

AWS
re:Invent

A N T 3 4 1

Amazon EMR: Optimize Transient Clusters for Data Processing & ETL

Anthony Virtuoso
Principal Engineer
AWS

Eric Mills
Senior Software Engineer
AWS EMR

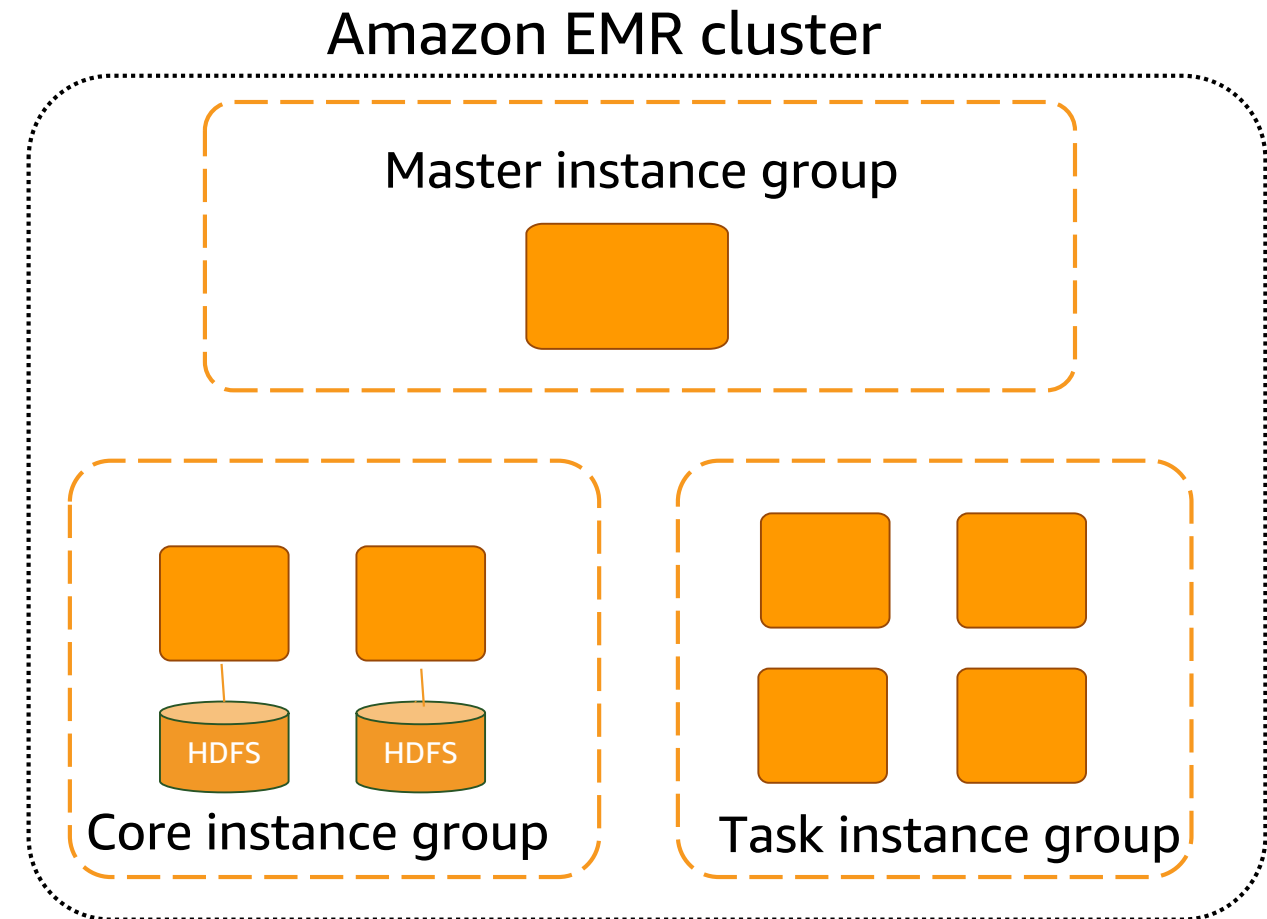
Esther Kundin
Senior Software Engineer
AWS EMR

Agenda

- Stateless clusters
- Scaling clusters
- Reducing costs
- Cluster orchestration

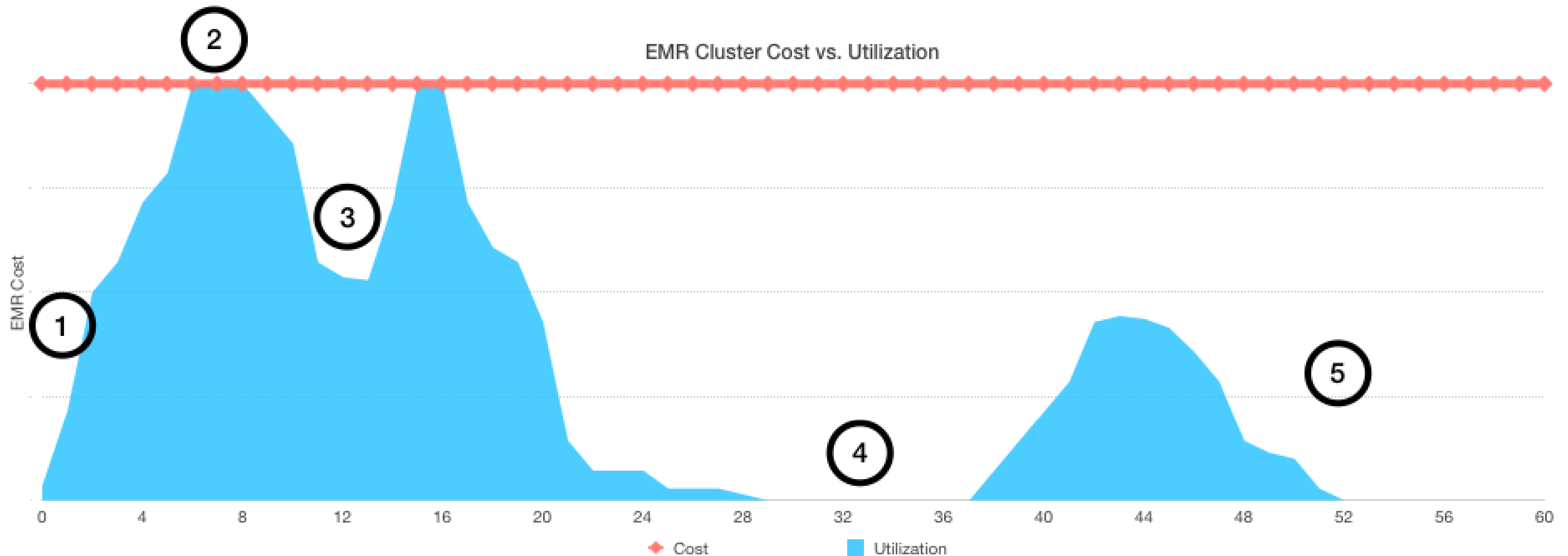
Amazon EMR basics

- **Master node**
 - Manages cluster
 - NameNode and JobTracker
- **Core nodes**
 - Task tracker (compute)
 - DataNode (HDFS)
- **Task nodes**
 - Task tracker only
 - No HDFS



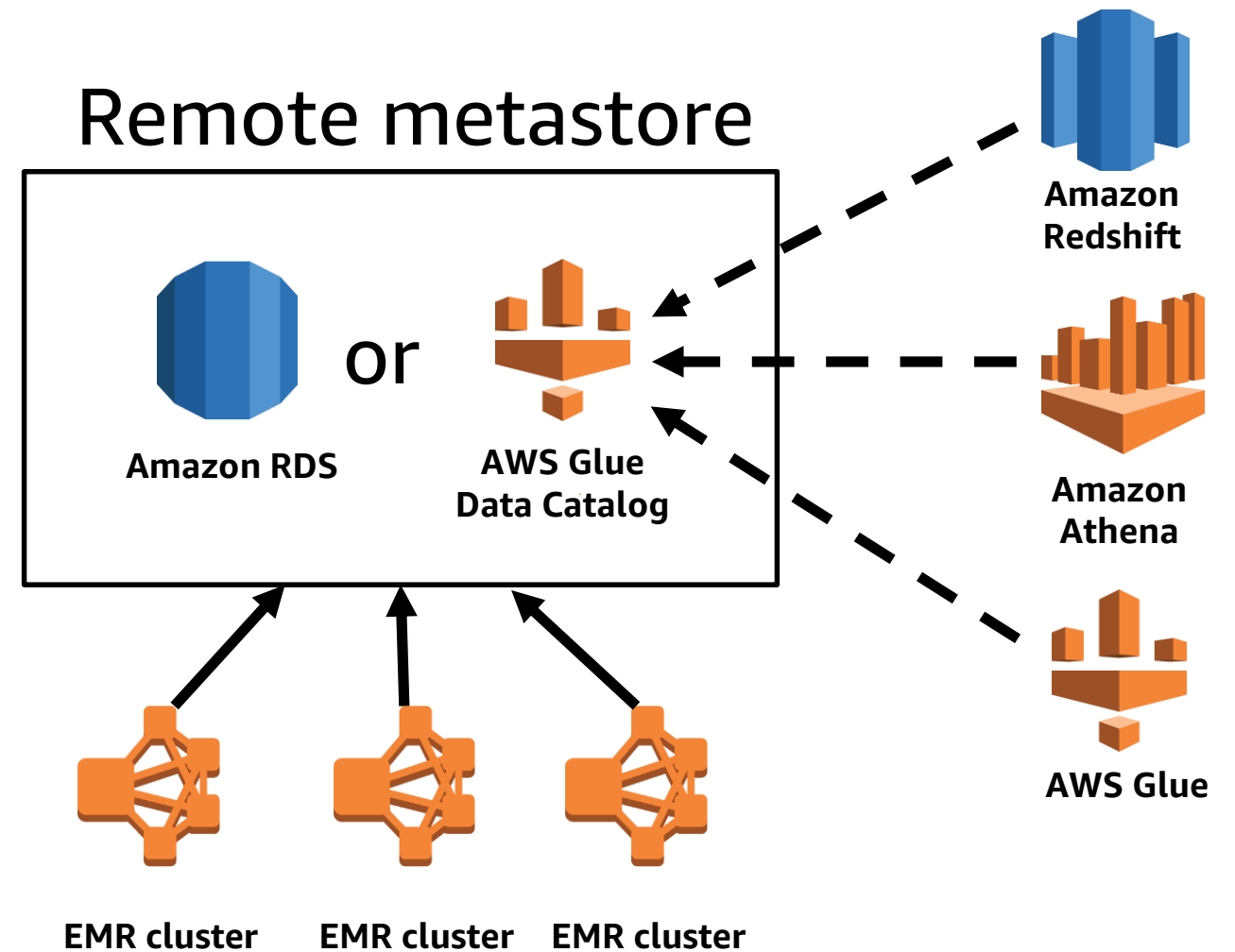
Example cluster

- Mixed workloads
- Capacity exhausted at peak
- Paying for idle time



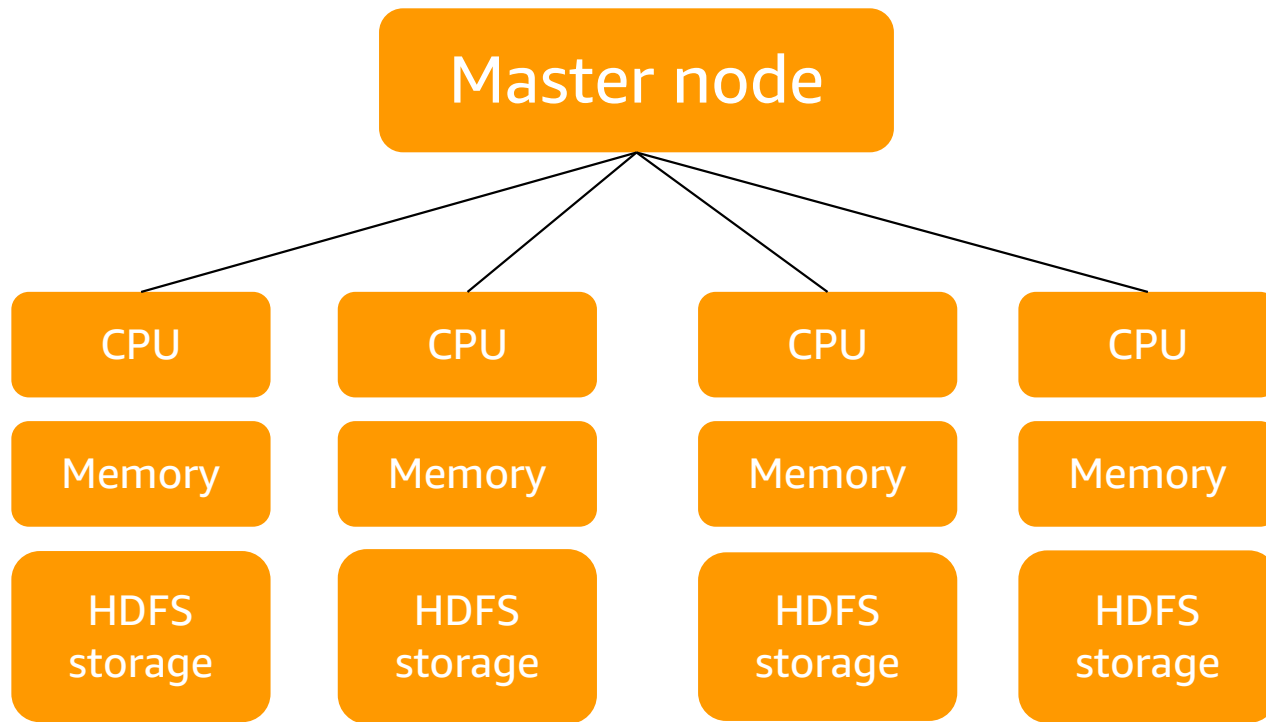
Making our cluster stateless

- Maintain metastores off cluster
- Faster startup time lowers cost



Making our cluster stateless

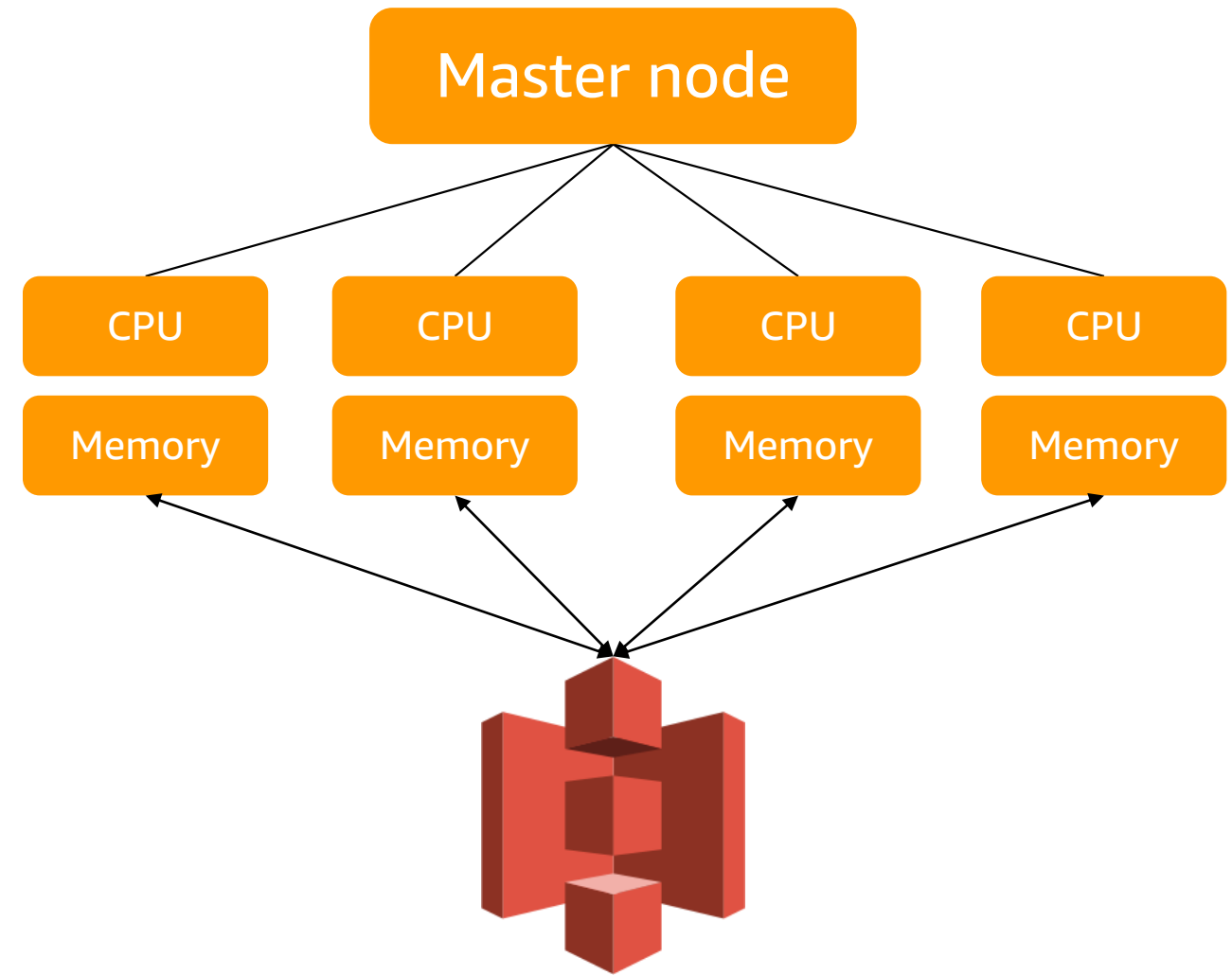
Old clustering/Localized model



HDFS has 3x replication

500 TB dataset equals 1.5 PB cluster with replication

Amazon EMR decoupled model



Amazon S3 as streaming HDFS through EMRFS

Scaling our cluster

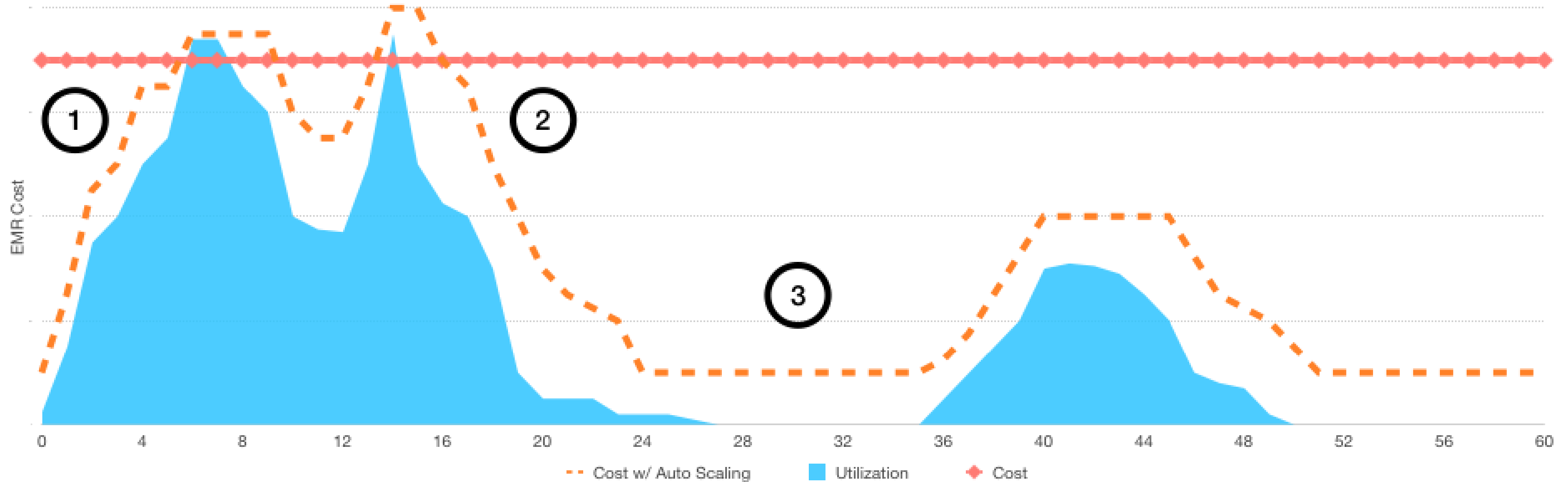
- Scale out or in cluster task instances using automatic scaling
 - Control through policies which monitor Amazon CloudWatch metrics
 - Popular metrics include 'YARNMemoryAvailablePercentage' and 'ContainerPendingRatio'



Example cluster with Auto Scaling

- Clusters adapt to demand needs
- Pay less for idle time
- Peak throughput: finish faster

EMR Cluster Cost vs. Utilization

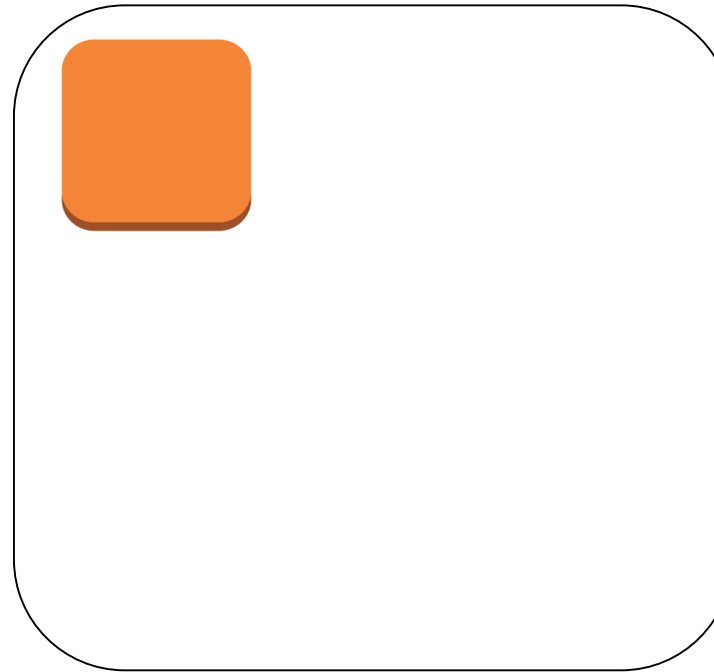


Turning on Spot Instances

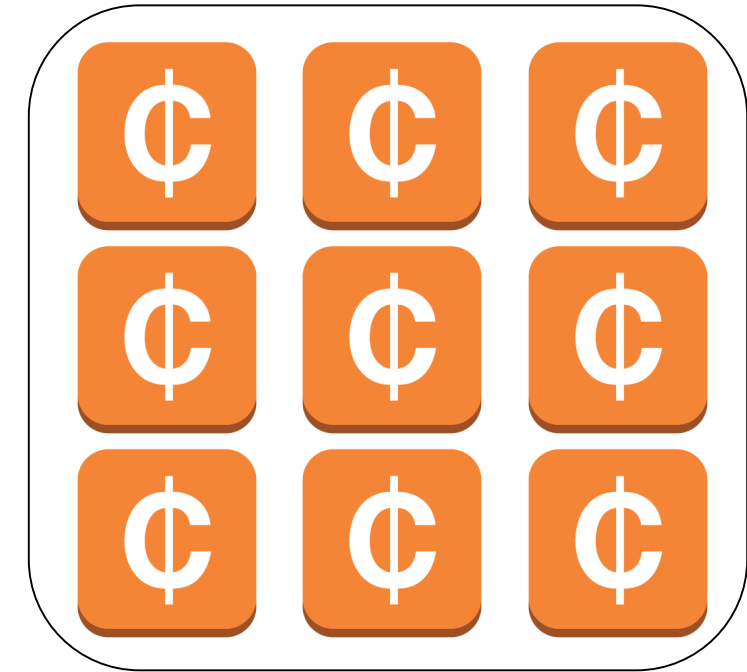
Master node



Core instances



Task instances



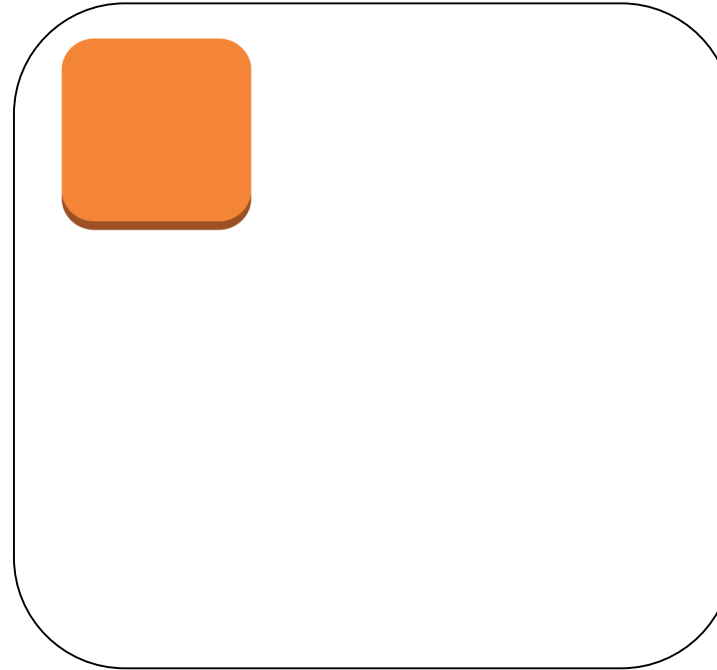
- Master node and at least one core node should be on-demand
- Launch clusters in the optimal availability zone based on capacity and price

Instance fleets for advanced Spot provisioning

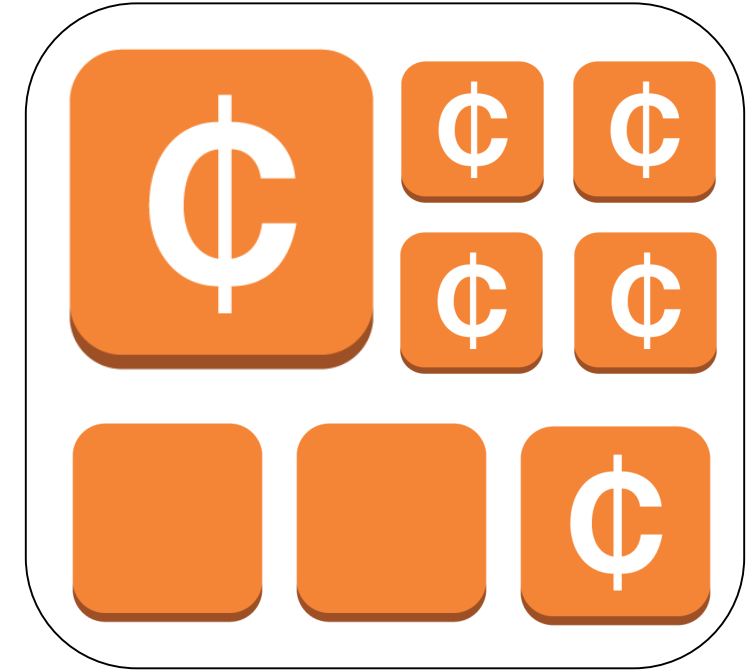
Master node



Core instances



Task instances

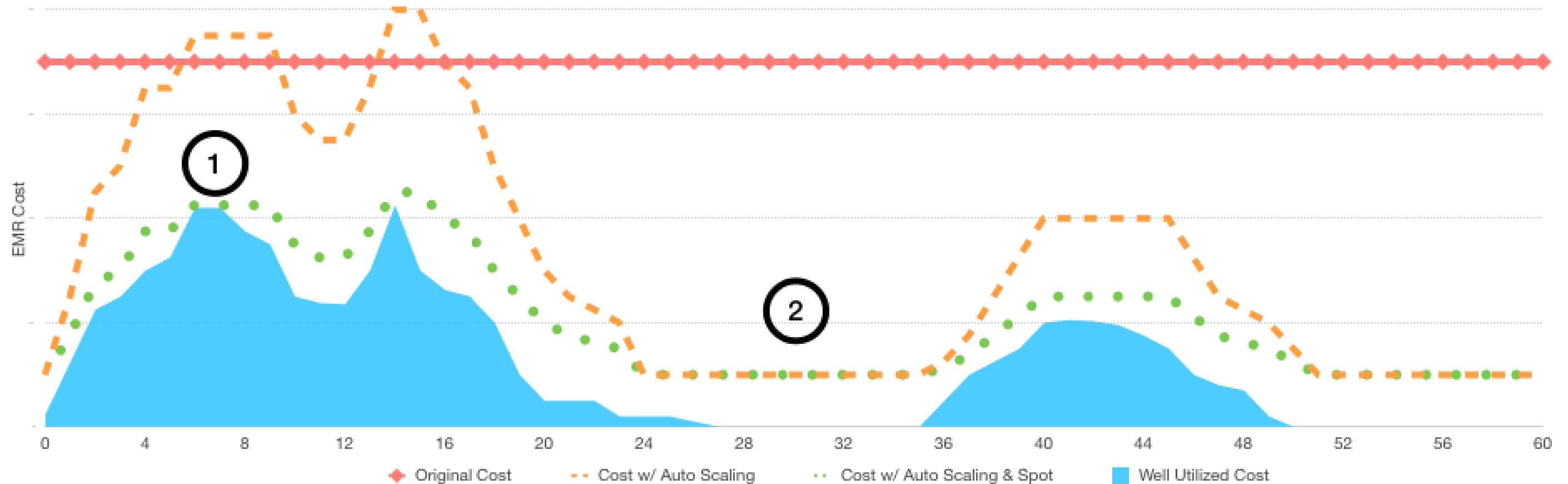


- Mix and match instance types and Spot versus on-demand

Example cluster with Spot Instances

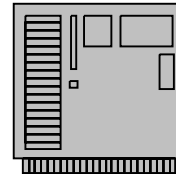
- Spot Instances can further reduce costs

EMR Cluster Cost vs. Utilization

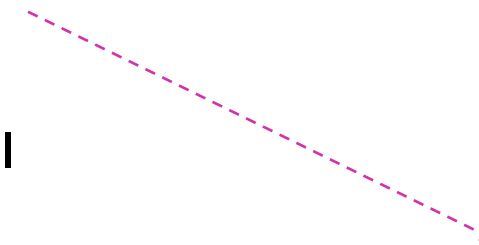


Options for orchestrating a cluster

Configure a cluster and launch a job using Step API



Amazon EMR JobFlow/Step API



Use AWS Lambda to launch clusters using the Amazon EMR Step API



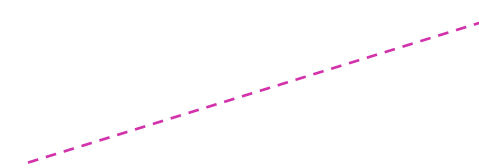
AWS Lambda



Create a pipeline to schedule cluster creation and job scheduling



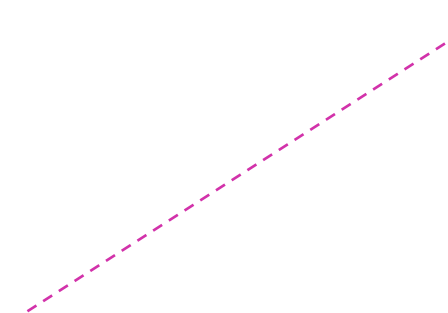
AWS Data Pipeline



Use an external orchestration system to launch clusters and jobs



Airflow, Luigi, or other schedulers on
Amazon Elastic Compute Cloud (Amazon EC2)



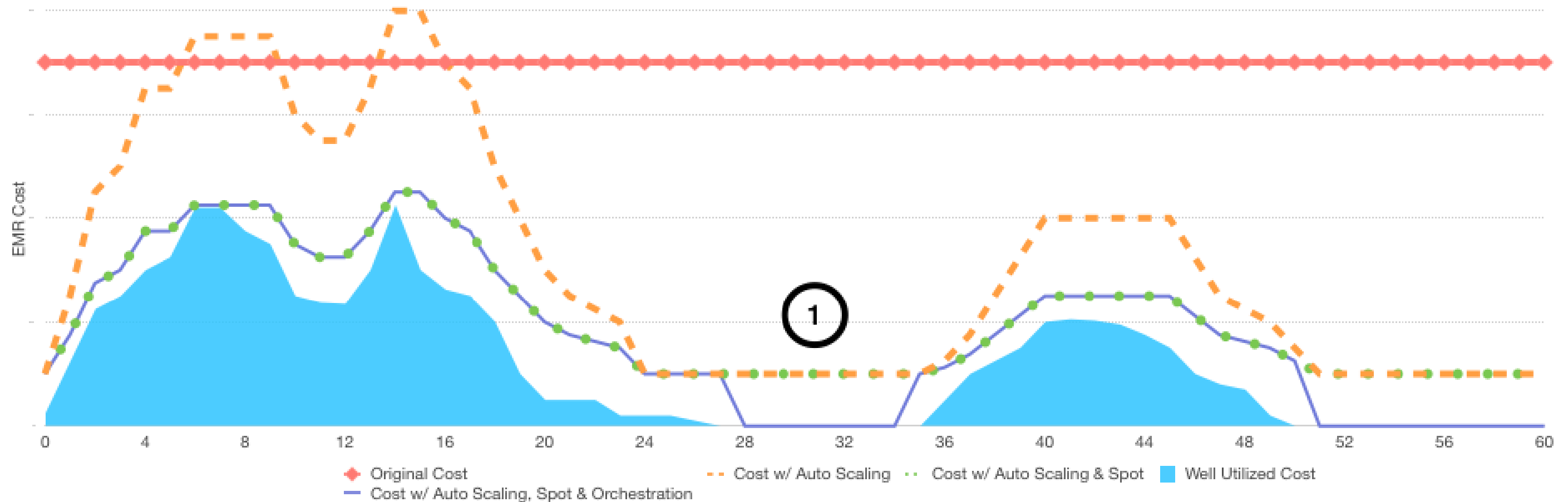
Amazon EMR

Use Oozie on your cluster to build DAGs of jobs

Example cluster with orchestration

- Stopping idle clusters lowers costs

EMR Cluster Cost vs. Utilization



Takeaways

- Use Amazon Simple Storage Service (Amazon S3) for storage
- Use Amazon Aurora or AWS Glue for remote metastore
- Auto Scaling for task instances
- Use Spot Instances for lower costs
- Interact with clusters and submit steps with JobFlow/Step API, AWS Data Pipeline, or Airflow on Amazon EC2

Questions?

Thank you!

Anthony Virtuoso
Eric Mills
Esther Kundin



Please complete the session
survey in the mobile app.

Appendix: Job optimization

Hadoop

- Compress input and output data
- Adjust number of mappers and reducers
- Skewed joins
- Tez

Spark

- Dynamic allocation settings
- RDD reuse
- Correct join type—broadcast join
- <https://spark.apache.org/docs/latest/tuning.html>

