Triton: A Continuous Query Translation Engine for Trident/Storm

Presented by
Zhiheng Li

Advisor: Amarnath Gupta

Table of Contents

Introduction

- Background
- Motivation

Background

- Continues Query Language
  - Sliding window
- Data Stream Management System (DSMS)
  - STREAM
- Complex Event Processing (CEP)
  - Esper
- Distributed streaming computation
  - Storm/Trident

Sliding window

- Row based
- Time based
- Time batch

Motivation

- Pig on Hadoop
- A system that combine the advantage of Esper and Storm.
- The Triton translation engine
  - compiles the *TQL* query into native Trident program written in JAVA.
- *TQL*: a query language for the Triton system.

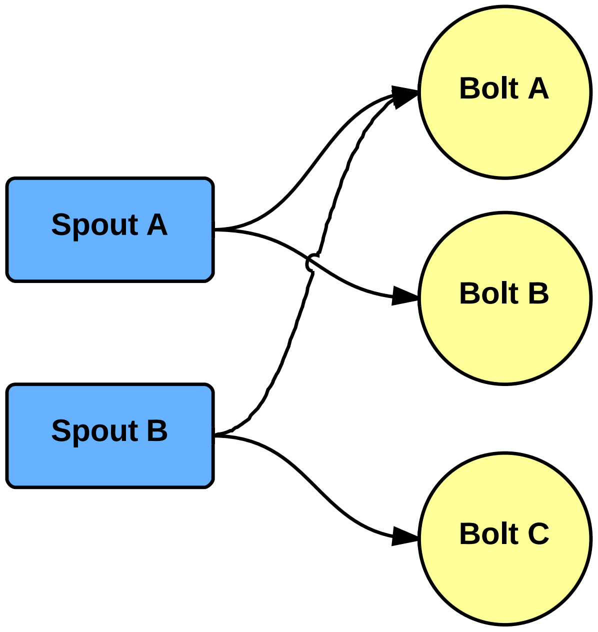
Storm and Trident

- Storm: A distributed real-time computation system.

- Trident: A high-level abstraction on Storm to easy the development of Storm program.

Storm Overview

- Topology: a graph of computation.
- Stream: key abstraction of data flow.
- Spout: source of stream.
- Bolt: computation node.



A Concrete Example of Storm Topology

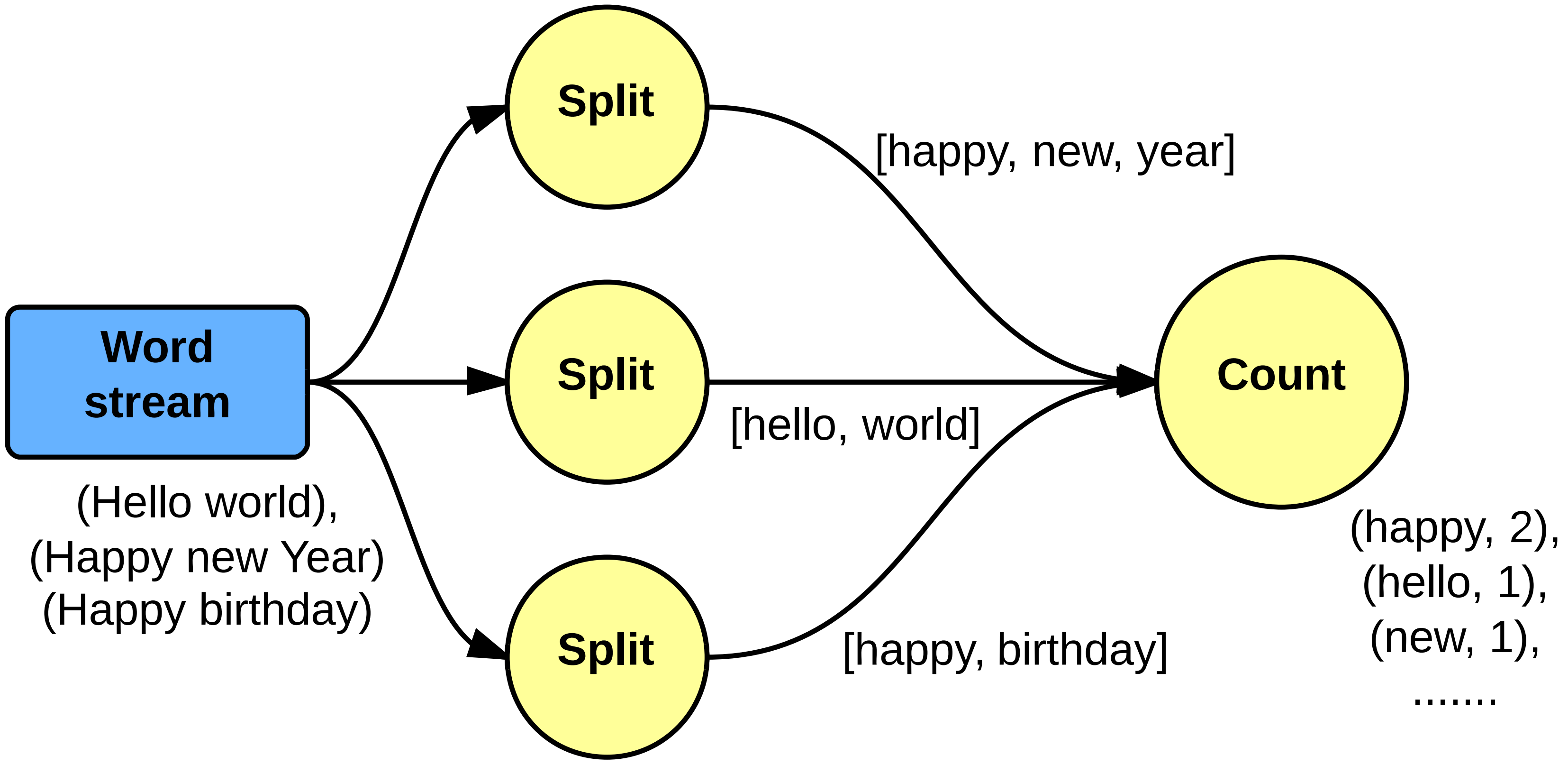


Table of Contents

Triton Overview

- Design Decision
- System Overview

Design Decisions

- System architecture
- Compliable and readable code generation

System Overview

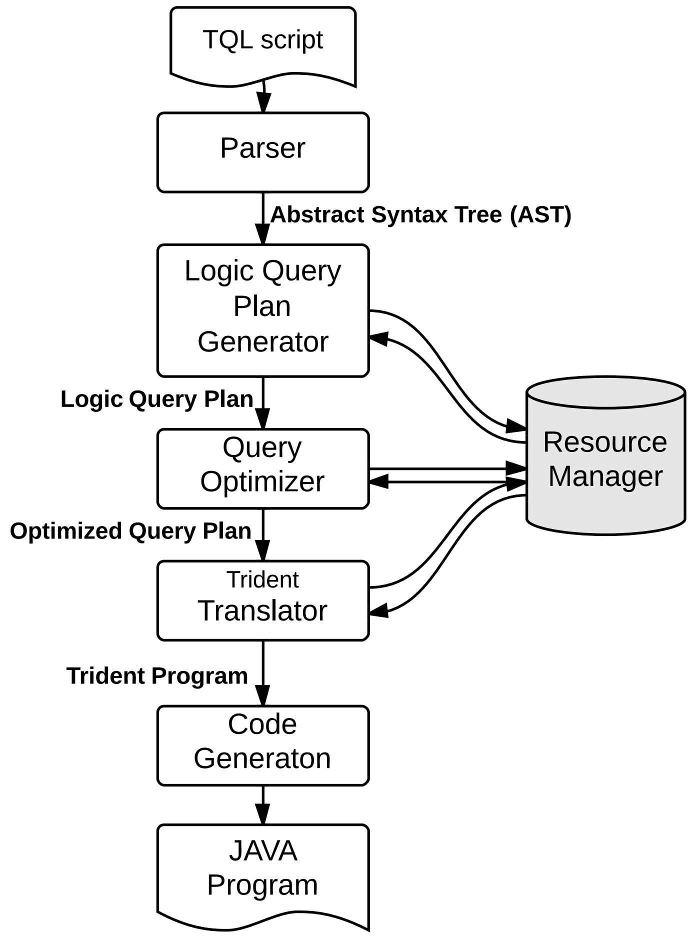


Table of Contents

Triton Query Language (TQL)

- Syntax Overview
- An example

Syntax Overview

- An extension to standard SQL
- Stream Registration

- Query definition

- Sliding window definition

Syntax Overview

- Named query definition
- A completed example

Table of Contents

Implementation Details

- Parser
- Meta-data management
- Logic query plan
- Optimization
- Code generation

Parser

- JJTree and JavaCC
- Left-recursion elimination
- Usage of LOOKAHEAD

Meta-data Management

- Stream definition management.
- Stream/attribute Renaming.
- Query dependencies.
- Named/Anonymous query.

Logic Query Plan

- Introduce window operator

- Translation from AST to logic query plan
  - simple case.
  - multiple streams involved.

Simple case

- A simple logic plan

- A logic plan involve multiple stream

Multiple stream

Optimization

- Selection push-down
- Join rewritten
- Optimization techniques

Code Generation

- Trident API Overview
  - newStream
  - each
    - function
    - filter
  - groupBy

Trident API Overview

- partitionPersistent
- aggregate
- applyAssembly
- newValueStream

Translation of Window Operator

- Eviction policy based ring buffer
- A combine of three Trident APIs
  - partitionPersistent
  - newValueStream
  - groupBy
- An example

Table of Trident API and logic operator mapping

Logic Operator	Trident API
Selection	each with Filter class
Projection	each with BaseFunction class project
InputStream	newStream
OutputStream	each
Aggregation	groupBy aggregation persistentAggregation
Window	partitionPersistent newValueStream groupBy
Join	join
OrderBy	applyAssembly(firstN)

JAVA Code Generation

- Query dependencies
  - Topological sort
- Import package management
- Default building script
  - MAVEN

Table of Contents

Future Work

- Query plan optimization
- Built-in sliding window support
- User Defined Function support(UDF)
- Performance benchmark on cluster

Table of Contents

Conclusion

- A query translation engine
- Real-time streaming process ecosystem

Demo

Thanks

Q & A