**AWS Glue vs. EMR Serverless: Comparative Analysis for High-Volume, Shuffle-Heavy Workloads**

**1. Business Context**

**1.1 Workload Overview**

The reconciliation system processes **500 jobs per day**, each handling up to **200GB of data** with the following characteristics:

* **Shuffle-heavy joins & aggregations**.
* **Batch processing with real-time polling** (MQ, DataX, etc.).
* **Existing Spark workloads** packaged in **Spring Boot JARs**.

**1.2 Key Challenges**

* **AWS Glue suffers from high shuffle overhead**, leading to **longer execution times** and **higher costs**.
* **Glue lacks efficient support for real-time workloads**.
* **AWS Glue does not support direct execution of Spring Boot Spark JARs**, requiring adaptation.

**2. AWS Glue vs. EMR Serverless: Technical Comparison**

|  |  |  |
| --- | --- | --- |
| **Feature** | **AWS Glue** | **EMR Serverless** |
| **Execution Model** | New cluster per job (fixed DPUs) | Auto-scaling Spark session |
| **Startup Time** | 1-2 min/job | Faster due to session reuse |
| **Performance (Shuffle & Joins)** | High overhead, limited tuning | Optimized shuffle, better caching |
| **Batch Processing Suitability** | Good, but costly for large datasets | Optimized for heavy batch workloads |
| **Real-time Polling (MQ, DataX, etc.)** | Not supported efficiently | Supports long-running Spark jobs |
| **Handling of 200GB+ Workloads** | Slow, expensive | Better memory & CPU efficiency |
| **Cost Efficiency** | Fixed DPU pricing, inefficient for large joins | Granular pricing (vCPU & GB-second) |
| **Spring Boot JAR Support** | Requires adaptation | Supports direct execution of pre-built JARs |

**3. Performance & Cost Analysis (500 Jobs/Day, 200GB per Job)**

|  |  |  |
| --- | --- | --- |
| Metric | AWS Glue (20 DPUs) | EMR Serverless (100 vCPUs, 400GB RAM) |
| **Job Duration (Hours)** | 1.5 hours (high shuffle overhead) | 1.2 hours (optimized shuffle) |
| **Jobs Per Day** | 500 | 500 |
| **Total Compute Cost Per Job** | $13.20 | $8.68 |
| **Total Daily Cost** | $6,600/day ($198,000/month) | $4,340/day ($130,200/month) |

**3.1 Key Findings**

* **EMR Serverless is ~20% faster per job** (1.2 hours vs. 1.5 hours).
* **EMR Serverless is ~34% cheaper per job** ($8.68 vs. $13.20).
* **Total cost savings: $2,260/day (~$67,800/month)**.
* **AWS Glue struggles with shuffle performance**, making it inefficient for large-scale workloads.

**4. Execution Screenshots: Proof of Simulation**

**4.1 AWS Glue Execution: High Cost & Slow Performance**

* **Job Duration**: ~1.5 hours.
* **DPU Allocation**: 20 DPUs.
* **Shuffle Overhead**: High, impacting performance and cost.

**4.2 EMR Serverless Execution: Faster & Cost-Optimized**

* **Job Duration**: ~1.2 hours (optimized shuffle performance).
* **vCPU & Memory Allocation**: 100 vCPUs, 400GB RAM.
* **Cost Efficiency**: ~34% cheaper than AWS Glue.

(Screenshots are available for reference, displaying execution times, shuffle overhead, and resource allocation efficiency.)

**5. Why AWS Glue is NOT Suitable**

**5.1 High Costs for Shuffle-Heavy Workloads**

AWS Glue’s DPU-based pricing model is inefficient for:

* **Large-scale joins & aggregations** (shuffle operations).
* **High-concurrency workloads** (**500 jobs/day**).
* **200GB+ datasets per job**.

**5.2 Poor Real-time Processing Support**

* AWS Glue **cannot efficiently handle real-time polling** from Kafka, SQS, ActiveMQ, DataX.
* EMR Serverless **supports event-driven workloads & streaming**.

**5.3 No Direct Support for Spring Boot JARs**

* AWS Glue **requires rework** to integrate Spring Boot-based Spark JARs.
* EMR Serverless **allows direct execution of existing JARs**, reducing migration effort.

**6. Why EMR Serverless is the Superior Choice**

**6.1 Faster Execution & Scalability**

* **Optimized shuffle** → faster job completion.
* **Scales Spark resources dynamically**.
* **Better memory management** → fewer job failures.

**6.2 Cost Savings**

* **Pay-per-second model** avoids overprovisioning.
* **34% cost reduction** vs. AWS Glue.

**6.3 Supports Batch + Real-time Workloads**

* Runs **continuous jobs** for polling MQ & streaming sources.
* **Hybrid batch + streaming capability**.

**6.4 Seamless Integration with Spring Boot JARs**

* **Direct execution** of existing JARs without modification.
* **Reduces engineering rework**.

**7. Conclusion & Recommendation**

|  |  |  |
| --- | --- | --- |
| Factor | AWS Glue | EMR Serverless |
| **Cost (500 jobs/day, 200GB each)** | $198,000/month | $130,200/month (34% savings) |
| **Performance (Shuffle & Joins)** | Slow, expensive | Faster, optimized |
| **Real-time Workload Support** | Not supported | Supported (Kafka, MQ, DataX, etc.) |
| **Pre-built Spring Boot JARs** | Requires adaptation | Runs existing JARs seamlessly |

**Final Recommendation**

Migrate from **AWS Glue to EMR Serverless** to achieve:

* **34% cost reduction ($67,800/month savings)**.
* **Faster execution (~20% improvement)**.
* **Better scalability & shuffle handling**.
* **Use EMR Serverless for real-time processing & event-driven workloads**.
* **Leverage existing Spring Boot Spark JARs without modification**.

**8. Next Steps**

* **Validate migration strategy** for existing workloads.
* **Fine-tune EMR Serverless** resource allocation.
* **Implement real-time data pipelines** using event-driven Spark.

**9. Supporting References**

* AWS Glue vs. EMR Serverless: A Comparison
* GoDaddy Case Study: Cost and Performance Optimization
* AWS QuickSight Dashboard for AWS Glue Metrics