**1,Apache hadoop là gì:**

**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. Hadoop hiện thực mô hình MapReduce, mô hình mà ứng dụng sẽ được chia nhỏ ra thành nhiều phân đoạn khác nhau được chạy song song trên nhiều node khác nhau. Hadoop được viết bằng Java tuy nhiên vẫn hỗ trợ C++, Python, Perl bằng cơ chế streaming.

### **2,HDFS (Hadoop Distributed File System)**

* Khái niệm:
  + Đây là hệ thống file phân tán cung cấp truy cập thông lượng cao cho ứng dụng khai thác dữ liệu. **Hadoop Distributed File System (HDFS)** là hệ thống tập tin ảo. Khi chúng ta di chuyển 1 tập tin trên HDFS, nó tự động chia thành nhiều mảnh nhỏ. Các đoạn nhỏ của tập tin sẽ được nhân rộng và lưu trữ trên nhiều máy chủ khác để tăng sức chịu lỗi và tính sẵn sàng cao.
* Kiến trúc:
  + HDFS sử dụng kiến trúc master/slave, trong đó master gồm một NameNode để quản lý hệ thống file metadata và một hay nhiều slave DataNodes để lưu trữ dữ liệu thực tại.
* Nguyên lí hoạt động:
  + Một tập tin với định dạng HDFS được chia thành nhiều khối và những khối này được lưu trữ trong một tập các DataNodes.
  + NameNode định nghĩa ánh xạ từ các khối đến các DataNode. Các DataNode điều hành các tác vụ đọc và ghi dữ liệu lên hệ thống file. Chúng cũng quản lý việc tạo, huỷ, và nhân rộng các khối thông qua các chỉ thị từ NameNode.

### **3, MapReduce**

* Khái niệm:
  + Đây là hệ thống dựa trên YARN dùng để xử lý song song các tập dữ liệu lớn. Là cách chia một vấn đề dữ liệu lớn hơn thành các đoạn nhỏ hơn và phân tán nó trên nhiều máy chủ. Mỗi máy chủ có 1 tập tài nguyên riêng và máy chủ xử lý dữ liệu trên cục bộ. Khi máy chủ xử lý xong dữ liệu, chúng sẽ gởi trở về máy chủ chính.
* Nguyên lí hoạt động:
  + **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.
  + JobTracker là một điểm yếu của Hadoop Mapreduce. Nếu JobTracker bị lỗi thì mọi công việc liên quan sẽ bị ngắt quãng.

**4, Hệ sinh thái Hadoop**

**a, Apache Pig:**

Apache Pig là một platform mã nguồn mở cho việc xử lý dữ liệu trên Apache Hadoop. Nó cho phép người dùng thực hiện các thao tác truy vấn và xử lý dữ liệu dễ dàng bằng cách sử dụng một ngôn ngữ lập trình tăng cường giống như SQL, gọi là Pig Latin.

**b, Apache Hive**

Apache Hive là một platform mã nguồn mở dành cho việc truy vấn và xử lý dữ liệu trên Apache Hadoop. Nó cho phép người dùng thực hiện các thao tác truy vấn và xử lý dữ liệu dễ dàng bằng cách sử dụng một ngôn ngữ lập trình giống như SQL, gọi là HiveQL. Nó có thể truy cập và xử lý dữ liệu lưu trữ trên hệ thống lưu trữ dữ liệu phân tán Hadoop Distributed File System (HDFS) và Hadoop Distributed Cache (HDC).

**c, Apache Hbase**

Apache HBase là một cơ sở dữ liệu phân tán, không quan hệ, NoSQL được xây dựng trên nền tảng Apache Hadoop. Nó được thiết kế để cung cấp quyền truy cập đọc/ghi ngẫu nhiên, theo thời gian thực vào các bộ dữ liệu lớn và rất phù hợp cho các trường hợp sử dụng dữ liệu lớn như truyền dữ liệu theo thời gian thực và xử lý khối lượng lớn, tốc độ cao và dữ liệu đa dạng.

HBase lưu trữ dữ liệu của mình theo kiểu phân tán trên một cụm máy chủ hàng hóa và sử dụng Hệ thống tệp phân tán Hadoop (HDFS) làm hệ thống lưu trữ cơ bản. Nó được thiết kế để mở rộng theo chiều ngang, nghĩa là nó có thể xử lý các tập dữ liệu rất lớn bằng cách thêm nhiều máy chủ vào cụm.

HBase hỗ trợ nhiều kiểu truy cập dữ liệu, bao gồm tra cứu và quét theo thời gian thực, cũng như xử lý hàng loạt bằng MapReduce. Nó cũng có hỗ trợ lập phiên bản tích hợp sẵn, cho phép lưu trữ và truy xuất nhiều phiên bản của cùng một dữ liệu.

Một trong những tính năng chính của Hbase là cơ sở dữ liệu hướng cột, nghĩa là nó tổ chức dữ liệu thành các cột chứ không phải hàng, giúp truy vấn các tập dữ liệu lớn với số lượng cột nhỏ hiệu quả hơn. Điều này khác với các cơ sở dữ liệu truyền thống, theo hàng như MySQL hoặc PostgreSQL tổ chức dữ liệu thành các hàng.

Hbase cũng có hỗ trợ tích hợp để tự động phân tách dữ liệu, cho phép dữ liệu được phân phối trên nhiều máy chủ trong một cụm để có khả năng mở rộng và hiệu suất tốt hơn. Ngoài ra, nó hỗ trợ chuyển đổi dự phòng tự động để cung cấp tính sẵn sàng cao.

Hbase được nhiều công ty và tổ chức lớn như Facebook, Adobe và Bộ Quốc phòng Hoa Kỳ sử dụng để xử lý và phân tích các tập dữ liệu lớn trong thời gian thực.

**d, Apache Sqoop**

Apache Sqoop là một công cụ được thiết kế để truyền một lượng lớn dữ liệu một cách hiệu quả giữa Apache Hadoop và các kho lưu trữ dữ liệu có cấu trúc, chẳng hạn như cơ sở dữ liệu quan hệ. Sqoop có thể được sử dụng để nhập dữ liệu từ các kho lưu trữ dữ liệu có cấu trúc vào Hệ thống tệp phân tán Hadoop (HDFS) hoặc các hệ thống tệp tương thích với Hadoop như Apache HBase và cũng để xuất dữ liệu từ Hadoop sang các kho lưu trữ dữ liệu có cấu trúc.

Sqoop sử dụng kiến trúc dựa trên trình kết nối, cho phép nó hỗ trợ nhiều loại kho lưu trữ dữ liệu bao gồm MySQL, PostgreSQL, Oracle và nhiều loại khác. Nó có thể nhập song song dữ liệu từ cơ sở dữ liệu sang HDFS và cũng hỗ trợ nhập gia tăng. Sqoop cũng có thể được sử dụng để nhập toàn bộ bảng hoặc cột cụ thể và cũng có thể được định cấu hình thành mệnh đề where để lọc dữ liệu.

Sqoop cũng có thể được sử dụng để xuất dữ liệu từ HDFS trở lại kho lưu trữ dữ liệu có cấu trúc, điều này có thể hữu ích cho việc lưu trữ hoặc tích hợp dữ liệu Hadoop với các hệ thống hiện có. Dữ liệu đã xuất cũng có thể được chuyển đổi hoặc sửa đổi trong quá trình xuất.

Nhìn chung, Sqoop là một công cụ mạnh mẽ để tích hợp Hadoop với các kho lưu trữ dữ liệu có cấu trúc và cung cấp một cách thuận tiện để truyền lượng lớn dữ liệu giữa Hadoop và các hệ thống khác.

**e, Apache Kafka**

Apache Kafka là một nền tảng phát trực tuyến phân tán được thiết kế để xử lý khối lượng dữ liệu lớn trong thời gian thực. Đây là một hệ thống có thông lượng cao, chịu lỗi và có thể mở rộng, rất phù hợp để xử lý các luồng dữ liệu thời gian thực, chẳng hạn như dữ liệu nhật ký, chỉ số và dữ liệu sự kiện.

Về cốt lõi, Kafka là một nhà môi giới tin nhắn cho phép nhà sản xuất gửi dữ liệu đến một chủ đề và người tiêu dùng đọc dữ liệu từ chủ đề đó. Các nhà sản xuất và người tiêu dùng có thể chạy trên các máy riêng biệt và có thể được viết bằng các ngôn ngữ khác nhau, làm cho nó trở thành một nền tảng rất linh hoạt và có thể mở rộng.

Kafka sử dụng mô hình đăng ký xuất bản, trong đó dữ liệu được sắp xếp thành các chủ đề và nhà sản xuất gửi dữ liệu đến một hoặc nhiều chủ đề, trong khi người tiêu dùng đăng ký một hoặc nhiều chủ đề. Mỗi chủ đề là một nhật ký bản ghi được phân vùng, được lưu trữ trên đĩa và được sao chép trên nhiều máy chủ trong cụm Kafka để có khả năng chịu lỗi.

Một trong những tính năng chính của Kafka là khả năng xử lý khối lượng dữ liệu lớn với độ trễ rất thấp. Điều này có thể thực hiện được nhờ thiết kế ưu tiên hiệu suất và khả năng mở rộng, tập trung vào sao chép dữ liệu tối thiểu và trừu tượng hóa chi phí thấp.

Kafka được sử dụng rộng rãi trong nhiều công ty và tổ chức, bao gồm LinkedIn, Uber và Netflix, cho các trường hợp sử dụng như truyền dữ liệu theo thời gian thực, tổng hợp nhật ký, thu thập số liệu và tìm nguồn sự kiện.

Ngoài chức năng cốt lõi, nó còn có một số thành phần và thư viện bổ sung, chẳng hạn như API kết nối Kafka và API luồng Kafka, cung cấp chức năng bổ sung như nhập dữ liệu, chuyển đổi dữ liệu và xử lý dữ liệu theo thời gian thực.

**f, Apache Oozie**

Apache Oozie là một hệ thống lập lịch trình công việc để quản lý và điều phối các công việc Apache Hadoop trên một cụm. Nó được thiết kế để đơn giản hóa quá trình tạo, quản lý và thực thi các đường dẫn dữ liệu phức tạp trên cụm Hadoop.

Oozie cho phép bạn xác định quy trình công việc ở định dạng XML đơn giản được gọi là Ngôn ngữ định nghĩa quy trình làm việc Oozie (OWDL), sau đó có thể được sử dụng để điều phối và thực thi một loạt công việc, chẳng hạn như công việc MapReduce, tập lệnh Pig và truy vấn Hive. Các quy trình công việc có thể được lên lịch để chạy vào những thời điểm cụ thể hoặc được kích hoạt bởi các sự kiện, chẳng hạn như dữ liệu mới xuất hiện trong HDFS.

Một trong những tính năng chính của Oozie là khả năng xử lý các phụ thuộc giữa các công việc, cho phép bạn tạo các đường dẫn dữ liệu phức tạp với nhiều giai đoạn. Ví dụ: bạn có thể sử dụng Oozie để lên lịch truy vấn Hive chạy sau khi công việc MapReduce hoàn thành hoặc lên lịch chạy tập lệnh Pig chỉ khi công việc trước đó đã hoàn tất thành công.

Oozie cũng cung cấp giao diện người dùng dựa trên web cho phép bạn theo dõi và quản lý quy trình công việc của mình, bao gồm khả năng xem trạng thái của các công việc đang chạy và đã hoàn thành, cũng như chạy lại các công việc bị lỗi. Nó cũng có một Restful API mạnh mẽ có thể được sử dụng để tương tác với dịch vụ Oozie.

Nhìn chung, Oozie là một công cụ mạnh mẽ để quản lý và điều phối các công việc xử lý dữ liệu trên cụm Hadoop và nó đơn giản hóa quá trình tạo, lên lịch và thực thi các đường dẫn dữ liệu phức tạp.

**g, Apache Zookeeper**

Apache ZooKeeper là một dịch vụ tập trung để duy trì thông tin cấu hình, đặt tên và cung cấp đồng bộ hóa phân tán trên một cụm máy. Nó thường được sử dụng trong các hệ thống phân tán để phối hợp các hoạt động của nhiều nút, cung cấp dịch vụ cấu hình phân tán, dịch vụ đồng bộ hóa và đăng ký đặt tên cho các hệ thống phân tán.

Một trong những tính năng chính của ZooKeeper là nó cung cấp một không gian tên phân cấp đơn giản, tương tự như một hệ thống tệp, để lưu trữ và sắp xếp dữ liệu. Dữ liệu được lưu trữ trong cấu trúc cây, với mỗi nút trong cây được gọi là "znode".

ZooKeeper cung cấp các hoạt động nguyên tử như tạo, đọc, cập nhật và xóa cho các znode. Ngoài ra, nó hỗ trợ khái niệm đồng hồ, cho phép khách hàng đăng ký nhận thông báo khi dữ liệu được lưu trữ trong một znode cụ thể thay đổi. Tính năng này rất hữu ích để triển khai các hệ thống phân tán cần nhận thức được các thay đổi về cấu hình hoặc trạng thái của các nút khác trong cụm.

ZooKeeper cũng cung cấp một cơ chế để duy trì tính nhất quán giữa các nút trong một hệ thống phân tán, thông qua một giao thức đồng thuận được gọi là giao thức "ZooKeeper Atomic Broadcast" (ZAB). Điều này cho phép nhiều nút đồng ý về một giá trị chung và đảm bảo rằng tất cả các nút đều nhìn thấy các bản cập nhật giống nhau theo cùng một thứ tự.

ZooKeeper được sử dụng rộng rãi trong nhiều công ty và tổ chức lớn như Yahoo, Cisco và Netflix, để cung cấp các dịch vụ điều phối và cấu hình cho các hệ thống phân tán, chẳng hạn như cơ sở dữ liệu phân tán, khám phá dịch vụ và hệ thống nhắn tin phân tán.

Nó cũng có thể được sử dụng làm dịch vụ điều phối cho các hệ thống phân tán như Hadoop và Kafka và nhiều hệ thống phân tán khác, để xử lý bầu chọn người lãnh đạo, quản lý cấu hình và cung cấp đồng bộ hóa cho các hệ thống phân tán.

**i, YARN**

YARN (Yet Another Resource Negotiator) là một khung quản lý tài nguyên cho Apache Hadoop. Nó chịu trách nhiệm quản lý việc phân bổ tài nguyên, chẳng hạn như CPU và bộ nhớ, trên một cụm máy và lên lịch các tác vụ sẽ được thực thi trên các máy đó.

YARN được thiết kế để trở thành một khung chung có thể được sử dụng để quản lý tài nguyên cho nhiều loại khối lượng công việc khác nhau, bao gồm xử lý hàng loạt, SQL tương tác và phát trực tuyến. Nó bao gồm hai thành phần chính: Trình quản lý tài nguyên (RM) và Trình quản lý nút (NM).

Trình quản lý tài nguyên là thành phần chính quản lý việc phân bổ tài nguyên tổng thể trên cụm. Nó giao tiếp với Trình quản lý nút đang chạy trên từng máy riêng lẻ trong cụm và lên lịch các tác vụ sẽ được thực thi trên các máy đó dựa trên các tài nguyên có sẵn.

Trình quản lý nút là một thành phần chạy trên mỗi máy trong cụm và chịu trách nhiệm quản lý tài nguyên trên máy đó. Nó giao tiếp với Trình quản lý tài nguyên để báo cáo các tài nguyên có sẵn trên máy và để nhận hướng dẫn về nhiệm vụ nào sẽ được thực thi.

YARN cũng cung cấp Giao diện lập trình ứng dụng (API) chung cho phép người dùng gửi và quản lý các ứng dụng của riêng họ, cung cấp một cách linh hoạt hơn để quản lý tài nguyên cụm. Điều này cho phép chạy các loại công cụ xử lý dữ liệu khác nhau như MapReduce, Spark và thành phần hệ sinh thái dữ liệu lớn khác trên YARN.

YARN được giới thiệu trong Hadoop phiên bản 2, tách biệt thành phần quản lý tài nguyên và xử lý dữ liệu, cho phép sử dụng tài nguyên cụm hiệu quả hơn, cải thiện khả năng mở rộng và thêm tính linh hoạt hơn cho thành phần hệ sinh thái dữ liệu lớn.