**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**Logo, company name

Description automatically generated**

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

TÌM HIỂU PARQUET VÀ VIẾT ỨNG DỤNG DEMO

**MÔN: ĐIỆN TOÁN ĐÁM MÂY  
MÃ LỚP: CLCO332779\_22\_1\_03CLC**

**GIẢNG VIÊN HƯỚNG DẪN: GV. Huỳnh Xuân Phụng**

**HỌC KÌ 1 NĂM HỌC 2022-2033**

**Thành phố Hồ Chí Minh, Tháng 12 Năm 2022**

**THÔNG TIN NHÓM**

**Tên nhóm:** Nhóm 8

**Nhóm Trưởng:** Đường Nguyễn An Khang

|  |  |  |
| --- | --- | --- |
| **STT** | **HỌ VÀ TÊN** | **MÃ SỐ SINH VIÊN** |
| 1 | Bùi Quang Huy | 20110492 |
| 2 | Đường Nguyễn An Khang | 20110501 |
| 3 | Đoàn Văn Hiếu | 20110476 |

**NỘI DUNG**

Giới thiệu về Apache Parquet 3

Giới thiệu về Hadoop MapReduce 4

Giới thiệu về các dịch vụ được sử dụng trong đồ án 8

Giới thiệu về Apache Spark 8

Giới thiệu về Apache Livy 9

1. **Giới thiệu về Apache Parquet**

Chắc hẳn, ai trong chúng ta cũng đã có người từng tiếp xúc, làm việc với CSV. Một loại định dạng tệp thông dụng và phổ biến sử dụng nhiều dòng chứa giá trị và được ngăn cách với nhau bằng dấu “,”.

Đồng nghĩa bạn phải truy vấn dữ liệu theo hàng. Việc này tưởng chừng như rất nhanh và đơn giản nhưng lại mất rất nhiều thời gian và khó lưu trữ kết quả.

Vì vậy đã có giải pháp thay thế CSV, đó chính là **Parpuet** – là một định dạng lưu trữ dữ liệu theo cột.

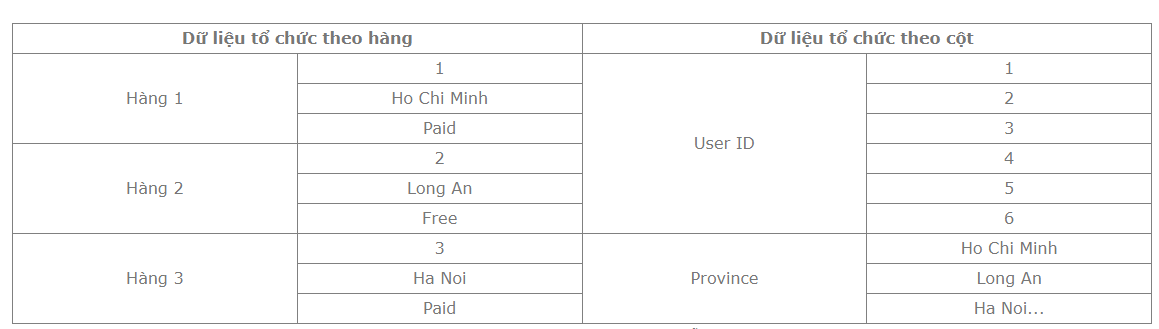
Về cơ bản thì cả CSV và Parquet đều được dùng để lưu trữ dữ liệu, nhưng chúng có sự khác biệt trong nội bộ. CSV được tổ chức dữ liệu theo hàng, còn Parquet thì theo cột.

Dưới đây là một vài ví dụ:

Dữ liệu mẫu:



Dữ liệu được tổ chức theo hàng và 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 để đọc trong số tất cả dữ liệu. Tuy nhiên đối với cơ sở dữ liệu cột thì điều này có nghĩa là chọn các cột cụ thể để đọc. Theo một bài viết của IBM, khi chuyển đổi CSV sang Parquet giúp tốc độ truy vấn tăng lên 30 lần (hoặc hơn) và tiết kiệm bộ nhớ lên đến 75%.

Tệp CSV dễ dàng sử dụng mà không cần thông qua một bước kiểm trả hay xác nhận nào. Điều này sẽ không đảm bảo tính bảo mật và an toàn nếu như chúng ta sử dụng nó làm cơ sở dụng liệu.

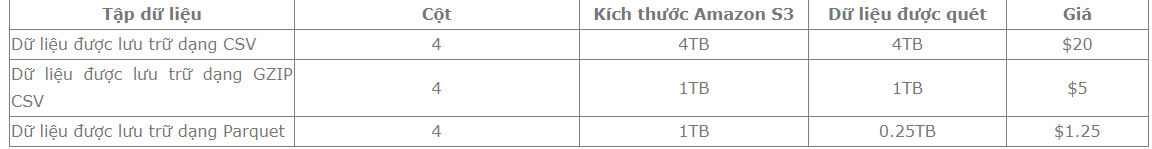
Nhưng điều khiến mọi người sử dụng parquet nhiều hơn chính là vì bài toán tiết kiệm chi phí. Ở mức độ doanh nghiệp khi học triển khai một dự án thì họ có thể lưu trữ trên hệ thống đám mây. Do đó các nhà cung cấp dịch vụ đám mây (ví dụ AWS) sẽ tính phí dựa trên dữ liệu được quét và lượng dữ liệu được lưu trữ.

Sau đây là bảng giá của dịch vụ Amazon S3:



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.

Và tiếp đến là bảng lưu trữ Amazon S3 và so sánh giá truy vấn cho các định dạng dữ liệu khác nhau:



Một lần nữa, bạn thấy Parquet sẽ tiết kiệm hơn 99,7% nếu bạn sử dụng một tệp CSV có kích thước 1TB. Do đó, CSV không phải là một lựa chọn tốt nhất khi dữ liệu của bạn lớn - cả về mặt chi phí và thời gian.

Vậy tóm lại Parquet là một định dạng dữ liệu theo cột mang lại hiệu quả hơn cho các tệp lớn hơn. Chúng ta sẽ tiết kiệm cả thời gian và tiền bạc bằng cách sử dụng Parquet thay vì CSV

1. **Giới thiệu về Hadoop MapReduce**

**Hadoop là gì?**

Hadoop là một Apache framework nguồn mở viết bằng Java cho phép phát triển các ứng dụng phân tán có cường độ dữ liệu lớn một cách miễn phí. Nó được thiết kế để mở rộng quy mô từ một máy chủ đơn sang hàng ngàn máy tính khác có tính toán và lưu trữ cục bộ (local computation and storage). Nhờ có cơ chế streaming mà Hadoop có thể phát triển trên các ứng dụng phân tán bằng cả java lẫn một số ngôn ngữ lập trình khác như C++, Pyhthon, Pearl,...

**Thành phần chính của Hadoop?**

* **HDFS**: Viết tắt của **H**adoop **D**istributed **F**ile **S**ystem, đây là thành phần xử lý việc lưu trữ file phân tán của Hadoop.
* **MapReduce**: Đây chính là thành phần để bạn “viết code” xử lý cho bài toán của bạn. Là nơi để bạn định nghĩa các job.
* **Yarn Framework**: Quản lý lập lịch các job và quản lý tài nguyên các node.

**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.

Trong một số trường hợp, Worker/Slave node được dùng để làm dữ liệu hoặc tính toán.

**MapReduce là gì?**

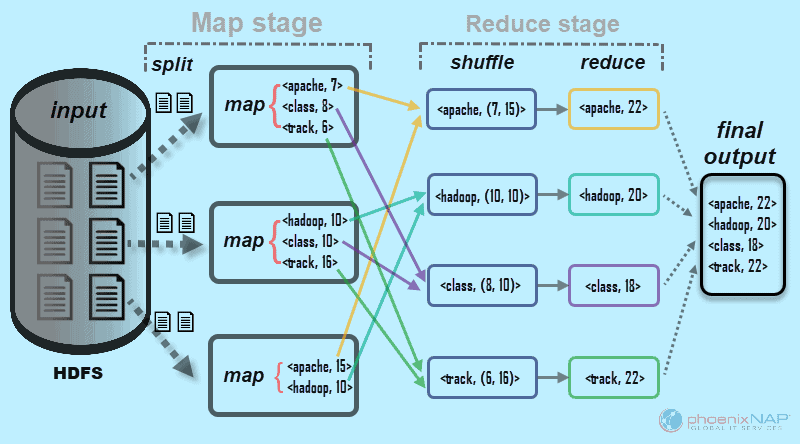
MapReduce là mô hình được thiết kế độc quyền bởi Google, nó có khả năng lập trình xử lý các tập dữ liệu lớn song song và phân tán thuật toán trên 1 cụm máy tính. Khi nhắc đến MapReduce là mọi người sẽ nghĩ ngay đến Hadoop MapReduce

**MapReduce** gồm một single master (máy chủ) JobTracker và các slave (máy trạm) TaskTracker trên mỗi cluster-node. Master có nhiệm vụ quản lý tài nguyên, theo dõi quá trình tiêu thụ tài nguyên và lập lịch quản lý các tác vụ trên các máy trạm, theo dõi chúng và thực thi lại các tác vụ bị lỗi. Những máy slave TaskTracker thực thi các tác vụ được master chỉ định và cung cấp thông tin trạng thái tác vụ (task-status) để master theo dõi.

**Các thành phần trong Hadoop MapReduce?**

MapReduce có 2 thành phần chính là ***Map*** và ***Reduce***. Đây là 2 thành phần được chính người dùng định nghĩa và nó cũng là 2 giai đoạn nối tiếp nhau trong quy trình xử lý dữ liệu của MapReduce. Ngoài ra ở giữa ***Map*** và ***Reduce*** còn một giai đoạn trung gian là ***Shuffle***.

Đây là ví dụ cách MapReduce hoạt động:



Chúng ta sẽ đi sâu vào cách chúng hoạt động:

* **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)***

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 các dịch vụ được sử dụng trong đồ án**

* ***AWS S3:***

Amazon S3 (Amazon Simple Storage) là một dịch vụ được cung cấp bởi Amazon

Web Services (AWS), cung cấp khả năng lưu trữ đối tượng (object storage) thông

qua giao diện web. Amazon S3 sử dụng cùng một cơ sở hạ tầng lưu trữ có thể mở

rộng. Amazon S3 có thể được sử dụng để lưu trữ bất kỳ loại đối tượng nào cho

phép sử dụng như lưu trữ cho các ứng dụng Internet, sao lưu và phục hồi, khôi

phục sự cố, lưu trữ dữ liệu, hồ dữ liệu (data lakes) để phân tích và lưu trữ đám mây kết hợp (Hybrid Cloud Storage).

* **Tác dụng:**
  + Tạo bucket chứa data đầu vào (file train.csv)
  + Tạo bucket chứa data đầu ra (các file được convert thành file parquet)
* ***AWS EMR***

Amazon EMR Serverless là một lựa chọn mới trong Amazon EMR, giúp các kỹ sư và nhà phân tích dữ liệu có thể chạy các ứng dụng được xây dựng bằng các khung dữ liệu lớn có nguồn mở một cách dễ dàng và tiết kiệm chi phí, ví dụ như Apache Spark, Hive hoặc Presto, mà không phải điều chỉnh, vận hành, tối ưu hóa, bảo mật hay quản lý các cụm.

* **Tác dụng:**
* Tạo các Spark Cluster
* Xử lý các Spark Job – đọc/ghi file Parquet
* ***AWS SageMaker***

Amazon SageMaker là một giải pháp AWS được quản lý hoàn toàn, cho phép các nhà khoa học và nhà phát triển dữ liệu nhanh chóng xây dựng, đào tạo và triển khai các mô hình học máy. Nó ở dạng môi trường phát triển tích hợp cho máy học, Amazon SageMaker Studio, hoạt động như một cơ sở để xây dựng dựa trên một bộ sưu tập các công cụ AWS SageMaker khác.

* **Tác dụng:**
  + Xây dụng notebooks instance được hỗ trợ bởi Amazon EMR
  + Đóng vai trò như interface giữa EMR và Livy

1. **Giới thiệu Apache Spark**

Apache spark cho phép bạn xây dựng những mô hình dự đoán nhanh chóng với khả năng thực hiện tính toán cùng lúc trên một nhóm các máy tính hay trên toàn bộ các tập dữ liệu mà không cần thiết phải trích xuất các mẫu tính toán thử nghiệm. Tốc độ xử lý dữ liệu của Apache spark có được là do khả năng thực hiện các tính toán trên nhiều máy khác nhau cùng một lúc tại bộ nhớ trong (in-memories) hay hoàn toàn trên RAM.

Apache spark cho phép bạn có thể xử lý các dữ liệu theo thời gian thực. Nghĩa là nó có thể vừa nhận dữ liệu từ các nguồn dữ liệu khác nhau, vừa có thể thực hiện xử lý ngay những dữ liệu mà nó vừa nhận được một cách đồng thời.

Apache spark không có hệ thống file cho riêng mình. Nó sử dụng nhiều hệ thống file khác nhau như S3, Cassandra, HDFS,.. Và hoàn toàn không phụ thuộc vào bất cứ hệ thống file nào nói trên.

1. **Giới thiệu Apache Livy**

Apache Livy là dịch vụ cho phép tương tác dễ dàng với Spark Cluster thông qua giao diện REST. Nó cho phép dễ dàng gửi các Spark Job hoặc các đoạn mã Spark. Tất cả thông qua giao diện REST đơn giản hoặc thư viện khách hàng RPC. Apache cũng đơn giản hóa sự tương tác giữa Spark các máy chủ ứng dụng, do đó cho phép sử dụng Spark cho các ứng dụng Web/Di động.