



**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



## **ĐỀ CƯƠNG NGHIÊN CỨU MÔN CHUYÊN ĐỀ NGHIÊN CỨU 3**

**TÊN ĐỀ TÀI: ĐÁNH GIÁ HIỆU NĂNG VÀ TÍNH MỞ RỘNG CỦA HỆ  
THỐNG HADOOP DISTRIBUTED FILE SYSTEM**

**TÊN ĐỀ TÀI TIẾNG ANH: EVALUATING THE PERFORMANCE AND  
SCALABILITY OF THE HADOOP DISTRIBUTED FILE SYSTEM**

**GIẢNG VIÊN HƯỚNG DẪN  
TS. LÊ DUY TÂN**

**HỌC VIÊN THỰC HIỆN  
NGUYỄN HỒNG SƠN**



# 1 Tóm tắt

Trong giai đoạn hiện nay, nhu cầu dữ liệu cho học máy cũng như các lĩnh vực khác liên tục tăng lên, do đó hệ thống lưu trữ chúng phải thích ứng với nhu cầu ngày càng tăng về hiệu suất, độ tin cậy và khả năng chịu lỗi. Điều này làm tăng độ phức tạp quản trị và chi phí. Cải thiện hiệu suất và tăng khả năng mở rộng của hệ thống trong khi vẫn duy trì chi phí thấp là rất quan trọng. Các giải pháp phần mềm lưu trữ được cho là hướng đi mới do giải pháp phần cứng vẫn quá đắt đỏ.

Hadoop Distributed File System (HDFS) do Apache Software Foundation phát triển đến nay đã có phiên bản thứ 3, là một giải pháp đáng tin cậy để lưu trữ và phân phối dữ liệu một cách đáng tin cậy trên nhiều node. Nghiên cứu này xem xét cách HDFS hoạt động trong thiết lập với các phần mềm hỗ trợ phân quyền và bảo mật khác như Kerberos, Openldap, Apache Ranger, YARN. Nghiên cứu thay đổi số lượng các node lưu trữ và tính toán để kiểm tra khả năng mở rộng của hệ thống. Nghiên cứu hướng đến việc thử nghiệm và kiểm tra tính chính xác các cam kết của HDFS, đồng thời phát hiện các điểm có tiềm năng cải tiến. Nghiên cứu này sẽ cải thiện hiểu biết về vệ thống HDFS cho cộng đồng, đồng thời thông qua các kịch bản được trình bày để giải quyết các hạn chế về hiệu suất của HDFS.

## 2 Giới thiệu

Sự ra đời của Big Data và Machine Learning đã thúc đẩy nhiều nhà khoa học và các tập đoàn tập trung sang phát triển lĩnh vực này, khiến nhu cầu về lưu trữ dữ liệu tăng nhanh. Do đó, các giải pháp quản lý và lưu trữ dữ liệu mới được phát triển. Trong nghiên cứu này, tác giả giới thiệu và thực sự một số đánh giá hiệu năng của HDFS[2], một phần mềm cung cấp hệ thống lưu trữ phân tán được thiết kế với khả năng dễ mở rộng, có hiệu quả cao, khả năng chịu lỗi tốt, đồng thời có chi phí triển khai thấp.

Sự phức tạp của hệ phân tán nói chung và HDFS nói riêng đòi hỏi các cấu hình và điều chỉnh cẩn thận để đạt hiệu suất tốt, các thay đổi được thực hiện ảnh hưởng tới toàn hệ thống và có khả năng làm mất mát dữ liệu nếu không được thao tác đúng. Các dữ liệu lưu trong hệ thống hầu hết đều là dữ liệu nhạy cảm như: dữ liệu thô, dữ liệu sau xử lý, dữ liệu huấn luyện và các mô hình sản sinh. Việc quản lý truy cập, vừa thuận tiện cho người sử dụng, đồng thời đảm bảo các tiêu chí của an toàn dữ liệu là rất quan trọng. Kết hợp các công cụ quản lý xác thực và định danh mã nguồn mở đã được kiểm chứng như OpenLDAP [8], Kerberos [1], Apache Ranger [4] sẽ khắc phục được những nhược điểm của HDFS.

Bên cạnh đó, do mức độ lớn và quan trọng của dữ liệu, nó chịu áp lực độ chịu lỗi đối với sự hao mòn không thể tránh khỏi của phần cứng, cũng chịu áp lực về hiệu suất khi áp dụng các phương thức xác thực và mã hóa dữ liệu. Tìm hiểu và đánh giá hiệu suất toàn diện của hệ thống là cần thiết. Trong nghiên cứu này, tác giả xây dựng và trả lời các câu hỏi sẽ giúp đáp ứng một số nhu cầu lưu trữ dữ liệu phân tán lớn, yêu cầu khả năng chịu lỗi cao, thuận tiện trong phân quyền và đồng thời tốc độ truy cập nhanh. Khả năng tích hợp với các ứng dụng khác cũng được khảo sát và đề xuất. Trong trường hợp nào thì hệ thống hoạt động tốt nhất? Nếu như vậy thì cái giá phải đánh đổi là gì? Hệ thống hoạt động thế nào khi dữ liệu tăng lên cao hoặc nhiều người dùng cùng truy cập? Khi xảy ra hư hỏng phần cứng thì ứng dụng sẽ điều hướng lưu trữ như thế nào? Các giải pháp theo dõi hiệu suất hệ thống khả dụng là gì? Khi nào thì hiệu suất của hệ thống giảm?

Phần còn lại của nghiên cứu được trình bày như sau: Phần 3 trình bày các thành phần cơ bản của HDFS và các nghiên cứu liên quan. Phần 4 trình bày thiết kế hệ thống và khác thực nghiệm. Phần 5 trình bày chi tiết các kết quả. Cuối cùng, kết luận và đề xuất được trình bày ở phần 6.

## 3 Cơ sở lý thuyết và các nghiên cứu liên quan

### 3.1 Các thành phần của hệ thống

#### 3.1.1 OpenLDAP

LDAP - Lightweight Directory Access Protocol là một dịch vụ xác thực dạng client-server để truy cập các thư mục trong dịch vụ X.500 nhưng theo thời gian, nó dần độc lập khỏi X.500 và được sử dụng mặc định cho một số hệ điều hành trong quản lý truy cập thư mục. Nó phổ biến bởi các đặc điểm tốt hơn X.500 [7] :

- LDAP chạy trên TCP/IP chứ không phải trên OSI protocol stack. TCP/IP ít tốn tài nguyên hơn và phổ biến trên máy tính cá nhân.
- Mô hình chức năng của LDAP đơn giản hơn. Nó bỏ qua các tính năng trùng lặp, hiếm khi được sử dụng.
- LDAP sử dụng chuỗi để biểu diễn dữ liệu thay vì các cú pháp có cấu trúc phức tạp như ASN.1 (Abstract Syntax Notation One).

#### 3.1.2 Kerberos

Kerberos là một giao thức xác thực được sử dụng trong hệ thống mạng không an toàn. Nó được Học viện kỹ thuật Massachusetts (MIT), Hoa Kỳ phát triển, sử dụng nội bộ, và sau đó đã được công khai để cộng đồng sử dụng. Kerberos được thiết kế dựa trên giao thức Needham-Schroeder. Nó cung cấp xác thực cho người sử dụng thông qua một khóa phiên tạm thời, sẽ hết hạn sau một thời gian. Sau khoảng này, người dùng phải xác thực lại với máy chủ để nhận khóa phiên mới.

#### 3.1.3 Hadoop

Apache Hadoop là một phần mềm quản lý lưu trữ file phân tán và tính toán dữ liệu. Nó được thiết kế để có thể tích hợp nhiều máy chủ vào để tạo thành hệ lưu trữ có tính sẵn sàng cao, chịu lỗi tốt và chi phí rẻ. Giống như các phần mềm được Apache Software Foundation phát triển, nó cho phép sử dụng miễn phí kể cả cho mục đích thương mại.

#### 3.1.4 Ranger

Sau khi Hadoop phát triển và có nhiều người sử dụng, ASF tiếp tục phát triển Apache Ranger nhằm mục đích giám sát, quản lý bảo mật và truy cập trên hệ thống thư mục của Hadoop. Ranger cung cấp giao diện web thân thiện, dễ sử dụng, dễ tích hợp với các thành phần của Hadoop cũng như có nhiều plugin do cộng đồng phát triển.

### 3.2 Các nghiên cứu liên quan

Các phần mềm mã nguồn mở được đề cập ở trên được nhiều người sử dụng, do đó thu hút rất nhiều nghiên cứu. Choi và cộng sự [3] đã nghiên cứu các giải pháp sử dụng bộ nhớ đệm để tăng hiệu suất của OpenLDAP, khiến nó tăng 126% thông lượng và giảm 59% độ trễ. Guta và cộng sự [6] đề xuất HeABAC, một giải pháp sử dụng attribute-base access control vào Apache Ranger để cung cấp một giải pháp quản lý truy cập cho toàn bộ hệ sinh thái Hadoop. Guo và cộng sự [5] đã đề xuất iShuffle, một giải pháp giúp giảm thời gian tính toán tới 30,2% thông qua việc đẩy dữ liệu từ reduce task tới các node bằng giải thuật shuffle-onwrite và flexible scheduling.

### 3.3 Câu hỏi nghiên cứu

Các nghiên cứu với Hadoop hiện tại còn rời rạc, chủ yếu chỉ nghiên cứu một vài thành phần trong Hadoop hoặc các thành phần khác có liên quan, mà chưa xem xét tổng thể việc sử dụng chúng đồng thời trong một hệ thống. Các giải pháp sử dụng cơ sở dữ liệu quan hệ để lưu nhật ký từ Ranger chưa xem xét đến tính phức tạp của quản lý, tạo thêm thành phần cho hệ thống. Việc sử dụng cơ chế tạo người dùng trên Ranger chưa xem xét tới việc hầu hết các công ty/tập đoàn lớn (là những nơi đầu tư sử dụng Hadoop lớn nhất) đã có hệ thống xác thực tập trung.

## 4 Phân tích thiết kế hệ thống

## 5 Thực nghiệm và kết quả

## 6 Kết luận và phương hướng

# Mục lục

1	Tóm tắt . . . . .	1
2	Giới thiệu . . . . .	1
3	Cơ sở lý thuyết và các nghiên cứu liên quan . . . . .	2
3.1	Các thành phần của hệ thống . . . . .	2
3.1.1	OpenLDAP . . . . .	2
3.1.2	Kerberos . . . . .	2
3.1.3	Hadoop . . . . .	2
3.1.4	Ranger . . . . .	2
3.2	Các nghiên cứu liên quan . . . . .	2
3.3	Câu hỏi nghiên cứu . . . . .	3
4	Phân tích thiết kế hệ thống . . . . .	3
5	Thực nghiệm và kết quả . . . . .	3
6	Kết luận và phương hướng . . . . .	3
	<b>Mục lục</b>	<b>4</b>
	<b>Tài liệu tham khảo</b>	<b>5</b>

## Tài liệu tham khảo

- [1] Steven M Bellovin and Michael Merritt. “Limitations of the Kerberos authentication system”. In: *ACM SIG-COMM Computer Communication Review* 20.5 (1990), pp. 119–132.
- [2] Dhruba Borthakur. “The hadoop distributed file system: Architecture and design”. In: *Hadoop Project Website* 11.2007 (2007), p. 21.
- [3] Jong Hyuk Choi, Hubertus Franke, and Kurt Zeilenga. “Enhancing the performance of OpenLDAP directory server with multiple caching”. In: *Simulation Series* 35.4 (2003), pp. 737–744.
- [4] Apache Software Foundation. *Apache Ranger*. 2014. URL: <https://ranger.apache.org/> (visited on 04/24/2024).
- [5] Yanfei Guo et al. “ishuffle: Improving hadoop performance with shuffle-on-write”. In: *IEEE transactions on parallel and distributed systems* 28.6 (2016), pp. 1649–1662.
- [6] Maanak Gupta, Farhan Patwa, and Ravi Sandhu. “An attribute-based access control model for secure big data processing in hadoop ecosystem”. In: *Proceedings of the Third ACM Workshop on Attribute-Based Access Control*. 2018, pp. 13–24.
- [7] Heinz Johner et al. *Understanding Ldap*. Vol. 6. IBM, 1998.
- [8] OpenLDAP: Community developed LDAP software. *The OpenLDAP Project*. 1998. URL: <https://git.openldap.org/openldap/openldap/> (visited on 04/24/2024).