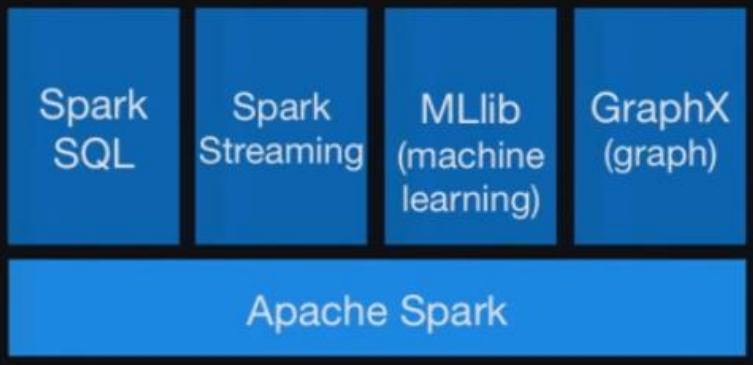
FINRA is a leader in the Financial Services industry who sought to move toward real-time data insights of billions of time-ordered market events by migrating from SQL batch processes on-prem, to Apache Spark in the cloud. By using Apache Spark on Amazon EMR, FINRA can now test on realistic data from market downturns, enhancing their ability to provide investor protection and promote market integrity (FINRA enacts rules and provides guidance that securities exchanges & brokers must follow). By using AWS Spot instances, FINRA has saved up to 50% from its on premises solution, increased elasticity/scalability, and accelerated reprocessing requests (from months to days). Learn best practices on how FINRA moves toward real-time data analytics with Spark and AWS, while managing production workloads in parallel, increasing performance and IT efficiency, reducing cost, and modernizing and scaling their infrastructure to prepare for real-time processing in the future.

AWS – Spark Overview

What is Spark?

- A fast and general engine for large-scale data processing
- Write applications quickly in Java, Scala, Python, R
- Run programs up to 100x faster than Hadoop MapReduce in memory, or 10x faster on disk



To learn more, check out http://spark.apache.org

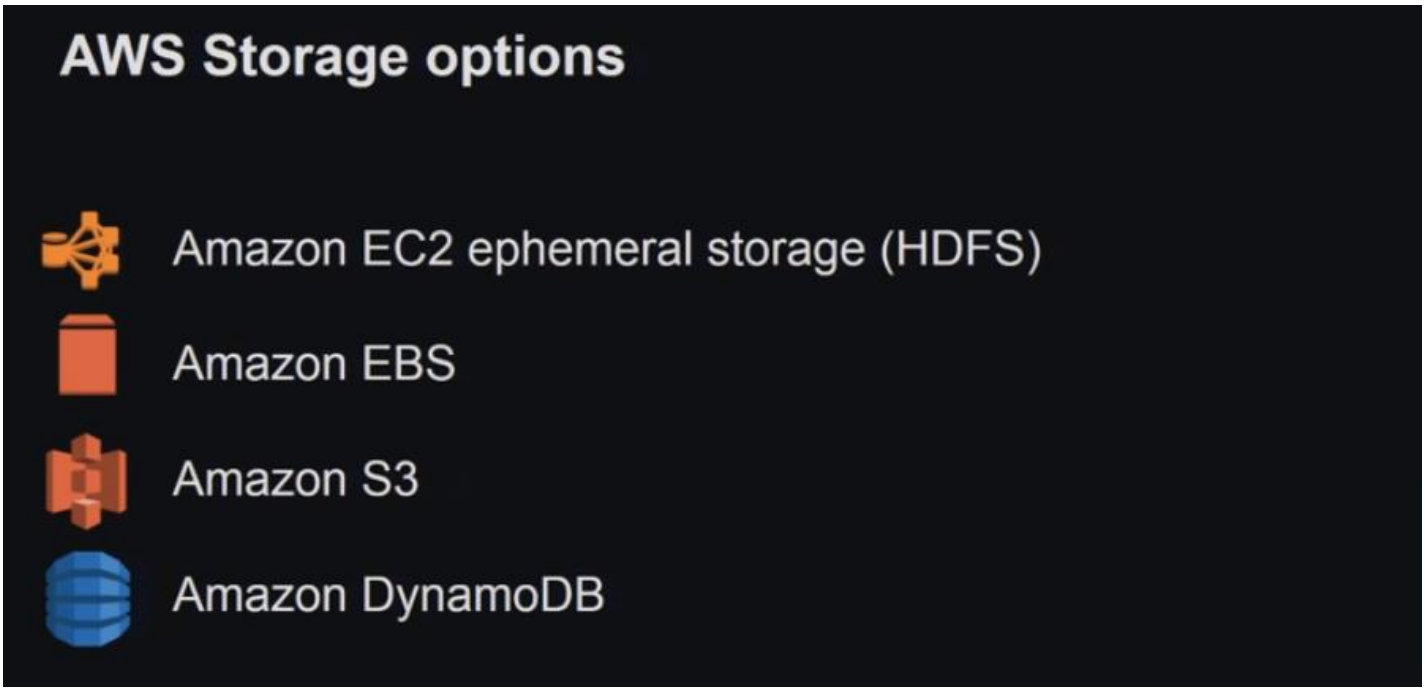
4

Spark on AWS Compute

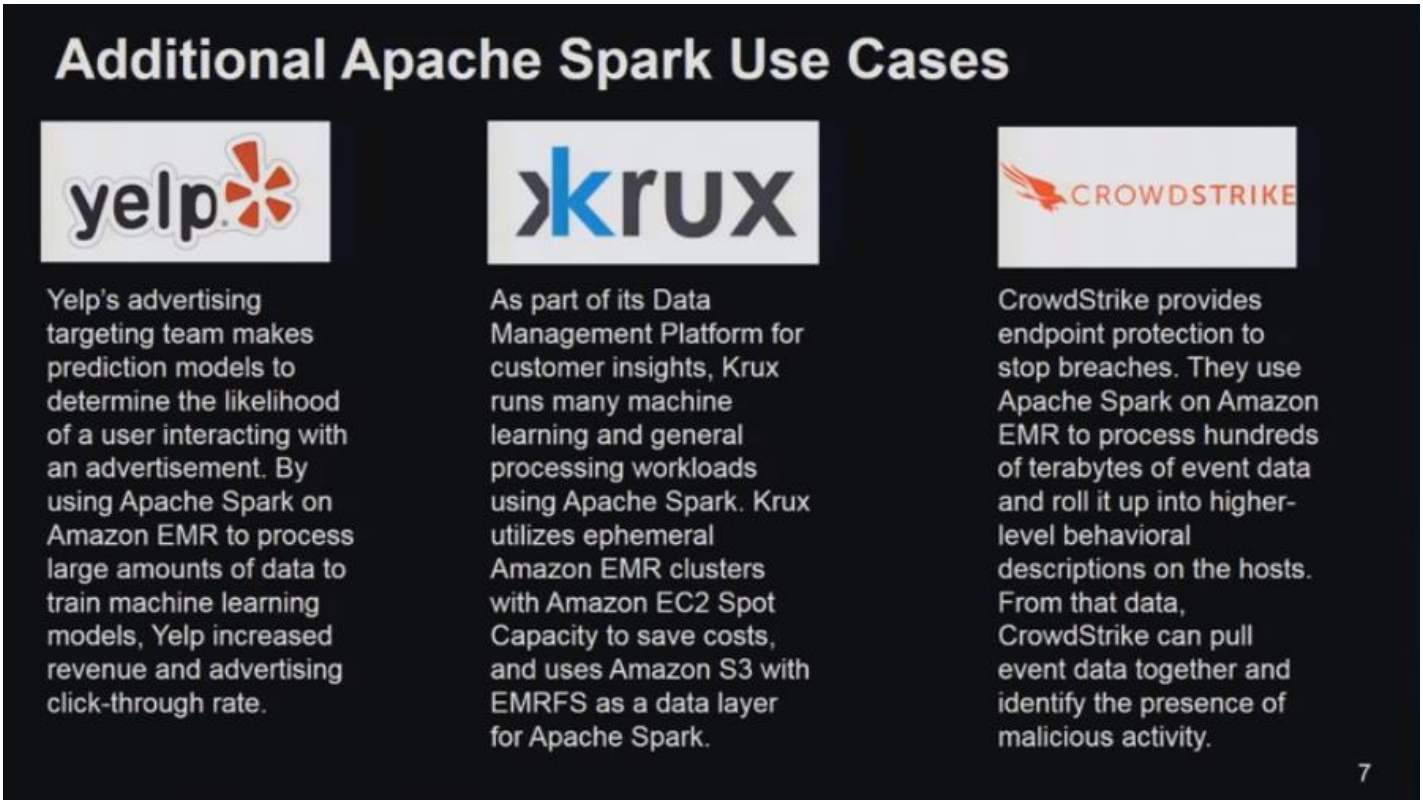
- Amazon EC2
  - Standalone cluster mode
  - Apache Mesos

- Amazon EMR
  - YARN

You can install and run Spark on EC2 instances that you spin up and use Mesos as the job scheduler. Spark is a first-class citizen on EMR and you can specify that you want to install Spark on your EMR cluster and it will use the YARN job scheduler



Spark supports many types of storage for doing data processing, some are listed above. You can run HDFS on ephemeral storage on your EC2 instances, you can also use EBS and attach them to your EC2 instances, S3 can also be used to store your data for your Spark jobs, DynamoDB can also be used also. You can write your Spark jobs and queries workloads to put data in any of the above listed data sources.



Spark on AWS Best Practices

Spark, Presto, Hive, Pig are all big data processing engines that you can use to run your workloads in AWS.

Separate Storage & Compute

- Optimize cluster size based on compute requirements
- Allows selection of optimal EC2 instance types
- Shut down your cluster when not in use
- Share data among multiple clusters
- Fault tolerance and disaster recovery

You can select R3 compute instances to get the best performance out of your job while leaving your data in S3, you can use S3 as your data lake. EMR comes with EMRFS that uses S3 like your local file system. You simply read your data from S3 and write to S3, this allows you to shut down your cluster when needed.

Spot instances ¢

- Bid on spare Amazon EC2 computing capacity
- Reduce operating costs by up to 50-90%
- Ideal for ephemeral compute clusters

FINRA: Spark Case Study

What Do We Do?

By the Numbers

- We oversee more than 3,900 securities firms with approximately 640,795 brokers.
- Every day, we watch over nearly 6 billion shares traded in U.S. listed equities markets—using technology powerful enough to detect potential abuses.
- In fact, FINRA processes approximately 6 terabytes of data and up to 75 billion events every day to build a complete, holistic picture of market trading in the U.S.
- In 2015, FINRA:
  - Referred over 800 fraud cases for prosecution
  - Levied more than $191 million in fines & restitution

Problem Statement

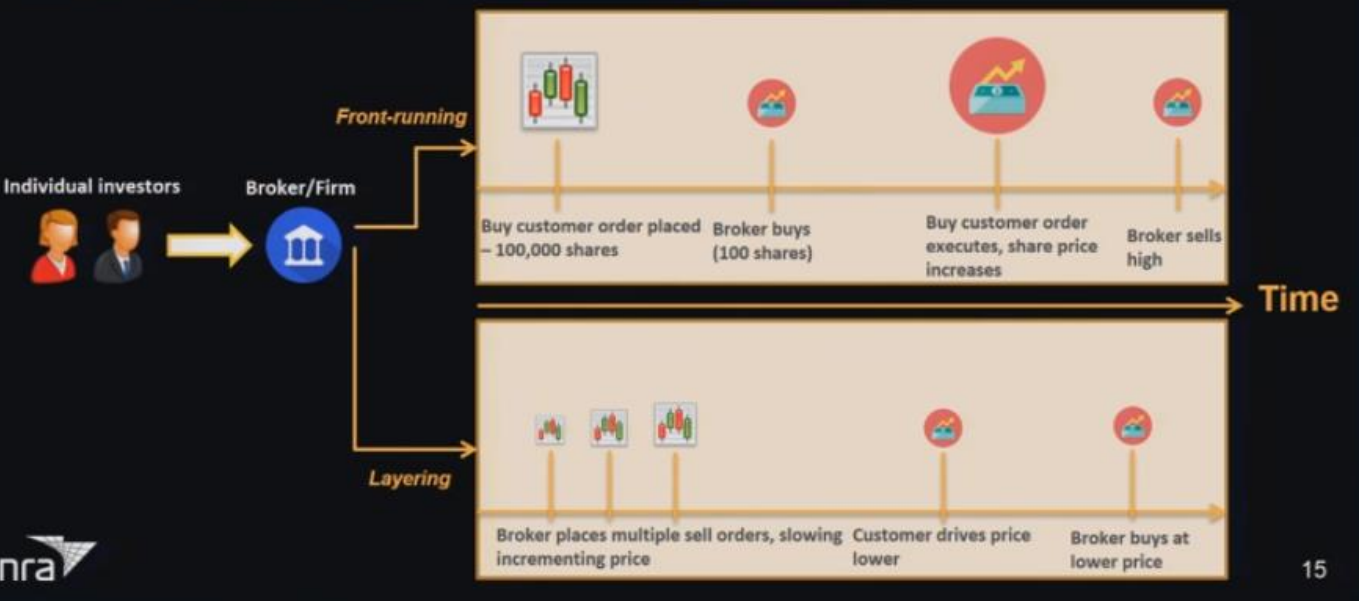
What data is FINRA tracking, and what do we do with it?

- Up to 75 billion events/day
- 12 markets/exchanges
- Surveillance – Compare the state of activity across different 'venues', such as quotation systems, exchanges, firms, or trade reporting facilities
  - Involves grouping events by stock symbol and venue then ordering by event time
- Alerts for 'venues of interest' where suspicious activity occurs

FINRA

Problem Statement (cont'd)

What qualifies as *suspicious* activity? Actions taken by traders/brokers which disadvantage customer orders or undermine fairness/efficiency of markets.



FINRA

Example 1 – Intermarket Price Protection

- ***Intermarket Price Protection*** - Restrict broker trading outside the best publicly available bid/offer to encourage best execution of customer orders

Best price = highest bid/lowest ask, broker is trading outside this band!

85K quotes/sec
< 1K trades/sec

Example 2 – Exchange side of the picture

Limit Order Display - Obligation of firm to **publish the full price/volume** of its received customer limit orders in an exchange where the firm is registered as a market-maker

Intra-day Exchange Feeds

- Exchange 1
- Exchange 2
- Exchange 3
- Exchange 4
- etc

4 billion order events/day across 11 exchanges

Best price per exchange – 'top of order book' – BUY side

Exchange #2

Time	Ticker	Price
13:00:00	STOCK XYZ	$57.42
13:00:00.1	STOCK XYZ	$57.41
13:00:00.156	STOCK XYZ	$57
13:00:00.243	STOCK XYZ	$57.10
13:00:00.244	STOCK XYZ	$57.11
13:00:00.260	STOCK XYZ	$57.15
13:00:00.287	STOCK XYZ	$57.92
13:00:00.29	STOCK XYZ	$57.90
13:00:00.293	STOCK XYZ	$57.40
13:00:00.297	STOCK XYZ	$57.41

90K order events/second

Universal best size/price

Time	Ticker	Exchange	Price	Size
13:00:00	STOCK XYZ	EX 1	$58	1000
13:00:00.1	STOCK XYZ	EX 2	$57.41	1200
13:00:00.156	STOCK XYZ	EX 1	$57.50	800
13:00:00.243	STOCK XYZ	EX 3	$57.13	700
13:00:00.244	STOCK XYZ	EX 4	$57.18	1200
13:00:00.260	STOCK XYZ	EX 2	$57.15	900

Accumulate quantity at the best price

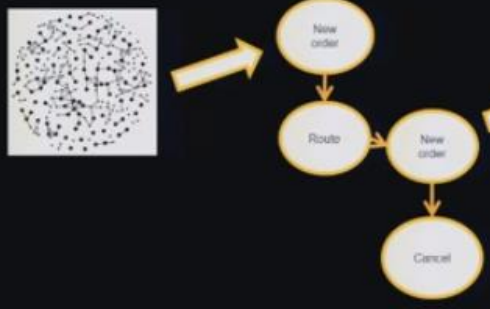
Best price at Exchange 2 at this point in time

Best universal price (Exchange 1)

Example 2 – Customer order side of the picture

4-5 billion customer orders/day

50 million displayable orders/day



BUY Limit Order Size/Price

Time	Firm	Size	Pr
13:00	FIRM ABC	1000	$60
13:05	FIRM ABC	0	$60
13:06	FIRM ABC	1000	$60

Published best size/price

Time	Firm	Ticker	Exchange	Price	Size
13:00:00	FIRM ABC	STOCK XYZ	EX	$58	1000
13:00:00.1	FIRM ABC	STOCK XYZ	EX 2	$57.41	1200
13:00:00.156	FIRM ABC	STOCK XYZ	EX 1	$57.50	800
13:00:00.243	FIRM ABC	STOCK XYZ	EX 3	$57.13	700
13:00:00.244	FIRM ABC	STOCK XYZ	EX 4	$57.18	1200
13:00:00.260	FIRM ABC	STOCK XYZ	EX 2	$57.15	900

At this point, the exchange price is worse than the customer's order, so in violation!

FINRA

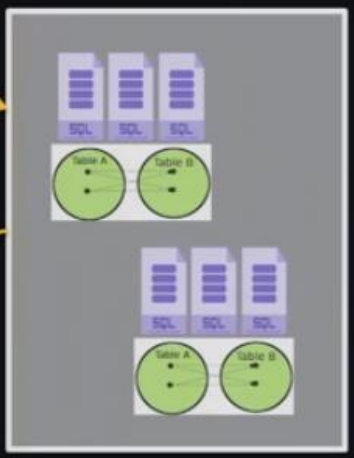
Legacy Solution

DR – Disaster Recovery

Production DB Architecture

Proprietary processors
380 cores, 380 TB disk

> 300 SQL-based surveillances



Data off-loading to make way for current data/processing

Data loads run day/night

FINRA

Pain Points

Pain points with the legacy solution – no longer present in AWS/Spark architecture

- 7 figures yearly to maintain and operate
- Difficult to scale during periods of market volatility
- Storage and compute tightly coupled
- Won't support industry movement toward real time
- Reprocessing is difficult (takes months)
- Unavailable during maintenance windows
- No control of managing internal compute model

FINRA

21

Alternatives

Key requirements of new solution:

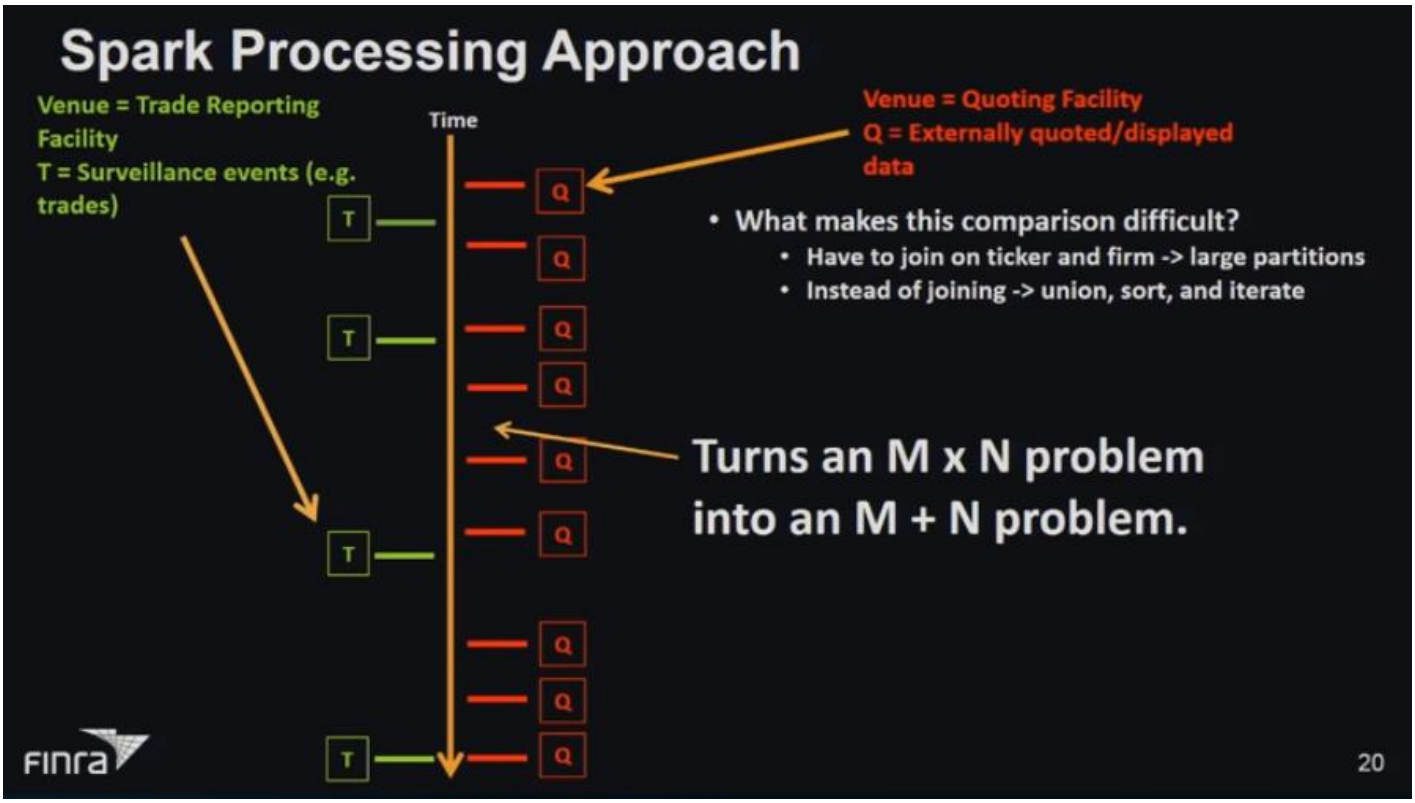
- Scalability / elasticity
- Cost effectiveness
- Ease of coding and testing
- Platform for future real-time processing
- Support for time-based iteration

What options were considered?

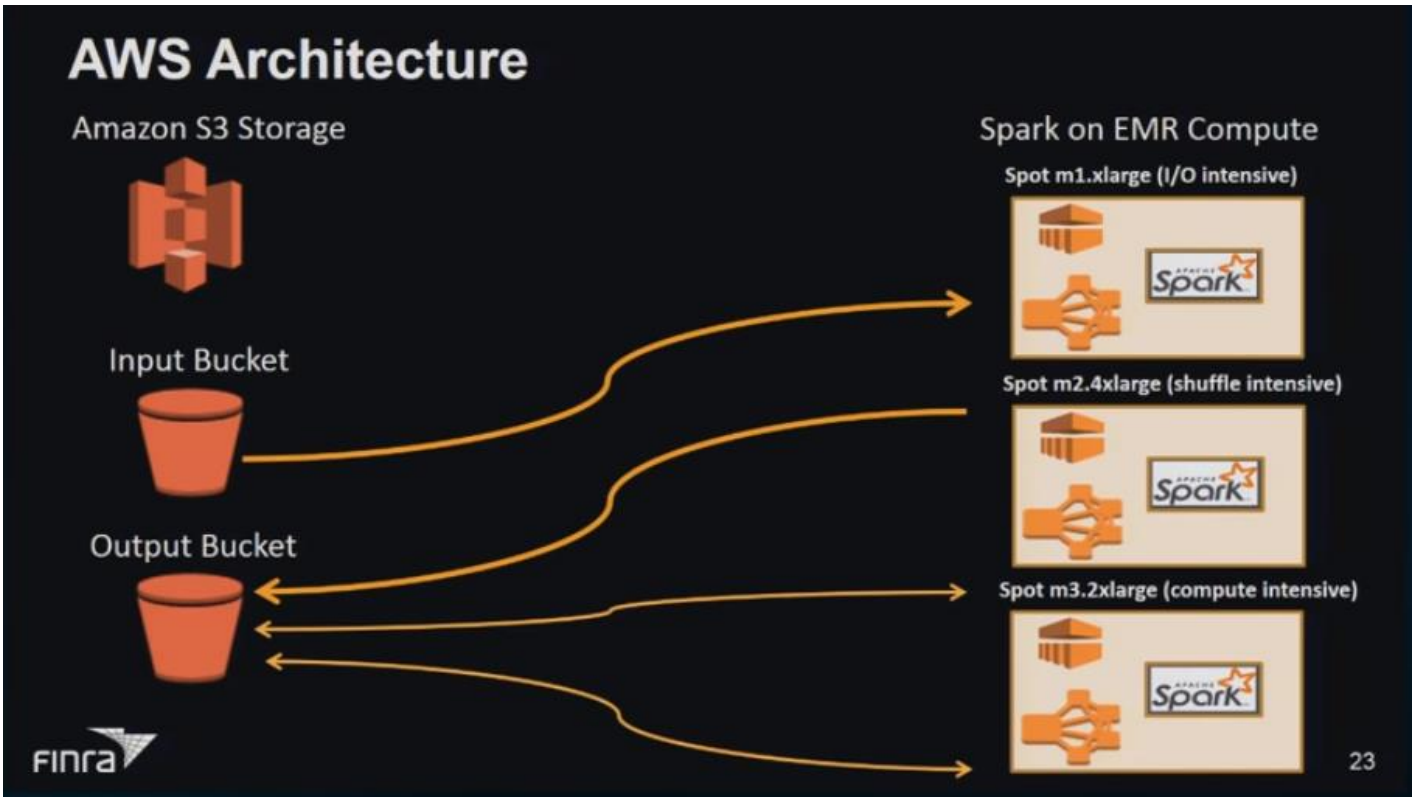
- Apache Spark on Amazon EMR
- Java MapReduce
- Apache Giraph
- Apache Crunch

FINRA

We decided to use AWS and choose to run AWS Spark on EMR for our time-ordered events problem use-case.



The data is coming in with a lot of quotes Q and some trades T. The trades T tends to be the trigger event for us to do some analysis for detecting manipulation. Using Spark, we are able to union the data, sort the data, and then walk the data for a specific analysis while keeping the state in memory and watching for data manipulation



AWS S3 is our primary storage as a single source of truth, it is cataloged and versioned. On the right are the computation steps that are done, first for getting the data, sorting and arranging the data, then running the specific computation on

the data. we can persist the intermediate results to S3 at any point in time and can also use the best instance type for the job and can use spot instances too.

Choosing Spot Instances

This architecture gets us:

- Flexibility / Elasticity
  - I/O intensive instances for portions of code (m1.xlarge)
  - Shuffle-heavy (m2.4x)
  - Time-based computations (compute optimized -> m3.2x)
- Cost Savings
  - Targeting AWS Spot Instance types to match these profiles **saved us 3X the cost.** We also pinpointed instances which were **not too volatile.**

FINRA

24

Benefits

Realized Benefits

- Order of magnitude cost savings vs. on-premises database architecture
- Increased speed on reprocessing requests
- Scalability, etc.

Expected Future Benefits

- Supports real-time processing (as data providers migrate to real time)
- Easier experimentation with new instance types
- Convert more surveillance to this model

FINRA

25

FINRA's Future Plans

- Capture additional benefits of Spark, such as use of data frames API for easier manipulations and optimization
- Migrate additional workloads onto Spark – currently 240 surveillance applications are implemented in Hive but would benefit from Spark
  - Some experiments at FINRA show 2X speed with Spark vs Hive for same design
- Use new AWS APIs
- Move toward real time (coordinating with data providers)

FINRA

Related Sessions

FINRA Sessions:

- BDM203 – Building a Secure Data Science Platform
- DAT302 – Best Practices for Migrating to RDS / Aurora
- ENT313 – FINRA in the Cloud, Big Data Enterprise
- STG308 – Data Lake for Big Data on Amazon S3
- SVR202 – What's new with Lambda