

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

-----oOo-----

BÁO CÁO TIỀU LUẬN



CƠ SỞ DỮ LIỆU PHÂN TÁN

**ĐỀ TÀI: TÌM HIỂU CƠ CHẾ PHÂN TÁN TRONG
HQT CSDL NOSQL - HADOOP/HBASE**

Giảng viên hướng dẫn:

Ths. Nguyễn Hồ Duy Tri

Nhóm: 3

Sinh viên thực hiện

Đoàn Ngọc Tuấn - 21521623

Doãn Công Trí - 21520492

Trần Quốc Hưng - 21522127

Trần Lê Tú - 21522746

Thành phố Hồ Chí Minh, ngày 25 tháng 12 năm 2023

LỜI CẢM ƠN

Lời đầu tiên nhóm em xin cảm ơn tập thể cán bộ, giảng viên trường Đại học Công Nghệ Thông Tin – ĐHQG Tp. Hồ Chí Minh đã tạo một môi trường tích cực cho sinh viên chúng em được thỏa sức sáng tạo, học tập và làm việc, cũng như trau dồi kiến thức cho chính bản thân mình.

Và đặc biệt, nhóm xin gửi lời cảm ơn chân thành và sự tri ân sâu sắc đối với thầy Nguyễn Hò Duy Tri - giảng viên lý thuyết và thực hành môn Cơ sở dữ liệu phân tán. Cảm ơn vì sự nhiệt tình, chu đáo, tận tâm của thầy trong công tác giảng dạy cũng như hỗ trợ nhóm em trong quá trình thực hiện đồ án môn học này.

Ngoài ra không thể không cảm ơn tập thể lớp IS211.O12 nói chung và những thành viên trong nhóm nói riêng đã có những đóng góp, ý kiến để nhóm có thể cải thiện chất lượng dự án. Cảm ơn vì các bạn đã đồng hành cùng chúng mình trong suốt quá trình thực hiện đồ án môn học.

Trong quá trình thực tập, cũng như là trong quá trình làm đồ án môn học, khó tránh khỏi sai sót, rất mong thầy bỏ qua. Đồng thời do trình độ lý luận cũng như kinh nghiệm thực tiễn còn hạn chế nên bài báo cáo không thể tránh khỏi những thiếu sót, nhóm em rất mong nhận được ý kiến đóng góp từ thầy để nhóm em học thêm được nhiều kinh nghiệm và sẽ hoàn thành tốt hơn những đồ án khác trong tương lai.

Em xin chân thành cảm ơn!

Nhóm sinh viên thực hiện

Nhóm 3

NHẬN XÉT CỦA GIẢNG VIÊN

MỤC LỤC

MỤC LỤC.....	4
TÓM TẮT NỘI DUNG ĐỀ TÀI.....	5
Chương 1: GIỚI THIỆU.....	6
1.1. TỔNG QUAN VỀ HỆ QUẢN TRỊ CSDL NOSQL.....	6
1.2. HỆ QUẢN TRỊ CSDL HBASE.....	8
1.2.1. Tổng quan về HBase.....	8
1.2.1.1 Hệ quản trị cơ sở dữ liệu Apache HBase.....	8
1.2.1.2 Lịch sử hình thành và tổ chức quản lý.....	9
1.2.1.3 Kiến trúc HBase.....	10
1.2.1.4 Data Flow trong HBase.....	12
1.2.3 Mô hình lưu trữ.....	13
1.2.5. Ngôn ngữ thao tác với dữ liệu.....	14
1.2.6. Cơ chế phân tán.....	15
Chương 2: HƯỚNG DẪN CÀI ĐẶT.....	16
2.1. YÊU CẦU CÀI ĐẶT.....	16
2.2. CÁC BƯỚC CÀI ĐẶT.....	16
2.2.1 Cấu hình máy ảo & cài đặt Hadoop.....	16
2.2.2 Cài đặt Hadoop.....	21
2.2.3 Cài đặt Zookeeper.....	40
2.2.4 Cài đặt HBase.....	46
Chương 3: THỰC NGHIỆM MÔ PHỎNG PHÂN TÁN.....	53
3.1. MÔ TẢ BÀI TOÁN ĐẶT RA VỚI DỮ LIỆU.....	53
3.2. MÔ TẢ CẤU TRÚC DỮ LIỆU SỬ DỤNG.....	55
3.3. CÁC BƯỚC THỰC NGHIỆM.....	59
3.3.1 Tạo Table và các column family.....	59
3.3.2 Thêm dữ liệu.....	61
3.3.3 Cập nhật dữ liệu.....	63
3.3.4 Xóa dữ liệu.....	64
3.3.5 Các thao tác với dữ liệu khác.....	64
TÀI LIỆU THAM KHẢO.....	68

TÓM TẮT NỘI DUNG ĐỀ TÀI

NoSQL (Non-relational Data Management System or Not Only SQL) is becoming increasingly popular and widely used in information technology applications. NoSQL systems provide flexible data storage and retrieval capabilities, suitable for applications with high scalability and performance requirements. One of these systems is HBase, a widely used database management system, especially in the field of Big Data.

HBase is a database management system based on Hadoop, an open-source project under Apache, developed and expanded from Google's Big Data storage project. HBase is written in Java and can store extremely large amounts of data, from terabytes to petabytes. The HBase prototype was created as a Hadoop contribution in February 2007. From October 2007 to September 2009, versions 0.81.1, 0.19.0, and 0.20.0 were released successively.

With outstanding features such as fast data filtering, storage of Big Data, the ability to store billions of rows and columns, real-time data querying, REST protocol support, and consistent data read and write mechanisms based on Hadoop, HBase is supported for various languages such as Java, PHP, and Python, etc.

The architecture of HBase consists of four basic components: HMaster, which is the central component in the HBase architecture, responsible for monitoring all RegionServers; HRegionServer, which directly manages the HRegions; HRegions, the fundamental architectural component of the HBase cluster, consisting of Memstore and Hfile; and Zookeeper, the monitoring center and configuration information storage.

HBase is a column-oriented database, and its tables are row-oriented. The table schema specifies column families, which are key-value pairs. A table can have multiple column families, and each column family can have any number of columns. The next column values are stored continuously on disk. Each cell of the table has its own metadata, such as a timestamp and other information.

The distributed mechanism of Apache's HBase follows a master-slave architecture. With this mechanism, ideally, the master receives all requests, and the actual work is performed by the slaves. However, in reality, for reading and writing data, HBase clients will directly communicate with specific Region Servers (slaves) responsible for handling row keys for all data operations. The master is only used by clients for table creation, modification, and deletion (HBaseAdmin).

With these advantages, HBase is used by technology companies worldwide on a large scale.

Chương 1: GIỚI THIỆU

1.1. TỔNG QUAN VỀ HỆ QUẢN TRỊ CSDL NOSQL

Cơ sở dữ liệu NoSQL (Non-relational Data Management System – Not Only SQL) là cơ sở dữ liệu không phải dạng bảng và lưu trữ dữ liệu khác với các bảng quan hệ. Cơ sở dữ liệu NoSQL có nhiều loại dựa trên mô hình dữ liệu của chúng. Các loại chính là document, key-value, column và graph. Chúng cung cấp các lược đồ linh hoạt và mở rộng quy mô một cách dễ dàng với lượng lớn dữ liệu và lượng người dùng tải cao.

- Cơ sở dữ liệu NoSQL xuất hiện vào cuối những năm 2000 khi chi phí lưu trữ giảm đáng kể. Đã qua rồi cái thời cần tạo ra một mô hình dữ liệu phức tạp, khó quản lý để tránh trùng lặp dữ liệu. Các nhà phát triển (thay vì lưu trữ) đang trở thành chi phí chính của việc phát triển phần mềm, do đó, cơ sở dữ liệu NoSQL được tối ưu hóa cho năng suất của nhà phát triển.
- Khi chi phí lưu trữ giảm nhanh chóng, lượng dữ liệu mà các ứng dụng cần để lưu trữ và truy vấn tăng lên. Dữ liệu này có đủ hình dạng và kích thước – có cấu trúc, bán cấu trúc và đa hình – và việc xác định trước lược đồ trở nên gần như không thể. Cơ sở dữ liệu NoSQL cho phép các nhà phát triển lưu trữ một lượng lớn dữ liệu phi cấu trúc, mang lại cho chúng rất nhiều tính linh hoạt.
- NoSQL đặc biệt nhấn mạnh đến mô hình lưu trữ cặp key - value và hệ thống lưu trữ phân tán:
 - + Phi quan hệ (Non-relational): relational là thuật ngữ sử dụng đến các mối quan hệ giữa các bảng trong cơ sở dữ liệu quan hệ (Relational Database Management System) sử dụng mô hình gồm 2 loại khóa: khóa chính (primary key) và khóa phụ (foreign key) để ràng buộc dữ liệu nhằm thể hiện tính nhất quán dữ liệu từ các bảng khác nhau. Non-relational là khái niệm không sử dụng các ràng buộc dữ liệu cho tính nhất quán dữ liệu.
 - + Lưu trữ dữ liệu phân tán.

- + Triển khai đơn giản, dễ nâng cấp và mở rộng.
- + Mô hình dữ liệu và truy vấn linh hoạt.
- Một số đặc điểm nhận dạng cho thể hệ CSDL mới này bao gồm: schema-free, hỗ trợ mở rộng dễ dàng, API đơn giản, nhất quán cuối (eventual consistency), không giới hạn khung gian dữ liệu,...
- Có nhiều cách phân loại các cơ sở dữ liệu NoSQL khác nhau, mỗi loại với các loại và loại con khác nhau, một số trong số đó có thể chồng chéo lên nhau. Một phân loại cơ bản dựa trên mô hình dữ liệu, với các ví dụ:
 - + Column: HBase, Accumulo, Cassandra, Druid, Vertica
 - + Document: Apache CouchDB, Clusterpoint, Couchbase, DocumentDB, HyperDex, Lotus Notes, MarkLogic, MongoDB, OrientDB, Qizx, RethinkDB
 - + Key-value: Aerospike, CouchDB, Dynamo, FairCom c-treeACE, FoundationDB, HyperDex, MemcacheDB, MUMPS, Oracle NoSQL Database, OrientDB, Redis, Riak, Berkeley DB.
 - + Graph: AllegroGraph, InfiniteGraph, MarkLogic, Neo4J, OrientDB, Virtuoso, Stardog.
 - + Multi-model: Alchemy Database, ArangoDB, CortexDB, FoundationDB, MarkLogic, OrientDB.
- Cơ sở dữ liệu NoSQL được sử dụng trong hầu hết mọi ngành. Các trường hợp sử dụng bao gồm từ mức độ quan trọng cao (ví dụ: lưu trữ dữ liệu tài chính và hồ sơ chăm sóc sức khỏe) đến thú vị hơn và phù phiếm hơn (ví dụ: lưu trữ các kết quả đọc IoT từ hộp vệ sinh mèo thông minh). Khi quyết định sử dụng cơ sở dữ liệu nào, những người ra quyết định thường tìm thấy một hoặc nhiều yếu tố sau đây dẫn họ đến việc chọn cơ sở dữ liệu NoSQL:
 - + Phát triển Agile tốc độ nhanh
 - + Lưu trữ dữ liệu có cấu trúc và bán cấu trúc
 - + Khối lượng dữ liệu khổng lồ

- + Yêu cầu đối với kiến trúc quy mô

1.2. HỆ QUẢN TRỊ CSDL HBASE

1.2.1. Tổng quan về HBase

1.2.1.1 Hệ quản trị cơ sở dữ liệu Apache HBase

Hbase là hệ quản trị cơ sở dữ liệu dựa trên Hadoop, đây là mã nguồn mở nằm trong dự án của Apache, phát triển và mở rộng từ dự án lưu trữ Big Data của google. (được xây dựng dựa trên Google Bigtable). Hbase được viết bằng ngôn ngữ Java có thể lưu trữ dữ liệu cực lớn từ terabytes đến petabytes.

HBase thực chất là một NoSQL điển hình nên vì thế các table của HBase không có một schemas cố định nào và cũng không có mối quan hệ giữa các bảng. Hiện nay, có rất nhiều công ty và tập đoàn công nghệ lớn trên thế giới sử dụng HBase, có thể kể đến: Facebook, Twitter, Yahoo, Adobe....

Các tính năng của Hbase

- Thời gian lọc dữ liệu nhanh
- Lưu trữ dữ liệu Big-Data, có thể lưu trữ hàng tỷ rows và columns
- Có độ ổn định và giảm thiểu rủi ro (failover) khi lưu một lượng lớn dữ liệu.
- Truy vấn dữ liệu theo thời gian thực
- Cung cấp giao thức REST, giúp trả về dữ liệu theo các định dạng khác nhau như plain text, json, xml. Nhờ đó chúng ta có thể khai thác dữ liệu không cần qua API từ phần mềm thứ 3.
- Nhất quán cơ chế đọc và ghi dữ liệu dựa trên Hadoop
- Nhiều extension hỗ trợ Hbase cho nhiều ngôn ngữ như Java, PHP, Python...
- Lưu trữ dữ liệu đáng tin cậy, được các hãng công nghệ trên thế giới sử dụng trên quy mô lớn.

1.2.1.2 Lịch sử hình thành và tổ chức quản lý

- **Năm phát hành:** 2007

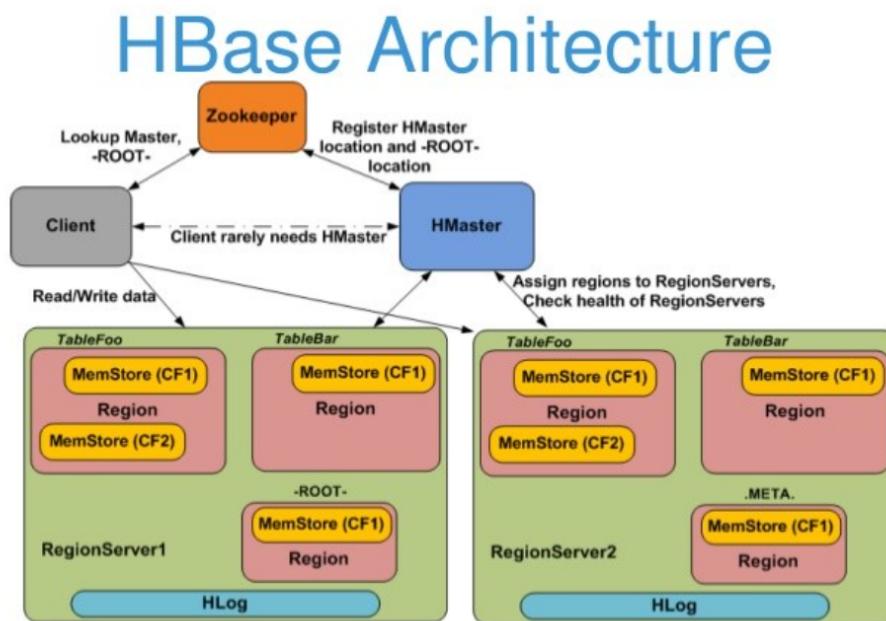
Apache HBase được phát hành lần đầu tiên vào tháng 2 năm 2007. Sau đó vào tháng 1 năm 2008, HBase trở thành một dự án con của Apache Hadoop. Năm 2010, HBase trở thành dự án cấp cao nhất của Apache.

Năm	Sự kiện
Tháng 11 năm 2006	Google đã phát hành bài báo trên BigTable.
Tháng 2 năm 2007	Nguyên mẫu HBase ban đầu được tạo ra như một đóng góp của Hadoop.
Tháng 10 năm 2007	HBase có thể sử dụng đầu tiên cùng với Hadoop 0.15.0 đã được phát hành.
Tháng 1 năm 2008	HBase trở thành dự án phụ của Hadoop.
Tháng 10 năm 2008	HBase 0.18.1 đã được phát hành.
Tháng 1 năm 2009	HBase 0.19.0 đã được phát hành.
Tháng 9 năm 2009	HBase 0.20.0 đã được phát hành.
Tháng 5 năm 2010	Hbase trở thành dự án cấp cao nhất của Apache

- **Tổ chức quản lý:** HBase ban đầu là một dự án của công ty Powerset, một công ty về tìm kiếm và ngôn ngữ tự nhiên có trụ sở tại San Francisco. Microsoft mua lại Powerset vào năm 2008.

- **Nguyên nhân ra đời:** Apache HBase bắt đầu như một dự án của công ty Powerset do nhu cầu xử lý lượng dữ liệu khổng lồ cho mục đích tìm kiếm ngôn ngữ tự nhiên.

1.2.1.3 Kiến trúc HBase

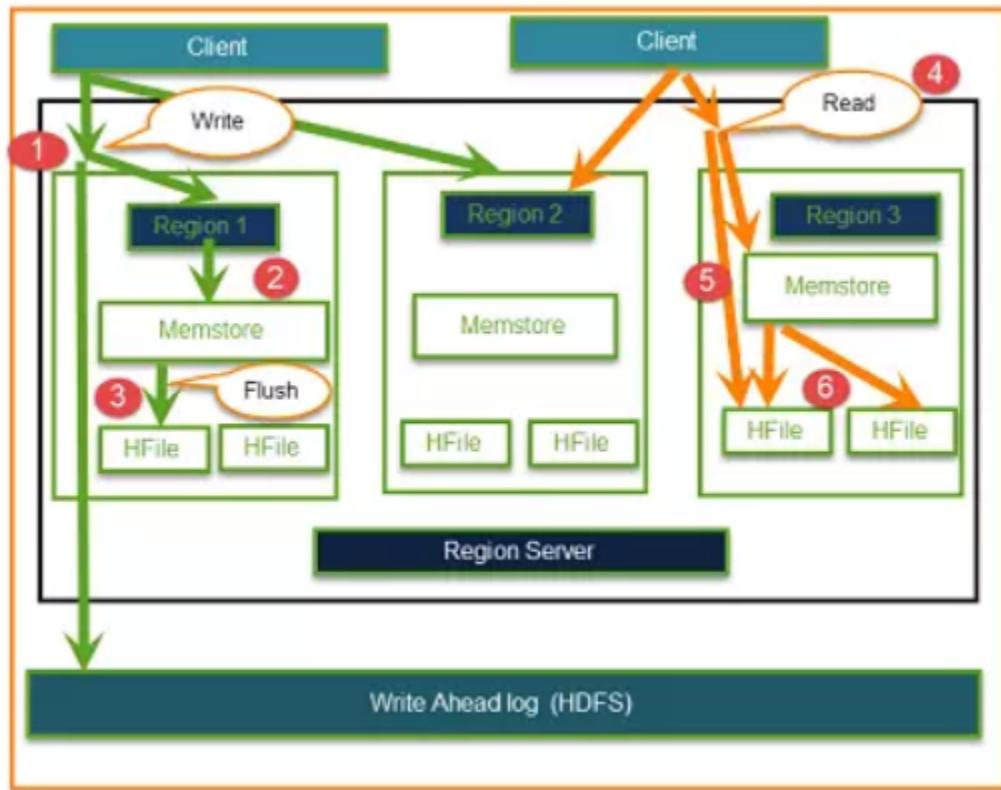


Kiến trúc HBase bao gồm 4 thành phần cơ bản:

- **HMaster:** chính là thành phần trung tâm trong kiến trúc của Hbase, nó giám sát tất cả các RegionServer trong cụm. Trong môi trường bao gồm nhiều cụm như Hadoop thì HMaster sẽ nằm ở NameNode, mọi thay đổi liên quan đến metadata đều phải thông qua HMaster, cụ thể:
 - + Cung cấp quyền admin, tính toán đến các Region Servers
 - + Gán các Regions cho Region Servers

- + HMaster cũng đảm nhận nhiệm vụ cân bằng tải hoặc xử lý lỗi ở các node con trong 1 cụm
- + Những thao tác liên quan đến metadata hoặc DDL tới cơ sở dữ liệu HBase
- **HBase RegionServer / HRegionServer:** Nhận trực tiếp yêu cầu DML (read, write) từ Client mà không cần thông qua HMaster. Khi HRegionServer nhận yêu cầu từ người dùng, nó thực hiện gán yêu cầu này cho các Regions tương ứng, HRegionServer còn chứa HLog dùng để chứa mọi log files. Trong môi trường bao gồm nhiều cụm như Hadoop thì HRegionServers sẽ nằm trên các DataNode, HMaster sẽ liên lạc với HRegionServers khi có các thao tác sau:
 - + Quản lý các Regions
 - + Phân phối các Regions tự động
 - + Nhận các lệnh DML
 - + Liên lạc trực tiếp với client
- **HRegions:** là thành phần kiến trúc cơ sở của Hbase cluster, nó bao gồm 2 thành phần chính là Memstore và Hfile. Memstore giống như một bộ nhớ cache, data đi vào đầu tiên sẽ nằm ở Memstore, được sắp xếp lại và cuối cùng là được đưa vào Hfile, nếu ta sử dụng Apache HBase trên một hệ thống Hadoop cluster, các Hfile này sẽ được lưu trữ vào trong Hadoop Distributed File System (HDFS).
- **Zookeeper:** ZooKeeper là trung tâm điều khiển của HBase, nó mang nhiệm vụ quan trọng là duy trì những thông tin cấu hình, cung cấp cơ chế đồng bộ phân tán cho toàn cơ sở dữ liệu. Cơ chế đồng bộ phân tán là việc truy cập phân tán đến các cụm đang chạy với nhiệm vụ điều phối nhiệm vụ giữa các node một cách chính xác, tránh xảy ra lỗi. Nếu Client muốn giao tiếp với Regions thì phải thông qua ZooKeeper trước, cụ thể của ZooKeeper như sau:
 - + Duy trì thông tin thiết lập.
 - + Cung cấp cơ chế đồng bộ phân tán.
 - + Thiết lập kết nối giữa Client với HregionServers.
 - + Kiểm tra liên tục các lỗi xảy ra với các cụm.

1.2.1.4 Data Flow trong HBase



Write operations

Step 1: Client muốn write data, tạo kết nối lần đầu tiên với Regions server và sau đó là regions.

Step 2: Regions liên lạc với memstore, lưu lại liên kết với column family.

Step 3: Dữ liệu trước tiên sẽ được lưu trữ tại Memstore. Tại đây dữ liệu sẽ được sắp xếp (sorted) trước khi chuyển tới HFile. Có 2 lý do chính cho việc sử dụng Memstore là :

- Hệ thống lưu trữ dữ liệu phân tán dựa trên row Key nên cần sắp xếp trước khi lưu trữ.
- Tối ưu hóa luồng ghi dữ liệu khi sử dụng kiến trúc The Log-Structured Merge Tree

Read operations

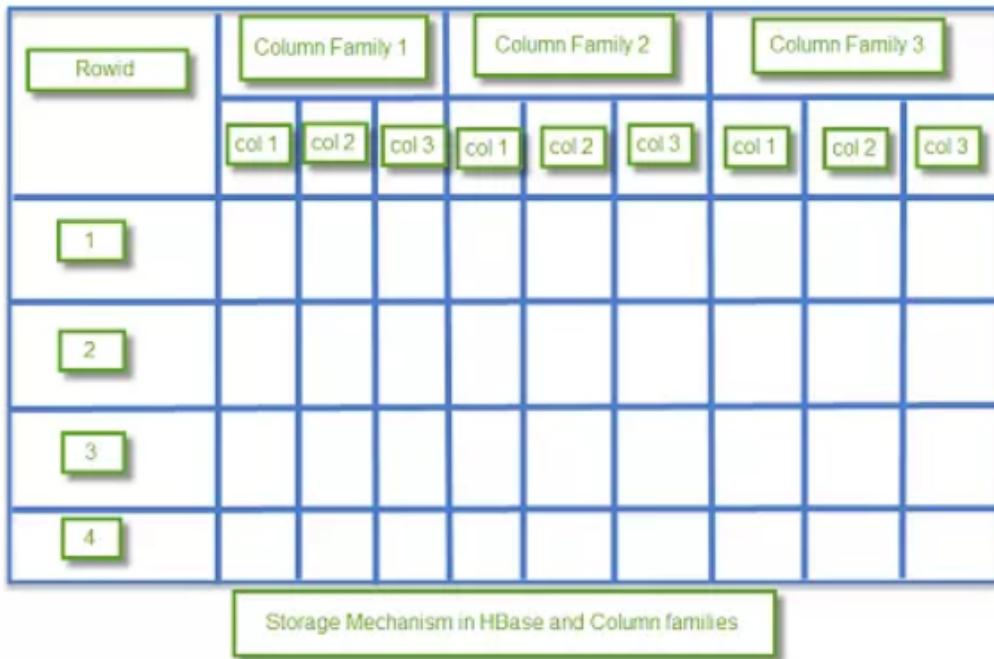
Step 4: Client muốn đọc data từ Regions

Step 5: Client có thể trực tiếp truy cập tới Mem store và yêu cầu dữ liệu.

Step 6: Client get data từ HFile.

1.2.2 Mô hình lưu trữ

HBase là một cơ sở dữ liệu theo hướng cột và dữ liệu được lưu trữ trong bảng. Các bảng được sắp xếp bởi RowId.



Các Column Family có trong lược đồ là các cặp key-value. Column Family có nhiều cột (col). Giá trị col được lưu trữ trong bộ nhớ đĩa. Mỗi ô của bảng có dữ liệu Meta riêng như timestamp và các thông tin khác.

Trong một HBase

- **Table** là một tập hợp các row
- **Rows** là tập hợp các Column family
- **Column family** là tập hợp các column
- **Column** là tập hợp các key - value

1.2.3. Ngôn ngữ thao tác với dữ liệu

Một số lệnh thao tác cơ bản trong HBase:

- **create**: tạo bảng mới

Để tạo một bảng mới với tên là **mytable** và hai cột là **col1** và **col2**, ta sử dụng cú pháp sau:

```
create 'mytable', 'col1', 'col2'
```

- **describe**: mô tả thuộc tính bảng
- **put**: thêm dữ liệu vào bảng

```
put <'tablename'>, <'rowname'>, <'columnvalue'>, <'value'>
```

Thêm dữ liệu vào bảng **mytable**:

```
put 'mytable', 'row1', 'col1', 'value1'
put 'mytable', 'row1', 'col2', 'value2'
```

Lệnh này sẽ thêm hai giá trị **value1** và **value2** vào hàng **row1** của bảng **mytable**, trong hai cột **col1** và **col2**.

- **get**: truy vấn dữ liệu từ bảng

```
get <'tablename'>, <'row key'>, <'filters'>
```

- **scan**: duyệt dữ liệu trong bảng

Thao tác scan được sử dụng để đọc nhiều hàng của một bảng. Nó khác với Get ở chỗ chúng ta cần chỉ định một tập hợp các hàng để đọc. Sử dụng Quét, chúng ta có thể lặp qua một dải hàng hoặc tất cả các hàng trong một bảng.

```
scan '<table name>'
```

- **delete**: xóa dữ liệu khỏi bảng

Sử dụng để xóa một hàng hoặc một tập hợp các hàng khỏi bảng HBase.

```
delete '<table name>', '<row>', '<column name >'
deleteall '<table name>', '<row>'
```

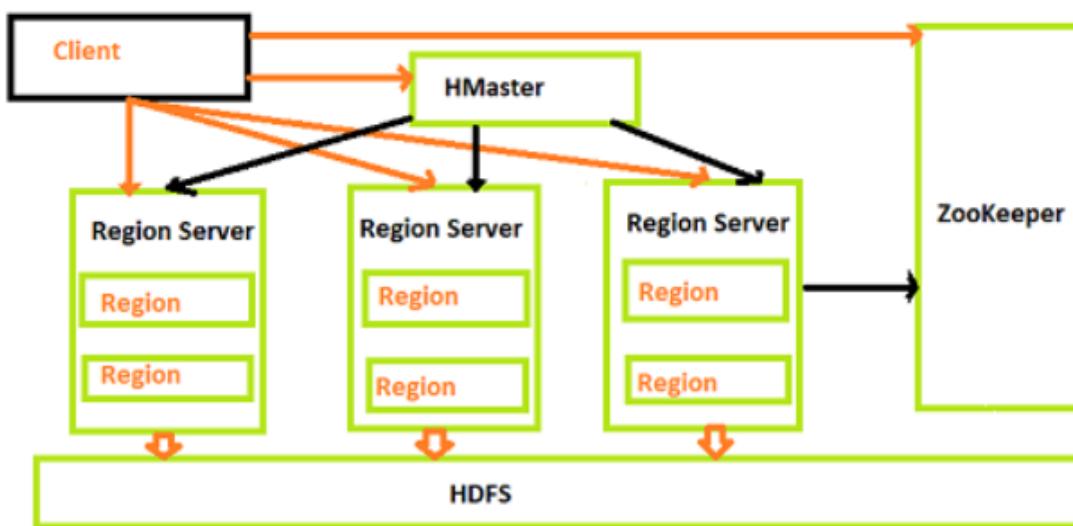
- **disable**: tạm dừng bảng
- **enable**: kích hoạt lại bảng
- **drop**: xóa bảng

• **Hbase Data Types**

Trong Apache HBase, không có khái niệm kiểu dữ liệu như vậy. Nó là một loại cơ sở dữ liệu byte-in và byte-out, trong đó, khi một giá trị được chèn vào, nó sẽ được chuyển đổi thành một mảng byte bằng giao diện **Put** và **Result**. Apache HBase sử dụng khung tuần tự hóa để chuyển đổi dữ liệu người dùng thành mảng byte.

1.2.4. Cơ chế phân tán

Cơ chế phân tán của HBase của Apache tuân theo cơ chế **master – slave**. Với cơ chế này, lẽ ra master nhận tất cả các yêu cầu và công việc thực sự được thực hiện bởi các slave, nhưng trên thực tế, để đọc và ghi dữ liệu, máy khách HBase sẽ chuyển trực tiếp đến Region Server (là slave) cụ thể chịu trách nhiệm xử lý các khóa hàng cho tất cả các hoạt động dữ liệu. Master chỉ được khách hàng sử dụng cho các hoạt động tạo, sửa đổi và xóa bảng (HBaseAdmin).



Chương 2: HƯỚNG DẪN CÀI ĐẶT

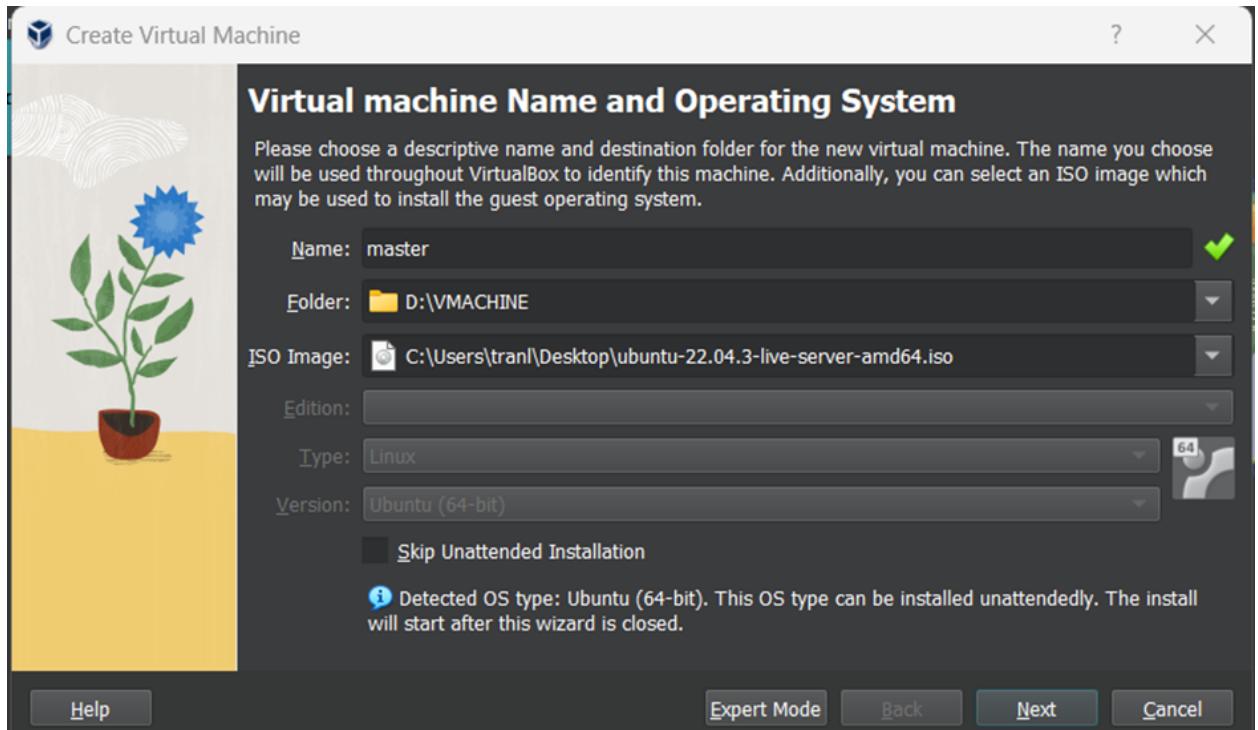
2.1. YÊU CẦU CÀI ĐẶT

- Oracle VM VirtualBox
- Ubuntu Server 22.04
- Openjdk-8-jdk
- Hadoop 3.3.2
- Apache Zookeeper 3.6.3
- Apache Storm 2.1.1
- Apache HBase 2.4.0

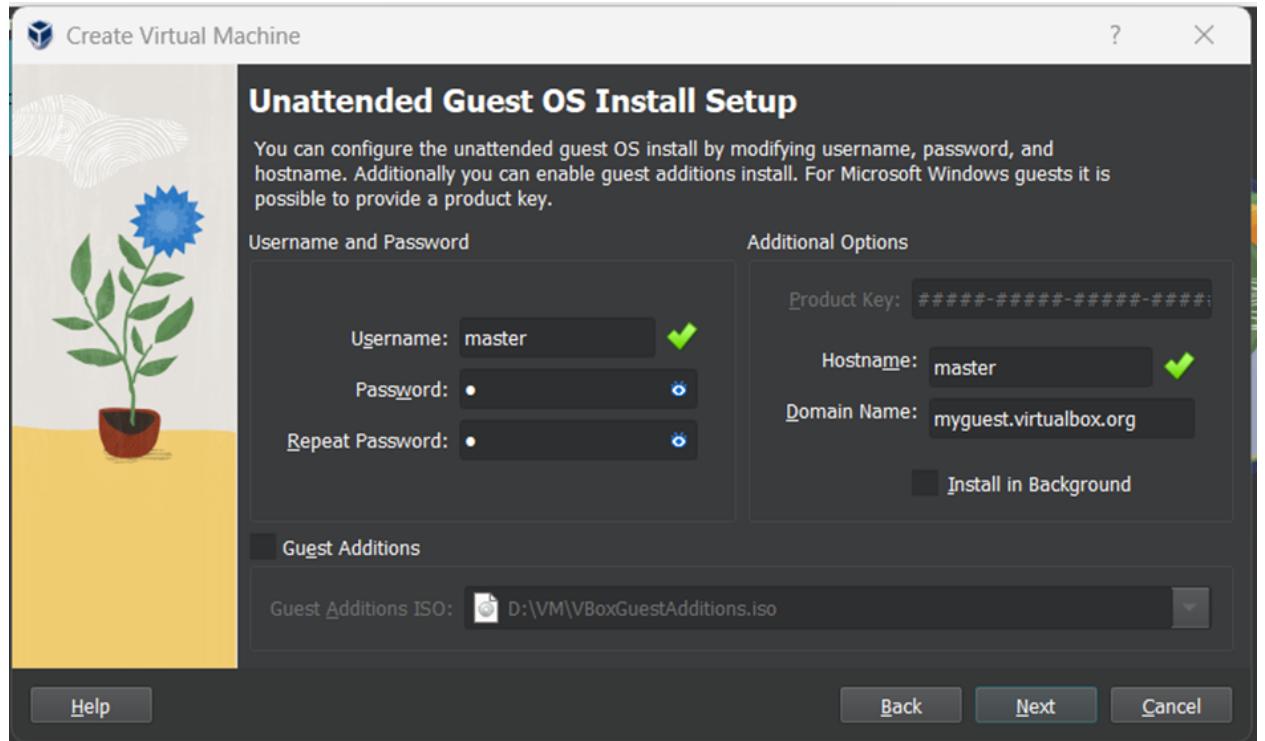
2.2. CÁC BƯỚC CÀI ĐẶT

2.2.1 Cấu hình máy ảo & cài đặt Hadoop

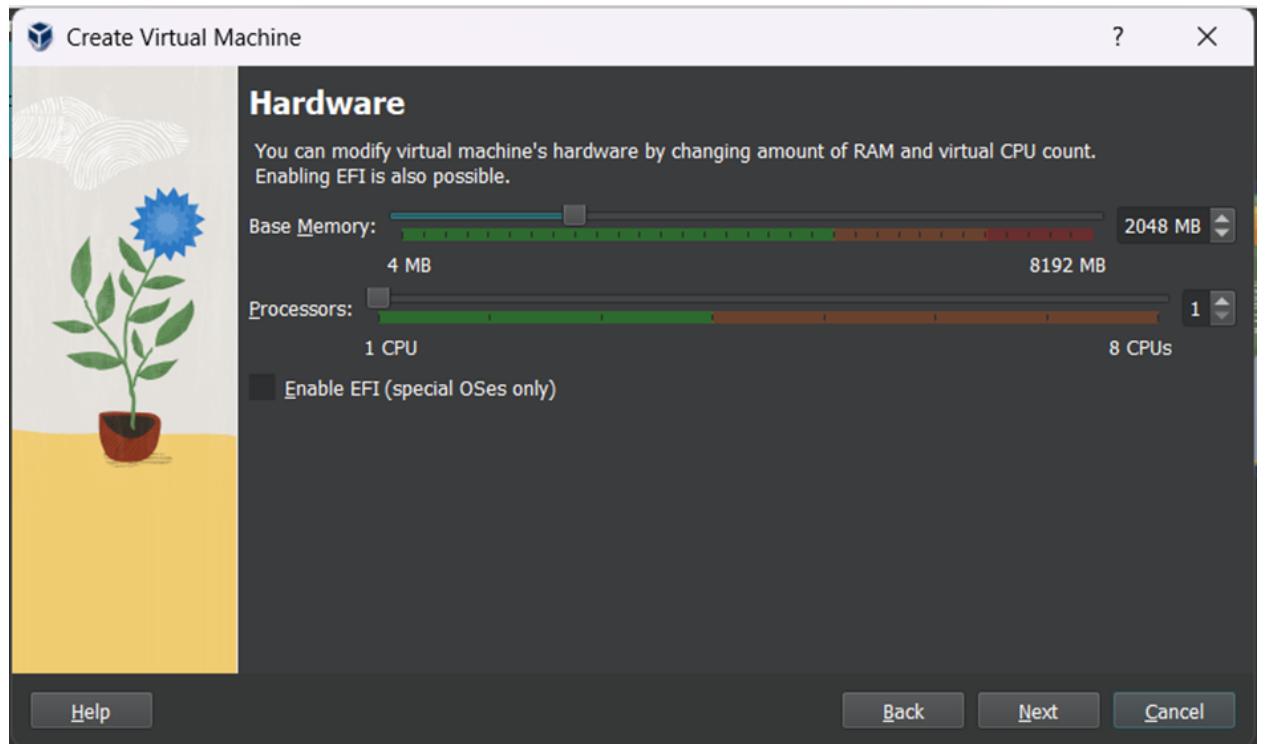
- Tạo máy ảo



- Tạo username & pass

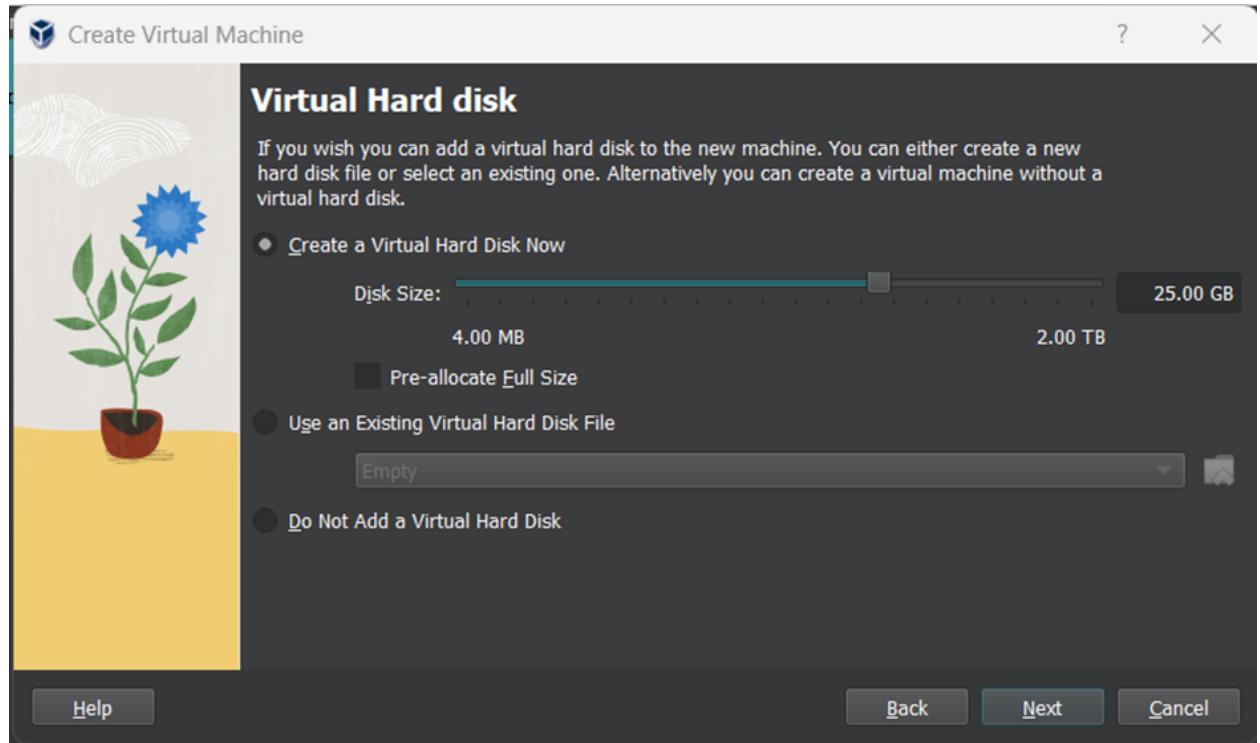


- Cấu hình bộ nhớ cho máy ảo

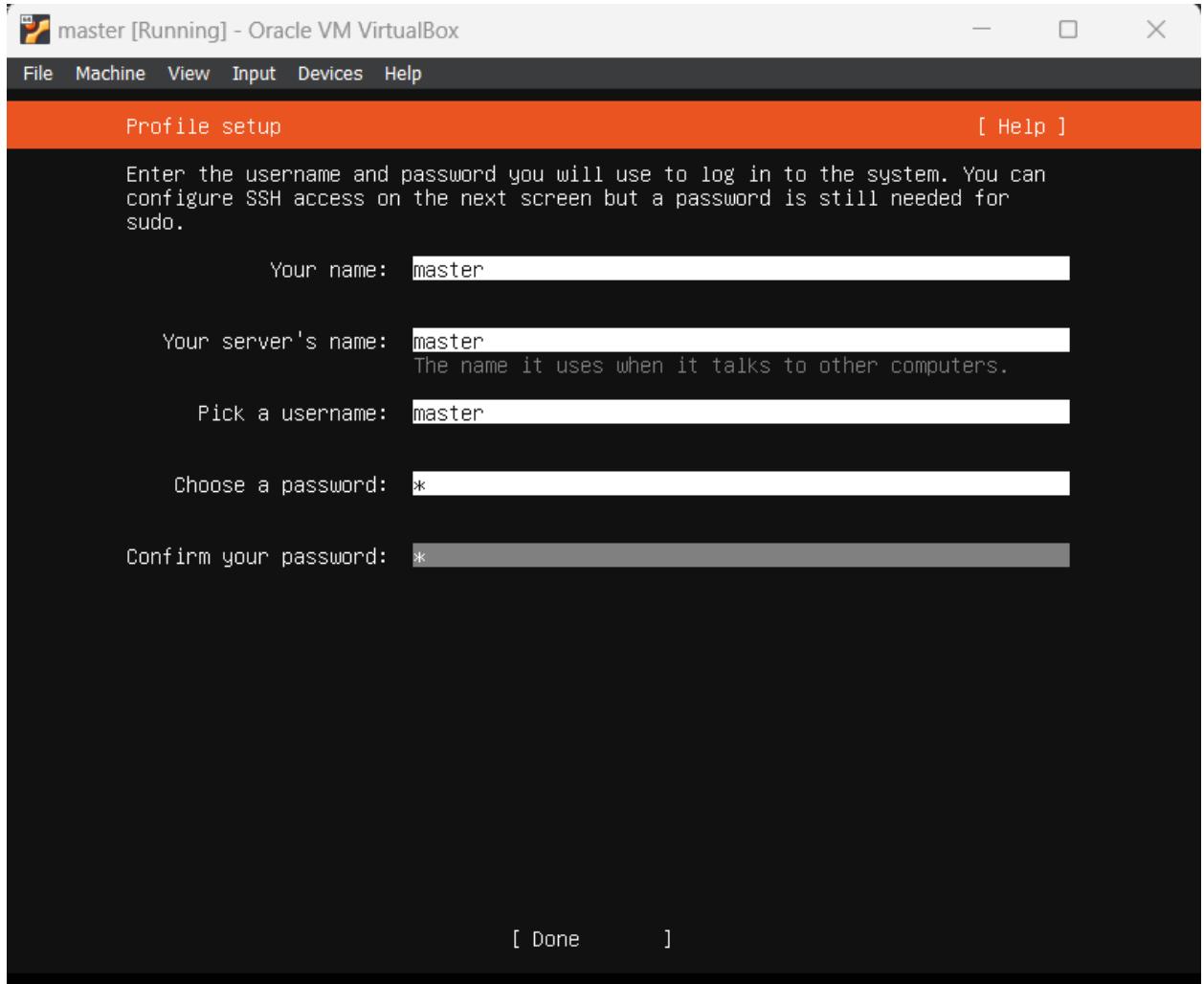


- Cấu hình ổ cứng

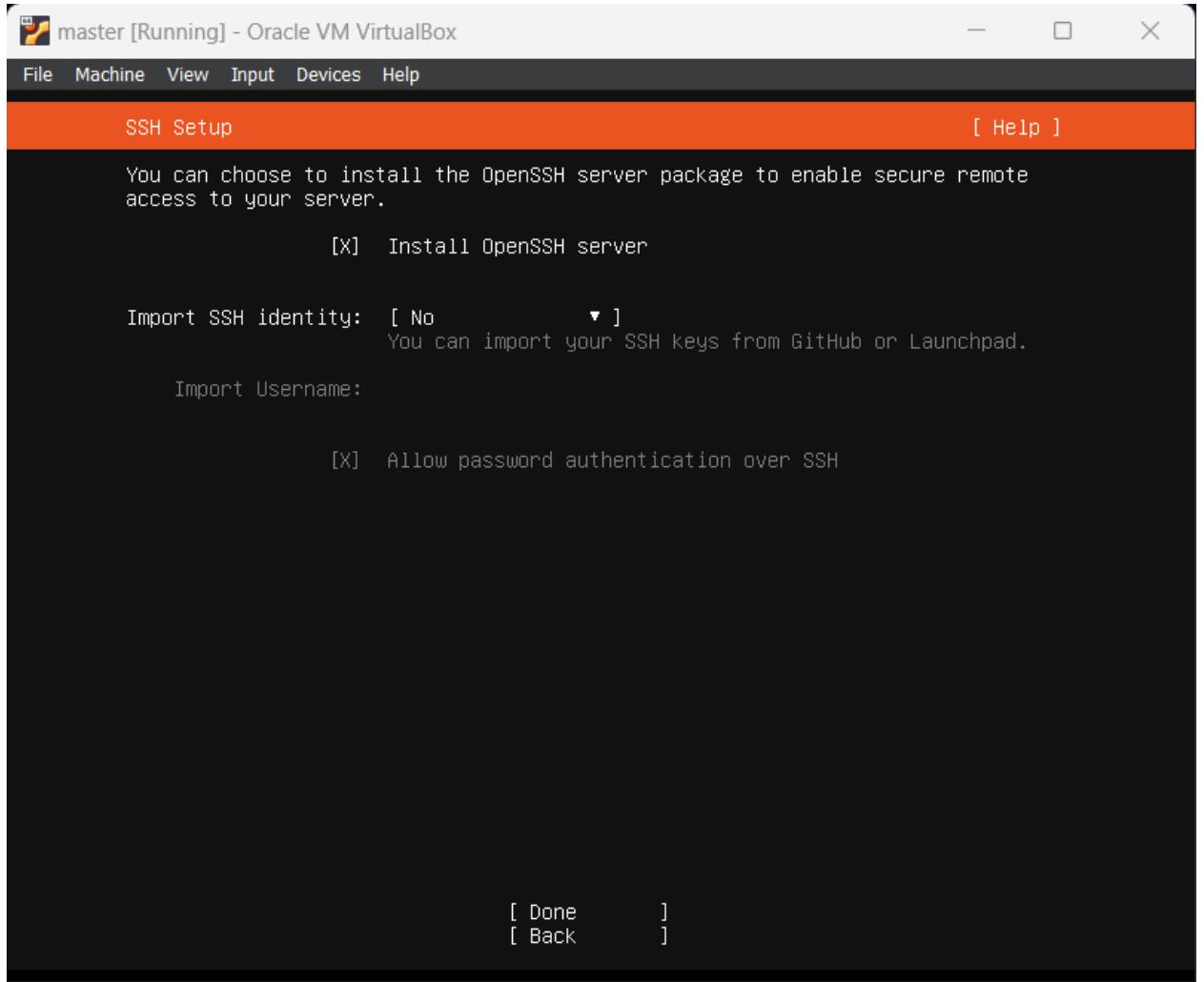
Đề tài: *Tìm hiểu cơ chế phân tán trong Hadoop/HBase*



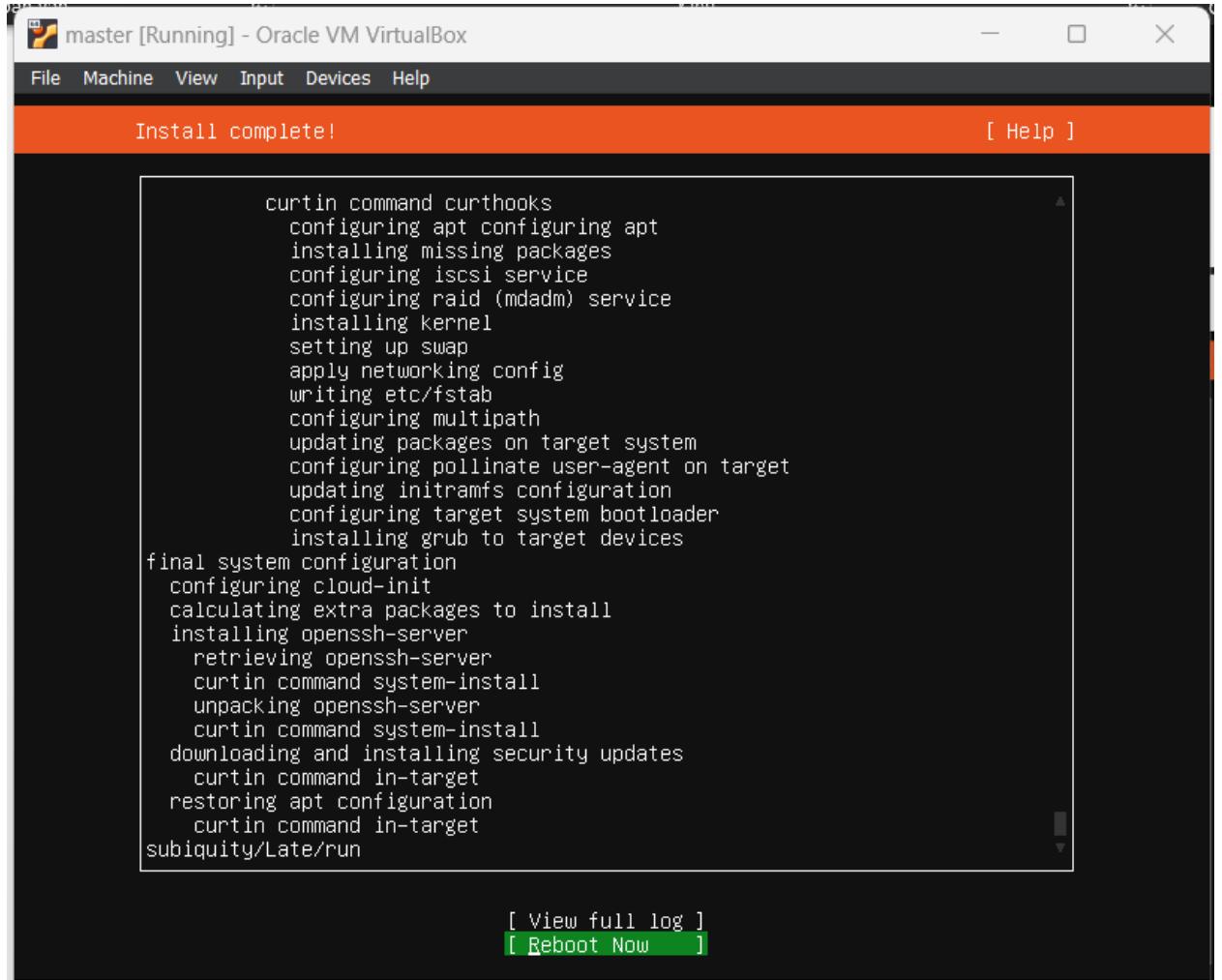
- **Đặt username và password cho máy ảo**



- Chọn Install OpenSSH Server và ấn Done



- Đợi cài đặt hoàn tất, chọn Reboot Now



2.2.2 Cài đặt Hadoop

- Tạo user hadoopuser và cấp tất cả quyền:

```
sudo adduser hadoopuser
```

```
sudo usermod -aG sudo hadoopuser
```

```
master@master:~$ sudo adduser hadoopuser_
master@master:~$ sudo usermod -aG sudo hadoopuser
```

- Thực hiện đăng nhập vào hadoopuser

```
master login: hadoopuser
Password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-91-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

 System information as of Thu Dec 21 08:51:40 AM UTC 2023

 System load: 0.38525390625    Processes:          103
 Usage of /: 43.6% of 11.21GB   Users logged in:      0
 Memory usage: 10%              IPv4 address for enp0s3: 10.0.2.15
 Swap usage:  0%

Expanded Security Maintenance for Applications is not enabled.

44 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Thu Dec 21 08:47:45 UTC 2023 on tty1
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

hadoopuser@master:~$ _
```

- **Thực hiện update và cài đặt jdk:**

sudo apt update

sudo apt install openjdk-8-jdk

Sudo apt update

```
hadoopuser@master:~$ sudo apt update
[sudo] password for hadoopuser:
Hit:1 http://vn.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://vn.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://vn.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://vn.archive.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
45 packages can be upgraded. Run 'apt list --upgradable' to see them.
hadoopuser@master:~$ _
```

Install openjdk

```

libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 libxcb-present0 libxcb-randr0 libxcb-render0
libxcb-shape0 libxcb-shm0 libxcb-sync1 libxcb-xfixes0 libxcb1-dev libxcomposite1 libxcursor1
libxdamage1 libxdmcp-dev libxfixes3 libxft2 libxi6 libxinerama1 libxkbfile1 libxmu6 libxpm4
libxrandr2 libxrender1 libxshmfence1 libxt-dev libxt6 libxtst6 libxvi libxxf86dga1 libxxf86vm1
openjdk-8-jdk-headless openjdk-8-jre openjdk-8-jre-headless session-migration ubuntu-mono
x11-common x11-utils x11proto-dev xorg-sgml-doctools xtrans-dev
Suggested packages:
default-jre libasound2-plugins alsamixer cups-common gvfs libice-doc liblcms2-utils opus-tools
pcscd pulseaudio librsvg2-bin lm-sensors libsm-doc libx11-doc libxcb-doc libxt-doc
openjdk-8-demo openjdk-8-source visualvm libnss-mdns fonts-nanum fonts-ipafont-gothic
fonts-ipafont-mincho fonts-wqy-microhei fonts-wqy-zenhei fonts-indic mesa-utils
The following NEW packages will be installed:
adwaita-icon-theme alsamixer-conf alsamixer-ucm-conf at-spi2-core ca-certificates-java
dconf-gsettings-backend dconf-service fontconfig fontconfig-config fonts-dejavu-core
fonts-dejavu-extra gsettings-desktop-schemas gtk-update-icon-cache hicolor-icon-theme
humanity-icon-theme java-common libasound2 libasound2-data libasyncns0 libatk-bridge2.0-0
libatk-wrapper-java libatk-wrapper-java-jni libatk1.0-0 libatk1.0-data libatspi2.0-0
libavahi-client3 libavahi-common-data libavahi-common3 libcairo-gobject2 libcairo2 libcups2
libdatrie1 libdconf1 libdeflate0 libdrm-amdgpu1 libdrm-intel1 libdrm-nouveau2 libdrm-radeon1
libflac8 libfontconfig1 libfontenc1 libgail-common libgail18 libgdk-pixbuf2.0-0
libgdk-pixbuf2.0-bin libgdk-pixbuf2.0-common libgif7 libgl1 libgl1-amber-dri libgl1-mesa-dri
libgl1-mesa-glx libglapi-mesa libglvnd0 libglx-mesa0 libglx0 libgraphite2-3 libgtk2.0-0
libgtk2.0-bin libgtk2.0-common libharfbuzz0b libice-dev libice6 libjbig0 libjpeg-turbo8 libjpeg8
liblcms2-2 libl10n15 libogg0 libopus0 libpango-1.0-0 libpangocairo-1.0-0 libpangoft2-1.0-0
libpcaccess0 libpcsc-lite1 libpixman-1-0 libpthread-stubs0-dev libpulse0 librsvg2-2
librsvg2-common libsensors-config libsensors5 libsm-dev libsm6 libsndfile1 libthai-data libthai0
libtiff5 libvorbis0a libvorbisenc2 libwebp7 libx11-dev libx11-xcb1 libxau-dev libxaw7
libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 libxcb-present0 libxcb-randr0 libxcb-render0
libxcb-shape0 libxcb-shm0 libxcb-sync1 libxcb-xfixes0 libxcb1-dev libxcomposite1 libxcursor1
libxdamage1 libxdmcp-dev libxfixes3 libxft2 libxi6 libxinerama1 libxkbfile1 libxmu6 libxpm4
libxrandr2 libxrender1 libxshmfence1 libxt-dev libxt6 libxtst6 libxvi libxxf86dga1 libxxf86vm1
openjdk-8-jdk openjdk-8-jdk-headless openjdk-8-jre openjdk-8-jre-headless session-migration
ubuntu-mono x11-common x11-utils x11proto-dev xorg-sgml-doctools xtrans-dev
0 upgraded, 136 newly installed, 0 to remove and 45 not upgraded.
Need to get 104 MB of archives.
After this operation, 406 MB of additional disk space will be used.
Do you want to continue? [Y/n] ^[S_]
```

- **Thực hiện download hadoop:**

wget

https://archive.apache.org/dist/hadoop/common/hadoop-3.3.2/hadoop-3.3.2.tar.gz

```

hadoopuser@master:~$ wget https://archive.apache.org/dist/hadoop/common/hadoop-3.3.2/hadoop-3.3.2.t
ar.gz
--2023-12-21 09:04:45-- https://archive.apache.org/dist/hadoop/common/hadoop-3.3.2/hadoop-3.3.2.t
ar.gz
Resolving archive.apache.org (archive.apache.org)... 65.108.204.189, 2a01:4f9:1a:a084::2
Connecting to archive.apache.org (archive.apache.org)|65.108.204.189|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 638660563 (609M) [application/x-gzip]
Saving to: ‘hadoop-3.3.2.tar.gz’

hadoop-3.3.2.tar.gz      100%[=====] 609.07M  8.70MB/s    in 8m 34s

2023-12-21 09:13:20 (1.18 MB/s) - ‘hadoop-3.3.2.tar.gz’ saved [638660563/638660563]

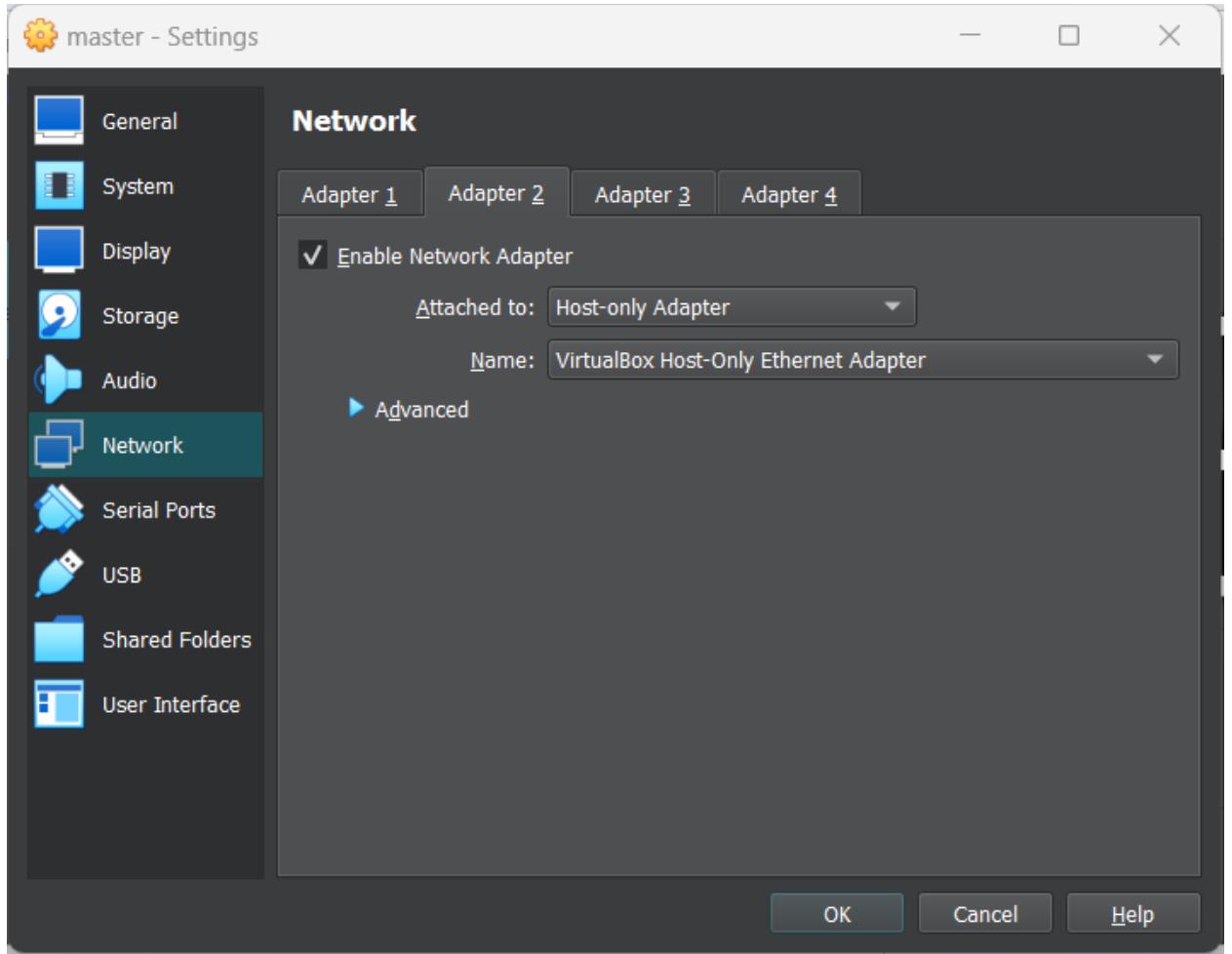
hadoopuser@master:~$ _
```

- **Giải nén file hadoop**

```
tar -zxvf hadoop-3.3.2.tar.gz
```

```
hadoop-3.3.2/share/hadoop/yarn/webapps/ui2/assets/images/ui-icons_222222_256x240.png  
hadoop-3.3.2/share/hadoop/yarn/webapps/ui2/assets/images/ui-icons_2e83ff_256x240.png  
hadoop-3.3.2/share/hadoop/yarn/webapps/ui2/assets/images/ui-icons_444444_256x240.png  
hadoop-3.3.2/share/hadoop/yarn/webapps/ui2/assets/images/ui-icons_454545_256x240.png  
hadoop-3.3.2/share/hadoop/yarn/webapps/ui2/assets/images/ui-icons_555555_256x240.png  
hadoop-3.3.2/share/hadoop/yarn/webapps/ui2/assets/images/ui-icons_777620_256x240.png  
hadoop-3.3.2/share/hadoop/yarn/webapps/ui2/assets/images/ui-icons_777777_256x240.png  
hadoop-3.3.2/share/hadoop/yarn/webapps/ui2/assets/images/ui-icons_888888_256x240.png  
hadoop-3.3.2/share/hadoop/yarn/webapps/ui2/assets/images/ui-icons_cc0000_256x240.png  
hadoop-3.3.2/share/hadoop/yarn/webapps/ui2/assets/images/ui-icons_cd0a0a_256x240.png  
hadoop-3.3.2/share/hadoop/yarn/webapps/ui2/assets/images/ui-icons_ffffff_256x240.png  
hadoop-3.3.2/share/hadoop/yarn/webapps/ui2/assets/vendor.css  
hadoop-3.3.2/share/hadoop/yarn/webapps/ui2/assets/vendor.js  
hadoop-3.3.2/share/hadoop/yarn/webapps/ui2/assets/yarn-ui.css  
hadoop-3.3.2/share/hadoop/yarn/webapps/ui2/assets/yarn-ui.js  
hadoop-3.3.2/share/hadoop/yarn/webapps/ui2/config/  
hadoop-3.3.2/share/hadoop/yarn/webapps/ui2/config/configs.env  
hadoop-3.3.2/share/hadoop/yarn/webapps/ui2/crossdomain.xml  
hadoop-3.3.2/share/hadoop/yarn/webapps/ui2/fonts/  
hadoop-3.3.2/share/hadoop/yarn/webapps/ui2/fonts/glyphicons-halflings-regular.eot  
hadoop-3.3.2/share/hadoop/yarn/webapps/ui2/fonts/glyphicons-halflings-regular.svg  
hadoop-3.3.2/share/hadoop/yarn/webapps/ui2/fonts/glyphicons-halflings-regular.ttf  
hadoop-3.3.2/share/hadoop/yarn/webapps/ui2/fonts/glyphicons-halflings-regular.woff  
hadoop-3.3.2/share/hadoop/yarn/webapps/ui2/fonts/glyphicons-halflings-regular.woff2  
hadoop-3.3.2/share/hadoop/yarn/webapps/ui2/index.html  
hadoop-3.3.2/share/hadoop/yarn/yarn-service-examples/  
hadoop-3.3.2/share/hadoop/yarn/yarn-service-examples/appcatalog/  
hadoop-3.3.2/share/hadoop/yarn/yarn-service-examples/appcatalog/appcatalog.json  
hadoop-3.3.2/share/hadoop/yarn/yarn-service-examples/httpd/  
hadoop-3.3.2/share/hadoop/yarn/yarn-service-examples/httpd/httpd-proxy.conf  
hadoop-3.3.2/share/hadoop/yarn/yarn-service-examples/httpd/httpd.json  
hadoop-3.3.2/share/hadoop/yarn/yarn-service-examples/httpd-no-dns/  
hadoop-3.3.2/share/hadoop/yarn/yarn-service-examples/httpd-no-dns/httpd-no-dns.json  
hadoop-3.3.2/share/hadoop/yarn/yarn-service-examples/httpd-no-dns/httpd-proxy-no-dns.conf  
hadoop-3.3.2/share/hadoop/yarn/yarn-service-examples/sleeper/  
hadoop-3.3.2/share/hadoop/yarn/yarn-service-examples/sleeper/sleeper.json  
hadoopuser@master:~$ _
```

- **Power off the Machine và Thay đổi adapter 2**



- Thực hiện tạo file 01-netcfg.yaml và cấu hình mạng như dưới đây

sudo vim /etc/netplan/01-netcfg.yaml

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: true
    enp0se8:
      addresses: [192.168.56.50/24]
      dhcp4: false
```

- Thực hiện apply netplan và kiểm tra ip:

sudo netplan apply

sudo ip a

```

hadoopuser@master:~$ sudo netplan apply
hadoopuser@master:~$ sudo ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
            inet6 ::1/128 scope host
                valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
0
    link/ether 08:00:27:c3:2a:4f brd ff:ff:ff:ff:ff:ff
        inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic enp0s3
            valid_lft 86395sec preferred_lft 86395sec
            inet6 fe80::a00:27ff:fec3:2a4f/64 scope link
                valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
0
    link/ether 08:00:27:ec:14:48 brd ff:ff:ff:ff:ff:ff
        inet 192.168.56.50/24 brd 192.168.56.255 scope global enp0s8
            valid_lft forever preferred_lft forever
            inet6 fe80::a00:27ff:feec:1443/64 scope link
                valid_lft forever preferred_lft forever
hadoopuser@master:~$
```

- **Thực hiện chỉnh sửa file host và thay đổi hostname**

sudo nano /etc/hosts

```

GNU nano 6.2                               /etc/hosts *
192.168.56.50 master
192.168.56.51 slave1
192.168.56.52 slave2
```

sudo hostname master

```

hadoopuser@master:~$ sudo hostname master
hadoopuser@master:~$ hostname
master
```

- **Thực hiện chỉnh sửa file ~/.bashrc**

sudo nano ~/.bashrc

thêm vào những lệnh như sau:

```

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre
export HADOOP_HOME=$HOME/hadoop-3.3.2
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin_
^G Help      ^O Write Out  ^W Where Is   ^K Cut       ^T Execute    ^C Location  M-U Undo
^X Exit      ^R Read File   ^Y Replace    ^U Paste     ^J Justify    ^V Go To Line M-E Redo
```

- **Thực hiện các lệnh sau:**

source ~/.bashrc

git clone <https://github.com/nilesh-g/hadoop-cluster-install.git>

```

hadoopuser@master:~$ source ~/.bashrc
hadoopuser@master:~$ git clone https://github.com/nilesh-g/hadoop-cluster-install.git
Cloning into 'hadoop-cluster-install'...
remote: Enumerating objects: 26, done.
remote: Counting objects: 100% (26/26), done.
remote: Compressing objects: 100% (20/20), done.
remote: Total 26 (delta 9), reused 12 (delta 4), pack-reused 0
Receiving objects: 100% (26/26), 11.47 KiB | 217.00 KiB/s, done.
Resolving deltas: 100% (9/9), done.
hadoopuser@master:~$ ls
hadoop-3.3.2  hadoop-3.3.2.tar.gz  hadoop-cluster-install
hadoopuser@master:~$ _

```

- **Copy từ hadoop-cluster-install/master qua máy master**

cp hadoop-cluster-install/master/ hadoop-3.3.2/etc/hadoop/*

```

hadoopuser@master:~$ cp hadoop-cluster-install/master/* hadoop-3.3.2/etc/hadoop/
hadoopuser@master:~$ 

```

- **Kiểm tra lại các file**

cd hadoop-3.3.2/etc/hadoop/

sudo nano hadoop-env.sh

```

GNU nano 6.2                                     hadoop-env.sh *
## {yarn-env.sh|hdfs-env.sh} > hadoop-env.sh > hard-coded defaults
##
## {YARN_xyz|HDFS_xyz} > HADOOP_xyZ > hard-coded defaults
## 

# Many of the options here are built from the perspective that users
# may want to provide OVERWRITING values on the command line.
# For example:
#
JAVA_HOME="/usr/lib/jvm/java-8-openjdk-amd64/jre"
#
# Therefore, the vast majority (BUT NOT ALL!) of these defaults
# are configured for substitution and not append. If append
# is preferable, modify this file accordingly.

```

sudo nano core-site.xml

```

GNU nano 6.2                                     core-site.xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://master:9000</value>
  </property>
</configuration>

```

sudo nano hdfs-site.xml

```
GNU nano 6.2                                     hdfs-site.xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>dfs.name.dir</name>
    <value>${user.home}/bigdata/hd-data/nn</value>
  </property>
  <property>
    <name>dfs.replication</name>
    <value>2</value>
  </property>
</configuration>
```

sudo nano mapred-site.xml

```
GNU nano 6.2                                     mapred-site.xml
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
  <property>
    <name>mapreduce.application.classpath</name>
    <value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$HADOOP_MAPRED_HOME/share/hadoop/mapred</value>
  </property>
</configuration>
```

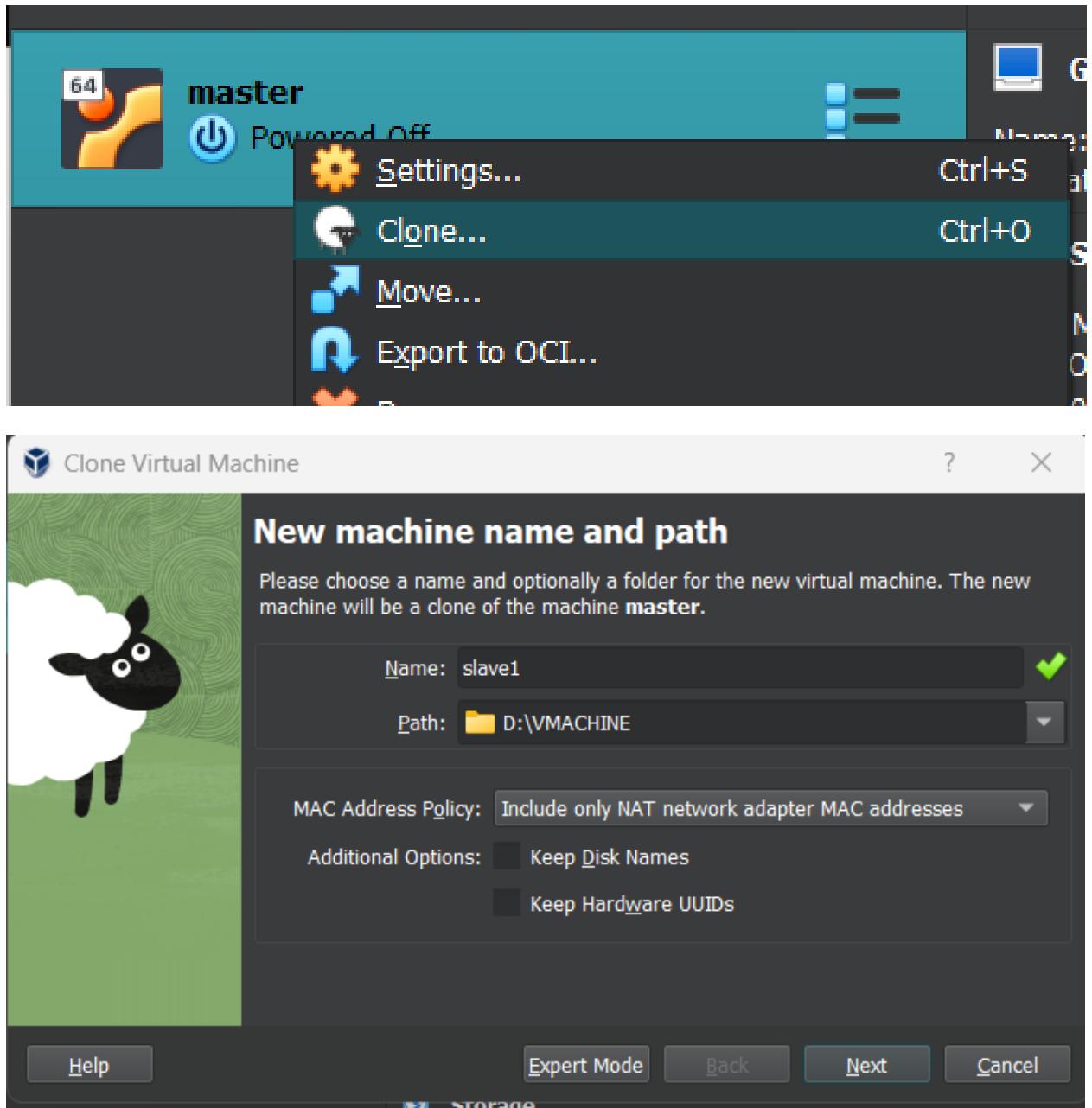
sudo nano yarn-site.xml

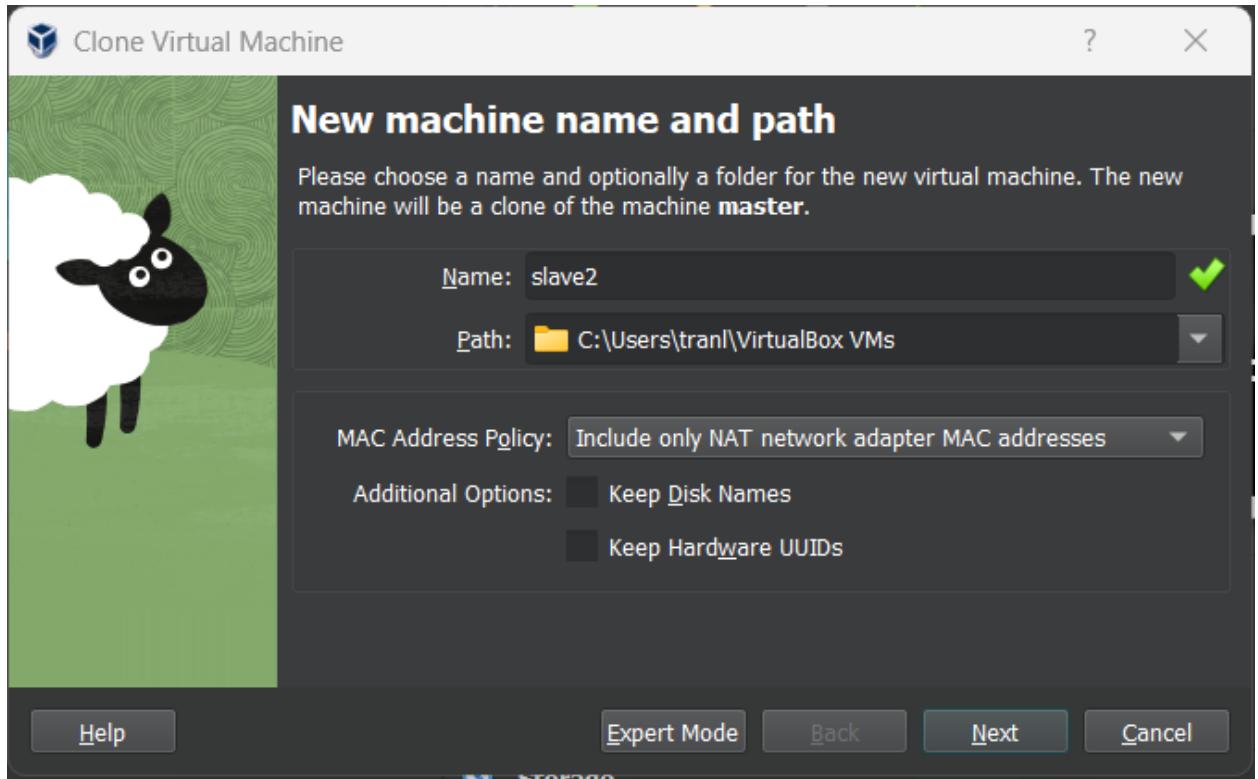
```
GNU nano 6.2                                     yarn-site.xml
<?xml version="1.0"?>
<configuration>
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>master</value>
  </property>
</configuration>
```

sudo nano workers

```
GNU nano 6.2                                     workers *
slave1
slave2_
```

- Thực hiện clone master thành 2 máy slave(sau khi đã cài hadoop) như sau:





- Thực hiện ở cả 2 máy slaves những bước sau:

Đăng nhập vào hadoopuser:

```
hadoopuser
Password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-91-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 System information as of Thu Dec 21 09:55:10 AM UTC 2023

 System load:  0.27294921875   Processes:          105
 Usage of /:   64.2% of 11.21GB  Users logged in:      0
 Memory usage: 10%              IPv4 address for enp0s3: 10.0.2.15
 Swap usage:   0%              IPv4 address for enp0s8: 192.168.56.50

Expanded Security Maintenance for Applications is not enabled.

44 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Thu Dec 21 09:18:08 UTC 2023 on tty1
hadoopuser@master:~$
```

Cấu hình file 01-netcfg.yaml:

Đề tài: *Tìm hiểu cơ chế phân tán trong Hadoop/HBase*

```
sudo nano /etc/netplan/01-netcfg.yaml
```

```
sudo netplan apply
```

Ở slave1:

```
GNU nano 6.2                               /etc/netplan/01-netcfg.yaml *
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      addresses: [192.168.56.51/24]
      dhcp4: false
```

Ở slave2:

```
GNU nano 6.2                               /etc/netplan/01-netcfg.yaml *
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      addresses: [192.168.56.52/24]
      dhcp4: false
```

Kiểm tra lại ip đã được chỉnh sửa:

```
sudo ip a
```

Ở slave1:

```
hadoopuser@master:~$ sudo netplan apply
hadoopuser@master:~$ sudo ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:c3:2a:4f brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 86398sec preferred_lft 86398sec
    inet6 fe80::a00:27ff:fec3:2a4f/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:28:91:bc brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.51/24 brd 192.168.56.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe28:91bc/64 scope link
        valid_lft forever preferred_lft forever
hadoopuser@master:~$ _
```

Ở slave2:

```
hadoopuser@master:~$ sudo netplan apply
hadoopuser@master:~$ sudo ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
0
    link/ether 08:00:27:c9:2a:4f brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 86399sec preferred_lft 86399sec
    inet6 fe80::a00:27ff:fec3:2a4f/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
0
    link/ether 08:00:27:37:4b:e6 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.52/24 brd 192.168.56.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe37:4be6/64 scope link
        valid_lft forever preferred_lft forever
hadoopuser@master:~$ _
```

Chỉnh sửa file hostname:

sudo nano /etc/hostname

Ở slave1:

```
GNU nano 6.2                               /etc/hostname *
slave1
```

Ở slave2:

```
GNU nano 6.2                               /etc/hostname *
slave2
```

Copy từ hadoop-cluster-install/worker qua:

cp hadoop-cluster-install/worker/ hadoop-3.3.2/etc/hadoop/*

```
hadoopuser@master:~$ cp hadoop-cluster-install/worker/* hadoop-3.3.2/etc/hadoop/
hadoopuser@master:~$ ls
hadoop-3.3.2  hadoop-3.3.2.tar.gz  hadoop-cluster-install
hadoopuser@master:~$ _
```

Ta thực hiện kiểm tra các file vừa copy:

*cd hadoop-3.3.2/etc/hadoop/
sudo nano hadoop-env.sh*

```
GNU nano 6.2                                     hadoop-env.sh *
##
## {yarn-env.sh|hdfs-env.sh} > hadoop-env.sh > hard-coded defaults
##
## {YARN_XYZ|HDFS_XYZ} > HADOOP_XYZ > hard-coded defaults
##

# Many of the options here are built from the perspective that users
# may want to provide OVERWRITING values on the command line.
# For example:
#
#JAVA_HOME="/usr/lib/jvm/java-8-openjdk-amd64/jre"
#
# Therefore, the vast majority (BUT NOT ALL!) of these defaults
# are configured for substitution and not append. If append
# is preferable, modify this file accordingly.
```

sudo nano core-site.xml

```
GNU nano 6.2                                     core-site.xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://master:9000</value>
  </property>
</configuration>
```

sudo nano hdfs-site.xml

```
GNU nano 6.2                                     hdfs-site.xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>dfs.name.dir</name>
    <value>${user.home}/bigdata/hd-data/nn</value>
  </property>
  <property>
    <name>dfs.replication</name>
    <value>2</value>
  </property>
</configuration>
```

sudo nano yarn-site.xml

```

GNU nano 6.2                               yarn-site.xml
<?xml version="1.0"?>
<configuration>
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>master</value>
  </property>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.nodemanager.local-dirs</name>
    <value>${user.home}/bigdata/hd-data/yarn/data</value>
  </property>
  <property>
    <name>yarn.nodemanager.logs-dirs</name>
    <value>${user.home}/bigdata/hd-data/yarn/logs</value>
  </property>
  <property>
    <name>yarn.nodemanager.disk-health-checker.max-disk-utilization-perdisk-percentage</name>
    <value>99.9</value>
  </property>
  <property>
    <name>yarn.nodemanager.vmem-check-enabled</name>
    <value>false</value>
  </property>
  <property>
    <name>yarn.nodemanager.env-whitelist</name>
    <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PREPEND_DISTDIR</value>
  </property>
</configuration>

```

- Mở 3 máy và thực hiện các lệnh sau (trên master):

ssh-keygen -t rsa -P “”

```

hadoopuser@master:~$ ssh-keygen -t rsa -P ""
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoopuser/.ssh/id_rsa):
Created directory '/home/hadoopuser/.ssh'.
Your identification has been saved in /home/hadoopuser/.ssh/id_rsa
Your public key has been saved in /home/hadoopuser/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:fG/KSsV1ZH/qn/nkBpQI5ojGQzn7IrJ4VQ1MNGHka70 hadoopuser@master
The key's randomart image is:
+---[RSA 3072]----+
|   =B. .   o   |
|   oo.+   oo .  |
|       .+ = + .o....|
|       oo....o. oo |
|       oo.oS + .. |
|       ..o ...o . ..|
|       . + .E..   o ...|
|       . o     . . o ++|
|       . .o     +=|
+---[SHA256]-----+
hadoopuser@master:~$ 

```

ssh-copy-id hadoopuser@master

```
hadoopuser@master:~$ ssh-copy-id hadoopuser@master
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/hadoopuser/.ssh/id_rsa.pub"
The authenticity of host 'master (192.168.56.50)' can't be established.
ED25519 key fingerprint is SHA256:+8bTegzz7VF4KeNyjNSlo/tnwcfbC+93Hsc4jCRKMOE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install all the new keys
hadoopuser@master's password:

Number of key(s) added: 1

Now try logging into the machine, with:    "ssh 'hadoopuser@master'"
and check to make sure that only the key(s) you wanted were added.

hadoopuser@master:~$
```

ssh-copy-id hadoopuser@slave1

```
hadoopuser@master:~$ ssh-copy-id hadoopuser@slave1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/hadoopuser/.ssh/id_rsa.pub"
The authenticity of host 'slave1 (192.168.56.51)' can't be established.
ED25519 key fingerprint is SHA256:+8bTegzz7VF4KeNyjNSlo/tnwcfbC+93Hsc4jCRKMOE.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install all the new keys
hadoopuser@slave1's password:

Number of key(s) added: 1

Now try logging into the machine, with:    "ssh 'hadoopuser@slave1'"
and check to make sure that only the key(s) you wanted were added.

hadoopuser@master:~$
```

ssh-copy-id hadoopuser@slave2

```
hadoopuser@master:~$ ssh-copy-id hadoopuser@slave2
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/hadoopuser/.ssh/id_rsa.pub"
The authenticity of host 'slave2 (192.168.56.52)' can't be established.
ED25519 key fingerprint is SHA256:+8bTegzz7VF4KeNyjNSlo/tnwcfbC+93Hsc4jCRKMOE.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:1: [hashed name]
  ~/.ssh/known_hosts:4: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install all the new keys
hadoopuser@slave2's password:

Number of key(s) added: 1

Now try logging into the machine, with:    "ssh 'hadoopuser@slave2'"
and check to make sure that only the key(s) you wanted were added.

hadoopuser@master:~$ _
```

- **Kiểm tra kết nối (trên master):**

ssh master

```
hadoopuser@master:~$ ssh master
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-91-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

 System information as of Thu Dec 21 10:32:32 AM UTC 2023

 System load: 0.0          Processes:          105
 Usage of /: 64.2% of 11.21GB  Users logged in:      1
 Memory usage: 11%          IPv4 address for enp0s3: 10.0.2.15
 Swap usage:  0%          IPv4 address for enp0s8: 192.168.56.50

Expanded Security Maintenance for Applications is not enabled.

44 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Thu Dec 21 10:00:30 2023
hadoopuser@master:~$
```

ssh slave1

```
hadoopuser@master:~$ ssh slave1
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-91-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

 System information as of Thu Dec 21 10:33:07 AM UTC 2023

 System load: 0.0          Processes:          107
 Usage of /: 64.2% of 11.21GB  Users logged in:      1
 Memory usage: 11%          IPv4 address for enp0s3: 10.0.2.15
 Swap usage:  0%          IPv4 address for enp0s8: 192.168.56.51

Expanded Security Maintenance for Applications is not enabled.

44 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Thu Dec 21 09:55:11 2023
hadoopuser@master:~$ _
```

ssh slave2

```
hadoopuser@master:~$ ssh slave2
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-91-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

 System information as of Thu Dec 21 10:33:43 AM UTC 2023

 System load: 0.0          Processes:           103
 Usage of /: 64.2% of 11.21GB  Users logged in:      1
 Memory usage: 11%          IPv4 address for enp0s3: 10.0.2.15
 Swap usage:  0%          IPv4 address for enp0s8: 192.168.56.52

Expanded Security Maintenance for Applications is not enabled.

44 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

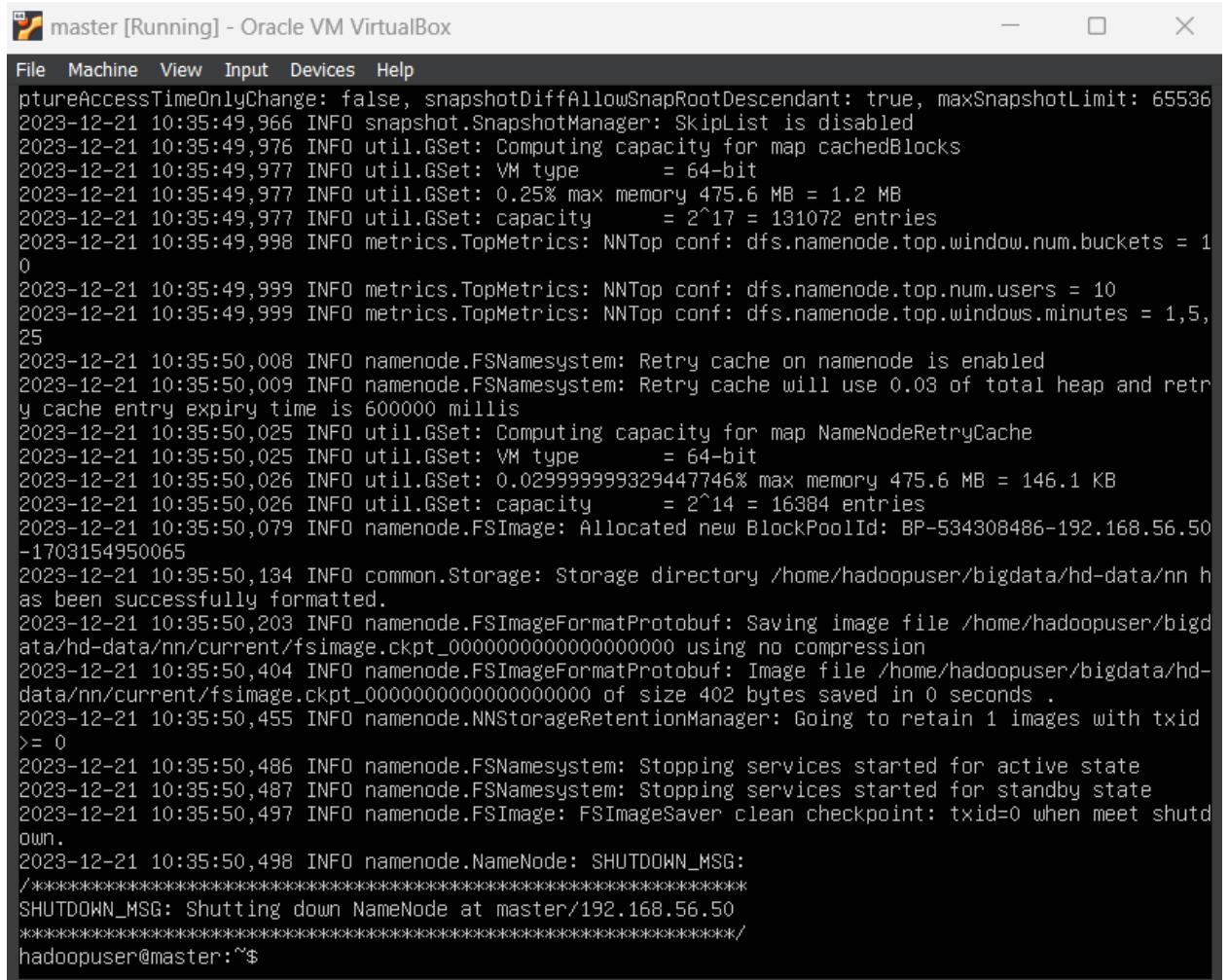
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Thu Dec 21 10:00:51 2023
hadoopuser@master:~$
```

- **Thực hiện các lệnh sau (trên master):**

cd ~

hdfs namenode -format



```

master [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
pictureAccessTimeOnlyChange: false, snapshotDiffAllowSnapRootDescendant: true, maxSnapshotLimit: 65536
2023-12-21 10:35:49,966 INFO snapshot.SnapshotManager: SkipList is disabled
2023-12-21 10:35:49,976 INFO util.GSet: Computing capacity for map cachedBlocks
2023-12-21 10:35:49,977 INFO util.GSet: VM type      = 64-bit
2023-12-21 10:35:49,977 INFO util.GSet: 0.25% max memory 475.6 MB = 1.2 MB
2023-12-21 10:35:49,977 INFO util.GSet: capacity      = 2^17 = 131072 entries
2023-12-21 10:35:49,998 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.window.num.buckets = 1
0
2023-12-21 10:35:49,999 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.num.users = 10
2023-12-21 10:35:49,999 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.windows.minutes = 1,5,
25
2023-12-21 10:35:50,008 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
2023-12-21 10:35:50,009 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retr
y cache entry expiry time is 600000 millis
2023-12-21 10:35:50,025 INFO util.GSet: Computing capacity for map NameNodeRetryCache
2023-12-21 10:35:50,025 INFO util.GSet: VM type      = 64-bit
2023-12-21 10:35:50,026 INFO util.GSet: 0.029999999329447746% max memory 475.6 MB = 146.1 KB
2023-12-21 10:35:50,026 INFO util.GSet: capacity      = 2^14 = 16384 entries
2023-12-21 10:35:50,079 INFO namenode.FSImage: Allocated new BlockPoolId: BP-534308486-192.168.56.50
-1703154950065
2023-12-21 10:35:50,134 INFO common.Storage: Storage directory /home/hadoopuser/bigdata/hd-data/nn h
as been successfully formatted.
2023-12-21 10:35:50,203 INFO namenode.FSImageFormatProtobuf: Saving image file /home/hadoopuser/bigd
ata/hd-data/nn/current/fsimage.ckpt_00000000000000000000 using no compression
2023-12-21 10:35:50,404 INFO namenode.FSImageFormatProtobuf: Image file /home/hadoopuser/bigdata/hd-
data/nn/current/fsimage.ckpt_00000000000000000000 of size 402 bytes saved in 0 seconds .
2023-12-21 10:35:50,455 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid
>= 0
2023-12-21 10:35:50,486 INFO namenode.FSNamesystem: Stopping services started for active state
2023-12-21 10:35:50,487 INFO namenode.FSNamesystem: Stopping services started for standby state
2023-12-21 10:35:50,497 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 when meet shutd
own.
2023-12-21 10:35:50,498 INFO namenode.NameNode: SHUTDOWN_MSG:
*****
SHUTDOWN_MSG: Shutting down NameNode at master/192.168.56.50
*****
hadoopuser@master:~$
```

ls

ls bigdata/

ls bigdata/hd-data/

ls bigdata/hd-data/nn

ls bigdata/hd-data/nn/current

```

hadoopuser@master:~$ ls
bigdata  hadoop-3.3.2  hadoop-3.3.2.tar.gz  hadoop-cluster-install
hadoopuser@master:~$ ls bigdata/
hd-data
hadoopuser@master:~$ ls bigdata/hd-data/
nn
hadoopuser@master:~$ ls bigdata/hd-data/nn
current
hadoopuser@master:~$ ls bigdata/hd-data/nn/current
fsimage_00000000000000000000  fsimage_00000000000000000000.md5  seen_txid  VERSION
hadoopuser@master:~$
```

- **Thực hiện khởi chạy hadoop (trên master):**

start-dfs.sh

start-yarn.sh

```
hadoopuser@master:~$ start-dfs.sh
Starting namenodes on [master]
Starting datanodes
Starting secondary namenodes [master]
hadoopuser@master:~$ start-yarn.sh
Starting resourcemanager
Starting nodemanagers
hadoopuser@master:~$ _
```

- **Kiểm tra quá trình hoàn tất (trên cả 3 máy):**

jps

Ở master:

```
hadoopuser@master:~$ jps
2610 SecondaryNameNode
2774 ResourceManager
3051 Jps
2396 NameNode
hadoopuser@master:~$ _
```

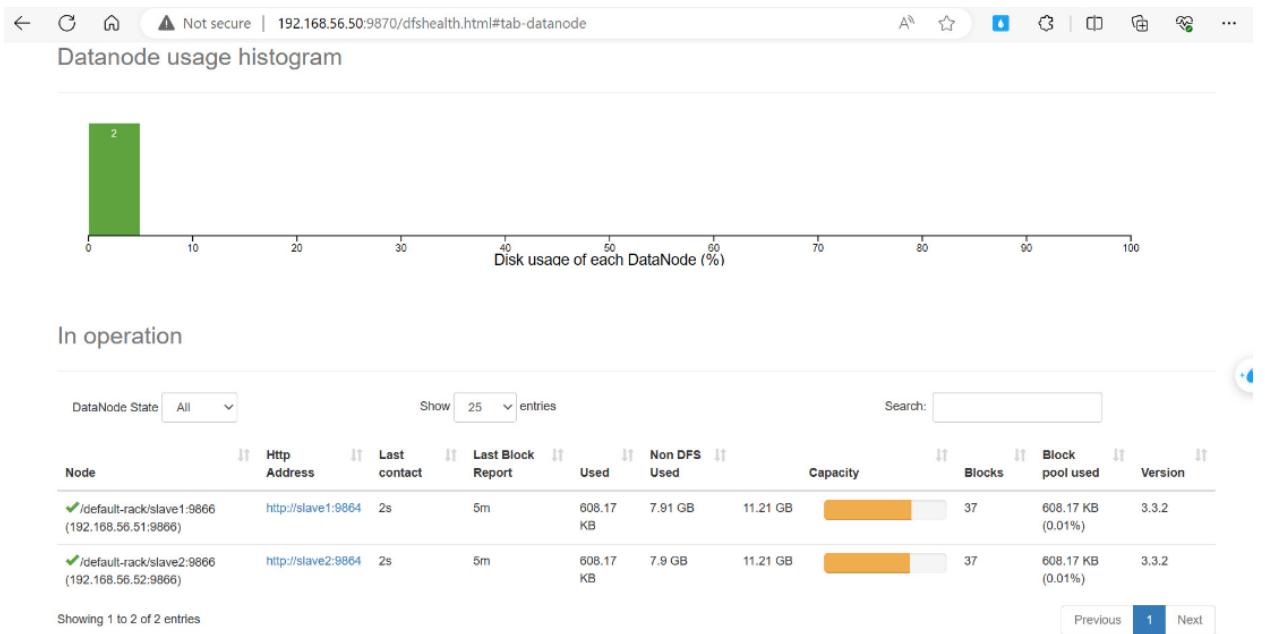
Ở slave1:

```
hadoopuser@master:~/hadoop-3.3.2/etc/hadoop$ jps
1810 Jps
1587 DataNode
1720 NodeManager
hadoopuser@master:~/hadoop-3.3.2/etc/hadoop$
```

Ở slave2:

```
hadoopuser@master:~/hadoop-3.3.2/etc/hadoop$ jps
1696 NodeManager
1786 Jps
1562 DataNode
hadoopuser@master:~/hadoop-3.3.2/etc/hadoop$ _
```

Kiểm tra trên Web UI của Hadoop



2.2.3 Cài đặt Zookeeper

- Thực hiện tải zookeeper và apache-storm về máy về máy (master):

wget

<https://archive.apache.org/dist/zookeeper/zookeeper-3.6.3/apache-zookeeper-3.6.3-bin.tar.gz>

wget

<https://archive.apache.org/dist/storm/apache-storm-2.2.1/apache-storm-2.2.1.tar.gz>

```

hadoopuser@master:~$ wget https://archive.apache.org/dist/zookeeper/zookeeper-3.6.3/apache-zookeeper-3.6.3-bin.tar.gz
--2023-12-21 10:48:44-- https://archive.apache.org/dist/zookeeper/zookeeper-3.6.3/apache-zookeeper-3.6.3-bin.tar.gz
Resolving archive.apache.org (archive.apache.org)... 65.108.204.189, 2a01:4f9:1a:a084::2
Connecting to archive.apache.org (archive.apache.org)|65.108.204.189|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 12516362 (12M) [application/x-gzip]
Saving to: 'apache-zookeeper-3.6.3-bin.tar.gz'

apache-zookeeper-3.6.3-b 100%[=====] 11.94M 2.25MB/s in 6.2s

2023-12-21 10:48:51 (1.93 MB/s) - 'apache-zookeeper-3.6.3-bin.tar.gz' saved [12516362/12516362]

hadoopuser@master:~$ 

```

Đề tài: Tìm hiểu cơ chế phân tán trong Hadoop/HBase

```
hadoopuser@master:~$ wget https://archive.apache.org/dist/storm/apache-storm-2.2.1/apache-storm-2.2.1.tar.gz
--2023-12-21 10:52:12-- https://archive.apache.org/dist/storm/apache-storm-2.2.1/apache-storm-2.2.1.tar.gz
Resolving archive.apache.org (archive.apache.org)... 65.108.204.189, 2a01:4f9:1a:a084::2
Connecting to archive.apache.org (archive.apache.org)|65.108.204.189|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 317302835 (303M) [application/x-gzip]
Saving to: 'apache-storm-2.2.1.tar.gz'

apache-storm-2.2.1.tar.gz 100%[=====] 302.60M 418KB/s in 5m 7s

2023-12-21 10:57:20 (1010 KB/s) - 'apache-storm-2.2.1.tar.gz' saved [317302835/317302835]

hadoopuser@master:~$ _
```

Thực hiện copy file zookeeper qua các máy slave (trên master)

```
scp apache-zookeeper-3.6.3-bin.tar.gz
hadoopuser@192.168.56.51:/home/hadoopuser
```

```
scp apache-zookeeper-3.6.3-bin.tar.gz
hadoopuser@192.168.56.52:/home/hadoopuser
```

```
hadoopuser@master:~$ scp apache-zookeeper-3.6.3-bin.tar.gz hadoopuser@192.168.56.51:/home/hadoopuser
The authenticity of host '192.168.56.51 (192.168.56.51)' can't be established.
ED25519 key fingerprint is SHA256:+8bTegzz7VF4KeNyjNS1o/tnwcfBc+93Hsc4jCRKM0E.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:1: [hashed name]
  ~/.ssh/known_hosts:4: [hashed name]
  ~/.ssh/known_hosts:5: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.51' (ED25519) to the list of known hosts.
apache-zookeeper-3.6.3-bin.tar.gz                                100%   12MB  29.6MB/s  00:00
hadoopuser@master:~$
```

```
hadoopuser@master:~$ scp apache-zookeeper-3.6.3-bin.tar.gz hadoopuser@192.168.56.52:/home/hadoopuser
The authenticity of host '192.168.56.52 (192.168.56.52)' can't be established.
ED25519 key fingerprint is SHA256:+8bTegzz7VF4KeNyjNS1o/tnwcfBc+93Hsc4jCRKM0E.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:1: [hashed name]
  ~/.ssh/known_hosts:4: [hashed name]
  ~/.ssh/known_hosts:5: [hashed name]
  ~/.ssh/known_hosts:6: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.52' (ED25519) to the list of known hosts.
apache-zookeeper-3.6.3-bin.tar.gz                                100%   12MB  27.2MB/s  00:00
hadoopuser@master:~$
```

Thực hiện copy file storm qua các máy slave (trên master):

```
scp apache-storm-2.2.1.tar.gz hadoopuser@192.168.56.51:/home/hadoopuser
```

```
scp apache-storm-2.2.1.tar.gz hadoopuser@192.168.56.51:/home/hadoopuser
```

```
hadoopuser@master:~$ scp apache-storm-2.2.1.tar.gz hadoopuser@192.168.56.51:/home/hadoopuser
apache-storm-2.2.1.tar.gz                                         100% 303MB 31.1MB/s  00:09
hadoopuser@master:~$ scp apache-storm-2.2.1.tar.gz hadoopuser@192.168.56.52:/home/hadoopuser
apache-storm-2.2.1.tar.gz                                         100% 303MB 31.0MB/s  00:09
hadoopuser@master:~$ _
```

Giải nén file zookeeper và move vào folder /usr/local/zookeeper (trên cả 3 máy):

```
tar -zxvf apache-zookeeper-3.6.3-bin.tar.gz
```

```
sudo mv apache-zookeeper-3.6.3-bin /usr/local/zookeeper
```

```
apache-zookeeper-3.6.3-bin/lib/audience-annotations-0.5.0.jar  
apache-zookeeper-3.6.3-bin/lib/zookeeper-3.6.3.jar  
apache-zookeeper-3.6.3-bin/lib/netty-handler-4.1.63.Final.jar  
apache-zookeeper-3.6.3-bin/lib/netty-common-4.1.63.Final.jar  
apache-zookeeper-3.6.3-bin/lib/netty-resolver-4.1.63.Final.jar  
apache-zookeeper-3.6.3-bin/lib/netty-buffer-4.1.63.Final.jar  
apache-zookeeper-3.6.3-bin/lib/netty-transport-4.1.63.Final.jar  
apache-zookeeper-3.6.3-bin/lib/netty-codec-4.1.63.Final.jar  
apache-zookeeper-3.6.3-bin/lib/netty-transport-native-epoll-4.1.63.Final.jar  
apache-zookeeper-3.6.3-bin/lib/netty-transport-native-unix-common-4.1.63.Final.jar  
apache-zookeeper-3.6.3-bin/lib/slf4j-api-1.7.25.jar  
apache-zookeeper-3.6.3-bin/lib/slf4j-log4j12-1.7.25.jar  
apache-zookeeper-3.6.3-bin/lib/log4j-1.2.17.jar  
apache-zookeeper-3.6.3-bin/lib/zookeeper-prometheus-metrics-3.6.3.jar  
apache-zookeeper-3.6.3-bin/lib/simpleclient-0.6.0.jar  
apache-zookeeper-3.6.3-bin/lib/simpleclient_hotspot-0.6.0.jar  
apache-zookeeper-3.6.3-bin/lib/simpleclient_servlet-0.6.0.jar  
apache-zookeeper-3.6.3-bin/lib/simpleclient_common-0.6.0.jar  
apache-zookeeper-3.6.3-bin/lib/commons-cli-1.2.jar  
apache-zookeeper-3.6.3-bin/lib/jetty-server-9.4.39.v20210325.jar  
apache-zookeeper-3.6.3-bin/lib/javax.servlet-api-3.1.0.jar  
apache-zookeeper-3.6.3-bin/lib/jetty-http-9.4.39.v20210325.jar  
apache-zookeeper-3.6.3-bin/lib/jetty-util-9.4.39.v20210325.jar  
apache-zookeeper-3.6.3-bin/lib/jetty-io-9.4.39.v20210325.jar  
apache-zookeeper-3.6.3-bin/lib/jetty-servlet-9.4.39.v20210325.jar  
apache-zookeeper-3.6.3-bin/lib/jetty-security-9.4.39.v20210325.jar  
apache-zookeeper-3.6.3-bin/lib/jetty-util-ajax-9.4.39.v20210325.jar  
apache-zookeeper-3.6.3-bin/lib/jackson-databind-2.10.5.1.jar  
apache-zookeeper-3.6.3-bin/lib/jackson-annotations-2.10.5.jar  
apache-zookeeper-3.6.3-bin/lib/jackson-core-2.10.5.jar  
apache-zookeeper-3.6.3-bin/lib/json-simple-1.1.1.jar  
apache-zookeeper-3.6.3-bin/lib/jline-2.14.6.jar  
apache-zookeeper-3.6.3-bin/lib/metrics-core-3.2.5.jar  
apache-zookeeper-3.6.3-bin/lib/snappy-Java-1.1.7.jar  
hadoopuser@master:~$ sudo mv apache-zookeeper-3.6.3-bin /usr/local/zookeeper  
[sudo] password for hadoopuser:  
hadoopuser@master:~$ _
```

Giải nén file storm và move vào folder /usr/local/storm (trên 3 máy):

```
tar -zxvf apache-storm-2.2.1.tar.gz
```

```
sudo mv apache-storm-2.2.1 /usr/local/storm
```

```
apache-storm-2.2.1/lib/httpclient-4.5.6.jar
apache-storm-2.2.1/lib/httpcore-4.4.10.jar
apache-storm-2.2.1/lib/commons-logging-1.2.jar
apache-storm-2.2.1/lib/commons-codec-1.11.jar
apache-storm-2.2.1/lib/commons-cli-1.4.jar
apache-storm-2.2.1/lib/guava-27.0.1-jre.jar
apache-storm-2.2.1/lib/failureaccess-1.0.1.jar
apache-storm-2.2.1/lib/listenablefuture-9999.0-empty-to-avoid-conflict-with-guava.jar
apache-storm-2.2.1/lib/jsr305-3.0.2.jar
apache-storm-2.2.1/lib/checker-qual-2.5.2.jar
apache-storm-2.2.1/lib/error_prone_annotations-2.2.0.jar
apache-storm-2.2.1/lib/j2objc-annotations-1.1.jar
apache-storm-2.2.1/lib/animal-sniffer-annotations-1.17.jar
apache-storm-2.2.1/lib/joda-time-2.3.jar
apache-storm-2.2.1/lib/jetty-server-9.4.14.v20181114.jar
apache-storm-2.2.1/lib/jetty-http-9.4.14.v20181114.jar
apache-storm-2.2.1/lib/jetty-util-9.4.14.v20181114.jar
apache-storm-2.2.1/lib/jetty-io-9.4.14.v20181114.jar
apache-storm-2.2.1/lib/jetty-servlet-9.4.14.v20181114.jar
apache-storm-2.2.1/lib/jetty-security-9.4.14.v20181114.jar
apache-storm-2.2.1/lib/jetty-servlets-9.4.14.v20181114.jar
apache-storm-2.2.1/lib/jetty-continuation-9.4.14.v20181114.jar
apache-storm-2.2.1/lib/jackson-core-2.9.8.jar
apache-storm-2.2.1/lib/jackson-dataformat-smile-2.9.8.jar
apache-storm-2.2.1/lib/commons-fileupload-1.3.3.jar
apache-storm-2.2.1/lib/hadoop-auth-2.8.5.jar
apache-storm-2.2.1/lib/nimbus-jose-jwt-4.41.1.jar
apache-storm-2.2.1/lib/jcip-annotations-1.0-1.jar
apache-storm-2.2.1/lib/json-smart-2.3.jar
apache-storm-2.2.1/lib/accessors-smart-1.2.jar
apache-storm-2.2.1/lib/zookeeper-3.4.14.jar
apache-storm-2.2.1/lib/audience-annotations-0.5.0.jar
apache-storm-2.2.1/lib/netty-3.10.6.Final.jar
apache-storm-2.2.1/lib/curator-framework-4.2.0.jar
apache-storm-2.2.1/lib/curator-client-4.2.0.jar
hadoopuser@master:~$ sudo mv apache-storm-2.2.1 /usr/local/storm
hadoopuser@master:~$
```

Chỉnh sửa file `~/.bashrc` (trên cả 3 máy):

```
sudo nano ~/.bashrc
```

Add những dòng sau:

```
export ZOOKEEPER_HOME=/usr/local/zookeeper
```

```
export PATH=$PATH:$ZOOKEEPER_HOME/bin
```

```
export STORM_HOME=/usr/local/storm
```

```
export PATH=$PATH:$STORM_HOME/bin
```

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre
export HADOOP_HOME=$HOME/hadoop-3.3.2
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export ZOOKEEPER_HOME=/usr/local/zookeeper
export PATH=$PATH:$ZOOKEEPER_HOME/bin
export STORM_HOME=/usr/local/storm
export PATH=$PATH:$STORM_HOME/bin
```

Đề tài: *Tìm hiểu cơ chế phân tán trong Hadoop/HBase*

Sau đó chạy `~/.bashrc` (trên cả 3 máy):

`source ~/.bashrc`

```
hadoopuser@master:~$ source ~/.bashrc
hadoopuser@master:~$ _
```

Di chuyển vào folder Zookeeper và tạo folder data và logs (trên cả 3 máy):

`cd $ZOOKEEPER_HOME/`

`mkdir data`

`mkdir logs`

```
hadoopuser@master:~$ source ~/.bashrc
hadoopuser@master:~$ cd $ZOOKEEPER_HOME/
hadoopuser@master:/usr/local/zookeeper$ mkdir data
hadoopuser@master:/usr/local/zookeeper$ mkdir logs
hadoopuser@master:/usr/local/zookeeper$ ls
bin  conf  data  docs  lib  LICENSE.txt  logs  NOTICE.txt  README.md  README_packaging.md
hadoopuser@master:/usr/local/zookeeper$ _
```

Di chuyển vào folder conf và thực hiện copy file `zoo_sample.cfg` (trên 3 máy)

`cd conf/`

`cp zoo_sample.cfg zoo.cfg`

```
hadoopuser@master:/usr/local/zookeeper$ cd conf/
hadoopuser@master:/usr/local/zookeeper/conf$ cp zoo_sample.cfg zoo.cfg
hadoopuser@master:/usr/local/zookeeper/conf$ _
```

Thực hiện chỉnh sửa file `zoo.cfg` (trên cả 3 máy):

`sudo nano zoo.cfg`

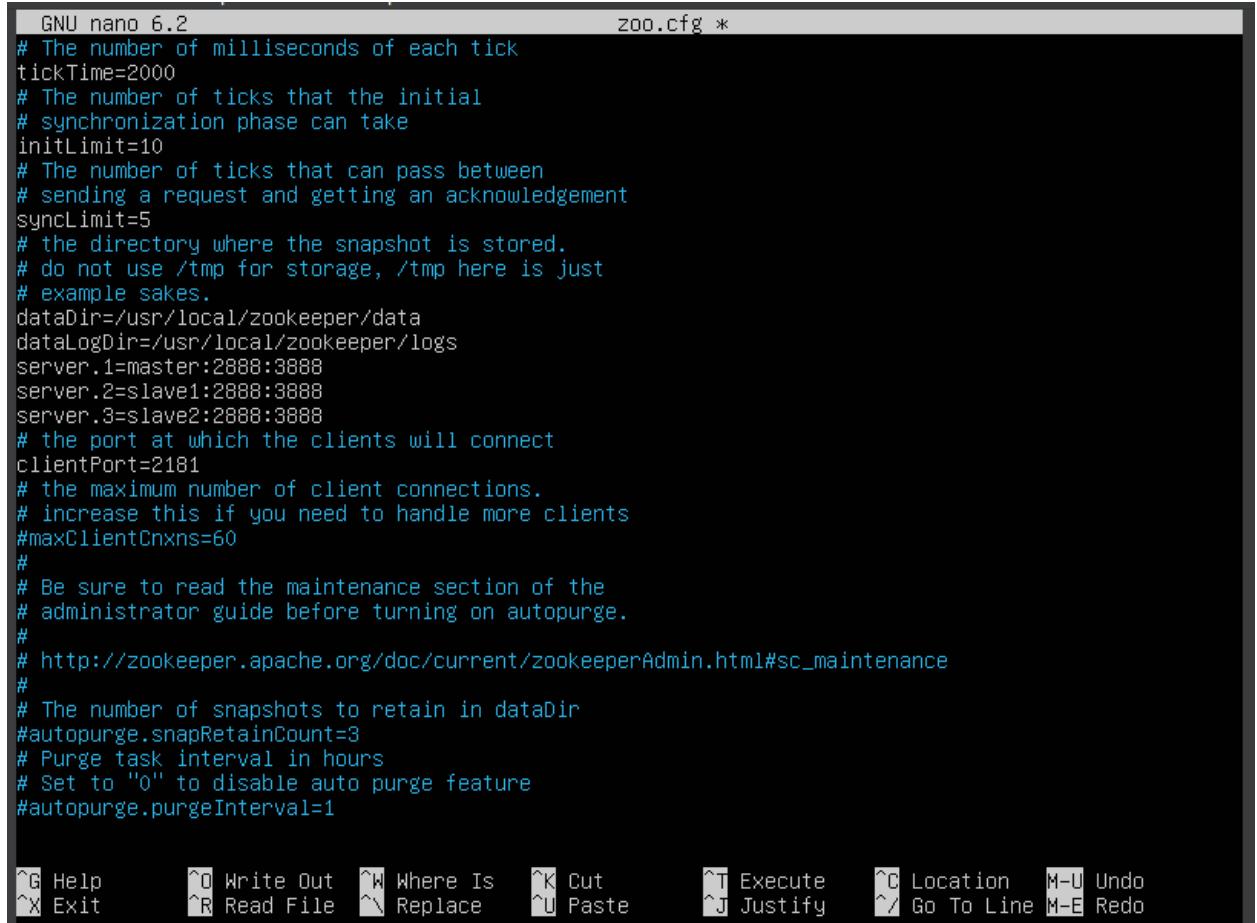
`dataDir=/usr/local/zookeeper/data`

`dataLogDir=/usr/local/zookeeper/logs`

`server.1=tnmaster:2888:3888`

`server.2=tnslave1:2888:3888`

`server.3=tnslave2:2888:3888`



```

GNU nano 6.2                                     zoo.cfg *
# The number of milliseconds of each tick
tickTime=2000
# The number of ticks that the initial
# synchronization phase can take
initLimit=10
# The number of ticks that can pass between
# sending a request and getting an acknowledgement
syncLimit=5
# the directory where the snapshot is stored.
# do not use /tmp for storage, /tmp here is just
# example sakes.
dataDir=/usr/local/zookeeper/data
dataLogDir=/usr/local/zookeeper/logs
server.1=master:2888:3888
server.2=slave1:2888:3888
server.3=slave2:2888:3888
# the port at which the clients will connect
clientPort=2181
# the maximum number of client connections.
# increase this if you need to handle more clients
#maxClientCnxns=60
#
# Be sure to read the maintenance section of the
# administrator guide before turning on autopurge.
#
# http://zookeeper.apache.org/doc/current/zookeeperAdmin.html#sc_maintenance
#
# The number of snapshots to retain in dataDir
#autopurge.snapRetainCount=3
# Purge task interval in hours
# Set to "0" to disable auto purge feature
#autopurge.purgeInterval=1

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute   ^D Location  M-U Undo
^X Exit      ^R Read File  ^N Replace    ^U Paste      ^J Justify   ^Y Go To Line M-B Redo

```

Thực hiện chỉnh sửa myid (trên cả 3 máy):

sudo nano /usr/local/zookeeper/data/myid

Ở master:

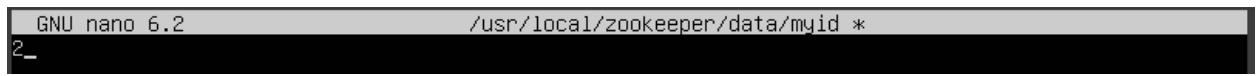


```

GNU nano 6.2                                     /usr/local/zookeeper/data/myid *
1

```

Ở slave1:



```

GNU nano 6.2                                     /usr/local/zookeeper/data/myid *
2-

```

Ở slave2:



```

GNU nano 6.2                                     /usr/local/zookeeper/data/myid *
3

```

Thực hiện chạy start zookeeper và kiểm tra tiến trình hoàn tất (trên 3 máy):

Đề tài: Tìm hiểu cơ chế phân tán trong Hadoop/HBase

zkServer.sh start

jps

Ở master:

```
hadoopuser@master:~$ zkServer.sh start
ZooKeeper JMX enabled by default
Using config: /usr/local/zookeeper/bin/..../conf/zoo.cfg
Starting zookeeper ... STARTED
hadoopuser@master:~$ jps
2610 SecondaryNameNode
3251 QuorumPeerMain
2774 ResourceManager
3288 Jps
2396 NameNode
hadoopuser@master:~$ _
```

Ở slave1:

```
hadoopuser@master:~$ zkServer.sh start
ZooKeeper JMX enabled by default
Using config: /usr/local/zookeeper/bin/..../conf/zoo.cfg
Starting zookeeper ... STARTED
hadoopuser@master:~$ jps
1587 DataNode
2088 Jps
1720 NodeManager
2073 QuorumPeerMain
hadoopuser@master:~$ _
```

Ở slave2:

```
hadoopuser@master:~$ zkServer.sh start
ZooKeeper JMX enabled by default
Using config: /usr/local/zookeeper/bin/..../conf/zoo.cfg
Starting zookeeper ... STARTED
hadoopuser@master:~$ jps
1696 NodeManager
2035 QuorumPeerMain
1562 DataNode
2062 Jps
hadoopuser@master:~$ _
```

2.2.4 Cài đặt HBase

Thực hiện download Hbase (trên master):

wget <https://archive.apache.org/dist/hbase/2.4.0/hbase-2.4.0-bin.tar.gz>

```
hadoopuser@master:~$ wget https://archive.apache.org/dist/hbase/2.4.0/hbase-2.4.0-bin.tar.gz
--2023-12-22 05:12:12-- https://archive.apache.org/dist/hbase/2.4.0/hbase-2.4.0-bin.tar.gz
Resolving archive.apache.org (archive.apache.org)... 65.108.204.189, 2a01:4f9:1a:a084::2
Connecting to archive.apache.org (archive.apache.org)|65.108.204.189|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 275211928 (262M) [application/x-gzip]
Saving to: 'hbase-2.4.0-bin.tar.gz'

hbase-2.4.0-bin.tar.gz    100%[=====] 262.46M 1.29MB/s   in 6m 12s

2023-12-22 05:18:24 (723 KB/s) - 'hbase-2.4.0-bin.tar.gz' saved [275211928/275211928]
```

Thực hiện copy qua các máy slave (trên master):

```
scp hbase-2.4.0-bin.tar.gz hadoopuser@192.168.56.51:/home/hadoopuser
```

```
scp hbase-2.4.0-bin.tar.gz hadoopuser@192.168.56.52:/home/hadoopuser
```

```
hadoopuser@master:~$ scp hbase-2.4.0-bin.tar.gz hadoopuser@192.168.56.51:/home/hadoopuser
hbase-2.4.0-bin.tar.gz                                100% 262MB 41.2MB/s 00:06
hadoopuser@master:~$ scp hbase-2.4.0-bin.tar.gz hadoopuser@192.168.56.52:/home/hadoopuser
hbase-2.4.0-bin.tar.gz                                100% 262MB 43.9MB/s 00:05
hadoopuser@master:~$
```

Giải nén file hbase và move vào folder /usr/local/hbase (trên cả 3 máy):

```
sudo tar -zxvf hbase-2.4.0-bin.tar.gz
```

```
sudo mv hbase-2.4.0 /usr/local/hbase
```

```

hbase-2.4.0/lib/jdk11/management-api-3.2.1.jar
hbase-2.4.0/lib/jdk11/pfl-basic-4.0.1.jar
hbase-2.4.0/lib/jdk11/pfl-tf-4.0.1.jar
hbase-2.4.0/lib/jdk11/pfl-asm-4.0.1.jar
hbase-2.4.0/lib/jdk11/pfl-dynamic-4.0.1.jar
hbase-2.4.0/lib/jdk11/pfl-basic-tools-4.0.1.jar
hbase-2.4.0/lib/jdk11/pfl-tf-tools-4.0.1.jar
hbase-2.4.0/lib/jdk11/stax-ex-1.8.1.jar
hbase-2.4.0/lib/jdk11/streambuffer-1.5.7.jar
hbase-2.4.0/lib/jdk11/mimepull-1.9.11.jar
hbase-2.4.0/lib/jdk11/FastInfoSet-1.2.16.jar
hbase-2.4.0/lib/jdk11/ha-api-3.1.12.jar
hbase-2.4.0/lib/jdk11/saaj-impl-1.5.1.jar
hbase-2.4.0/lib/jdk11/jakarta.activation-api-1.2.1.jar
hbase-2.4.0/lib/jdk11/jaxws-tools-2.3.2.jar
hbase-2.4.0/lib/jdk11/jaxb-xjc-2.3.2.jar
hbase-2.4.0/lib/jdk11/jaxb-jxc-2.3.2.jar
hbase-2.4.0/lib/jdk11/jaxws-eclipselink-plugin-2.3.2.jar
hbase-2.4.0/lib/jdk11/jakarta.mail-api-1.6.3.jar
hbase-2.4.0/lib/jdk11/jakarta.persistence-api-2.2.2.jar
hbase-2.4.0/lib/jdk11/org.eclipse.persistence.moxy-2.7.4.jar
hbase-2.4.0/lib/jdk11/org.eclipse.persistence.core-2.7.4.jar
hbase-2.4.0/lib/jdk11/org.eclipse.persistence.asm-2.7.4.jar
hbase-2.4.0/lib/jdk11/sdo-eclipselink-plugin-2.3.2.jar
hbase-2.4.0/lib/jdk11/org.eclipse.persistence.sdo-2.7.4.jar
hbase-2.4.0/lib/jdk11/commonj.sdo-2.1.1.jar
hbase-2.4.0/lib/jdk11/release-documentation-2.3.2-docbook.zip
hbase-2.4.0/lib/jdk11/samples-2.3.2.zip
hbase-2.4.0/lib/jdk11/jakarta.xml.ws-api-2.3.2.jar
hbase-2.4.0/lib/jdk11/jakarta.xml.bind-api-2.3.2.jar
hbase-2.4.0/lib/jdk11/jakarta.xml.soap-api-1.4.1.jar
hbase-2.4.0/lib/jdk11/jakarta.jws-api-1.1.1.jar
hadoopuser@master:~$ ls
apache-storm-2.2.1.tar.gz      bigdata      hadoop-3.3.2.tar.gz    hbase-2.4.0
apache-zookeeper-3.6.3-bin.tar.gz hadoop-3.3.2  hadoop-cluster-install  hbase-2.4.0-bin.tar.gz
hadoopuser@master:~$ sudo mv hbase-2.4.0 /usr/local/hbase
hadoopuser@master:~$
```

Thêm các dòng sau vào file .bashrc (trên cả 3 máy):

sudo nano ~/.bashrc

```
export HBASE_HOME=/usr/local/hbase
export PATH=$PATH:$HBASE_HOME/bin
```

```

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre
export HADOOP_HOME=$HOME/hadoop-3.3.2
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export ZOOKEEPER_HOME=/usr/local/zookeeper
export PATH=$PATH:$ZOOKEEPER_HOME/bin
export STORM_HOME=/usr/local/storm
export PATH=$PATH:$STORM_HOME/bin
export HBASE_HOME=/usr/local/hbase
export PATH=$PATH:$HBASE_HOME/bin_
```

$\wedge G$ Help $\wedge O$ Write Out $\wedge W$ Where Is $\wedge K$ Cut $\wedge T$ Execute $\wedge C$ Location $M-U$ Undo
 $\wedge X$ Exit $\wedge R$ Read File $\wedge N$ Replace $\wedge U$ Paste $\wedge J$ Justify $\wedge /$ Go To Line $M-E$ Redo

Sau đó chạy lệnh source ~/bashrc

Đề tài: Tìm hiểu cơ chế phân tán trong Hadoop/HBase

```
hadoopuser@master:~$ source ~/.bashrc  
hadoopuser@master:~$
```

Thực hiện chỉnh sửa hbase-env.sh (trên cả 3 máy)

sudo nano /usr/local/hbase/conf/hbase-env.sh

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre
```

```
export HBASE_PID_DIR=/usr/local/hbase/pids
```

```
export HBASE_MANAGES_ZK=false
```

```
# Override text processing tools for use by these launch scripts.  
# export GREP="\${GREP-grep}"  
# export SED="\${SED-sed}"  
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre  
export HBASE_PID_DIR=/usr/local/hbase/pids  
export HBASE_MANAGES_ZK=false
```

^G Help ^D Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^A Replace ^U Paste ^J Justify ^Z Go To Line M-E Redo

Chỉnh sửa file hbase-site.xml (trên master):

sudo nano /usr/local/hbase/conf/hbase-site.xml

```
GNU nano 6.2          /usr/local/hbase/conf/hbase-site.xml *
HBase will refuse to run in such an environment. Setting
`hbase.unsafe.stream.capability.enforce` to `false` overrides this behavior,
permitting operation. This configuration is for the developer workstation
only and __should not be used in production!__

See also https://hbase.apache.org/book.html#standalone_dist
-->
<property>
  <name>hbase.rootdir</name>
  <value>hdfs://master:9000/hbase</value>
</property>
<property>
  <name>hbase.master.info.port</name>
  <value>60010</value>
</property>
<property>
  <name>hbase.cluster.distributed</name>
  <value>true</value>
</property>
<property>
  <name>hbase.zookeeper.quorum</name>
  <value>master,slave1,slave2</value>
</property>
<property>
  <name>hbase.tmp.dir</name>
  <value>/usr/local/hbase/pids</value>
</property>
<property>
  <name>hbase.zookeeper.property.dataDir</name>
  <value>/usr/local/zookeeper</value>
</property>
</configuration>

[ Soft wrapping of overlong lines enabled ]
^G Help      ^D Write Out   ^W Where Is   ^K Cut       ^T Execute   ^C Location   M-U Undo
^X Exit      ^R Read File   ^Y Replace    ^U Paste     ^J Justify   ^Z Go To Line M-E Redo
```

Chỉnh sửa file hbase-site.xml (trên slave1, slave2):

sudo nano /usr/local/hbase/conf/hbase-site.xml

```

slave1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
GNU nano 6.2          /usr/local/hbase/conf/hbase-site.xml *
<!--
The following properties are set for running HBase as a single process on a
developer workstation. With this configuration, HBase is running in
"stand-alone" mode and without a distributed file system. In this mode, and
without further configuration, HBase and ZooKeeper data are stored on the
local filesystem, in a path under the value configured for `hbase.tmp.dir`.
This value is overridden from its default value of `/tmp` because many
systems clean `/tmp` on a regular basis. Instead, it points to a path within
this HBase installation directory.

Running against the `LocalFileSystem`, as opposed to a distributed
filesystem, runs the risk of data integrity issues and data loss. Normally
HBase will refuse to run in such an environment. Setting
`hbase.unsafe.stream.capability.enforce` to `false` overrides this behavior,
permitting operation. This configuration is for the developer workstation
only and __should not be used in production!__

See also https://hbase.apache.org/book.html#standalone\_dist
-->
<property>
  <name>hbase.cluster.distributed</name>
  <value>true</value>
</property>
<property>
  <name>hbase.rootdir</name>
  <value>hdfs://master:9000/hbase</value>
</property>
<property>
  <name>hbase.master.info.port</name>
  <value>60010</value>
</property>
</configuration>

^G Help      ^O Write Out  ^W Where Is  ^K Cut      ^T Execute  ^C Location  M-U Undo
^X Exit      ^R Read File  ^Y Replace   ^U Paste    ^J Justify   ^V Go To Line M-E Redo

```

Thực hiện chỉnh sửa file regionservers (trên master):

`sudo nano /usr/local/hbase/conf/regionservers`

```

GNU nano 6.2          /usr/local/hbase/conf/regionservers *
slave1
slave2

```

Thực hiện khởi động Hbase (trên master):

`start-all.sh` (*khởi động hadoop cluster nếu chưa khởi động*)

`zkServer.sh start` (*khởi động zookeeper ở cả 3 máy nếu chưa khởi động*)

`start-hbase.sh` (*khởi động hbase*)

```
hadoopuser@master:~$ start-hbase.sh
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hadoopuser/hadoop-3.3.2/share/hadoop/common/lib/slf4j-log4j1
2-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.30
.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
running master, logging to /usr/local/hbase/logs/hbase-hadoopuser-master-master.out
slave1: running regionserver, logging to /usr/local/hbase/bin/../logs/hbase-hadoopuser-regionserver-
slave1.out
slave2: running regionserver, logging to /usr/local/hbase/bin/../logs/hbase-hadoopuser-regionserver-
slave2.out
hadoopuser@master:~$
```

Kiểm tra quá trình hoàn tất (trên 3 máy):

jps

Ở master:

```
hadoopuser@master:~$ jps
1511 ResourceManager
1115 NameNode
1804 QuorumPeerMain
1341 SecondaryNameNode
2013 HMaster
2253 Jps
hadoopuser@master:~$
```

Ở slave1:

```
hadoopuser@slave1:~$ jps
1576 Jps
1032 DataNode
1305 QuorumPeerMain
1178 NodeManager
1423 HRegionServer
hadoopuser@slave1:~$
```

Ở slave2:

```
hadoopuser@slave2:~$ jps
1588 Jps
1301 QuorumPeerMain
1174 NodeManager
1435 HRegionServer
1037 DataNode
hadoopuser@slave2:~$ _
```

hbase shell (ở master)

```

hadoopuser@master:~$ hbase shell
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hadoopuser/hadoop-3.3.2/share/hadoop/common/lib/slf4j-log4j1
2-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.30
.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.4.0, r282ab70012ae843af54a6779543ff20acbcb629, Thu Dec 3 09:58:52 PST 2020
Took 0.0028 seconds
hbase:001:0> _

```

Kiểm tra active trên web UI HBase

The screenshot shows the Apache HBase master status interface. At the top, there's a navigation bar with links like Home, Table Details, Procedures & Locks, HBCK Report, Process Metrics, Local Logs, Log Level, Debug Dump, Metrics Dump, and Profiler. Below the navigation bar, it says "HBase Configuration". The main content area has two sections: "Region Servers" and "Backup Masters".

Region Servers:

Base Stats	Memory	Requests	Storefiles	Compactions	Replications
ServerName	Start time	Last contact	Version	Requests Per Second	Num. Regions
slave1,16020,1703171226694	2023-12-21T15:07:06.694Z	1 s	2.4.0	0	0
slave2,16020,1703171226207	2023-12-21T15:07:06.207Z	1 s	2.4.0	0	2
Total:2				0	2

Backup Masters:

ServerName	Port	Start Time
slave1	16000	Thu Dec 21 15:07:11 UTC 2023

Chương 3: THỰC NGHIỆM MÔ PHỎNG PHÂN TÁN

3.1. MÔ TẢ BÀI TOÁN ĐẶT RA VỚI DỮ LIỆU

CSDL Quản lý hệ thống cửa hàng điện tử TechnoShop:

Hệ thống cửa hàng điện tử TechnoShop có 3 chi nhánh, với trụ sở chính nằm ở TP. Hồ Chí Minh, 2 chi nhánh còn lại ở Khánh Hòa và Hà Nội.

Các chức năng của hệ thống:

Đề tài: Tìm hiểu cơ chế phân tán trong Hadoop/HBase

- Đăng nhập hệ thống
- Quản lý danh mục cửa hàng
- Quản lý danh sách khách hàng
- Quản lý danh sách nhân viên
- Quản lý bán hàng
- Thống kê doanh thu

Lý do phân tán dữ liệu:

- Tăng hiệu suất và khả năng mở rộng: Phân tán dữ liệu cho phép dữ liệu được lưu trữ và xử lý trên nhiều nút máy chủ, giúp tăng hiệu suất và khả năng mở rộng của hệ thống. Khi dữ liệu được phân tán, các tác vụ xử lý có thể được chia nhỏ và phân phối trên nhiều nút, giúp tối ưu hóa tài nguyên và giảm thời gian xử lý.
- Đơn vị có nhiều cửa hàng nằm ở các tỉnh thành khác nhau trong cả nước, có nhu cầu trao đổi và xử lý thông tin giữa các cửa hàng.
- Trong thực tế, có cửa hàng được phân tán khắp nơi, trong khi đó, dữ liệu quản lý ngày càng lớn và phục vụ cho đa người dùng phân tán, vì vậy CSDL phân tán là con đường thích hợp nhất.
- Tối ưu hóa việc truy cập dữ liệu: Phân tán dữ liệu có thể giúp tối ưu hóa việc truy cập dữ liệu bằng cách đặt dữ liệu gần người dùng hoặc ứng dụng sử dụng nó, giúp giảm độ trễ và tăng tốc độ truy cập.
- Tính linh hoạt: Phân tán dữ liệu cho phép hệ thống mở rộng dễ dàng bằng cách thêm nút mới vào cụm, giúp tăng khả năng chịu tải và đáp ứng nhu cầu mở rộng của ứng dụng.
- Cần kết nối các CSDL có sẵn: CSDL phân tán là giải pháp tự nhiên khi có các CSDL đang tồn tại và sự cần thiết xây dựng một ứng dụng toàn cục.
- Giảm chi phí truyền thông: Tăng ứng dụng cục bộ làm giảm chi phí truyền thông.
- Nâng cao hiệu suất: Có cơ chế xử lý song song và phân mảnh dữ liệu theo ứng dụng làm cực đại hóa tính cục bộ của ứng dụng.

- Tăng độ tin cậy: Phân tán dữ liệu giúp tăng độ tin cậy của hệ thống. Khi dữ liệu được sao chép và phân phối trên nhiều nút, nếu một nút gặp sự cố, dữ liệu vẫn có thể được truy cập từ các nút khác, giúp giảm thiểu nguy cơ mất mát dữ liệu.

3.2. MÔ TẢ CẤU TRÚC DỮ LIỆU SỬ DỤNG

Database TechnoShop

CUAHANG (MaCH, TenCH, SDT, DiaDiem, ChiNhanh)

Tân từ: Mỗi cửa hàng có mã cửa hàng (MaCH) dùng để phân biệt các cửa hàng với nhau, ngoài ra còn lưu tên cửa hàng (TenCH), số điện thoại của cửa hàng đó (SDT), địa điểm cửa hàng(DiaDiem), chi nhánh cửa hàng hoạt động (ChiNhanh). Trường mã cửa hàng, tên cửa hàng, số điện thoại là duy nhất.

KHACHHANG (MaKH, TenKH, DiaChi, GT, SDT, DiemTichLuy)

Tân từ: Mỗi khách hàng có mã khách hàng (MaKH) dùng để phân biệt với các khách hàng khác, ngoài ra còn có lưu tên khách hàng (TenKH), địa chỉ để giao hàng (DiaChi), giới tính của khách hàng (GioiTinh), số điện thoại (SDT) và số điểm tích lũy dựa vào số lần mua hàng (DiemTichLuy).

NHANVIEN (MaNV, TenNV, NgSinh, NgayVL, SDT, GioiTinh)

Tân từ: Mỗi nhân viên có mã nhân viên (MaNV) để phân biệt với các nhân viên khác. Ngoài ra còn có lưu tên nhân viên (TenNV), ngày sinh (NgaySinh), ngày vào làm (NgayVL), số điện thoại liên lạc (DienThoai) và giới tính của nhân viên (GioiTinh).

SANPHAM (MaSP, TenSP, LoaiSP, Gia, ThuongHieu)

Tân từ: Mỗi sản phẩm có mã sản phẩm (MaSP) là duy nhất, ngoài ra còn có lưu tên sản phẩm, tên loại sản phẩm (LoaiSP), giá tiền (Gia) và thương hiệu của sản phẩm đó (ThuongHieu).

HOADON (MaHD, MaKH, MaCH, MaNV, NgayHD, ThanhTien)

Tân từ: Mỗi hóa đơn sẽ có một mã hóa đơn (MaHD) là duy nhất, ngoài ra còn lưu thông tin ngày lập hóa đơn (NgayHD), mã khách hàng mua sản phẩm (MaKH), mã cửa hàng nơi sản phẩm được bán (MaCH), mã nhân viên lập hóa đơn (MaNV) và tổng thành tiền của hóa đơn đó (ThanhTien).

CTHD (SoHD, MaSP, SoLuong)

Tân từ: Chi tiết hóa đơn lưu giữ thông tin sản phẩm và số lượng mà hóa đơn sở hữu nó có. Thông tin bao gồm số hóa đơn (SoHD), mã sản phẩm (MaSP), số lượng sản phẩm được mua (SoLuong).

Dữ liệu mẫu ở các bảng

- **Bảng CUAHANG**

RowID	ThongTinCH		DiaChi		
	MaCH	TenCH	SDT	DiaDiem	ChiNhanh
CH01	TechnoShop HCM	0389417014	Khu pho 6, Phuong Linh Trung, Thu Duc, TP HCM		Ho Chi Minh
CH02	TechnoShop HaNoi	0951761491	213 P. Thai Ha, Lang Ha, Dong Da, Ha Noi		Ha Noi
CH03	TechnoShop KhanhHoa	0780141751	Ninh Dien, Ninh Tho, Ninh Hoa, Khanh Hoa		Khanh Hoa

- **Bảng KHACHHANG**

RowID	ThongTinKH				TichLuy
	MaKH	TenKH	DiaChi	GT	
					DiemTichLuy

KH01	Đoàn Ngọc Tuấn	P. Linh Chiểu, TP.Thủ Đức - TP. Hồ Chí Minh	Nam	0987164192	590
KH02	Doãn Công Trí	Phường 6, Quận 3, TP, Hồ Chí Minh	Nam	0983615411	268
KH03	Trần Quốc Hưng	Phường Bến Nghé, Quận 1, TP. Hồ Chí Minh	Nam	0356719331	1190
KH04	Trần Lê Tú	Huyện Hóc Môn, TP. Hồ Chí Minh	Nam	0782561511	873

- Bảng SANPHAM

RowID	ThongTinSP		ChiTiet		
	MaSP	TenSP	ThuongHieu	Gia	LoaiSP
SP01	Apple iPad 10.2-inch (9th Gen) Wi-Fi, 2021	Apple		6990000	Điện thoại
SP02	Điện thoại Samsung Galaxy S23 5G	Samsung		6600000	Điện thoại
SP03	Điện Thoại Oppo A57 4GB/128GB - Hàng Chính Hãng	Oppo		4500000	Điện thoại
SP04	MacBook Air M1 13 inch 2020	Apple		18900000	Laptop

- Bảng NHANVIEN

RowID	TTCaNhan					TTLamViec
	MaNV	TenNV	NgaySinh	SDT	GioiTinh	
NV01	Nguyễn Thùy Duyên		05/09/2000	0397172623	Nu	12/04/2023
NV02	Trần Lê Thúy Anh		04/07/1997	0326786536	Nu	06/11/2023
NV03	Huỳnh Khắc Nam		12/01/2003	0982544013	Nam	01/01/2023

- Bảng HOADON

RowID	DoiTuongLK				ChiTiet	
	MaHD	MaKH	MaCH	MaNV	NgayHD	ThanhTien
HD01		KH01	CH01	NV01	10/12/2022	445000
HD02		KH02	CH03	NV02	10/12/2022	2250000
HD03		KH03	CH02	NV03	10/12/2022	250000

- Bảng CTHD

RowID	SoHD	MaSP	SoLuong
1	HD01	SP01	1
2	HD02	SP02	6
3	HD01	SP03	3

4	HD02	SP04	2
---	------	------	---

3.3. CÁC BƯỚC THỰC NGHIỆM

3.3.1 Tạo Table và các column family

Câu lệnh:

```
create '<table name>', '<column family>'
```

- Tạo các table ở máy **master**:

```

hadoopuser@master:~$ hbase shell
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hadoopuser/hadoop-3.3.2/share/hadoop/common/lib/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.4.0, r282ab70012ae843af54a6779543ff20acbb629, Thu Dec 3 09:58:52 PST 2020
Took 0.0049 seconds
hbase:001:0> create 'CUAHANG', 'ThongTinCH', 'DiaChi'
Created table CUAHANG
Took 3.9392 seconds
=> Hbase::Table - CUAHANG
hbase:002:0> create 'SANPHAM', 'ThongTinSP', 'ChiTiet'
Created table SANPHAM
Took 2.3256 seconds
=> Hbase::Table - SANPHAM
hbase:003:0> create 'NV', 'TTCaNhan', 'TTLamViec'
Created table NV
Took 2.3123 seconds
=> Hbase::Table - NV
hbase:004:0> create 'KHACHHANG', 'ThonTinKH', 'TichLuy'
Created table KHACHHANG
Took 2.2988 seconds
=> Hbase::Table - KHACHHANG
hbase:005:0> create 'HOADON', 'DoiTuongLK', 'ChiTiet'
Created table HOADON
Took 2.2776 seconds
=> Hbase::Table - HOADON
hbase:006:0> create 'CTHD', 'SoHD', 'MaSP', 'SoLuong'
Created table CTHD
Took 2.2625 seconds
=> Hbase::Table - CTHD
hbase:007:0> █

```

- Qua 2 máy slave kiểm tra ta thấy các table đã được cập nhật qua 2 máy này:

```

Did you mean? caller
hbase:002:0> list
TABLE
CTHD
CUAHANG
HOADON
KHACHHANG
NV
SANPHAM
6 row(s)
Took 1.7140 seconds
=> ["CTHD", "CUAHANG", "HOADON", "KHACHHANG", "NV", "SANPHAM"]
hbase:003:0> █

```

```
Took 0.0074 seconds
hbase:001:0> list
TABLE
CTHD
CUAHANG
HOADON
KHACHHANG
NV
SANPHAM
6 row(s)
Took 2.4319 seconds
=> ["CTHD", "CUAHANG", "HOADON", "KHACHHANG", "NV", "SANPHAM"]
hbase:002:0> █
```

Ta có thể kiểm tra thông tin table trên Web UI:

Table	Description
CTHD	'CTHD', {NAME => 'MaSP', BLOOMFILTER => 'ROW', IN_MEMORY => 'false', VERSIONS => '1', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', COMPRESSION => 'NONE', TTL => 'FOREVER', MIN VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}, {NAME => 'SoHD', BLOOMFILTER => 'ROW', IN_MEMORY => 'false', VERSIONS => '1', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', COMPRESSION => 'NONE', TTL => 'FOREVER', MIN VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}, {NAME => 'SoLuong', BLOOMFILTER => 'ROW', IN_MEMORY => 'false', VERSIONS => '1', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', COMPRESSION => 'NONE', TTL => 'FOREVER', MIN VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
CUAHANG	'CUAHANG', {NAME => 'DiaChi', BLOOMFILTER => 'ROW', IN_MEMORY => 'false', VERSIONS => '1', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', COMPRESSION => 'NONE', TTL => 'FOREVER', MIN VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}, {NAME => 'ThongTinCH', BLOOMFILTER => 'ROW', IN_MEMORY => 'false', VERSIONS => '1', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', COMPRESSION => 'NONE', TTL => 'FOREVER', MIN VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
HOADON	'HOADON', {NAME => 'ChiTiet', BLOOMFILTER => 'ROW', IN_MEMORY => 'false', VERSIONS => '1', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', COMPRESSION => 'NONE', TTL => 'FOREVER', MIN VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}, {NAME => 'DoiTuongLK', BLOOMFILTER => 'ROW', IN_MEMORY => 'false', VERSIONS => '1', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', COMPRESSION => 'NONE', TTL => 'FOREVER', MIN VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
KHACHHANG	'KHACHHANG', {NAME => 'ThonTinKH', BLOOMFILTER => 'ROW', IN_MEMORY => 'false', VERSIONS => '1', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', COMPRESSION => 'NONE', TTL => 'FOREVER', MIN VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}, {NAME => 'TichLuy', BLOOMFILTER => 'ROW', IN_MEMORY => 'false', VERSIONS => '1', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', COMPRESSION => 'NONE', TTL => 'FOREVER', MIN VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
NV	'NV', {NAME => 'TTCaNhan', BLOOMFILTER => 'ROW', IN_MEMORY => 'false', VERSIONS => '1', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', COMPRESSION => 'NONE', TTL => 'FOREVER', MIN VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}, {NAME => 'TTLaM Viec', BLOOMFILTER => 'ROW', IN_MEMORY => 'false', VERSIONS => '1', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', COMPRESSION => 'NONE', TTL => 'FOREVER', MIN VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}

3.3.2 Thêm dữ liệu

Câu lệnh:

```
put '<table name>','row1','<colfamily:colname>','<value>'
```

Đề tài: Tìm hiểu cơ chế phân tán trong Hadoop/HBase

- Thêm dữ liệu trên máy master:

```

pacce
hbase:008:0> put 'CUAHANG', 'CH01', 'ThongTinCH: TenCH', 'TechnoShop HCM'
Took 0.7191 seconds
hbase:009:0> scan 'CUAHANG'
ROW                                COLUMN+CELL
  CH01                             column=ThongTinCH: TenCH, timestamp=2023-12-21T15:28:58.438, value=
                                         TechnoShop HCM
1 row(s)
Took 0.2840 seconds
hbase:010:0> █

```

- Kiểm tra trên 2 máy slave:

```

For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.4.0, r282ab70012ae843af54a6779543ff20acbcbb629, Thu Dec 3 09:58:52 PST 2020
Took 0.0061 seconds
hbase:001:0> clear
Traceback (most recent call last):
NameError: undefined local variable or method `clear' for #<HBaseReceiver:0x6009cd34>
Did you mean? caller
hbase:002:0> list
TABLE
CTHD
CUAHANG
HOADON
KHACHHANG
NV
SANPHAM
6 row(s)
Took 1.7140 seconds
=> ["CTHD", "CUAHANG", "HOADON", "KHACHHANG", "NV", "SANPHAM"]
hbase:003:0> scan 'CUAHANG'
ROW                                COLUMN+CELL
  CH01                             column=ThongTinCH: TenCH, timestamp=2023-12-21T15:28:58.438, value=
                                         TechnoShop HCM
1 row(s)
Took 0.6122 seconds
hbase:004:0>

SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.4.0, r282ab70012ae843af54a6779543ff20acbcbb629, Thu Dec 3 09:58:52 PST 2020
Took 0.0074 seconds
hbase:001:0> list
TABLE
CTHD
CUAHANG
HOADON
KHACHHANG
NV
SANPHAM
6 row(s)
Took 2.4319 seconds
=> ["CTHD", "CUAHANG", "HOADON", "KHACHHANG", "NV", "SANPHAM"]
hbase:002:0> scan 'CUAHANG'
ROW                                COLUMN+CELL
  CH01                             column=ThongTinCH: TenCH, timestamp=2023-12-21T15:28:58.438, value=
                                         TechnoShop HCM
1 row(s)
Took 0.5840 seconds
hbase:003:0>

```

3.3.3 Cập nhật dữ liệu

Câu lệnh:

```
put 'table name','row ','Column family:column name','new value'
```

- Sửa dữ liệu trên máy slave1:

Ta thấy, ban đầu sản phẩm SP10 có giá là 6990000, thực hiện sửa giá SP10 thành 9000000.

```
=> 699000
hbase:004:0> put 'SANPHAM', 'SP01', 'ChiTiet:Gia', '9000000'
Took 0.1671 seconds
hbase:005:0> ■
```

Kết quả sau khi sửa giá (giá được cập nhật lại 1000000 ở cả 3 máy):

<pre>hbase:152:0> hbase:153:0> put 'HOADON', 'HD03', 'DoiTuongLK:KhachHang', 'KH03' Took 0.0193 seconds hbase:154:0> put 'HOADON', 'HD03', 'ChiTiet:ChiTietHD', 'CH02' Took 0.0244 seconds hbase:155:0> put 'HOADON', 'HD03', 'ChiTiet:MaNV', 'NV03' Took 0.0173 seconds hbase:156:0> put 'HOADON', 'HD03', 'ChiTiet:NgayHD', '10/12/2022' Took 0.0202 seconds hbase:157:0> put 'HOADON', 'HD03', 'ChiTiet:ThanhTien', '250000' Took 0.0198 seconds hbase:158:0> put 'CTHD', '1', 'SoHD:SoHD', 'HD01' Took 0.0495 seconds hbase:159:0> put 'CTHD', '1', 'MaSP:MaSP', 'SP01' Took 0.0536 seconds hbase:160:0> put 'CTHD', '1', 'SoLuong:SoLuong', '1' Took 0.0202 seconds hbase:161:0> hbase:162:0> put 'CTHD', '2', 'SoHD:SoHD', 'HD02' Took 0.0178 seconds hbase:163:0> put 'CTHD', '2', 'MaSP:MaSP', 'SP02' Took 0.0176 seconds hbase:164:0> put 'CTHD', '2', 'SoLuong:SoLuong', '6' Took 0.0193 seconds hbase:165:0> hbase:166:0> put 'CTHD', '3', 'SoHD:SoHD', 'HD01' Took 0.0173 seconds hbase:167:0> put 'CTHD', '3', 'MaSP:MaSP', 'SP03' Took 0.0174 seconds hbase:168:0> put 'CTHD', '3', 'SoLuong:SoLuong', '3' Took 0.0193 seconds hbase:169:0> hbase:170:0> put 'CTHD', '4', 'SoHD:SoHD', 'HD02' Took 0.0294 seconds hbase:171:0> put 'CTHD', '4', 'MaSP:MaSP', 'SP04' Took 0.0223 seconds hbase:172:0> put 'CTHD', '4', 'SoLuong:SoLuong', '2' Took 0.0151 seconds hbase:173:0> scan 'SANPHAM', {FILTER => "PrefixFilter('SP01')"} ROW COLUMN+CELL SP01 column=ChiTiet:LoaiSP, timestamp=2023-12-21T15:48:14.444, value=Lap top SP04 column=ThongTinSP:TenSP, timestamp=2023-12-21T15:48:14.141, value=M acBook Air M1 13 inch 2020 SP04 column=ThongTinSP:ThuongHieu, timestamp=2023-12-21T15:48:14.239, va lue=Apple 4 row(s) Took 0.3591 seconds hbase:003:0> 699000 => 699000 hbase:004:0> put 'SANPHAM', 'SP01', 'ChiTiet:Gia', '9000000' Took 0.1671 seconds hbase:005:0> scan 'SANPHAM', {FILTER => "PrefixFilter('SP01')"} ROW COLUMN+CELL SP01 column=ChiTiet:Gia, timestamp=2023-12-21T16:11:48.298, value=9000000 0 SP01 column=ChiTiet:LoaiSP, timestamp=2023-12-21T15:48:13.093, value=\xC 4\x90i\xE1\xB8\x87n tho\xE1\xBA\xA1 SP01 column=ThongTinSP:TenSP, timestamp=2023-12-21T15:48:12.777, value=A pple iPad 10.2-inch (9th Gen) Wi-Fi, 2021 SP01 column=ThongTinSP:ThuongHieu, timestamp=2023-12-21T15:48:12.880, va lue=Apple 1 row(s) Took 0.1141 seconds hbase:006:0> ■</pre>	<pre>SP04 column=ChiTiet:LoaiSP, timestamp=2023-12-21T15:48:14.444, value=Lap top SP04 column=ThongTinSP:TenSP, timestamp=2023-12-21T15:48:14.141, value=M acBook Air M1 13 inch 2020 SP04 column=ThongTinSP:ThuongHieu, timestamp=2023-12-21T15:48:14.239, va lue=Apple 4 row(s) Took 0.3591 seconds hbase:003:0> 699000 => 699000 hbase:004:0> put 'SANPHAM', 'SP01', 'ChiTiet:Gia', '9000000' Took 0.1671 seconds hbase:005:0> scan 'SANPHAM', {FILTER => "PrefixFilter('SP01')"} ROW COLUMN+CELL SP01 column=ChiTiet:Gia, timestamp=2023-12-21T16:11:48.298, value=9000000 0 SP01 column=ChiTiet:LoaiSP, timestamp=2023-12-21T15:48:13.093, value=\xC 4\x90i\xE1\xB8\x87n tho\xE1\xBA\xA1 SP01 column=ThongTinSP:TenSP, timestamp=2023-12-21T15:48:12.777, value=A pple iPad 10.2-inch (9th Gen) Wi-Fi, 2021 SP01 column=ThongTinSP:ThuongHieu, timestamp=2023-12-21T15:48:12.880, va lue=Apple 1 row(s) Took 0.1141 seconds hbase:006:0> ■</pre>
---	--

Đề tài: Tìm hiểu cơ chế phân tán trong Hadoop/HBase

3.3.4 Xóa dữ liệu

Câu lệnh:

```
delete '<table name>', '<row>', '<column name >'
```

- Thực hiện xóa, ta có kết quả:

```
Took 0.1971 seconds
hbase:174:0> delete 'SANPHAM', 'SP01', 'ThongTinSP:ThuongHieu'
Took 0.0634 seconds
hbase:175:0> scan 'SANPHAM', {FILTER => "PrefixFilter('SP01')"}
ROW                                COLUMN+CELL
  SP01                             column=ChiTiet:Gia, timestamp=2023-12-21T16:11:48.298, value=900000
                                      0
  SP01                             column=ChiTiet:LoaiSP, timestamp=2023-12-21T15:48:13.093, value=\xC
                                      4\x90i\xE1\xBB\x87n tho\xE1\xBA\xA1i
  SP01                             column=ThongTinSP:TenSP, timestamp=2023-12-21T15:48:12.777, value=A
                                      pple iPad 10.2-inch (9th Gen) Wi-Fi, 2021
1 row(s)
Took 0.0469 seconds
hbase:176:0> █
```

- Câu lệnh xóa tất cả các giá trị của sản phẩm SP01

```
ppcc 1.1.0 10.2 inch (9th Gen) Wi-Fi, 2021
1 row(s)
Took 0.0469 seconds
hbase:176:0> deleteall 'SANPHAM', 'SP01'
Took 0.0225 seconds
hbase:177:0> scan 'SANPHAM', {FILTER => "PrefixFilter('SP01')"}
ROW                                COLUMN+CELL
  0 row(s)
Took 0.0220 seconds
hbase:178:0> █
```

3.3.5 Các thao tác với dữ liệu khác

- Count

Đếm số hàng của một bảng:

```

hbase:178:0> count 'SANPHAM'
3 row(s)
Took 0.0768 seconds
=> 3
hbase:179:0> scan 'SANPHAM'
ROW                                COLUMN+CELL
  SP02                             column=ChiTiet:Gia, timestamp=2023-12-21T15:48:13.402, value=660000
                                      0
  SP02                             column=ChiTiet:LoaiSP, timestamp=2023-12-21T15:48:13.480, value=\xC
                                      4\x90i\xE1\xBB\x87n tho\xE1\xBA\xA1i
  SP02                             column=ThongTinSP:TenSP, timestamp=2023-12-21T15:48:13.220, value=\
                                      xC4\x90i\xE1\xBB\x87n tho\xE1\xBA\xA1i Samsung Galaxy S23 5G
  SP02                             column=ThongTinSP:ThuongHieu, timestamp=2023-12-21T15:48:13.310, va
                                      lue=Samsung
  SP03                             column=ChiTiet:Gia, timestamp=2023-12-21T15:48:13.879, value=450000
                                      0
  SP03                             column=ChiTiet:LoaiSP, timestamp=2023-12-21T15:48:14.003, value=\xC
                                      4\x90i\xE1\xBB\x87n tho\xE1\xBA\xA1i
  SP03                             column=ThongTinSP:TenSP, timestamp=2023-12-21T15:48:13.609, value=\
                                      xC4\x90i\xE1\xBB\x87n Tho\xE1\xBA\xA1i Oppo A57 4GB/128GB - H\xC3\x
                                      A0ng Ch\xC3\xADnh H\xC3\xA3ng
  SP03                             column=ThongTinSP:ThuongHieu, timestamp=2023-12-21T15:48:13.721, va
                                      lue=Oppo
  SP04                             column=ChiTiet:Gia, timestamp=2023-12-21T15:48:14.329, value=189000
                                      00
  SP04                             column=ChiTiet:LoaiSP, timestamp=2023-12-21T15:48:14.444, value=Lap
                                      top
  SP04                             column=ThongTinSP:TenSP, timestamp=2023-12-21T15:48:14.141, value=M
                                      acBook Air M1 13 inch 2020
  SP04                             column=ThongTinSP:ThuongHieu, timestamp=2023-12-21T15:48:14.239, va
                                      lue=Apple
3 row(s)
Took 0.1708 seconds
hbase:180:0> ■

```

- **KeyOnlyFilter**

Bộ lọc sẽ chỉ trả về các giá trị key trong cặp key-value (value sẽ được viết lại thành trống). Bộ lọc này có thể được sử dụng để lấy tất cả các key mà không cần phải lấy cả các value.

```
hbase:180:0> scan 'SANPHAM', {FILTER => "KeyOnlyFilter()"}
ROW          COLUMN+CELL
SP02        column=ChiTiet:Gia, timestamp=2023-12-21T15:48:13.402, value=
SP02        column=ChiTiet:LoaiSP, timestamp=2023-12-21T15:48:13.480, value=
SP02        column=ThongTinSP:TenSP, timestamp=2023-12-21T15:48:13.220, value=
SP02        column=ThongTinSP:ThuongHieu, timestamp=2023-12-21T15:48:13.310, va
lue=
SP03        column=ChiTiet:Gia, timestamp=2023-12-21T15:48:13.879, value=
SP03        column=ChiTiet:LoaiSP, timestamp=2023-12-21T15:48:14.003, value=
SP03        column=ThongTinSP:TenSP, timestamp=2023-12-21T15:48:13.609, value=
SP03        column=ThongTinSP:ThuongHieu, timestamp=2023-12-21T15:48:13.721, va
lue=
SP04        column=ChiTiet:Gia, timestamp=2023-12-21T15:48:14.329, value=
SP04        column=ChiTiet:LoaiSP, timestamp=2023-12-21T15:48:14.444, value=
SP04        column=ThongTinSP:TenSP, timestamp=2023-12-21T15:48:14.141, value=
SP04        column=ThongTinSP:ThuongHieu, timestamp=2023-12-21T15:48:14.239, va
lue=
3 row(s)
Took 0.1309 seconds
hbase:181:0>
```

- **PrefixFilter**

Dùng để lọc các record dựa trên RowKey

```
hbase:181:0> scan 'SANPHAM', {FILTER => "PrefixFilter('SP03')"}
ROW          COLUMN+CELL
SP03        column=ChiTiet:Gia, timestamp=2023-12-21T15:48:13.879, value=450000
0
SP03        column=ChiTiet:LoaiSP, timestamp=2023-12-21T15:48:14.003, value=\xC
4\x90i\xE1\xBB\x87n tho\xE1\xBA\xA1i
SP03        column=ThongTinSP:TenSP, timestamp=2023-12-21T15:48:13.609, value=\
xC4\x90i\xE1\xBB\x87n Tho\xE1\xBA\xA1i Oppo A57 4GB/128GB - H\xC3\x
A0ng Ch\xC3\xADnh H\xC3\xA3ng
SP03        column=ThongTinSP:ThuongHieu, timestamp=2023-12-21T15:48:13.721, va
lue=oppo
1 row(s)
Took 0.0459 seconds
hbase:182:0>
```

- **ColumnPrefixFilter**

Bộ lọc này được sử dụng để chỉ chọn những key có cột khớp với một tiền tố cụ thể. Dưới đây ta có tiền tố là ‘T’ nên các dòng có tên cột bắt đầu bằng ‘T’ được lọc ra, đó là cột **TenKH**.

```
Took 0.0609 seconds
hbase:200:0> scan 'KHACHHANG', {FILTER => "ColumnPrefixFilter('T')"}
ROW
  KH01
    COLUMN+CELL
      column=ThongTinKH:TenKH, timestamp=2023-12-21T16:01:37.346, value=\x
      xC4\x90o\xC3\xA0n Ng\xE1\xBB\x8Dc Tu\xE1\xBA\xA5n
  KH02
    COLUMN+CELL
      column=ThongTinKH:TenKH, timestamp=2023-12-21T16:01:37.783, value=D
      o\xC3\xA3n C\xC3\xB4ng Tr\xC3\xAD
  KH03
    COLUMN+CELL
      column=ThongTinKH:TenKH, timestamp=2023-12-21T16:01:38.216, value=T
      r\xE1\xBA\xA7n Qu\xE1\xBB\x91c H\xC6\xB0ng
  KH04
    COLUMN+CELL
      column=ThongTinKH:TenKH, timestamp=2023-12-21T16:01:38.647, value=T
      r\xE1\xBA\xA7n L\xC3\xAA T\xE1\xBB\xA9
4 row(s)
Took 0.0427 seconds
hbase:201:0>
```

● **MutipleColumnPrefixFilter**

Nó hoạt động giống như bộ lọc ColumnPrefixFilter nhưng cho phép chỉ định nhiều tiền tố.

```
hbase:182:0> scan 'KHACHHANG', {FILTER => "MultipleColumnPrefixFilter('T')"}
ROW
  KH01
    COLUMN+CELL
      column=ThongTinKH:TenKH, timestamp=2023-12-21T16:01:37.346, value=\x
      xC4\x90o\xC3\xA0n Ng\xE1\xBB\x8Dc Tu\xE1\xBA\xA5n
  KH02
    COLUMN+CELL
      column=ThongTinKH:TenKH, timestamp=2023-12-21T16:01:37.783, value=D
      o\xC3\xA3n C\xC3\xB4ng Tr\xC3\xAD
  KH03
    COLUMN+CELL
      column=ThongTinKH:TenKH, timestamp=2023-12-21T16:01:38.216, value=T
      r\xE1\xBA\xA7n Qu\xE1\xBB\x91c H\xC6\xB0ng
  KH04
    COLUMN+CELL
      column=ThongTinKH:TenKH, timestamp=2023-12-21T16:01:38.647, value=T
      r\xE1\xBA\xA7n L\xC3\xAA T\xE1\xBB\xA9
4 row(s)
Took 0.3331 seconds
hbase:183:0>
```

● **ValueFilter**

Dùng để lọc các records theo value đưa ra

```
Took 0.0254 seconds
hbase:196:0> scan 'KHACHHANG', {COLUMNS => 'ThongTinKH:TenKH', FILTER =>
hbase:197:1* "ValueFilter(=,'binary:Đoãn Công Trí')"}
ROW
    KH02
        COLUMN+CELL
            column=ThongTinKH:TenKH, timestamp=2023-12-21T16:01:37.783, value=D
o\xC3\xA3n C\xC3\xB4ng Tr\xC3\xAD
1 row(s)
Took 0.0334 seconds
hbase:198:0> █
```

TÀI LIỆU THAM KHẢO

1. <https://itnavi.com.vn/blog/hbase-la-gi#e752>
2. <https://blog.vietnamlab.vn/tong-quan-ve-hbase/>
3. <https://viblo.asia/p/hbase-overview-architecture-va-data-flow-63vKj6J6K2R>
4. <https://www.oreilly.com/library/view/hbase-the-definitive/9781449314682/ch01.html>
5. <https://www.simplilearn.com/tutorials/hadoop-tutorial/hbase#:~:text=Back%20in%20November%202006%2C%20Google,of%20Hadoop%20in%20January%2008.>
6. <https://fr.slideshare.net/tuanbv/hbase-29536077>
7. https://youtu.be/--lax6MCs6k?si=rbnw2z_6e0BwFtwL
8. <https://www.youtube.com/watch?v=PEL0e51oeaE>
9. https://youtu.be/c2Lg5c8v4YQ?si=tQHaz4h2LW_LYhhB
10. https://blog.csdn.net/qq_45811072/article/details/121693142