Spark-Neo4j Processing Pipeline Project Summary

Ishaan Madan
Sep 13th, 2018

AMOBEE

Agenda

- Context / Problem Definition
- Assessing Technology Options
  - » Neo4j Highlights
  - » Key Requirements
- Solution Overview – Layout & Sequence
- Implementation Considerations
- Demo
- Next Steps

AMOBEE

Context / Problem Definition

- Insightful (Valuable) Information
  - » Which ***superID*** does this specific ***deviceID*** belong to?
  - » Which/how many ***superID***'s have at least two or more devices?
  - » In Redwood City, how many people have a certain ***deviceType***?

- Current Approach (Pain-Points)
  - » Cheetah QL queries are slow (take anywhere from 30–120 minutes to execute)
  - » …and multi-staged
    - » when unsure about partition, required to scan entire table-format AUP for answers
    - » involve extensive post-processing (storing, filtering, sorting)

AMOBEE

Neo4j Highlights

- Graph database system with Create, Read, Update, and Delete functionality
- Property Graph Model
  - » Nodes
  - » Relationships
  - » Properties
- Cypher Querying Language
- Architectural Advantages
  - » Native graph architecture (efficient node traversal – relationship, not index driven)
  - » Index free adjacency (performance proportional to search size, not population size)

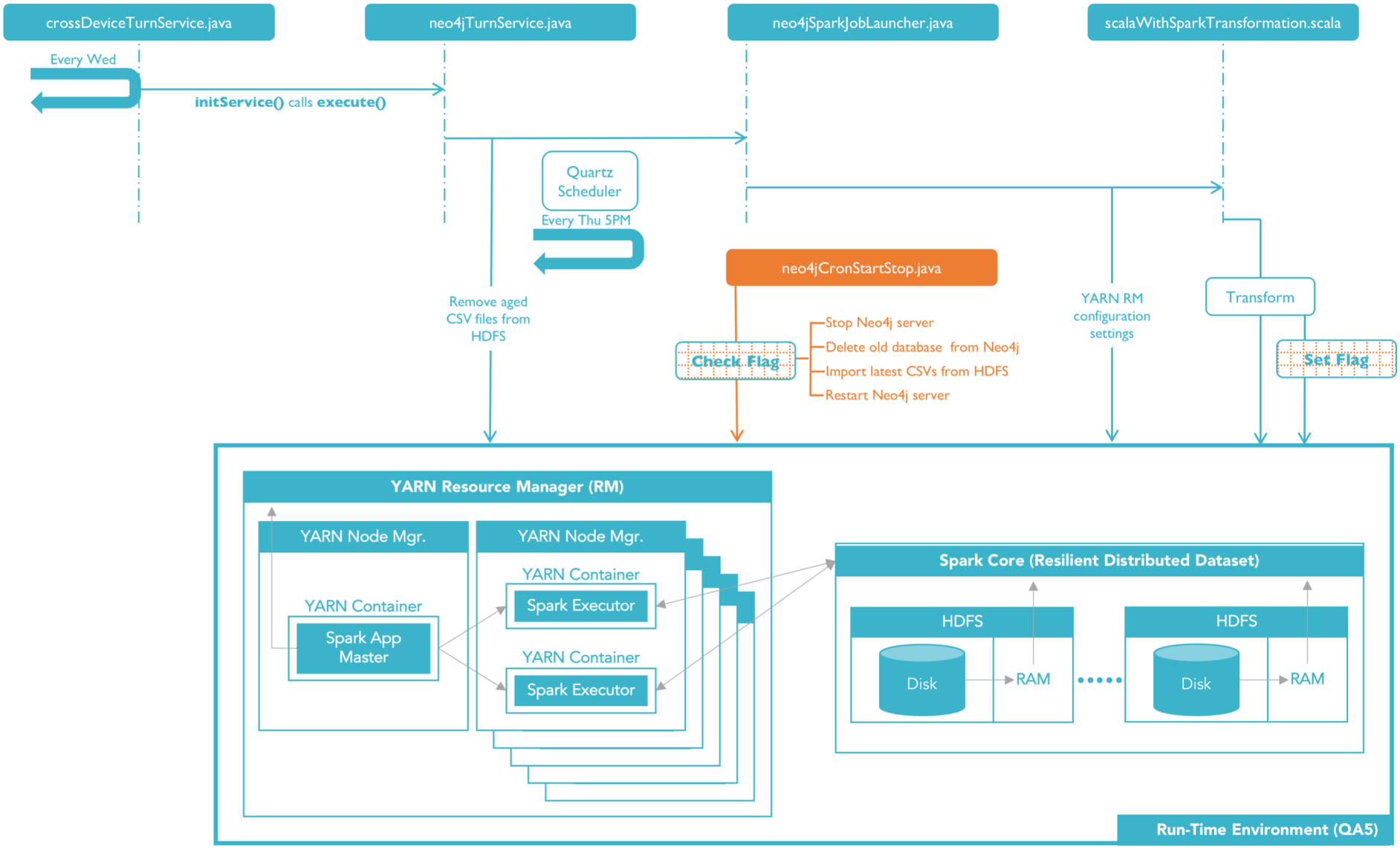
AMOBEE

Key Requirements

- Multi-threaded processing
  - » Need for scale – 30GB data weekly
  - » **SPARK**
- Pattern-based file transformation
  - » Ability to control transformations with functional-style programming
  - » **SCALA**
- Relationship-based data
  - » Ability to visually navigate dataset by multiple/variable/ad-hoc relationships
  - » **NEO4J**

AMOBEE

Solution Overview – Layout & Sequence



AMOBEE

Implementation Considerations

- Performance
  - » Local machine (16GB RAM) ~50 million nodes; queries run in 10-180 seconds
  - » app006.qa5 (64GB RAM, ~20GB usable) ~350 million nodes; queries fail
  - » No node limit – need to figure out server configuration to optimize performance
- Costs
  - » Community Edition for project (free)
  - » Enterprise Edition allows for following "industrial grade" features -
    - » Server clustering
    - » *Active page cache warmup*, which shortens the page fault spike and makes page cache warm up faster
    - » Better logging features (queries/access security)
    - » Granular access control simplifying app design and maintenance by providing *reader, publisher, architect* and *admin* security roles, eliminating the need to code security into application logic.

AMOBEE

Demo

- Insightful (Valuable) Information
  - » Which ***superID*** does this specific ***deviceID*** belong to?
  - » Which/how many ***superID***'s have at least two or more devices?
  - » In Redwood City, how many people have a certain ***deviceType***?

- Current Approach (Pain-Points)
  - » Cheetah QL queries are slow (take anywhere from 30–120 minutes to execute)
  - » …and multi-staged
    - » when unsure about partition, required to scan entire table-format AUP for answers
    - » involve extensive post-processing (storing, filtering, sorting)

Query time reduced to 30-600 seconds

Traverses graph containing only key relationships between users and devices

Succinct queries (often single-line), and single stage

AMOBEE

Next Steps – Program Rollout

- Immediate Term – Scale
  - » Fine tune server configuration for optimizing neo4j performance
  - » Better understand indexing and memory allocation configuration for graph
  - » Create query templates for ease-of-use for frequently used Neo4j queries
  - » Deploy Enterprise Edition to production cluster
- Longer Term – Extend & Expand Usage
  - » Extend usage
    - » by adding cross-device data properties in graph (location, impressions, clicks, etc.)
    - » by allowing front-end users to query dataset directly
    - » by ingesting additional Cross Device Data from new providers (more than TAPAD)
  - » Expand to include other use-cases/datasets
    - » by introducing a project pertaining to adType clicks across users

AMOBEE

Thank You For The Opportunity !

AMOBEE