẵng; dovannho@gmail.com

³Trường Cao đẳng CNTT Hữu nghị Việt-Hàn; vungq@viethanit.edu.vn

⁴Trường Đại học Bách khoa, Đại học Đà Nẵng; ntbinh@dut.udn.vn

Tóm tắt - Hầu hết các ứng dụng hiện nay được phát triển và triển khai trên môi trường mạng Internet. Hơn nữa, các ứng dụng web này thường được số lượng lớn người sử dụng truy cập đồng thời. Vì vậy, hiệu năng của ứng dụng web đóng vai trò rất quan trọng. Kiểm thử hiệu năng cho phép đánh giá hiệu năng của ứng dụng, từ đó có các giải pháp kỹ thuật cải tiến ứng dụng. Trong bài báo này, chúng tôi trình bày phân tích hiện trạng của hệ thống đăng ký tín chỉ tại Trường Đại học Kinh tế - Đại học Đà Nẵng; tiến hành kiểm thử hiệu năng của hệ thống; sau đó, đề xuất giải pháp kỹ thuật cải tiến hiệu năng trên nhiều mặt, gồm thiết kế lại giao diện và cải tiến việc truy cập cơ sở dữ liệu. Kết quả kiểm thử hiệu năng sau khi thực hiện giải pháp cải tiến cho thấy thời gian thực thi các chức năng đăng ký tín chỉ đã giảm đáng kể.

Từ khóa - kiểm thử phần mềm; kiểm thử ứng dụng web; kiểm thử hiệu năng; cải tiến hiệu năng; cải tiến hiệu năng ứng dụng web.

1. Đặt vấn đề

Ứng dụng công nghệ thông tin từ lâu đã là một phần không thể thiếu trong mọi lĩnh vực, đặc biệt là trong công tác quản lý. Ở hầu hết các nước trên thế giới, để quản lý các công việc giáo dục và đào tạo thì việc ứng dụng công nghệ thông tin luôn được đặt ra như một giải pháp hiệu quả nhất.

Ở Việt Nam, trong những năm vừa qua, việc ứng dụng công nghệ thông tin vào các lĩnh vực quản lý giáo dục đang từng bước được triển khai. Cùng với việc chuyển đổi đào tạo đại học theo học chế tín chỉ, Trường Đại học Kinh tế - Đại học Đà Nẵng đã triển khai ứng dụng *Hệ thống đăng ký tín chỉ* trực tuyến để đáp ứng yêu cầu quản lý việc dạy và học theo đặc thù của hệ thống đào tạo này. Một trong những công việc khá nặng nề và mất công sức là việc tổ chức đăng ký học các học phần qua mạng Internet cho sinh viên các lớp tín chỉ. Với số lượng sinh viên nhập học ngày càng tăng, trong khi máy chủ, đường truyền chưa được nâng cấp tương xứng. Tuy nhiên, yêu cầu đặt ra là phải đảm bảo khả năng vận hành ổn định của hệ thống cơ sở hạ tầng mạng cho phần mềm đăng ký tín chỉ, không gây ra tình trạng nghẽn mạng, để phục vụ tốt nhất cho nhu cầu đăng ký học của từng sinh viên.

Với những lý do nêu trên, việc nghiên cứu đánh giá và cải tiến hiệu năng của *Hệ thống đăng ký tín chỉ* tại Trường Đại học Kinh tế là một nhu cầu cần thiết, và là một trong những biện pháp nâng cao chất lượng và hiệu quả của hệ thống.

Nội dung của bài báo này được trình bày trong 6 phần. Trong Mục 2, các tác giả sẽ trình bày khái quát về kiểm thử

Abstract - Nowadays, most applications are developed and deployed on Internet. In addition, such web applications are usually used by a huge number of users at the same time. Hence, web applications' performance plays an important role. Performance testing allows evaluating application performance and some technical solutions then should be applied to improve application. In this paper, we present the analysis of the credit registration system at University of Economics - The University of Danang; the realization of performance testing of the system; then a proposal of technical solutions improving its performance on different aspects, which includes redesigning the interface and improving the access databases. The results of performance testing after the improvement shows the time execution of the functions are significantly reduced.

Key words - software testing; web application testing; performance testing; performance improvement; web application performance improvement.

hiệu năng và ứng dụng của nó. Mục 3 sẽ giới thiệu tổng quan về hệ thống đăng ký tín chỉ tại Trường Đại học Kinh tế - Đại học Đà Nẵng. Mục 4 trình bày việc kiểm thử hiệu năng để đánh giá hệ thống hiện tại và từ đó đề xuất các giải pháp cải tiến hiệu năng của hệ thống. Trong Mục 5, các tác giả sẽ tiến hành phân tích và đánh giá kết quả kiểm thử của hệ thống sau khi được cải tiến. Mục cuối cùng là kết luận và đề xuất hướng nghiên cứu tiếp theo trong tương lai.

2. Kiểm thử hiệu năng

Kiểm thử hiệu năng (performance testing) là một trong những tiêu chí quan trọng để thẩm định một hệ thống hoạt động có thực sự hiệu quả khi nhiều người truy cập đồng thời cùng sử dụng một dịch vụ nào đó.

Hiện nay, có nhiều định nghĩa về kiểm thử hiệu năng. Theo Moorthy [1], kiểm thử hiệu năng là hành động kiểm định, đánh giá phần mềm, một thành phần hoặc phần cứng có đúng với yêu cầu về hiệu năng và tối ưu hóa hiệu năng của nó. Một cách định nghĩa khác về kiểm thử hiệu năng của Meier và các cộng sự [2], kiểm thử hiệu năng là kiểm thử xác định thời gian phản hồi (responsiveness), thông lượng (throughput), mức độ tin cậy (reliability) hoặc khả năng mở rộng (scalability) của hệ thống theo khối lượng công việc (workload). Như vậy, có thể nói, kiểm thử hiệu năng là hoạt động cần thiết cho việc phát triển những giải pháp tối ưu hóa hiệu năng cho phần mềm. Kiểm thử hiệu năng giúp chúng ta tránh được các tình huống không lường trước khi triển khai ứng dụng trong môi trường thực tế. Nói một cách khác, kiểm thử hiệu năng là tiến trình thu thập và phân tích thông tin, mà trong đó dữ liệu đo lường được tập

hợp để dự đoán khi nào các mức tải sẽ vượt quá khả năng chịu đựng của nguồn tài nguyên hệ thống.

Kiểm thử hiệu năng được thực hiện nhằm đánh giá các tính chất về tốc độ, khả năng ổn định và khả năng chịu đựng của phần mềm dưới một sức tải xác định trước [1, 2, 4].

Kiểm thử hiệu năng có thể xác định các vấn đề liên quan đến tốc độ như: Phần mềm có thỏa mãn người sử dụng về tốc độ? Các dữ liệu có được xử lý và sử dụng trước khi bị lạc hậu? Phần mềm có trình bày cho người sử dụng thông tin mới nhất? Đối với ứng dụng web, các dịch vụ web có trả lời trong khoảng thời gian cho phép trước khi thông báo lỗi xuất hiện?

Khả năng ổn định nhằm hướng đến sức tải lớn, kiểm thử quá tải, thời gian quá dài hay sự ổn định của phần mềm. Các vấn đề liên quan như: Phần mềm có thể thực thi thời gian dài mà không mất dữ liệu, bị chậm hay phải khởi động lại? Nếu phần mềm kết thúc một cách không dự báo trước, cái gì xảy ra đối với các giao tác chỉ mới hoàn thành một phần? Sự kết hợp các lỗi hay lặp lại các lỗi có làm cho hệ thống sụp đổ? Cái gì sẽ xảy ra nếu có nhiều người sử dụng hơn mong đợi? Cái gì sẽ xảy ra nếu tất cả những người sử dụng thực hiện cùng công việc?

Kiểm thử hiệu năng có thể xác định số lượng người sử dụng mà phần mềm hỗ trợ, hay khối lượng dữ liệu mà phần mềm có thể lưu trữ và xử lý, hay còn gọi là xác định khả năng chịu đựng của phần mềm. Các vấn đề liên quan cần xem xét như: Phần mềm có cung cấp thời gian trả lời nhất quán và chấp nhận được đối với người sử dụng? Phần mềm có lưu trữ tất cả dữ liệu thu thập được trong suốt thời gian hoạt động? Có những khuyến cáo khi phần mềm đạt đến khả năng đỉnh điểm? Phần mềm vẫn sẽ an toàn dưới sức tải rất lớn? Chức năng phần mềm vẫn sẽ đảm bảo dưới sức tải rất lớn? Phần mềm vẫn có thể chịu đựng được sức tải lớn không dự báo trước?

Trên cơ sở xác định các vấn đề liên quan được đề cập ở trên, kiểm thử viên cần có và phối hợp các chiến lược khác nhau nhằm cải tiến hiệu năng [5].

Trên thực tế, kiểm thử hiệu năng thường nhằm thực hiện:

- Đánh giá sự sẵn sàng vận hành của phần mềm.
- Đánh giá phần mềm so với các tiêu chí về hiệu năng.
- So sánh các tính chất hiệu năng của nhiều hệ thống hay cấu hình hệ thống.
- Tìm nguồn gốc của các vấn đề về hiệu năng.
- Hỗ trợ cho việc điều chỉnh hiệu năng của hệ thống.
- Xác định mức lưu lượng thông tin.

Hiện nay, có nhiều loại kiểm thử hiệu năng khác nhau, cụ thể như:

- Kiểm thử cơ sở (baseline test): Kiểm thử cơ sở là kiểm thử được xây dựng để đánh giá hiệu năng ứng dụng với sức tải một người sử dụng [1, 3]. Kịch bản kiểm thử có thể được tạo ra với thời gian suy nghĩ (think time) trong thực tế và những cài đặt khác giống như sử dụng trong thời gian thực.

- Kiểm thử tải (load test) nhằm đánh giá hiệu năng của phần mềm dưới các điều kiện về sức tải được dự báo trước, nghĩa là điều kiện bình thường. Kiểm thử tải cho phép đo lường thời gian trả lời, lưu lượng thông tin, tài nguyên sử dụng và xác định điểm bị sụp đổ của hệ thống [1, 3, 4].

- Kiểm thử quá tải, hay còn gọi là kiểm thử áp lực (stress test), nhằm xác định các điều kiện thất bại hay những điểm bị sụp đổ của phần mềm dưới những điều kiện vượt quá bình thường. Kiểm thử quá tải nên được thiết kế để đẩy giới hạn nguồn tài nguyên của các hệ thống lên đến điểm mà tại đó các liên kết yếu của chúng xuất hiện. Kiểm thử quá tải cho phép chỉ ra hoạt động của phần mềm dưới các điều kiện vượt quá về sức tải dự báo trước [1, 4].

- Kiểm thử spike (spike test): Kiểm thử này gần giống kiểm thử quá tải (stress test) nhưng hệ thống được đặt dưới sức tải cực cao trong một thời gian ngắn [1, 4]. Kiểm thử này giúp xác nhận hiệu năng hệ thống trong điều kiện tải cao đột ngột trong giờ giao dịch cao điểm của ứng dụng.

- Kiểm thử chịu đựng (endurance test): Kiểm thử chịu đựng tập trung vào đánh giá hiệu năng của hệ thống với mức tải sử dụng được định trước trong khoảng thời gian kéo dài [1]. Kiểm thử chịu đựng được thực hiện với 70% - 80% của sức tải mục tiêu, trong kịch bản có cài đặt thời gian suy nghĩ giống như trong thực tế.

- Kiểm thử cô lập nghẽn cổ chai (bottleneck isolation test): Kiểm thử cô lập nghẽn cổ chai là kiểm thử được thực hiện trên hệ thống hoặc một thành phần cụ thể để tìm ra các vấn đề và nguyên nhân ảnh hưởng đến hiệu năng của hệ thống [1].

- Kiểm thử khối lượng (volume test): Kiểm thử khối lượng là kiểm thử hiệu năng của hệ thống khi nó phải xử lý một lượng dữ liệu nhất định [1, 2]. Khối lượng này có thể là kích thước bản ghi dữ liệu hoặc nó cũng có thể là kích thước của một tập tin.

- Kiểm thử khả năng (capacity test): Nhằm xác định số lượng người sử dụng hay giao tác mà hệ thống sẽ hỗ trợ và vẫn đảm bảo đáp ứng yêu cầu về hiệu năng [1, 2].

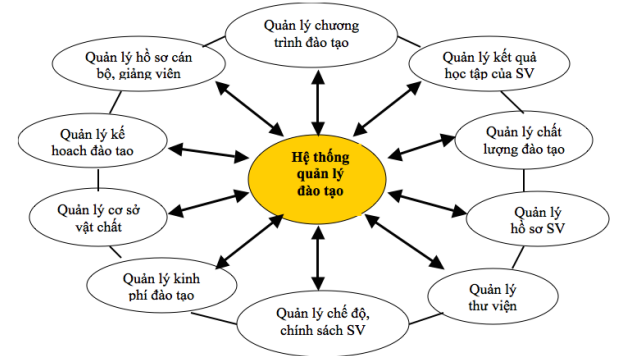
Mục này trình bày tổng quan về kiểm thử hiệu năng. Trong các mục tiếp theo, chúng tôi sẽ trình bày về hiện trạng hệ thống đăng ký tín chỉ tại Trường Đại học Kinh tế - Đại học Đà Nẵng cũng như việc áp dụng kiểm thử hiệu năng nhằm đánh giá hiệu năng và các giải pháp nhằm nâng cao hiệu năng của hệ thống.

3. Hệ thống đăng ký tín chỉ tại Trường Đại học Kinh tế

Trường Đại học Kinh tế - Đại học Đà Nẵng bắt đầu thực hiện việc đào tạo theo học chế tín chỉ cho các sinh viên được tuyển sinh năm 2006. Để triển khai việc quản lý quá trình đào tạo theo học chế tín chỉ, điều cần thiết là phải phát triển hệ thống thông tin và ứng dụng công nghệ thông tin vào quản lý đào tạo theo hệ thống tín chỉ. Trong năm học 2006-2007, Trường Đại học Kinh tế đã xây dựng và phát triển thành công hệ thống đào tạo tín chỉ qua mạng internet

và đã đưa vào sử dụng khá hiệu quả, góp phần rất lớn vào công tác quản lý học tập đối với sinh viên của nhà trường một cách nhanh chóng, chuẩn xác.

Hệ thống quản lý đào tạo (Hình 1) của Trường Đại học Kinh tế gồm các phân hệ chính sau: Quản lý kế hoạch đào tạo; Quản lý chương trình đào tạo; Đăng ký tín chỉ; Quản lý chất lượng đào tạo; Quản lý kinh phí đào tạo; Quản lý kết quả học tập của sinh viên; Quản lý hồ sơ sinh viên; Quản lý hồ sơ cán bộ, giảng viên; Quản lý cơ sở vật chất; Quản lý chế độ, chính sách cho sinh viên; Quản lý hồ sơ cán bộ, giảng viên; Quản lý cơ sở vật chất; Quản lý chế độ, chính sách SV



Hình 1. Hệ thống quản lý đào tạo tín chỉ

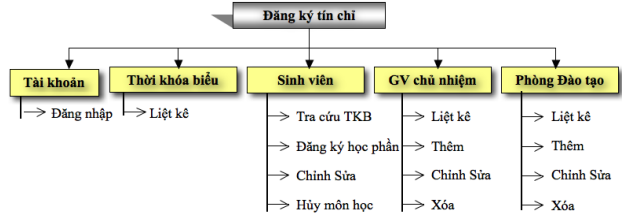
Trong đào tạo theo tín chỉ, sinh viên có quyền chọn học phần trong mỗi học kỳ phù hợp với năng lực và điều kiện học tập. Đăng ký học phần trong mỗi học kỳ của sinh viên phải đảm bảo điều kiện học phần tiên quyết, học phần học trước, học phần song hành trong chương trình đào tạo. Lập kế hoạch học tập và đăng ký học phần cho học kỳ có ý nghĩa rất quan trọng đối với sinh viên. Đăng ký phù hợp với năng lực và điều kiện của bản thân dẫn đến kết quả học tập tốt. Đăng ký học phần vượt quá năng lực có thể dẫn đến kết quả học tập kém, ảnh hưởng trực tiếp đến học tập trong các học kỳ sau.

Việc tổ chức đăng ký học phần cho sinh viên một cách thủ công là vấn đề phức tạp và chi phí cao, vì số lượng sinh viên của mỗi khóa khá cao (khoảng 2-3 nghìn sinh viên). Phân hệ đăng ký tín chỉ trực tuyến thuộc hệ thống quản lý đào tạo tín chỉ của Trường góp phần giải quyết vấn đề này. Hình 2 mô tả các chức năng cơ bản của phân hệ đăng ký tín chỉ. Trong đó, có các chức năng dành cho sinh viên, giảng viên chủ nhiệm (vai trò cố vấn) và nhân viên phòng đào tạo.

Trong quá trình đăng ký, hai chức năng con được sinh viên sử dụng chủ yếu để đăng ký học phần gồm: chọn học phần và chọn thời khóa biểu. Vì vậy, trong quá trình xây dựng hệ thống đăng ký, hai chức năng này rất được chú trọng nhằm đảm bảo hiệu quả sử dụng.

Hệ thống đăng ký tín chỉ đã được triển khai trên mạng internet và tự động hóa việc đăng ký cũng như xếp thời khóa biểu cho sinh viên. Trong điều kiện bình thường, khi số lượng truy cập không cao (khoảng vài trăm sinh viên), hệ thống vận hành khá tốt và ổn định. Tuy nhiên, khi số lượng sinh viên truy cập đồng thời rất lớn, thì tình trạng nghẽn mạng cục bộ hay không truy cập được hệ thống thường xuyên xảy ra. Vận hành thực tế cho thấy hiệu năng hệ thống không ổn định và có nhiều hạn chế trong những khoảng thời gian có số lượng truy cập lớn. Việc này gây

nhều khó khăn cho đơn vị quản lý cũng như người sử dụng.



Hình 2. Phân hệ đăng ký tín chỉ

Từ những hạn chế nêu trên, chúng tôi tiến hành kiểm thử hiệu năng của hệ thống đăng ký tín chỉ trực tuyến để có cơ sở nghiên cứu các giải pháp nâng cao hiệu năng của hệ thống.

4. Kiểm thử và cải tiến hiệu năng hệ thống đăng ký tín chỉ

Trong mục này, chúng tôi trình bày việc kiểm thử hiệu năng hệ thống đăng ký tín chỉ. Từ đó, chúng tôi đề xuất các giải pháp kỹ thuật cải tiến hiệu năng hệ thống.

4.1. Kiểm thử hiệu năng

Hệ thống đăng ký tín chỉ được cài đặt trong môi trường kiểm thử có cấu hình như sau:

- Bộ vi xử lý: Core Duo P8400 2,26 GHz;
- Bộ nhớ RAM: 2 GB;
- Hệ điều hành: Window Server 2008 và IIS 7.0;
- Hệ quản trị cơ sở dữ liệu: MS SQL Server 2008 R2.

Có nhiều công cụ [6, 7, 8, 9] khác nhau cho phép áp dụng các loại kiểm thử hiệu năng để kiểm thử các ứng dụng web. Trong bài báo này, chúng tôi sử dụng công cụ Firebug của Firefox [9] như một công cụ chính để kiểm thử dung lượng tải và thời gian thực thi của hai chức năng chính (được sử dụng chủ yếu bởi người sử dụng để đăng ký tín chỉ) của hệ thống đăng ký tín chỉ: chọn học phần và chọn thời khóa biểu. Kết quả kiểm thử của hai chức năng này về dung lượng tải và thời gian thực thi được trình bày trong Bảng 1.

Bảng 1. Kết quả kiểm thử các chức năng chọn học phần và chọn thời khóa biểu

	Dung lượng tải (Kilo Byte - KB)	Thời gian thực thi (giây - s)
Chức năng chọn học phần	61,3	46,26
Chức năng chọn thời khóa biểu	51,1	3,79

Bảng 1 cho thấy thời gian thực thi các chức năng này khá nhiều. Đặc biệt, chức năng chọn học phần đòi hỏi rất nhiều thời gian. Nếu số lượng người sử dụng truy cập đồng thời tăng lên thì hệ thống sẽ rất chậm và thậm chí không thể xử lý được.

Ngoài ra, để đánh giá tốt hơn hiệu năng cũng như phát hiện các yếu tố làm giảm hiệu năng của hệ thống, chúng tôi còn sử dụng các công cụ sau:

- ClickHeat [10]: là công cụ mã nguồn mở thể hiện trực quan những vùng trên trang web mà người dùng kích chuột vào nhiều nhất hoặc ít nhất.

- Crazy Egg [11]: cung cấp bộ công cụ phân tích giúp xác định được người dùng đang làm gì và bị tắc nghẽn ở vị trí nào trên trang web.

- YSlow [12]: chỉ ra những vấn đề có thể ảnh hưởng đến tốc độ tải về trong thiết kế.

Kết hợp các kết quả đánh giá từ các công cụ trên, ngoài những yếu tố khách quan, như số lượng người truy cập, điều kiện hạ tầng phần cứng và mạng, chúng tôi đã xác định được một số nguyên nhân chính làm giảm hiệu năng của hệ thống đăng ký tín chỉ như sau:

- Giao diện của hệ thống khá phức tạp, một số thành phần chiếm nhiều thời gian tải.
- Giao diện đăng ký học phần hiển thị đồng thời nhiều thông tin, như thông tin cá nhân, danh sách môn học, danh sách thời khóa biểu.
- Nhiều hoạt động diễn ra trong quá trình đăng ký, như hệ thống tự động đưa ra các học phần theo lộ trình học của từng chuyên ngành, theo từng khóa học; sinh viên thực hiện các thao tác đăng ký học phần, xóa học phần, đăng ký lại học phần; hệ thống phải thường xuyên kiểm tra để khóa không cho đăng ký vào những lớp đã được đăng ký đủ số lượng...
- Các thủ tục xử lý cơ sở dữ liệu khi số lượng bản ghi lớn chiếm nhiều thời gian.

4.2. Cải tiến hiệu năng

Từ các nguyên nhân chỉ ra ở trên, chúng tôi xác định giải pháp cải tiến tập trung vào ba khía cạnh: giao diện người dùng; mã nguồn; mô hình cơ sở dữ liệu và xử lý cơ sở dữ liệu.

4.2.1. Cải tiến giao diện người dùng

Giao diện người dùng được thiết kế đơn giản hơn, tối ưu mã HTML. Ngoài ra, DIV layout có tốc độ nhanh hơn TABLE layout, nên thay phương pháp dựng trang web TABLE layout bởi DIV layout. Cấu trúc lại thẻ <div> một cách gọn gàng, mạch lạc. Sử dụng thẻ <div> tránh được việc sử dụng các thẻ <td> và <tr> của TABLE lồng vào nhau nhiều cấp nên mã nguồn dễ bảo trì hơn, sinh ra mã HTML ít hơn. Nhờ vậy, thời gian tải trang web sẽ nhanh hơn. Hơn nữa, thiết kế chia lại bố cục của giao diện màn hình đăng ký theo các khối. Thay vì phải tải một lần tất cả các thông tin xuất hiện cùng một lúc trên một trang web, thiết kế chia nhỏ các khối hiển thị kết hợp với công nghệ Ajax, chỉ khi kích chuột vào thì khối đó mới được thực thi.

4.2.2. Cải tiến mã nguồn

Để mã nguồn được thực thi tối ưu hơn, nghĩa là nhanh hơn, chúng tôi đã thực hiện một số điều chỉnh:

- Sử dụng cấu trúc (structure) thay cho lớp (class);
- Sử dụng mảng (array) thay cho tập hợp (collection);
- Hạn chế việc gọi thực thi dữ liệu từ cơ sở dữ liệu;
- Đưa dữ liệu ra các mảng, việc sử dụng mảng sẽ giúp truy xuất nhanh hơn, sử dụng mảng để tối ưu cho chương trình khi xử lý.

4.2.3. Cải tiến xử lý cơ sở dữ liệu

Hệ thống đăng ký tín chỉ sử dụng hệ quản trị cơ sở dữ liệu (CSDL) MS SQL Server. Hệ thống sử dụng nhiều câu lệnh SQL để truy vấn CSDL. Các câu lệnh truy vấn được cài đặt trong các thủ tục (stored procedure) nhằm cải thiện tốc độ. Tuy nhiên, thời gian thực thi của các thủ tục đối với CSDL hiện tại cũng vẫn còn khá cao. Cụ thể, thời gian thực thi một số thủ tục truy vấn được trình bày trong Bảng 2.

Bảng 2. Thời gian thực thi các thủ tục truy vấn CSDL trước cải tiến

Thủ tục	Thời gian thực thi (giây - s)
MA_GetAverageMark	5
LE_UpdateOpenRegister	11
LE_GetSubjectCanRegister	21
LE_GetSubjectCreditRegister	25
SH_ScheduleChoiceRegister	26
LE_GetStudentScheduleCreditRegister	5
LE_DeleteSubjectRegisterOfStudent	25
Tổng cộng	118

Thời gian thực thi của các thủ tục truy vấn CSDL cho kết quả khác nhau, phụ thuộc vào độ phức tạp của thủ tục và độ lớn của CSDL. Nhưng nhìn chung, thời gian thực thi của mỗi thủ tục khá lớn, như các thủ tục: SH_ScheduleChoiceRegister(), LE_GetSubjectCreditRegister() và LE_DeleteSubjectRegisterOfStudent().

Vì vậy, chúng tôi đề xuất sử dụng bảng trung gian để lưu dữ liệu đăng ký tín chỉ. Dữ liệu của bảng trung gian này chỉ tồn tại trong thời gian đăng ký tín chỉ cho đến khi kết thúc việc đăng ký. Sau đó, toàn bộ dữ liệu sẽ được chuyển về bảng chính trong CSDL để phục vụ cho các chức năng khác của hệ thống. Việc sử dụng bảng trung gian giúp cho việc truy xuất dữ liệu nhanh hơn, vì các câu lệnh truy vấn sẽ không phải dò từ hàng trăm ngàn thậm chí là hàng triệu bản ghi để cho ra kết quả. Thật vậy, thời gian thực thi của các thủ tục trên cùng CSDL và môi trường phần cứng sau khi cải tiến được trình bày trong Bảng 3.

Bảng 3. Thời gian thực thi các thủ tục truy vấn CSDL sau khi cải tiến

Thủ tục	Thời gian thực thi (giây - s)
MA_GetAverageMark	1
LE_UpdateOpenRegister	1
LE_GetSubjectCanRegister	5
LE_GetSubjectCreditRegister	10
SH_ScheduleChoiceRegister	10
LE_GetStudentScheduleCreditRegister	3
LE_DeleteSubjectRegisterOfStudent	1
Tổng cộng	31

Bảng 3 cho thấy thời gian thực thi của các thủ tục đã giảm đáng kể, chỉ còn xấp xỉ 25% so với thời gian thực thi của các thủ tục trước khi cải tiến.

5. Kết quả kiểm thử

Sau khi thực hiện các cải tiến đề xuất ở trên, hệ thống được kiểm thử hiệu năng trong môi trường có cùng cấu hình và sử dụng cùng các công cụ kiểm thử như đã nêu trong Mục 4.1. Kết quả kiểm thử trước và sau khi cải tiến của các chức năng chọn môn học và chọn thời khóa biểu được trình bày trong các Bảng 4 và Bảng 5.

Bảng 4. So sánh kết quả kiểm thử trước và sau khi cải tiến chức năng chọn học phần

	Chức năng chọn học phần	
	Dung lượng tải	Thời gian thực thi
Trước cải tiến	61,3 KB	46,26 s
Sau cải tiến	6,2 KB	65 ms

Bảng 5. So sánh kết quả kiểm thử trước và sau khi cải tiến chức năng chọn thời khóa biểu

	Chức năng chọn thời khóa biểu	
	Dung lượng tải	Thời gian thực thi
Trước cải tiến	51,1 KB	3,79 s
Sau cải tiến	17,1 KB	272 ms

Kết quả thử nghiệm này được thực hiện trong cùng môi trường kiểm thử (cấu hình phần cứng và phần mềm) được trình bày trong Mục 4.1 và cùng một người sử dụng. Kết quả này đo thời gian thực thi trung bình sau 10 lần thực hiện.

Kết quả các Bảng 4 và Bảng 5 cho thấy dung lượng tải và thời gian thực thi của các chức năng chọn học phần cũng như chọn thời khóa biểu giảm đi rất đáng kể.

Ngoài ra, chúng tôi cũng đã tiến hành khảo sát trên hơn 150 sinh viên đã sử dụng hệ thống đăng ký tín chỉ trước và sau khi cải tiến hiệu năng tại Trường Đại học Kinh tế. Kết quả khảo sát cho thấy sinh viên nhận xét hệ thống vận hành tốt hơn nhiều so với trước khi thực hiện các cải tiến.

6. Kết luận

Kiểm thử và cải tiến hiệu năng các ứng dụng web luôn là thách thức đối người phát triển phần mềm cũng như những người nghiên cứu. Trong bài báo này, chúng tôi nghiên cứu về các kỹ thuật kiểm thử và cải tiến hiệu năng của ứng dụng web. Trên cơ sở đó, chúng tôi đã phân tích các hạn chế về hiệu năng của hệ thống đăng ký tín chỉ của Trường Đại học Kinh tế - là một ứng dụng có số lượng người sử dụng cùng một thời điểm rất cao. Từ đó, chúng tôi đã kiểm thử hiệu năng của hệ thống và đề xuất các giải pháp kỹ thuật nhằm cải tiến hiệu năng. Kết quả thử nghiệm cho thấy hiệu năng của hệ thống đăng ký tín chỉ đã được vận hành tốt hơn.

Trong tương lai, chúng tôi sẽ tiếp tục kiểm thử hiệu năng của hệ thống trên các môi trường có cấu hình khác nhau nhằm đánh giá tốt hơn về hiệu năng, cũng như sẽ tiếp tục nghiên cứu và ứng dụng tiếp tục các cải tiến khác, như kiến trúc phần mềm, giải thuật, cấu trúc và xử lý dữ liệu, nhằm nâng cao hơn nữa hiệu năng của hệ thống.

TÀI LIỆU THAM KHẢO

[1] R. R. Moorthy, *Software Performance Testing Handbook - A Comprehensive guide for beginners*, 2000.

[2] J.D. Meier, Carlos Farre, Prashant Bansode, Scott Barber, Dennis Rea, *Performance Testing Guidance for Web Applications*, Microsoft Corporation, 2007.

[3] Ian Molyneaux, *The Art of Application Performance Testing*, O'Reilly Media, Inc., 2009.

[4] J.D. Meier, S. Vasireddy, A. Babbar, and A. Mackman, *Improving.NET Application Performance and Scalability*, Microsoft Corporation.

[5] Nguyễn Thanh Bình, *Kiểm thử phần mềm*, Nhà xuất bản Giáo dục Việt Nam, 2013.

[6] E. H. Halili, *Apache JMeter*, Ltd, Packt Publishing, 2008.

[7] S. Joines, R. Willenborg, K. Hygh, *Performance Analysis for Java™ Web Sites*, Addison Wesley, 2002.

[8] IBM, *Using Rational Performance Tester Version 7*, IBM corporation, 2008.

[9] Firebug, <http://getfirebug.com>, 24/3/2017.

[10] ClickHeat, <http://www.dugwood.com/clickheat>, 24/3/2017.

[11] Crazy Egg, <http://www.crazyegg.com>, 24/3/2017.

[12] Yslow, <http://yslow.org>, 24/3/2017.

(BBT nhận bài: 24/03/2017, hoàn tất thủ tục phản biện: 25/04/2017)