

Google Trace Data Streaming System

Nhóm 1

Ngày 21 tháng 12 năm 2020

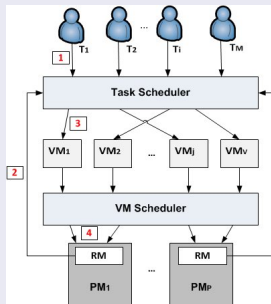
- 1 Giới thiệu bài toán
- 2 Đặc điểm dữ liệu
- 3 Mô hình hệ thống
- 4 Batch Layer - Xây dựng mô hình học máy
- 5 Speech Layer - Phân tích dữ liệu thời gian thực
- 6 Kiểm thử hiệu năng hệ thống

Giới thiệu

Bài toán thực tế

Thực hiện mô phỏng hệ thống theo dõi trạng thái của một trung tâm máy chủ và quản lý hoạt động của các máy tính trong trung tâm máy chủ.

Mô hình hoạt động thực tế



Dữ liệu mô phỏng

Nguồn

Là các file csv google publish, phiên bản đầu tiên vào ngày 27/10/2011

Đặc điểm

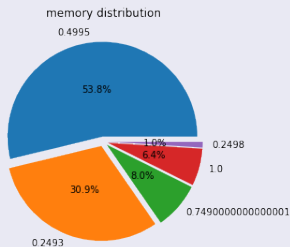
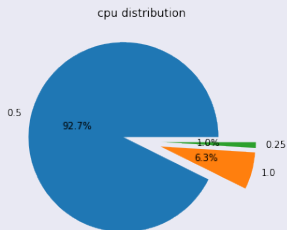
Gồm 6 loại dữ liệu:

- **job-event**: lưu lại các thông tin job được submit
- **machine-attributes**: chứa thông tin của các máy tính trong trung tâm máy chủ
- **machine-events**: chứa thông tin cấu hình của các máy tính trong trung tâm máy chủ
- **task-event**: chứa thông tin các task được submit
- **task-constraints**: chứa các điều kiện ràng buộc của các task
- **task-usage**: chứa thông tin trong thời gian thực thi các task

Số lượng máy tính

12583

Phân phối cấu hình

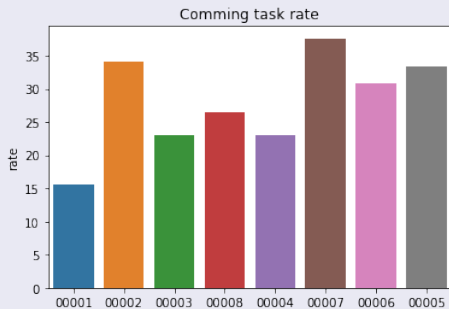


Thông tin task được submit

#	Column	Non-Null Count	Dtype
---	-----	-----	-----
0	time	167470 non-null	int64
1	missing_info	0 non-null	float64
2	job_id	167470 non-null	int64
3	task_index	167470 non-null	int64
4	machine_id	95973 non-null	float64
5	event_type	167470 non-null	int64
6	user	167470 non-null	object
7	scheduling_class	167470 non-null	int64
8	priority	167470 non-null	int64
9	cpu_request	167470 non-null	float64
10	memory_request	167470 non-null	float64
11	disk_space_request	167470 non-null	float64
12	different_machine_restriction	167470 non-null	int64

Thông tin task được submit

Số lượng task đến hệ thống trong 1 giây



Trung bình

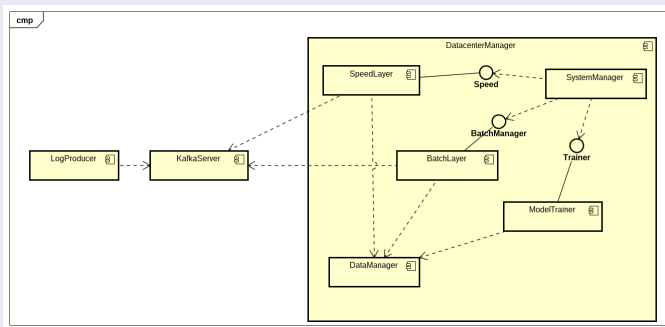
28 tasks / 1 giây được gửi đến hệ thống

Mô hình giải quyết bài toán

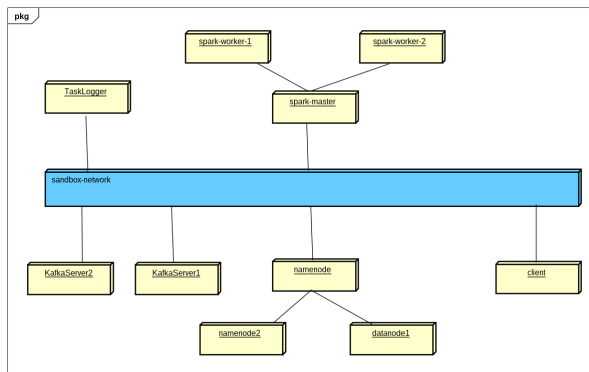
Mục tiêu

Xây dựng mô hình theo dõi được lượng task đang đến hệ thống và thực hiện phân tích, đánh giá và lưu trữ theo thời gian thực.

Mô hình



Các thành phần cài đặt



Tại sao sử dụng mô hình học máy

Đặc điểm của các máy tính trong hệ thống

Hệ thống gồm rất nhiều máy tính được phân chia vào các cấu hình khác nhau, do đó các task đến hệ thống cần được phân chia vào các nhóm có cấu hình phù hợp.

Đặc điểm của các task

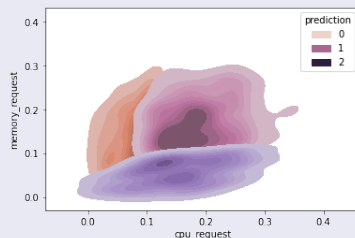
Từng task sẽ có yêu cầu tài nguyên khác nhau, nên sẽ được chia vào các nhóm máy tính khác nhau.

Giảm thời gian lập lịch

Thay vì tìm một máy tính thích hợp cho một task trên toàn bộ máy tính, ta chia các máy tính thành từng nhóm, phân task vào một nhóm và thực hiện thuật toán lập lịch trên số lượng máy tính nhỏ hơn.

Mô hình phân cụm task

Phân cụm



Kết quả

- $memory_request < 0.1$
- $cpu_request < 0.1$ và $memory_request \geq 0.1$
- $cpu_request \geq 0.1$ và $memory_request \geq 0.1$

Các công việc monitor

Xử lý dữ liệu

- Nhận dữ liệu từ broker
- Extract các thông tin cần thiết

Trung bình các thông số

- cpu request
- memory request
- cpu request per machine
- memory request per machine

Spark

Train clustering model (spark-worker config: 1 core 1g ram)

- 2 worker 4 partitions: 27.3 s
- 1 worker 2 partitions: 22.8 s
- 0 worker: Lost

Do đặc điểm của môi trường mô phỏng cụm phân tán nên có thể dẫn đến việc train mô hình với 2 worker chậm hơn so với 1 worker.

Kafka

Kịch bản

Tạo topic TASK-EVENT với partitions = 2, TASK-USAGE với partitions = 1. Cả hai đều có replication-factor=1.

Kiểm tra xem TASK-USAGE được gán với server kafka nào, sau đó tắt nó đi.

Kết quả

Những consumer hay producer của TASK-EVENT và TASK-USAGE đều không hoạt động

TASK-EVENT không còn hoạt động là vì replication-factor=1 nên nếu 1 partition ở một máy bị ngắt kết nối thì topic đó sẽ không còn tìm thấy leader cho partition tương ứng.

T

ạo topic TASK-EVENT với partitions = 2, TASK-USAGE với partitions = 1, replication-factor=2.

Kiểm tra xem TASK-USAGE được gán với server kafka nào, sau đó tắt nó đi.

Kết quả

Các consumer hay producer của TASK-USAGE ngừng hoạt động do bị lỗi còn TASK-EVENT vẫn có thể hoạt động bình thường.