

**TẬP ĐOÀN VIỄN THÔNG QUÂN ĐỘI**

BIỂU MẪU

TÀI LIỆU THIẾT KẾ CHI TIẾT

**Mã hiệu dự án: Triển khai AQE tối ưu Spark SQL**

**Hà Nội, ngày 04 tháng 09 năm 2023**

**BẢNG GHI NHẬN THAY ĐỔI TÀI LIỆU**

\*A – Tạo mới, M – Sửa đổi, D – Xóa bỏ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Ngày  thay đổi | Vị trí  thay đổi | A\*  M, D | Nguồn gốc | Phiên bản cũ | Mô tả thay đổi | Phiên  bản mới |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**Trang ký**

Người lập: Nguyễn Viết Long

Data Engineer-Gem

Người xem xét: Phạm Văn Ngọc

Người xem xét: <Ngày>

<Chức danh>

Người phê duyệt: <Ngày>

<Chức danh>

MỤC LỤC

I. [GIỚI THIỆU 5](#_Toc144306179)

[1.1 Spark Adaptive Query Execution là gì? 5](#_Toc144306180)

[1.2 Mục đích 5](#_Toc144306181)

[1.3 Phạm vi 5](#_Toc144306182)

[1.4 Khái niệm, thuật ngữ 5](#_Toc144306183)

[1.5 Tài liệu tham khảo 5](#_Toc144306184)

[1.6 Mô tả tài liệu 5](#_Toc144306185)

II. các ưu điểm của khi sử dụng spark aqe

III. KẾT QUẢ THEO DÕI

# GIỚI THIỆU

## Spark Adaptive Query Execution là gì?

Spark AQE là một nâng cấp để tối ưu khi sử dụng SparkSQL

## Mục đích

Tài liệu này giới thiệu các nâng cấp và nêu kết quả

## Phạm vi

## Khái niệm, thuật ngữ

| Thuật ngữ | Định nghĩa | Ghi chú |
| --- | --- | --- |
|  |  |  |

## Tài liệu tham khảo

<https://karcuta.medium.com/adaptive-query-execution-in-spark-3-0-7b2887a212c5>

<https://kyuubi.readthedocs.io/en/v1.5.0-incubating/deployment/spark/aqe.html>

## Mô tả tài liệu

Tài liệu gồm 5 phần:

* Giới thiệu;
* Nêu chi tiết các ưu điểm khi sử dụng AQE
* Kết quả theo dõi

# 3 Ưu điểm khi của Spark AQE

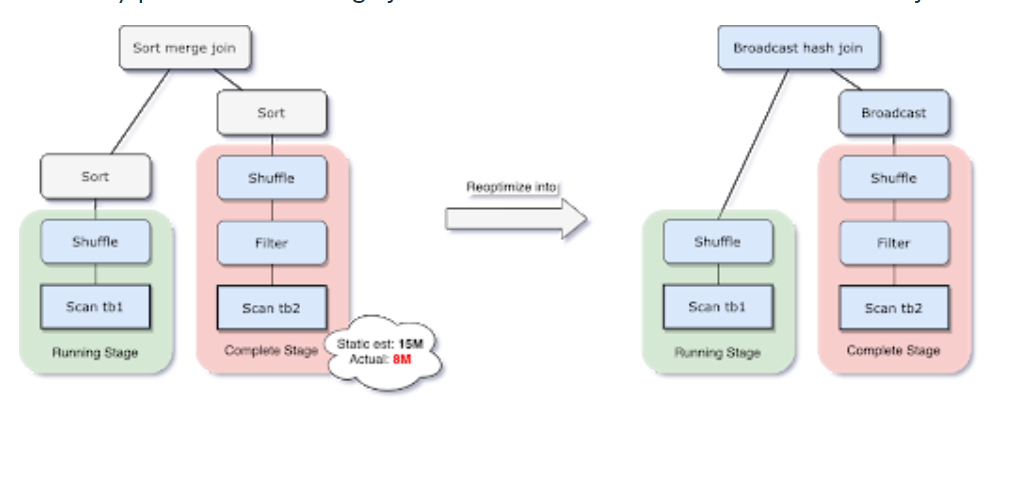
## Dynamically Switch Join Strategies

Ưu điểm đầu tiên là sẽ tự đọng thay đổi phương pháp join giữa **BroadcastHash** **và MergeSort** **Join**

(Link tham khảo về 2 loại join <https://medium.com/swlh/spark-joins-tuning-part-1-sort-merge-vs-broadcast-a98d82610cf0>

)

* Vấn đề : Như trên size estimate trên spark là 15mb nhưng thực tế lại là 8mb lúc trước sẽ không thể sử dụng broadcast join (giả sử cấu hình 10mb để Spark thực hiện BroadcastHash join)
* Giải pháp: Với **AQE** thì spark hỗ trợ chuyển đổi ngay trong lúc run time. Khi dữ liệu được đọc ra sẽ được quét lại nếu đáp ứng đúng yêu cầu là nhỏ hơn 10mb thì sẽ có thể sử dụng broadcast join



*Hình 1*

Các tham số liên quan :

**spark.sql.autoBroadcastJoinThreshold = 10mb**

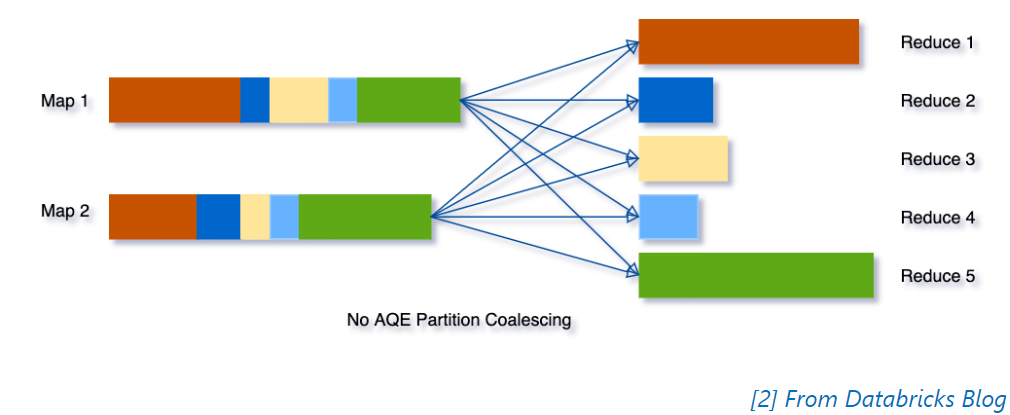
**spark.sql.adaptive.enabled=true**

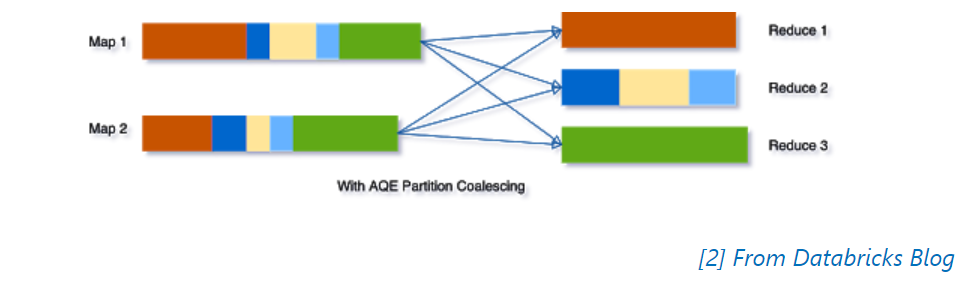
**spark.sql.adaptive.localShuffleReader.enabled=true**

## Dynamically Coalesce Shuffle Partitions

Ưu điểm tiếp theo tự động giảm partition khi suffle

* Vấn đề : Khi suffle Spark sẽ tự tạo số partition theo tham số **spark.sql.shuffle.partitions** dẫn đến việc các partition không bằng nhau và không tối ưu được tài nguyên.
* Giải pháp : Spark sẽ gộp các partition nhỏ thành một phần lớn để tránh tạo nhiều task.





Hình 2

Các tham số liên quan:

**spark.sql.adaptive.coalescePartitions.enabled**

**spark.sql.adaptive.coalescePartitions.initialPartitionNum**

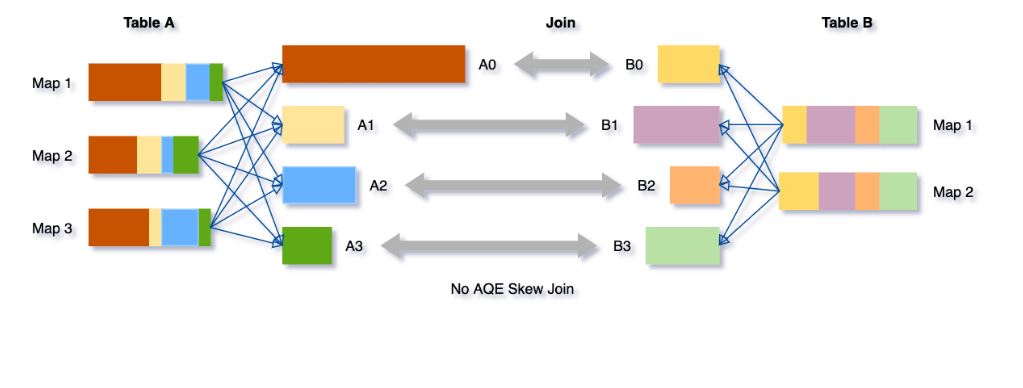
**spark.sql.adaptive.advisoryPartitionSizeInBytes**

**spark.sql.adaptive.coalescePartitions.initialPartitionNum** số partition khởi tạo khi suffle ta set lớn để có thể tối ưu đc phần chạy sau đó vs aqe sẽ thu lại để cho mỗi partition sẽ có size là 128m (**spark.sql.adaptive.advisoryPartitionSizeInBytes=128mb** mục đích là khớp với kích thước mỗi block trên HDFS tuy nhiên với hệ thống VDP hiện tại thì set **spark.sql.adaptive.advisoryPartitionSizeInBytes = 64mb** sẽ chạy nhanh hơn)

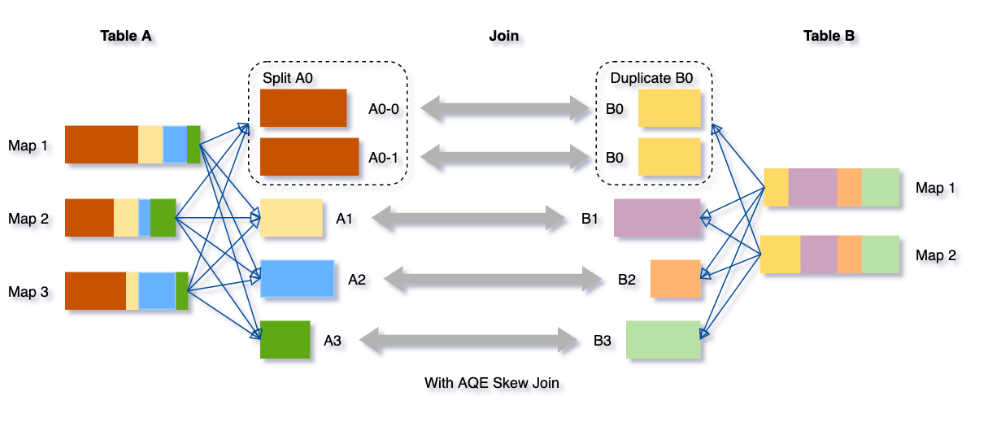
## Dynamically Handle Skew Joins

Ưu điểm cuối cùng là giải quyết vấn đề skew join

* Vấn đề : Khi thực hiện MergeSort Join có thể bị một partition lớn hơn nhiều so với phần còn lại . Như hình ta thấy phần A0 và B0 đang làm việc nhiều hơn so vs các partition khác.



* Giải pháp : là duplicate B0 thành 2 phần sau đó join thì A0 cx sẽ chia làm 2. Điều này sẽ tạo thêm một task nữa nhưng thời gian sẽ nhanh hơn.



Các tham số liên quan

**spark.sql.adaptive.skewJoin.skewedPartitionFactor**

**spark.sql.adaptive.skewJoin.skewedPartitionThresholdInBytes**

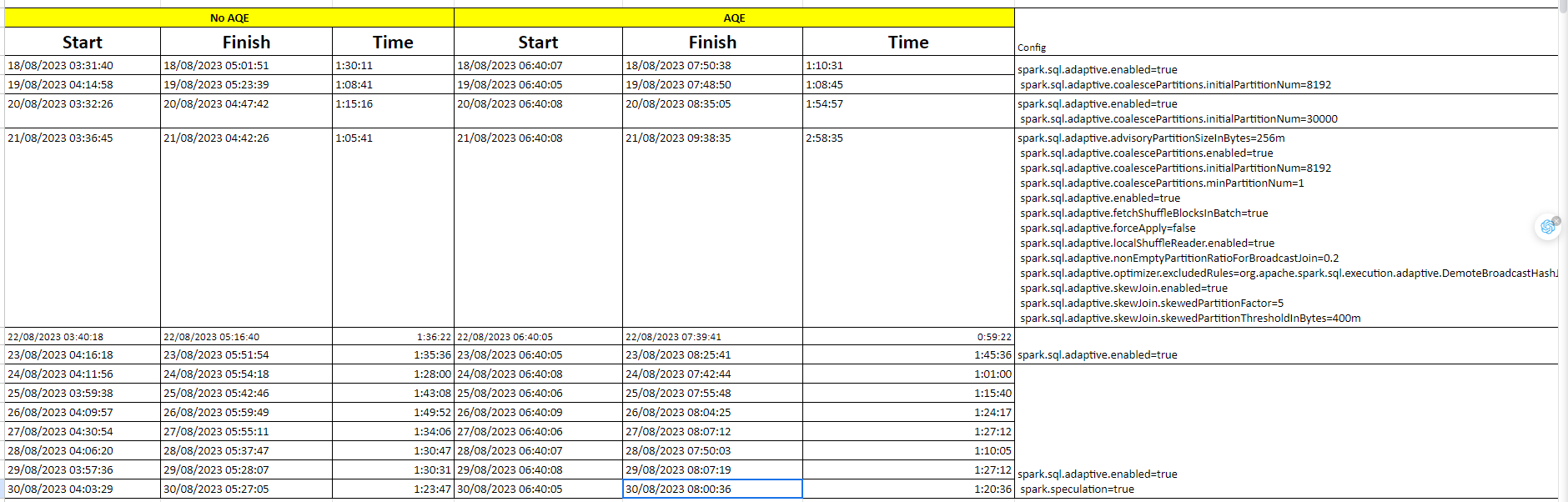
**spark.sql.adaptive.advisoryPartitionSizeInBytes**

Một partition bị coi là skew và sẽ áp dụng tối ưu trên nếu :

* Nếu một **size\_partition** lớn hơn **median\_partition** \* **skewedPartitionFactor**
* Nếu **size\_partition** > **skewedPartitionThresholdInBytes**

# Kết quả theo dõi

<https://docs.google.com/spreadsheets/d/1W_3jDXNIeuHy7-BO77Gb-5E_vYNo0ILk3ZTChgv7YYk/edit#gid=0>



Từ kết quả trên ta sẽ

Cấu hình phù hợp :

* **spark.sql.adaptive.enabled=true**
* **spark.speculation=true**
* **spark.sql.adaptive.enabled=true**
* **spark.sql.adaptive.coalescePartitions.enabled=true**
* **spark.sql.adaptive.localShuffleReader.enabled=true**
* **spark.sql.adaptive.skewJoin.enabled=true**
* **spark.sql.autoBroadcastJoinThreshold=-1**
* **spark.sql.adaptive.skewJoin.skewedPartitionFactor=5**
* **spark.sql.adaptive.advisoryPartitionSizeInBytes=64M**

Tham số cần config:

* **spark.sql.adaptive.enabled=true**
* **spark.speculation=true**
* **spark.sql.autoBroadcastJoinThreshold=-1**

Giải thích :

**spark.sql.adaptive.enabled:** Tham số để tắt bật AQE

**spark.speculation :** Tham số này giúp cho spark sẽ tự restart nếu một task chạy lâu hơn các task trong một stage

**spark.sql.autoBroadcastJoinThreshold:** Tham số điều chỉnh size BroadcastJoin

Vì AQE cần shuffle để có thể thục hiện tối ưu nên ta sẽ không cho thực hiện **BroadcastJoin**

bằng cách xét tham số **spark.sql.autoBroadcastJoinThreshold=-1**