Building Applications With Hadoop

(Xây dựng ứng dụng Với Hadoop)

*Khi xây dựng các ứng dụng sử dụng Hadoop, người ta thường có dữ liệu đầu vào từ các nguồn khác nhau, trong các định dạng khác nhau.****Trong bài trình bày của mình, "***[*New Tools for Building Applications on Apache Hadoop*](http://www.infoq.com/presentations/hadoop-frameworks-api)*", Eli Collins, người dẫn đầu công nghệ của nhóm Cloudera miêu tả tổng quát làm thế nào để xây dựng sản phẩm tốt hơn với Hadoop và các công cụ khác nhau mà nó có thể hỗ trợ, như Apache Avro, Apache Crunch, Cloudera ML và Cloudera Development Kit.*

[**Avro**](https://github.com/apache/avro)

Apache Avro là một dự án dành cho dữ liệu tuần tự trong các định dạng. Bạn có thể giải quyết vấn đề về bản ghi, mảng, liên kết, enums; có được các tập tin dữ liệu nhỏ hơn nhiều, có thể đọc và ghi dữ liệu mà không tạo ra bất kỳ mã nào. Avro có khả năng phù hợp với rất nhiều ngôn ngữ khác nhau như Java, C, C ++, C #, Python, Ruby, vv… và có thể sử dụng bởi rất nhiều công cụ. Tất cả các khía cạnh truyền thống của các định dạng dữ liệu Hadoop, như các dữ liệu nén hoặc chia tách dữ liệu đều là của Avro.

Một trong những mục tiêu của Avro đó là có thể sử dụng thông suốt nền tảng dữ liệu mà bạn đang sử dụng, không chỉ trong một tập hợp con mà còn của các thành phần. Vì vậy, MapReduce, Pig, Hive, Crunch, Flume, Sqoop… tất cả đều được Avro hỗ trợ.

Avro là một yếu tố quan trọng trong một hệ thống dữ liệu lớn khi bạn cần tạo ra rất nhiều dữ liệu, trong khi bạn không muốn tái xử lý nó liên tục.

**Crunch**

Apache Crunch Là một thư viện Java API giúp cho việc như thu thập và tập hợp dữ liệu trên MapReduce dễ dàng hơn.

Có lẽ bạn đã quen thuộc với Pig và Hive cũng như làm thế nào để xử lý dữ liệu với chúng. Tuy nhiên, không phải tất cả các định dạng dữ liệu mà bạn sử dụng sẽ phù hợp với Pig và Hive. Pig và Hive phù hợp để xử lí dữ liệu đã được ghi chép hoặc dữ liệu quan hệ, nhưng bạn vẫn có thể xử lý dữ liệu khác với Pig và Hive, mà không buộc bạn phải có một mô hình quan hệ hoặc một cấu trúc nhập, bù lại bạn phải làm rất nhiều việc liên quan. Đó là lí do Crunch được tạo ra.

Apache Crunch phát triển và hỗ trợ Java API đơn giản hóa quá trình tạo đường truyền dữ liệu trên Apache Hadoop. Các API Crunch được dựa trên mô hình [FlumeJava (PDF)](http://pages.cs.wisc.edu/~akella/CS838/F12/838-CloudPapers/FlumeJava.pdf) , là thư viện mà Google sử dụng để xây dựng đường truyền dữ liệu MapReduce của riêng của họ.

Ưu điểm của Crunch.

* Nó chỉ là Java. Bạn có thể truy cập một ngôn ngữ lập trình một cách đầy đủ.
* Bạn không cần phải tìm hiểu về Pig.
* Tương thích tốt với nhiều loại hệ thống.
* Nó được xây dựng như một thư viện mô-đun để tái sử dụng.

**Cloudera ML**

[Cloudera ML](https://github.com/cloudera/ml) (machine learning) là một thư viện mã nguồn mở, tích hợp nhiều công cụ nhằm mục đích phân tích, mô hình hóa dữ liệu như các lệnh để: tổng hợp, lấy mẫu, tiêu chuẩn hóa, và xoay vòng dữ liệu; hiện nay có thêm thuật toán phân nhóm cho k-mean.

Nó được xây dựng bằng cách sử dụng Crunch, từ đó thúc đẩy nhiều dự án hiện có. Ví dụ, định dạng vector: rất nhiều ML liên quan đến việc chuyển đổi dữ liệu thô, đó là danh sách các định dạng vector cho các thuật toán trong machine learning. Nó thúc đẩy giao diện vector Mahout và các classes cho mục đích đó. Các định dạng kỷ lục chỉ là một wrapper mỏng Avro, và HCatalog là hồ sơ và giản đồ định dạng để bạn có thể dễ dàng tích hợp với các nguồn dữ liệu hiện có.

**Cloudera Development Kit**

[Cloudera Development Kit](https://github.com/cloudera/cdk) (CDK) là một tập hợp các thư viện mã nguồn mở và các công cụ giúp việc viết các ứng dụng trên Hadoop dễ dàng hơn. Nó nhắm vào các nhà phát triển đang cố gắng để xây dựng các ứng dụng trên Hadoop. CDK cung cấp các API cao cấp trên đầu trang của các thành phần Hadoop.

CDK có kiến ​​trúc là một hệ thống mô-đun lỏng, bạn có thể sử dụng các mô-đun độc lập với nhau. Nó không bắt buộc bạn vào bất kỳ mô hình lập trình cụ thể nào, không buộc bạn phải chấp nhận cả tấn ràng buộc. Bạn có thể chỉ áp cần dụng các module mà bạn muốn. Các module dữ liệu CDK sẽ xử lý tất cả điều này cho bạn. Nó sẽ tự động serializes và deserializes dữ liệu từ Java POJO. Nó đã được xây dựng trong dạng nén, và trong các chính sách xung quanh tập tin cũng như thư mục bố trí, để bạn không phải lặp lại nhiều lần những lựa chọn. Nó sẽ tự động phân vùng dữ liệu trong những bố cục. Nó cho phép bạn tập trung vào làm việc trên một tập dữ liệu trên HDFS thay vì tất cả các chi tiết thực hiện. Nó cũng có các nhà cung cấp plugin cho hệ thống hiện có. CDK cung cấp một sơ đồ quản lí cho khối lượng dữ liệu khổng lồ mà người dùng không cần phải xác định lại toàn bộ khối dữ liệu.

Tóm lại, khi làm việc với dữ liệu từ nhiều nguồn khác nhau, thì việc chuẩn bị, làm sạch dữ liệu và xử lý chúng thông qua Hadoop liên quan đến rất nhiều công việc. Các công cụ như Crunch, Cloudera ML và CDK làm cho công việc trở lên dễ dàng hơn và tận dụng Hadoop hiệu quả hơn.

Tác giả **Eli Collins**