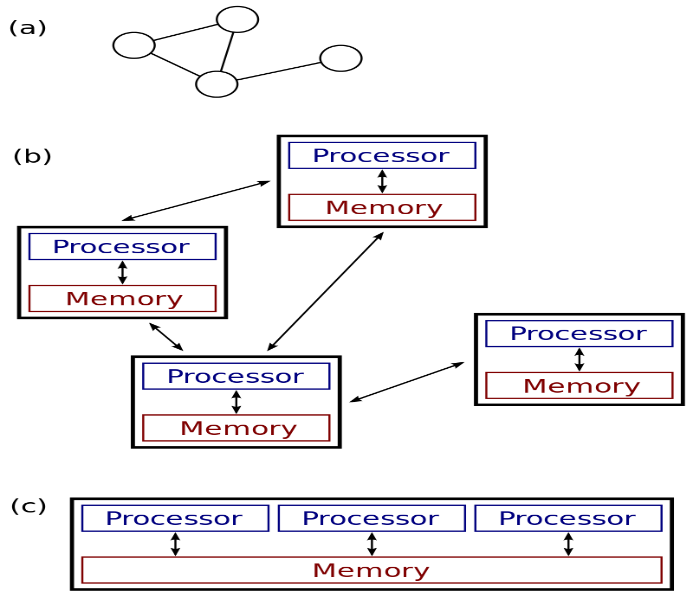
**TỔNG QUAN APACHE SPARK**

1. Định nghĩa

* Apache spark: là framework giúp thao tác với hệ thống phân tán trên một cụm máy tính có mã nguồn mở. Apache spark phục vụ cho việc xử lý dữ liệu lớn.
* Distributed computing – hệ thống phân tán: là hệ thống có các thành phần được đặt trên các máy tính nối mạng khác nhau, các máy này giao tiếp và phối hợp hành động bằng cách chuyển các thông điệp cho nhau. Nó khác với parallel computing (hình c).



* Computer cluster – cụm máy tính: là tập hợp các máy tính được kết nối hoạt động cùng nhau, chúng có thể được xem như một hệ thống duy nhất. Cụm máy tính có mỗi nút được thiết lập để thực hiện cùng một nhiệm vụ, được điều khiển và lên lịch bởi phần mềm.
* RDDs (resilient distributed dataset): là tập hợp các phần tử có khả năng chịu được lỗi và có thể hoạt động song song. Có 2 cách để tạo RDDs: tham chiếu đến một tập dữ liệu có sẵn (paralleized collection) trong chương trình điều khiển (driver programe) hoặc tham chiếu đến taapj dữ liệu bên ngoài
* Driver program (chương trình điều khiển)

1. RDDs, Accumulators, Broadcasts Vars
   1. Parallelized Collections

Parallelized được tạo ra bởi method *sparkcontext.parallelize.* Các phần tử của collection được sao chép và tạo thành một tập dữ liệu phân tán có thể được vận hành song song.

Một tham số quan trọng đối với method *parallezied* là *partitions.* Spark chạy một tác vụ cho mỗi phân vùng (partition) của cụm (cluster). Thông thường sẽ chia 2 dến 4 partition cho mỗi CPU trong Cluster. Spark cố gắng đặt số lượng phân vùng tự động dựa trên cluster, tuy nhiên mình có thể tùy chỉnh số lượng phân vùng.

* 1. Extenal Dataset

Spark có thể tạo tập dữ liệu phân tán từ bất kỳ nguồn lưu trữ nào được Hadoop hỗ trợ, bao gồm các tệp trên local, HDFS, Casandra,..

Textfile RDD có thể được tạo bằng method *sparkcontext.textFile* Method này lấy một URI cho tệp (đường dẫn cục bộ trên máy hoặc URI hdfs://, …) và đọc tệp như một tập hợp các dòng.

Một số lưu ý khi đọc file với spark:

* Nếu đọc file trên local thì đường dẫn phải được sử dụng tương tự ở các worker.
* Có thể tùy chỉnh số lượng phân vùng cho mỗi khối của tệp. Mặc đinh, spark tạo một phân vùng cho mỗi khối của tệp (128MB theo mặc định trong HDFS). Lưu ý là số lượng phân vùng không được ít hơn số lượng khối.

Ngoài textFile thì Spark python APi còn hỗ trợ một số định dạng dữ liệu khác:

* SparkContext.wholeTextFiles cho phép bạn đọc một thư mục chứa nhiều tệp văn bản nhỏ và trả về từng tệp dưới dạng cặp (tên tệp, nội dung)
* SequenceFile and Hadoop Input/Output Formats
  1. RDD Operation

RDD hỗ trợ 2 loại hoạt động: transformation (biến đổi) và actions. Transformation là tạo một dataset mới dựa vào cái có sẵn. Actions là trả về giá trị cho chương trình điều khiển sau khi tính toán trên các tập dữ liệu

Transformation được biến đổi khi có yêu cầu trả về kết quả cho chương trình điều khiển. Thiết kế này giúp spark chạy nhanh và hiệu quả hơn. Theo mặc đinh mỗi RDD đã chuyển đổi có thể được tính toán lại sau khi chạy chương trình điều khiển. Tuy nhiên, chúng ta có thể duy trì một RDD đã được chuyển đổi bằng phương pháp persist or cache

* 1. Passing Function to Spark

Spark hoạt động chủ yếu dựa vào việc chuyển các hàm trong các chương trình điều khiển để chạy trên cụm. Có 3 cách tối ưu nhất để làm việc này:

* Sử dụng biểu thức Lamda
* Viết hàm sau đó gọi vào Spark
* Sử dụng các hàm có sẵn trong các module

1. SQL, DataFrames, Datasets
2. Spark Streaming
3. MLlib
4. Security