Big Data Analytics made simple and easy with Amazon EMR

Easily scale and query with Apache Spark, Hive, Presto, and others

Shwetha Radhakrishnan
Solutions Architect
shwrad@amazon.com

Manpinder Singh Panesar
Solutions Architect
panesam@amazon.com
Trends and challenges in Big Data
Customers want more value from data

Growing Exponentially

From new sources

Increasingly diverse

Used by people with diverse skillsets

Accessed by many applications
Self managing analytics services is time consuming, complex, and expensive.

**Non Scalable Infrastructure**

On-premise big data deployments are complex to install, manage and scale.

**High Cost**

Advanced features such as centralized data catalog, handling streaming data need to be licensed separately.

**Lock in**

Data stored can only be consumed by the proprietary platform and needs to be duplicated to extend it to other tools.
What is Amazon EMR and how do customers use it?
Amazon EMR

Big Data analytics using open-source frameworks: Apache Spark, Presto, Trino, Hive, HBase, Hudi and Flink

Differentiated performance for Runtimes
Performance optimized runtime for popular frameworks like Spark, Hive, Presto, and Flink with 100% open source API compatibility

Latest open source features
New open source features available within 30-60 days of release in open source

Best price performance for big data analytics
Reduce cost using EC2 Spot, EMR Managed Scaling and per-second billing

Self service data science
Data Science IDE with EMR Studio and Deep integration with Sagemaker Studio provides ability to use open source UX and frameworks to build, visualize and debug applications

Multiple deployment models
EMR provides flexibility to run big data workloads on EC2, EKS, EMR Serverless, and on-premises with Outpost

S3 Data Lake Integration
Fine grained access controls with AWS Lake Formation and Apache Ranger, and Integrations with Apache HUDI, Apache Iceberg, and Delta Lake to enable S3 data lake use cases
Why do customers use Amazon EMR?

- To build Data Lakes as part of modern data architecture for scalable data analysis
- To query petabytes of data both in batch and real-time using Apache Spark, Hive, and Presto
- To migrate from expensive Big Data solutions to reduce costs and gain flexibility
- To gain insights using BI tools and prepare data for Machine Learning
- To build big data applications using Notebooks and leverage other AWS Analytics services
Lower costs with Amazon EMR

LOWER TCO

On-premises

Support costs

Server costs
Hardware–Server, Rack, Chassis, PDUs, Tor Switches (+Maintenance)
Software–OS, virtualization licenses (+Maintenance)

Network costs
Network hardware – LAN switches, Load Balancer bandwidth costs
Software–Network Monitoring

IT labor costs
Server admin, virtualization admin, storage admin, network admin, support team

Extras
Project planning, advisors, legal, contractors, managed services, training, cost of capital

Amazon EMR

Subscription fee
Support costs

• Less admin time to manage, and support Hadoop clusters
• No up-front costs – hardware acquisition, installation
• Save on operating costs – data center space, power, cooling
• Business value: Cost of delays, risk premium, competitive abilities, governance, etc.
High impact results with Amazon EMR

- near real-time analytics for 250M players
- scales 3,000 transient clusters on a daily basis
- powers the Predix solution processing 1,000,000 data executions/day
- achieves costs savings of 55% when compared to on-demand pricing and 40% savings when compared to Reserved Instances
- computes Zestimates on 100M +homes in hours instead of 1 day
Challenge:
FINRA's legacy system was not able to scale to handle 150 billion events per day. They needed to run complex surveillance queries over 20+ PB of data to detect and analyze illegal market activity.

Solution:
FINRA migrated their big data appliance to a S3 Data Lake and uses Lambda and EMR for data ingestion and EMR and Redshift for data processing.

Result:
FINRA has been able to increase agility, speed, and cost savings while allowing them to operate at scale. The company estimates it will save $10 to $20 million annually by using AWS.
Glue Vs. EMR

Data Integration and ETL on AWS

AWS Glue

- Data Catalog
- Data Crawlers, bookmarks, small file handling
- Glue Workflows (data pipeline)
- Glue Studio – WYSIWYG ETL Development
- Support for native and 3rd party data connectors
- PII Detection and duplicate record match

Amazon EMR

- Big data analytics
- Open source framework support – Hadoop, Spark, Presto, Trino, Hive, HBase, Flink, Pig, Hue, TensorFlow etc.
- Multiple deployment options EC2, EKS and Serverless
- Support wide array of Instance types
- Support for latest versions of OSS within weeks
What’s New in Amazon EMR
Capabilities available to reduce costs

With EMR, you can do way more with way less!

Performance optimizations
- Runtime improvements
- Transactions in data lakes

Compute optimizations
- Graviton instances
- Spot instances
- Instance fleets

Cluster management
- Managed Scaling
- Cluster auto-termination
Amazon EMR innovates constantly in these areas

- Cost & Performance
- Ease of use
- Transactional Data lakes
- Security
Differentiated Spark runtime performance

**Over 3X faster than standard Apache Spark 3.0 in Derived TPC-DS 3TB benchmark**

---

**Takes advantage of AWS native Graviton2 instances to provide the best performance**

---

**100% compliance with open source APIs makes moving applications to EMR easy**

---

**Performance improvements are enabled by default**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic Sized Executors</td>
<td></td>
</tr>
<tr>
<td>Adaptive Join Selection</td>
<td></td>
</tr>
<tr>
<td>Dynamic Pruning of Data Columns</td>
<td></td>
</tr>
<tr>
<td>Operator Optimization</td>
<td></td>
</tr>
<tr>
<td>Early Worker Allocation</td>
<td></td>
</tr>
<tr>
<td>Intelligent Filtering</td>
<td></td>
</tr>
<tr>
<td>Parallel/Async Initialization</td>
<td></td>
</tr>
<tr>
<td>Redundant Scan Elimination</td>
<td></td>
</tr>
<tr>
<td>Broadcast Join w/o Statistics</td>
<td></td>
</tr>
<tr>
<td>Stats Inference</td>
<td></td>
</tr>
<tr>
<td>Optimized Metadata Fetch</td>
<td></td>
</tr>
<tr>
<td>Data Pre-Fetch</td>
<td></td>
</tr>
<tr>
<td>Intelligent Filtering</td>
<td></td>
</tr>
<tr>
<td>Redundant Scan Elimination</td>
<td></td>
</tr>
<tr>
<td>Optimized Metadata Fetch</td>
<td></td>
</tr>
</tbody>
</table>
Managed Scaling feature overview

Automatically reduce cost by 60% shaping cluster size

- Constantly improving EMR managed algorithm that gives you a fully managed experience
- High Resolution Metrics enabled with Managed scaling
- Only min/max cost constraints configurations required
- More data points and faster reaction time than autoscaling
- Save 20-60% costs
Improvements in cluster startup times

Starting or scaling an EMR on EC2 cluster is now 35% faster

Task nodes provisioned alongside core and master nodes

Have ready to use proxy instances available to improve start time for cluster launched in private subnet

Optimized retry policies for EC2 throttling
Amazon EMR innovates constantly in these areas

- Cost & Performance
- Ease of use
- Transactional Data lakes
- Security
EMR Studio

FULLY MANAGED IDE FOR INTERACTIVE DATA ANALYTICS: DEVELOP, VISUALIZE, AND DEBUG APPLICATIONS

- Single sign-on integration with IdP
- Fully-managed Jupyter Notebooks
- Integrated with Git Repositories
- Simplified debugging with Spark UI and YARN UI
- Browse, create, or delete EMR clusters
- Run Notebooks in workflows using APIs
- Run interactive data analysis using EMR on EKS clusters
EMR Studio features

FULLY MANAGED IDE FOR INTERACTIVE DATA ANALYTICS: DEVELOP, VISUALIZE, AND DEBUG APPLICATIONS

- IAM authentication and federation support
- Multi-language support (R, PySpark, Scala, SQL)
- Auto-terminate idle clusters
- Real-time co-authoring of notebooks
- SQL explorer with Presto support
- Mount workspace directories to EMR clusters
- Latest JupyterLab, JEG, Livy, SparkMagic
Amazon EMR innovates constantly in these areas

- Cost & Performance
- Ease of use
- Transactional Data lakes
- Security
Transactional data lakes

CHOICE OF FRAMEWORK FOR EACH WORKLOAD

EMR 6.9 includes:

- Apache Hudi 0.12
- Apache Iceberg 0.14.1
- OSS Delta Lake 2.1.0
Transactional data lakes features

Transactions, Record-level Updates/Deletes, and Change Streams to Data Lakes

**Ingestion**
- Transactions (ACID) - Reader and writer isolation
- Transactions (ACID) - Concurrent write support
- Record level upserts and deletes
- High throughput streaming ingestion
- Spark, Flink, and Java Writer Support
- Automatic compaction of small files
- SQL DML support

**Query**
- Spark, PrestoDB/Trino, Flink, Hive Support
- Efficient queries across partitions and files
- Incremental query support
- Time travel query support
Transactional data lakes features

TRANSACTIONS, RECORD-LEVEL UPDATES/DELETES AND CHANGE STREAMS TO DATA LAKES!

- Async background compaction of files
- Async background sorting and clustering of keys
- Automatically clean up files beyond retention period
- Metrics for past commits or rollbacks

Administration
EMR 6.9 includes Hudi 0.12.

Key new features include:

- **Multi-modal indexes**: Improve the lookup performance in file index and query latency with data skipping
- **Async indexer service**: Index columns in the background without affecting writes
- **Schema-on-read for Spark**: Improved Schema evolution support
Apache Iceberg

Open Table Format for Huge Analytic Datasets

Apache Iceberg 0.14.1 is packaged as a library for Spark3 Runtime, Trino, Flink, and Hive in EMR 6.9.0.

Key new features include:

- Time travel support with Spark SQL and Trino SQL
- Merge on Read (MoR) support
- Optimistic concurrency with AWS Glue Data Catalog
- Disaster recovery with S3 access points
- Flink and Hive integration (EMR 6.9.0)
OSS Delta Lake

OPEN-SOURCE STORAGE FRAMEWORK THAT ENABLES BUILDING A LAKEHOUSE ARCHITECTURE

OSS Delta Lake 2.1.0 is packaged as a library in EMR 6.9.0.

Engines supported: Spark3 and Trino

To learn more:

Amazon EMR innovates constantly in these areas

- Cost & Performance
- Ease of use
- Transactional Data lakes
- Security
Security pillars:

- Isolation
- Authentication
- Authorization
- Encryption
- Audit

VPC
- Private subnets
- Security groups

LDAP
- Kerberos

AWS IAM Identity Center (EMR Studio)

Cluster IAM Role
- FGAC using Apache Ranger
- FGAC using AWS Lake Formation
- Job runtime role

Encryption at rest
- Encryption in transit

Audit using Ranger
- Audit using AWS Lake Formation

NEW!
Amazon EMR Deployment Options
EMR Deployment Options

Amazon EMR on Amazon EC2
Choose instances that offer the best price performance for your workload

Amazon EMR on AWS Outposts
Set up, manage, and scale EMR in your on-premises environments, just as you would in the cloud

Amazon EMR on Amazon EKS
Automate provisioning, management, and scaling of Apache Spark jobs on EKS

Amazon EMR Serverless
Run petabyte-scale data analytics in the cloud without managing and operating clusters
**Amazon EMR Serverless**

All the benefits of EMR without managing clusters and servers

- Run frameworks more easily; just pick a version and run
- Automatically scale; don’t guess cluster sizes
- Optimize cost; Automatic and fine-grained scaling reduces cost
- Performance-optimized version delivers 2x better performance
- Multi-AZ resiliency from day 1
- Integration with familiar tools like Apache Airflow
Jobs
- Run jobs on applications
- Can run multiple jobs on an application
- Can control authorization using per-job execution role

Workers
- Internally used to execute your workloads
- Workers run the OSS framework you choose
- You can change the size of workers to control performance

Pre-initialized workers
- Optional feature to pre-initialize workers
- Jobs start immediately
- Helps you maintain a warm pool
<table>
<thead>
<tr>
<th><strong>AWS Glue</strong></th>
<th><strong>Amazon EMR Serverless</strong></th>
<th><strong>Amazon EMR on EC2</strong></th>
<th><strong>Amazon EMR on EKS</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Serverless</td>
<td>Yes (Acquires capacity for jobs at launch)</td>
<td>Yes (Virtual cluster model, with pre-initialized capacity)</td>
<td>No</td>
</tr>
<tr>
<td>Frameworks</td>
<td>Spark and Python</td>
<td>Spark, Hive, Trino (coming soon)</td>
<td>Spark, Hive, Trino, Hbase, Flink, …</td>
</tr>
<tr>
<td>Pricing unit of measure</td>
<td>DPUs (4vCPU/16GB)</td>
<td>Workers aggregated consumed resources (aggregated vCPU, memory, storage)</td>
<td>EC2 instances + EMR price (per instance) + EBS</td>
</tr>
<tr>
<td>Startup time</td>
<td>~10 seconds</td>
<td>~2 minutes OR ~few seconds if pre-initialized capacity</td>
<td>~5 minutes with only Spark installed (no kerberos) (this considers the time to create the cluster as subsequent jobs submissions are almost immediate)</td>
</tr>
<tr>
<td>Scaling</td>
<td>Fully managed (both for batch and streaming)</td>
<td>Fully managed</td>
<td>Autoscaling with custom policies (Cloudwatch metrics based) OR Managed scaling (only YARN based applications, fully-managed)</td>
</tr>
<tr>
<td>Interactive Analysis</td>
<td>Glue interactive sessions, Glue Studio, Glue Notebooks, SageMaker Studio and desktop-based Notebooks</td>
<td>EMR Studio</td>
<td>Jupyterhub, Hue, Zeppelin, EMR Studio, SageMaker Studio and desktop-based Notebooks</td>
</tr>
</tbody>
</table>
Amazon SageMaker + Amazon EMR
Amazon SageMaker helps organizations harness ML

**Amazon SageMaker**

- Infrastructure, tools, visual interfaces, workflows, orchestration, and collaboration

**Business analysts**

Make ML predictions using a visual interface with Amazon SageMaker Canvas

**ML engineers**

Deploy and manage models at scale with Amazon SageMaker MLOps

**Data scientists**

Prepare data and build, train, and deploy ML models with Amazon SageMaker Studio
Amazon SageMaker Studio Notebook + EMR

- Discover, connect to, create and terminate EMR clusters (Hive, Spark and Presto)
- Collaborate using Scala-based Spark and PySpark notebook kernels
- Bring your own image and customize notebook lifecycle configuration
- Enforce fine-grained data access
- Automate EMR, Glue and ML pipelines in production

SageMaker Studio Notebook

EMR
SageMaker
Lake Formation (Governance)
Thank you

Manpinder Singh Panesar
Solutions Architect
panesam@amazon.com

Shwetha Radhakrishnan
Solutions Architect
shwrad@amazon.com