Customer Showcase: John Deere
Perform Real-time ETL from IoT Devices into your Data Lake with Amazon Kinesis

Greg Finch, Data Platform Senior Product Lead, John Deere
Ryan Nienhuis, Senior Product Manager, AWS

03/20/2019
Learning Objectives

1. Learn how to decide between what processing to do in real-time and what to do in batch

2. Learn how to perform real-time ETL using Amazon Kinesis Data Firehose, AWS Lambda, and AWS Glue, compared to Amazon Kinesis Data Streams and Amazon Kinesis Data Analytics

3. Learn the pros and cons of the two solutions, and how knowledgeable AWS customers make the decision about how to use one over the other
To create value companies must derive insights from a variety of data sources that are producing data at high velocity and volume.
To create value companies must derive insights from a variety of data sources that are producing data at high velocity and volume.

Data warehouse  Data lake  Data streams
To create value companies must derive insights from a variety of data sources that are producing data at high velocity and volume.

Data integration requirements … now demand real-time streaming, replication and virtualized capabilities …

—Gartner 2019 Planning Guide for Data and Analytics
Common streaming use cases

- **milliseconds**
  - Messaging between micro-services
  - Response analytics (Web and mobile app notifications)

- **seconds**
  - Log ingestion
  - IoT device maintenance
  - Change Data Capture (CDC)

- **minutes**
  - Streaming ETL into data lakes and data warehouses
Why streaming ETL?

1. Efficient ingest for high velocity data
2. Increased volume to your data lake
3. Continuous versus periodic
4. Reduce time to information
End to end streaming architecture

Data streaming technology enables a customer to ingest, process and analyze high volumes of high velocity data from a variety of sources in real time.

Source → Stream ingestion → Stream storage → Stream processing → Destination
Streaming data with AWS

Easily collect, process, and analyze data streams in real time

Capture and store data streams

Analyze data streams in real time

Load streaming data into destinations

Capture and store data streams
Super simple load to your data lake

Source

IoT thing

Amazon Simple Storage Service (S3)
Super simple load to your data lake

**Streaming**

- Lots of unorganized small files on data lake
- No processing occurs in real-time
- Data must be buffered on devices that are not reliable – adds latency

**Batch**

- Lots of steps required to clean and organize data
- Processing is more complex and takes more time, adding latency
Simple ETL to your data lake

Source → Stream ingestion → Stream storage → Stream processing → Destination

IoT thing generic → AWS IoT Core → IoT rule → Amazon Kinesis Data Firehose → Optional convert to Parquet or ORC → Amazon Simple Storage Service (S3)

Lambda function → AWS Glue Data Catalog → AWS Glue
Simple ETL to your data lake

**Streaming**
- Purpose is to acquire data and setup for batch processing
- Stateless stream processing
- Processing includes buffering to larger files, format conversion, filtering, compression, encryption, and more
- Easy to setup and run with no to little code

**Batch**
- One to many steps are used after data lands in lake
- Bulk of processing occurs here including joins, correlations, deduplication, and other stateful functions
- Micro-batching can be used to reduce latency to a point
Sophisticated streaming ETL to your data lake

Source → Stream ingestion → Stream storage → Stream processing → Destination

IoT thing generic → Elastic Load Balancing (ELB) → Amazon EC2 Instances → Amazon Kinesis Data Streams → Amazon Kinesis Data Analytics → Amazon Simple Storage Service (S3)
Sophisticated ETL to your data lake

**Streaming**

- Reduce latency to useful data to minutes or seconds
- Stateful stream processing
- Most processing occurs in near real-time including complex joins, correlations
- Lose efficiencies that are gained in the batch approach

**Batch**

- Reduce or eliminate the majority of batch processing steps
- Processing may include data consolidating objects into larger files (e.g. larger files) and joins against larger data sets
KDA for Java for sophisticated applications

Utilizes Apache Flink, a Framework and distributed engine for stateful processing of data streams

- **Simple programming**
  - Easy to use and flexible APIs make building apps fast

- **High performance**
  - In-memory computing provides low latency & high throughput

- **Stateful Processing**
  - Durable application state saves

- **Strong data integrity**
  - Exactly-once processing and consistent state
Simple or Sophisticated Streaming?

• Latency decision
• Not a binary decision, many use both
• Early adopters moving to pure streaming
• AWS has services regardless of your decision
Customer Spotlight: John Deere
About John Deere

Agricultural Equipment

Construction Equipment

Turf Equipment

Forestry Equipment
Our Purpose: Committed to Those Linked to the Land

There are 7 billion people on this planet. And by 2050, there will be 2 billion more... many moving into urban centers at an unprecedented rate. Making sure there is enough food, fiber and infrastructure for our rapidly growing world is what we’re all about at John Deere.

Global population is increasing.

Arable land is not.
John Deere Intelligent Solutions Group
ExactEmerge™ Planter
ExactEmerge™ Planter

- 36,000 seeds / acre
- Dozens of acres / hr
- 10 miles / hr
- In the right place
- At the right depth
- With the right spacing
ExactEmerge™ Planter

15 sensor readings
×
5 hertz
×
32 row units
=

2400 readings / sec
One Minute

5738 active machines
734 million measurements
12.2 million per second
Serving billions of field maps to applications for:

- Tracking
- Monitoring
- Alerting
- Dashboards
- Analysis
Ingestion

Constant Stream

Micro-batches

Large Batch
Stream or Batch Processing?

- Zip up the stream and process it as a batch?
- Unzip the batch and process it as a stream?
- Some of both?
Streaming – The Lowest Common Denominator

Kinesis Data Stream

… but not always the best choice
Retaining Cohesion

Kinesis Data Stream
Mostly batch or mostly streaming?

1. Mostly streaming when input is a stream and latency matters

2. Mostly batch if input arrives in batches and you want to gain processing efficiencies
Transforming Streaming Data
Stateless Stream Processing

Concerns:
1. Can I keep up?
2. Can I recover?
Keeping Up - Options

1. More Shards
2. Bigger Decoder Instances
3. Fan Out
Stateful Stream Processing

More Concerns:
1. How do I group related data?
2. How do I handle late arriving data?
3. How do I ensure exactly once processing?

512,107 seeds

4,804,347 seeds
Archive – Amazon Kinesis Firehose

Creates a permanent archive of all data passing through the stream.
Lower latency stream processing is usually more expensive than higher latency or batch processing.*

Batch processing is usually cheaper, easier, and more efficient than stream processing.

* Except constant steady flow, stateless stream processing
Trends

- More Machines
- Better Connectivity
- More Sensors
- Higher Density
- Higher Frequency
- Lower Latency
- Actionable Insights

Customer requirements will demand more stream processing at higher scale.
John Deere Careers

Now hiring:

- ML / AI
- Vision and Perception
- Data Science
- Telematics
- Robotics
- Mobile Software
- Embedded Software
- Software Engineering
- Architecture

http://jobs.deere.com
Thank You!