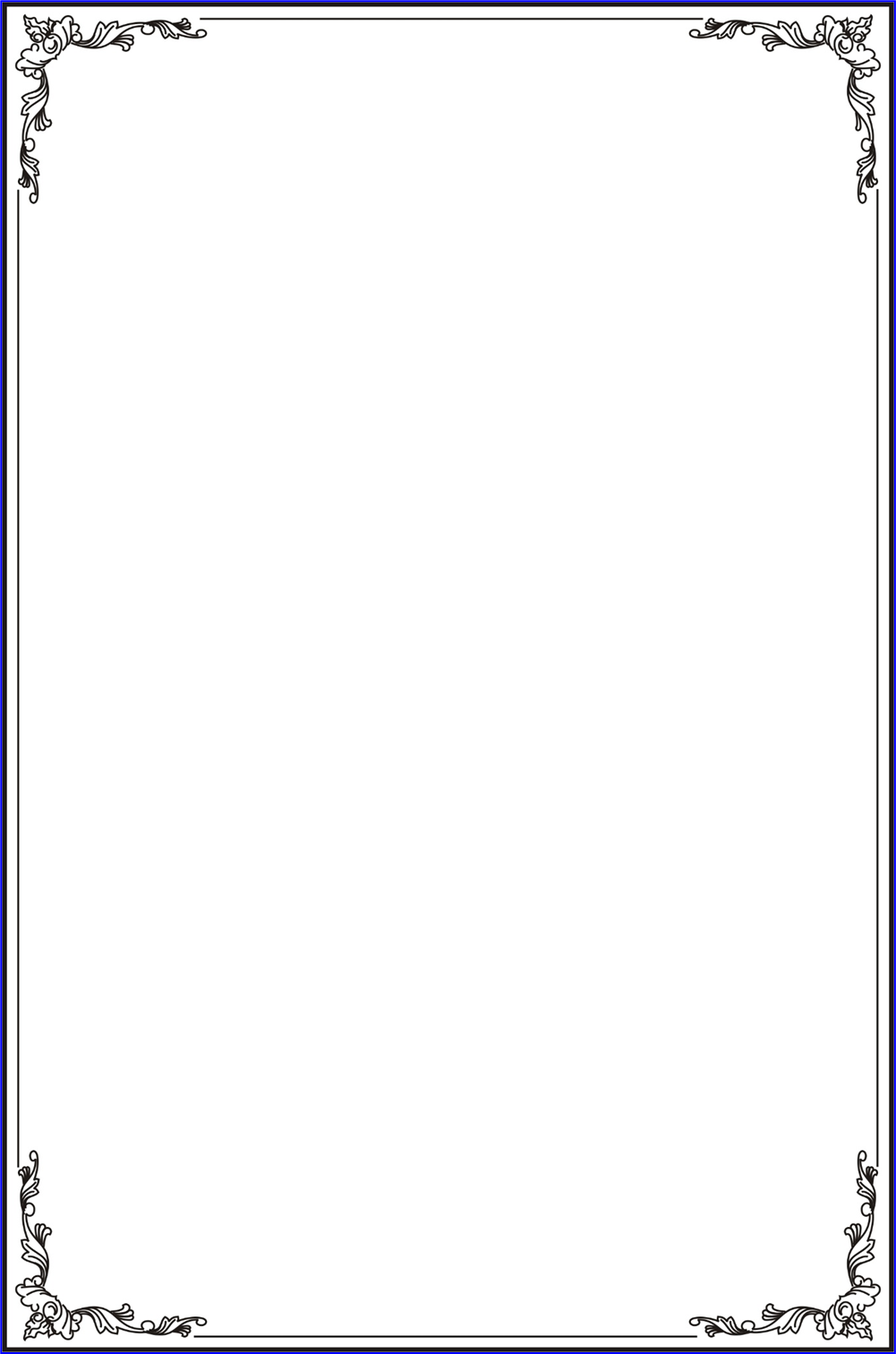
**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**🕯✡🕮🕮✡🕯**



**BÁO CÁO GIỮA KỲ**

**Môn học: BIGDATA SENSTIAL**

**Mã lớp học phần: BDES333877\_23\_1\_01**

**GVHD:ThS. Lê Thị Minh Châu**

**Sinh viên thực hiện: - Trần Thị Ngọc Trang - 21133109**

TP Hồ Chí Minh, tháng 10 năm 2023

*Read the original MapReduce paper by Dean and Ghemawat [1]. You can find the paper at* [*http://research.google.com/archive/mapreduce.html*](http://research.google.com/archive/mapreduce.html)

*(Tài liệu Tiếng Việt)*

[*https://www.slideshare.net/microlife/mapreduce-simplifieddataprocessing?fbclid=IwAR2aJcC43RG-YbanOMWQL\_RpxEDkIMQZpf9n5VrYczglvDQ7-CBuQ19go0o*](https://www.slideshare.net/microlife/mapreduce-simplifieddataprocessing?fbclid=IwAR2aJcC43RG-YbanOMWQL_RpxEDkIMQZpf9n5VrYczglvDQ7-CBuQ19go0o)

*Answer the following questions:*

*a) How do the input keys/values, the intermediate keys/values and the output keys/values relate?*

*b) How does MapReduce deal with node failures?*

*c) What is the meaning and the implication of locality? How is it used?*

*d) Which problem is addressed by introducing a combiner function to the MapReduce model?*

*Answer:*

1. **Các keys/values đầu vào, keys/values trung gian và keys/values đầu ra liên quan với nhau như thế nào?**

MapReduce là một mô hình lập trình và một phương thức triển khai liên kết gồm nhiều máy tính kết nối với nhau để tính toán và xử lí trên các bộ dữ liệu lớn. Người dùng chỉ định một hàm map xử lí một cặp keys/values đầu vào và tạo ra một tập các cặp keys/values trung gian. Thư viện MapReduce nhóm tất cả các cặp key/value tạm thời mà có key giống nhau sau đó chuyển cho hàm Reduce.

Sau đó hàm Reduce do người dùng tạo ra sẽ xử lí đầu vào là một key và một danh sách các value của key đó. Nó kết hợp các giá trị này lại với nhau cho ra một tập các keys/values output (nó có thể nhỏ hơn những giá trị đầu vào). Các giá trị trung gian được cấp cho Reduce thông qua một trình lặp. Điều này cho phép chúng ta xử lý tập giá trị lớn để phù hợp với bộ nhớ.

1. **MapReduce đối phó với lỗi nút (node) như thế nào?**

MapReduce đảm bảo tính chịu lỗi bằng cách thực hiện lại các tác vụ cần thiết và duy trì tính nhất quán của dữ liệu để đảm bảo kết quả chương trình khi gặp sự cố giống với trường hợp không có lỗi.

Sự cố trên Worker Node

* Master node (nút quản lý) thường xuyên liên lạc với các worker node (nút thực hiện công việc).
* Nếu Master node không nhận được phản hồi từ một worker node sau một khoảng thời gian nhất định, Master sẽ đánh dấu worker đó bị lỗi.
* Các tác vụ map đã hoàn thành bởi worker lỗi sẽ được đặt lại trạng thái chưa thực hiện, cho phép chúng thực hiện lại bởi một worker khác.
* Việc thực hiện lại là cần thiết vì dữ liệu key/value tạm thời chỉ được lưu trên đĩa địa phương của máy bị lỗi và không thể truy cập được.
* Tuy nhiên, tác vụ reduce không cần thực hiện lại vì dữ liệu đầu ra của chúng đã được lưu trên hệ thống file toàn cục có sao lưu trên các máy khác.
* Khi một tác vụ map được thực hiện lại bởi worker khác (vì worker ban đầu bị lỗi), tất cả các tác vụ reduce sẽ được thông báo để đọc dữ liệu từ worker mới.

Sự cố trên Master Node

* Dữ liệu cấu trúc trên Master node có thể được định kỳ sao lưu, giúp dễ dàng khôi phục trạng thái trước đó.
* Nếu Master node gặp sự cố, một Master node khác có thể được khởi động bằng dữ liệu sao lưu gần đây nhất.
* Tuy nhiên, vì chỉ có một Master, nếu Master gặp sự cố, chương trình MapReduce sẽ không thể hoàn thành. Người dùng có thể kiểm tra trạng thái và khởi động lại chương trình nếu cần.

1. **Nêu ý nghĩa và hàm ý của từ locality? Và cách nó được sử dụng như thế nào?**

Băng thông mạng là tài nguyên khó có thể cải thiện, nhưng có thể tìm cách để sử dụng nó một cách hợp lý. Dữ liệu lưu trên hệ thống file phân tán thực chất được lưu trên đĩa cứng của các máy tính trên cụm. Hệ thống file chia nhỏ mỗi file thành các block 64 MB và lưu một vài bản sao (thường là 3) của mỗi block trên các máy tính khác nhau.

1. **Vấn đề nào được giải quyết bằng cách đưa một hàm kết hợp vào mô hình MapReduce?**

*Vấn đề của hiệu suất trong trường hợp dữ liệu lặp lại nhiều lần*: Trong một số tình huống, khi chúng ta thực hiện tác vụ Map, các key trung gian có thể được tạo ra một cách lặp đi lặp lại. Điều này thường xảy ra khi xử lý dữ liệu có phân bố Zipf, nơi một số ít phần tử xuất hiện rất nhiều lần trong dữ liệu đầu vào.

*Giải pháp bằng cách sử dụng hàm kết hợp (Combiner):* Để giải quyết vấn đề này, MapReduce cho phép người dùng định nghĩa một hàm kết hợp tùy chọn. Hàm kết hợp này thường giống với hàm Reduce, và nó được thực hiện trên các máy thực hiện tác vụ Map. Hàm kết hợp hoạt động bên trong máy thực hiện Map, chẳng hạn như trên cùng một nút, để kết hợp các phần dữ liệu trung gian trước khi chúng được gửi qua mạng đến tác vụ Reduce.

Khi dữ liệu được kết hợp bởi hàm Combiner trên các máy thực hiện Map, kích thước của dữ liệu cần truyền đi mạng giảm đi đáng kể, làm tăng tốc độ truyền dữ liệu và giảm áp lực trên hệ thống. Hàm kết hợp được sử dụng để tối ưu hóa hiệu suất trong trường hợp lặp lại nhiều key trung gian và giúp MapReduce hoạt động hiệu quả hơn trong xử lý dữ liệu lớn.