TỐI ƯU HÓA CẤU HÌNH APACHE SPARK BẰNG MÔ HÌNH NGÔN NGỮ LỚN: MỘT CÁCH TIẾP CẬN DỰA TRÊN DỮ LIỆU

Lăng Huỳnh Đăng Khoa - 230101051

UIT-HCM

Tóm tắt

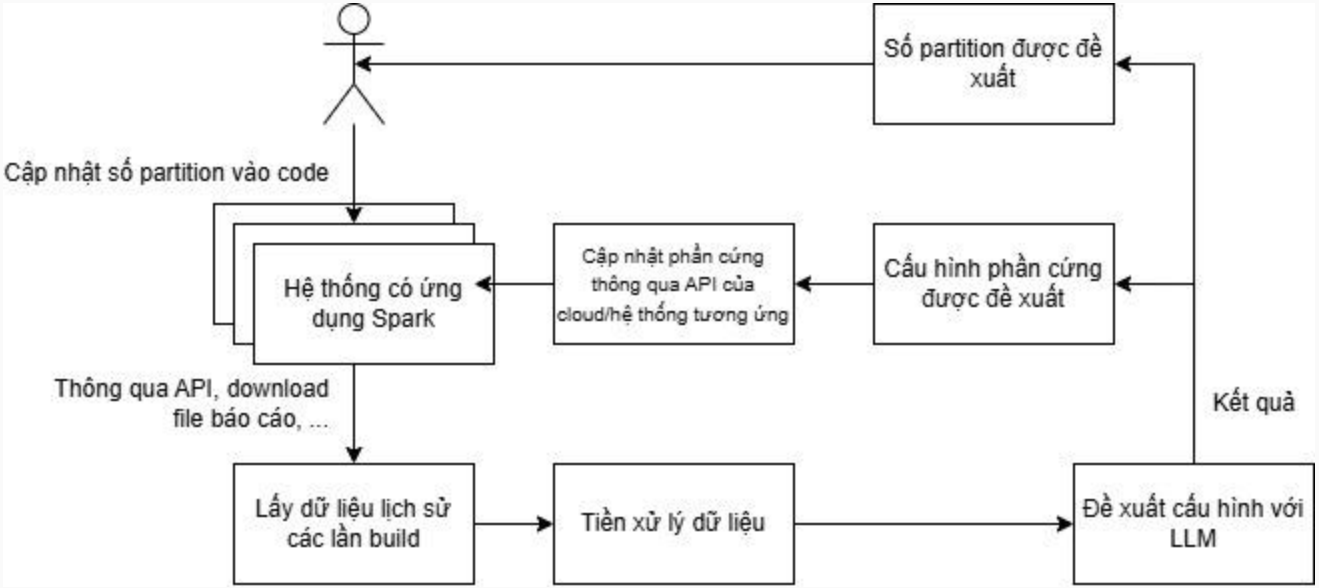
- Link Github của nhóm: https://github.com/khoalhd18/CS2205.CH183
- Link YouTube video: https://youtu.be/wOxwvdY4x8A
- Ảnh + Họ và Tên của các thành viên

Lăng Huỳnh Đăng Khoa - 230101051

Giới thiệu

- **Thách thức tối ưu hóa Spark**: Việc lựa chọn cấu hình phù hợp cho Apache Spark rất quan trọng, ảnh hưởng trực tiếp đến hiệu suất hệ thống, chi phí tài nguyên và thời gian xử lý dữ liệu lớn.
- **Tự động hóa đề xuất cấu hình**: Hệ thống tận dụng dữ liệu lịch sử về tài nguyên, dung lượng dữ liệu, số file xử lý,... để đưa ra thông số Spark phù hợp, giúp giảm thời gian thực thi và tối ưu hóa tài nguyên.
- **Ứng dụng AI và LLM trong tối ưu hóa**: Mô hình ngôn ngữ lớn (LLM) có thể phân tích dữ liệu lịch sử và tự động đề xuất cấu hình tối ưu, giúp giảm sự phụ thuộc vào kinh nghiệm kỹ sư và cải thiện hiệu suất hệ thống.

Giới thiệu


Số partition được đề xuất
Cập nhật số partition vào code
Hệ thống có ứng dụng Spark
Cập nhật phần cứng thông qua API của cloud/hệ thống tương ứng
Cấu hình phần cứng được đề xuất
Thông qua API, download file báo cáo, ...
Kết quả
Lấy dữ liệu lịch sử các lần build
Tiền xử lý dữ liệu
Đề xuất cấu hình với LLM

Mục tiêu

- **Tối ưu cấu hình phần cứng Spark**: Hệ thống sẽ đề xuất cấu hình Spark tối ưu nhằm giảm lãng phí tài nguyên và tăng hiệu suất xử lý dữ liệu lớn.
- **Tối ưu số partition một lần build**: Hệ thống sẽ đề xuất số partition trong một lần build nhằm giúp spark engine có thể điều phối, sử dụng các executor một cách hiệu quả hơn.
- **Đánh giá hiệu suất**: Sử dụng các chỉ số như thời gian thực thi, tài nguyên tiêu thụ, số lần tái tính toán và mức độ cải thiện so với cấu hình mặc định để đo lường hiệu quả của hệ thống.

Nội dung và Phương pháp

- **Nội dung 1**: Nghiên cứu cách thức hoạt động của Apache Spark và các cấu hình quan trọng.
  - **Phương pháp**: Nghiên cứu tài liệu chính thức của Apache Spark, phân tích các nghiên cứu liên quan, đồng thời thực hiện nhiều thử nghiệm để đánh giá tác động của các tham số.
- **Nội dung 2**: Ứng dụng LLM trong tối ưu cấu hình Spark.
  - **Phương pháp**:
    - Sử dụng LLM để phân tích dữ liệu đầu vào, học từ các cấu hình trước đó để đề xuất cấu hình tối ưu.
    - So sánh kết quả với các phương pháp tối ưu truyền thống như heuristic-based tuning hoặc grid search.

Nội dung và Phương pháp

- **Nội dung 3**: Triển khai hệ thống trên dịch vụ đám mây.
  - **Phương pháp**:
    - Sử dụng Amazon EC2 để triển khai môi trường Spark với các cấu hình linh hoạt.
    - Xây dựng API hoặc giao diện web để người dùng có thể nhập thông tin và nhận khuyến nghị từ LLM.
    - Đảm bảo hệ thống có khả năng chịu tải cao, phục hồi sau lỗi và bảo mật dữ liệu.
- **Nội dung 4**: Đánh giá hiệu suất của hệ thống.
  - **Phương pháp**:
    - Sử dụng các chỉ số như thời gian thực thi trung bình, mức giảm lãng phí tài nguyên, số lần tái tính toán và độ chính xác của dự đoán so với cấu hình tối ưu thực tế.
    - So sánh kết quả của hệ thống với các phương pháp truyền thống để xác định mức độ cải thiện.

Kết quả dự kiến

- Hệ thống có thể đưa ra khuyến nghị cấu hình Spark chính xác dựa trên dữ liệu lịch sử, giúp giảm thời gian xử lý và tối ưu hóa tài nguyên.
- So sánh với các phương pháp tối ưu truyền thống cho thấy LLM có khả năng cải thiện hiệu suất đáng kể.
- Hệ thống triển khai trên đám mây, cho phép người dùng tương tác dễ dàng và áp dụng vào thực tế.

Tài liệu tham khảo

[1] J. Doe and A. Smith, "Auto-Tuning Apache Spark Parameters for Processing Large Datasets," KTH Royal Institute of Technology, 2024. Available: https://kth.diva-portal.org/smash/record.jsf?pid=diva2%3A1798104&dswid=-8982

[2] M. Johnson and L. Wang, "AI-Enhanced Compute Resource Management for Apache Spark," International Journal of Future Modern Research (IJFMR), vol. 6, no. 3, pp. 45–56, 2024. Available: https://www.ijfmr.com/research-paper.php?id=33716

[3] S. Lee and K. Patel, "Reinforcement Learning for Automatic Parameter Tuning in Apache Spark: A Q-Learning Approach," in Proceedings of IEEE International Conference on Big Data (IEEE BigData), 2024. Available: https://ieeexplore.ieee.org/document/10665567

[4] G. Wang, J. Xu, and B. He, "Towards Automatic Tuning of Apache Spark Configuration," in Proceedings of the ACM Symposium on Cloud Computing (SoCC), 2018. Available: https://www.researchgate.net/publication/327812093_Towards_Automatic_Tuning_of_Apache_Spark_Configuration

[5] Y. Zhang and H. Liu, "A Novel Reinforcement Learning Approach for Spark Configuration Optimization," Sensors, vol. 22, no. 15, p. 5930, 2022. Available: https://www.mdpi.com/1424-8220/22/15/5930