Logo, company name

Description automatically generated **BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC SƯ PHẠM KĨ THUẬT TP.HCM  
KHOA CÔNG NGHỆ THÔNG TIN**

**PHÂN TÍCH YÊU CẦU, CHỨC NĂNG ĐỒ ÁN CUỐI KỲ**

TÌM HIỂU PARQUET VÀ VIẾT ỨNG DỤNG DEMO

**MÃ MÔN HỌC: CLCO332779\_21\_1\_09  
HỌC KỲ 1 – NĂM HỌC 2022-2023  
Thực hiện: Nhóm 08.   
Giảng viên hướng dẫn: GV. Huỳnh Xuân Phụng**

***Thành phố Hồ Chí Minh, Tháng 10 năm 2022***

DANH SÁCH NHÓM

1. Mã lớp môn học: CLCO332779\_21\_1\_092. Giảng viên hướng dẫn: GV. Huỳnh Xuân Phụng  
3. Tên đề tài: Tìm hiểu Parquet và viết ứng dụng demo

4. Danh sách nhóm viết tiểu luận cuối kỳ:

|  |  |  |
| --- | --- | --- |
| **STT** | **HỌ VÀ TÊN SINH VIÊN** | **Mã số sinh viên** |
| 1 | Nguyễn Công Thành | 20110267 |
| 2 | Trần Văn Kiệt | 20112665 |
| 3 | Nguyễn Đình Thiện | 20110263 |

- Trưởng nhóm: Nguyễn Công Thành

1. **Giới thiệu về Parquet**

Chắc hẳn, trong chúng ta ai cũng từng làm việc với tệp CSV. Và cũng rất nhiều người trong chúng ta nghĩ rằng: “CSV là một giải pháp hữu hiệu và tối ưu”.

Nhưng CSV được định dạng theo hàng, đều này có nghĩa khi muốn sử dụng chúng,

bạn phải truy vấn vào từng hàng. Việc này tưởng chừng như rất nhanh và đơn giản, nhưng nó lại khá chậm và khó lưu trữ kết quả.

Parquet - là một định dạng lưu trữ dữ liệu theo cột

Đối với các tệp CSV hoặc cơ sở dữ liệu SQL thông thường, khi đọc điều này có nghĩa là chọn các hàng cụ thể trong số tất cả dữ liệu. Tuy nhiên, đối với cơ sở dữ liệu cột, điều này có nghĩa là chọn các cột cụ thể để đọc. Khi đọc một phần rất nhỏ các cột, giả sử < 10 trên 100, tỷ lệ tốc độ đọc sẽ lớn bằng > 50, tức là **với** Parquet **nhận được tốc độ tăng gấp 50 lần so với việc đọc tệp CSV thông thường**

Tệp CSV dễ dang sử dụng mà không cần qua một bước kiểm tra, xác nhận nào. Đều này sẽ không đảm bảo tính bảo mật và an toàn nếu bạn sử dụng nó làm cơ sở dữ liệu.

Nhưng điều làm người ta chọn parquet nhiều hơn là vì bài toán tiết kiệm chi phí. Nếu bạn sử dụng Parquet - là một định dạng lưu trữ dữ liệu theo cột sẽ có sự khác biệt rất lớn về kích thước nếu các tập dữ liệu giống hệt nhau. Ở mức độ doanh nghiệp thì khi họ triển khai 1 dự án thì họ có thể sẽ lưu trữ dữ liệu trên hệ thống đám mây, làm cơ sở cho các ứng dụng web 🡪 Do đó, các nhà cung cấp dịch vụ đám mây sẽ tính phí dựa trên lượng dữ liệu được quét và lượng dữ liệu được lưu trữ.

Bạn hãy xem giá của dịch vụ Amazon S3 sau đây:

Table

Description automatically generated

Bảng 1 - Giá Amazon S3 với các định dạng dữ liệu khác nhau

Qua bản dữ liệu bên trên. Chắc chắn, bạn sẽ thấy rằng: "Tệp Parquet sử dụng ít dung lượng ổ đĩa hơn nhiều sơ với CSV, thời gian quét nhanh hơn. Và nếu bạn có một tập dữ liệu giống hệt nhau thì sẽ rẻ hơn 16 lần khi lưu trữ dữ liệu ở định dạng Parquet.

Bây giờ ta xem đến Lưu trữ Amazon S3 và so sánh giá truy vấn cho các định dạng dữ liệu khác nhau

Table

Description automatically generated

Bảng 2 - Lưu trữ Amazon S3 và so sánh giá truy vấn cho các định dạng dữ liệu khác nhau

Bạn thấy Parquet sẽ tiết kiệm hơn nếu bạn sử dụng một tệp CSV có kích thước 1TB.

Vậy Parquet là gì?

Parquet là một định dạng thay thế để lưu trữ dữ liệu. Về cơ bản thì cả CSV và Parquet đều dùng để lưu trữ dữ liệu, nhưng chúng có sự khác biệt nhau trong nội bộ. Như đã nói ở trên, CSV được tổ chức dữ liệu trong hàng, còn Parquet thì theo cột.

Dưới đây là một ví dụ, Ta có bảng dữ liệu mẫu:

Graphical user interface, application

Description automatically generated

Bảng 3 - Dữ liệu mẫu

Với cùng 1 dữ liệu như trên thì ta sẽ có 2 cách tổ chức khác nhau

Table

Description automatically generated

Bảng 4 - Dữ liệu được tổ chức trong cả vùng lưu trữ cả hàng và cột

Qua đây, bạn sẽ thấy các tệp lưu trữ theo cột nhẹ hơn, vì có thể thực hiện nén đủ dữ liệu cho mỗi cột.

Apache Parquet bỏ qua dữ liệu không liên quan nên ta truy vấn và tổng hợp dữ liệu đều nhanh hơn dẫn đến tiết kiệm phần cứng.

* Tóm lại, Parquet là một định dạng dữ liệu theo cột mang lại hiệu quả hơn cho các tệp lớn hơn. Bạn sẽ tiết kiệm cả thời gian và tiền bạc bằng cách sử dụng Apache Parquet thay vì CSV.

1. **Giới thiệu về Hadoop Mapreduce**

# Hadoop

Hadoop là một Apache framework mã nguồn mở cho phép phát triển các ứng dụng phân tán (distributed processing) để lưu trữ và quản lý các tập dữ liệu lớn.

# Các thành phần chính của Hadoop

Apache Hadoop có nguồn gốc là để quản lý và truy cập dữ liệu, và chỉ bao gồm 2 thành phần là: Hadoop Distributed File System (HDFS) và MapReduce, một khuôn khổ xử lý cho dữ liệu lưu trữ trong HDFS.

# Kiến trúc Hadoop là gì?

Một cụm Hadoop nhỏ gồm 1 master node và nhiều worker/slave node.

Mỗi lớp có các thành phần liên quan riêng. Master node gồm JobTracker, TaskTracker, NameNode, và DataNode. Slave node gồm DataNode, và TaskTracker.

Cũng có thể slave node chỉ là dữ liệu hoặc node để tính toán.

# MapReduce

MapReduce là một kỹ thuật xử lý và là một mô hình lập trình cho tính toán phân tán để triển khai và xử lý dữ liệu lớn. Hadoop MapReduce là một khung xử lý dữ liệu của Hadoop xây dựng dựa trên ý tưởng của MapReduce, khi nói về MapReduce là chúng ta sẽ nghĩ ngay tới Hadoop MapReduce

# Các thành phần Hadoop MapReduce

Khi lập trình với MapReduce các bạn chỉ cần phải để ý tới 3 lớp sau:

1. Mapper

2. Shuffle and sorting

3. Reducer

# Tổng quan thực thi

Diagram

Description automatically generated

• Máy master phân phối M Map task cho các máy vào theo dõi tiến trình của chúng

• Các Map task đọc dữ liệu và tiến hành xử lý, kết quả được lưu trữ tại local buffer

• Pha Shuffle chỉ định các reducers tới các vùng nhớ đệm nơi mà chúng được đọc từ xa và xử lý bởi các reducers

• Reducer xuất kết quả và lưu trữ trên HDFS

# Mapper

Đây là pha đầu tiên của chương trình. Có hai bước trong pha này: splitting and mapping. Một tập dữ liệu được chia thành các đơn vị bằng nhau được gọi là chunks trong bước phân tách (splitting). Hadoop bao gồm một RecordReader sử dụng TextInputFormat để chuyển đổi các phân tách đầu vào thành các cặp key-value.

# Shuffle

Đây là pha thứ hai diễn ra sau khi hoàn thành Mapper. Nó bao gồm hai bước chính: sắp xếp và hợp nhất (sorting and merging). Trong pha này, các cặp key-value được sắp xếp bằng cách sử dụng các key. Việc hợp nhất đảm bảo rằng các cặp key-value được kết hợp.

# Reduce

Trong pha Reduce, đầu ra của pha Shuffle được sử dụng làm đầu vào. Reducer xử lý đầu vào này hơn nữa để giảm các giá trị trung gian thành các giá trị nhỏ hơn. Nó cung cấp một bản tóm tắt của toàn bộ tập dữ liệu, ví dụ như là tính tổng, tìm max, min,… . Đầu ra của pha này được lưu trữ trong HDFS.

# Ưu điểm của Hadoop MapReduce

### Hỗ trợ xử lý và tính toán song song (Parallel Processing)

Trong MapReduce, công việc được phân chia giữa nhiều node và mỗi node hoạt động đồng thời với một phần công việc. Mô hình MapReduce cho phép công việc được phân chia ra thành các công việc nhỏ hơn hoàn toàn riêng biệt.

### Data Locality

Mặc dù các máy đã kết hợp với nhau thành một cụm, tuy nhiên khi dữ liệu càng lớn lên thì việc di chuyển dữ liệu giữa các máy là rất mất thời gian và có thể gây ra các vấn đề như tắc nghẽn đường truyền.

Hadoop khắc phục vấn đề trên bằng cách phân phối dữ liệu ở nhiều node và mỗi node xử lý các phần dữ liệu nằm trên chính nó.

### Khả năng mở rộng (Scalability)

Hadoop có khả năng mở rộng cao, có thể lên tới hàng nghìn node mà không ảnh hưởng tới hiệu năng cũng như phát sinh lỗi.

Ví dụ để scan 1000TB dữ liệu trên 1 node với tốc độ 100MB/s thì sẽ mất 24 ngày, và khi mở rộng cụm lên 1000 node chúng ta cũng mất tương đương 35 phút để scan xong 1000TB dữ liệu này (hiệu năng hoàn toàn không bị giảm sút và không có phát sinh lỗi trong quá trình mở rộng)

Tính sẵn có và khả năng chịu lỗi (Availability & Fault Tolerance)

Hadoop lưu trữ các bản sao của dữ liệu trên các node khác nhau, vì thế trong trường hợp bị lỗi bản sao dữ liệu luôn sẵn sàng sử dụng bất cứ khi nào được yêu cầu để đảm bảo tính sẵn có của dữ liệu.

Nhờ tính năng sẵn có của dữ liệu mà Hadoop có khả năng chịu lỗi cao, khi một Task bị kill hay là một node bị mất kết nối dẫn tới Task đó không được hoàn thành thì Hadoop sẽ nhanh chóng phát hiện và chỉ định một node mới có chứa bản sao dữ liệu thực hiện Task đó (đảm bảo tính locality)

### Chi phí thấp (Cost-effective)

Như đã được đề cập tới trong bài Giới thiệu tổng quan về Hadoop, Hadoop chạy trên các máy có phần cứng phổ thông (commodity hardware), là các máy rẻ, băng thông không cao. Hadoop có khả năng chịu lỗi cao vì vậy cần ít các quản trị viên hơn. Hadoop là dễ học, dễ sử dụng nên cũng tốn ít chi phí trong việc đào tạo cũng như thuê nhân công.

### Bảo mật và xác thực (Security & Authentication)

Mô hình lập trình MapReduce giải quyết rủi ro về bảo mật bằng cách làm việc với HDFS và HBase có tính mật cao chỉ cho phép người dùng được phê duyệt mới có thể thao tác trên dữ liệu được lưu trữ trong hệ thống.

### Mô hình lập trình đơn giản

Các bạn có thể thấy mô hình lập trình MapReduce là cực kì đơn giản, ngoài ra thì Hadoop MapReduce sử dụng ngôn ngữ Java là một ngôn ngữ phổ biến và dễ học

1. **Giới thiệu về dịch vụ được sử dụng trong đồ án**
2. Dịch vụ AWS IAM: Quản lý danh tính và truy cập vào các tài nguyên và dịch vụ AWS một cách bảo mật

Ở đây nhóm em sử dụng dịch vụ này để chỉ định những tài khoản cá nhân hoặc thiết bị có thể truy cập dịch vụ và tài nguyên trong AWS, quản lý tập trung các quyền chi tiết.

1. Dịch vụ AWS S3: Dịch vụ lưu trữ đối tượng được xây dựng để truy xuất dữ liệu thuộc mọi kích thước và từ mọi nơi

Ở dịch vụ này, chúng em sử dụng để upload data và sử dụng S3 để chứa data

1. Databrick: Databricks là một nền tảng phân tích dữ liệu thống nhất dành cho kỹ thuật dữ liệu, máy học và khoa học dữ liệu cộng tác.

Sử dụng databrick như môi trường để query đến S3. Và để sử dụng dịch vụ này chúng ta cần đến AWS CloudFormation để tạo Workspace.