ApacheCon North America
Portland, Oregon 26th - 28th February 2013

A New Generation of Data Transfer Tools for Hadoop: Sqoop 2

Arvind Prabhakar | Kathleen Ting

Who Are We?

Arvind Prabhakar

Apache Sqoop Committer, PMC Chair, ASF Member

Engineering Manager, Cloudera

arvind@apache.org, @aprabhakar

Kathleen Ting

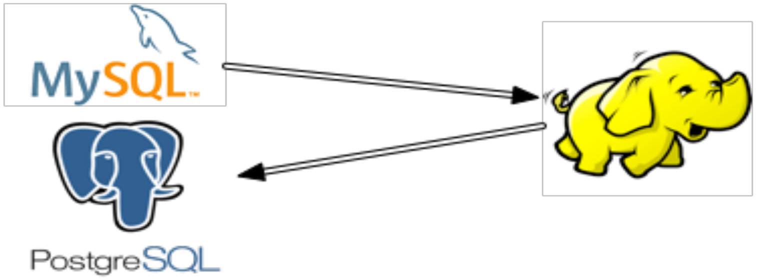
Apache Sqoop Committer, PMC Member

Customer Operations Engineering Manager, Cloudera

kathleen@apache.org, @kate_ting

What is Sqoop?

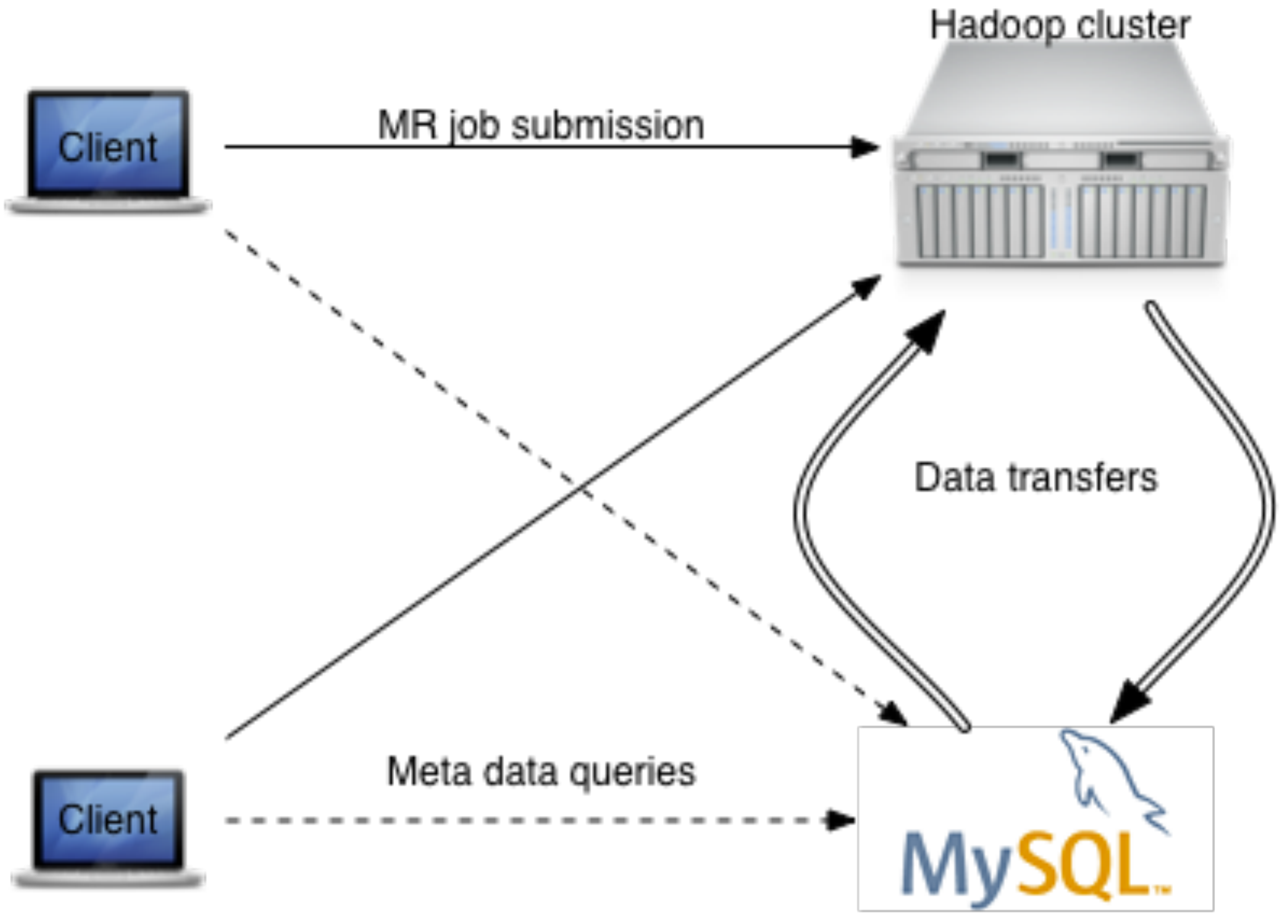
- Apache Top-Level Project
- SQl to hadOOP
- Tool to transfer data from relational databases
  - Teradata, MySQL, PostgreSQL, Oracle, Netezza
- To Hadoop ecosystem
  - HDFS (text, sequence file), Hive, HBase, Avro
- And vice versa



Why Sqoop?

- Efficient/Controlled resource utilization
  - Concurrent connections, Time of operation
- Datatype mapping and conversion
  - Automatic, and User override
- Metadata propagation
  - Sqoop Record
  - Hive Metastore
  - Avro

Sqoop 1


Hadoop cluster
Client
MR job submission
Data transfers
Client
Meta data queries
MySQL

Sqoop 1

- Based on Connectors
  - Responsible for Metadata lookups, and Data Transfer
  - Majority of connectors are JDBC based
  - Non-JDBC (direct) connectors for optimized data transfer
- Connectors responsible for all supported functionality
  - HBase Import, Avro Support, ...

Sqoop 1 Challenges

Cryptic, contextual command line arguments

Security concerns

Type mapping is not clearly defined

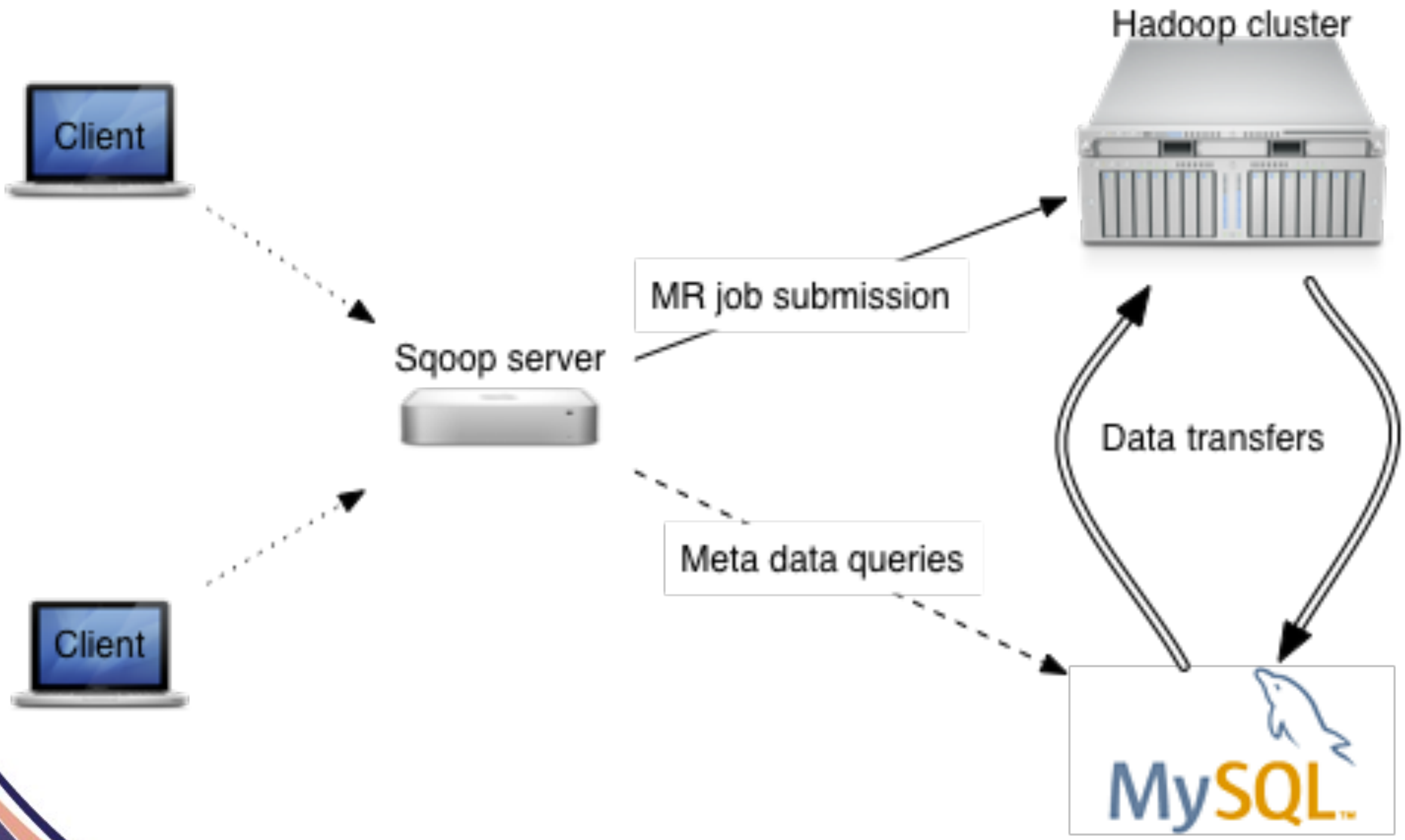
Client needs access to Hadoop binaries/ configuration and database

JDBC model is enforced

Sqoop 1 Challenges

- Non-uniform functionality
  - Different connectors support different capabilities
- Overlap/Duplicated functionality
  - Different connectors may implement same capabilities differently
- High Coupling with Hadoop
  - Database vendors required to understand Hadoop idiosyncrasies in order to build connectors.

Sqoop 2


Hadoop cluster
Client
Client
MR job submission
Sqoop server
Data transfers
Meta data queries
MySQL

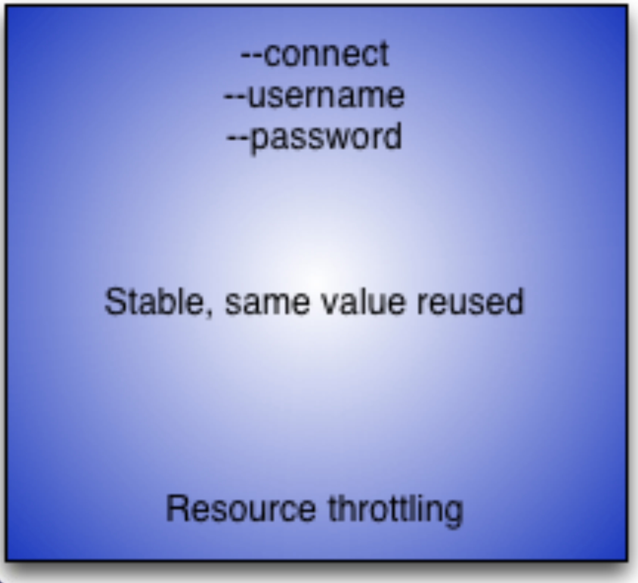
Sqoop 2 – Design Goals

- Ease of Use
  - Uniform functionality
  - Domain Specific Interactions
- Ease of Extension
  - No low-level Hadoop Knowledge Needed
  - No functional overlap between Connectors
- Security and Separation of Concerns
  - Role based access and use

Sqoop 2: Connection vs Job metadata

There are two distinct sets of options to pass in to Sqoop:

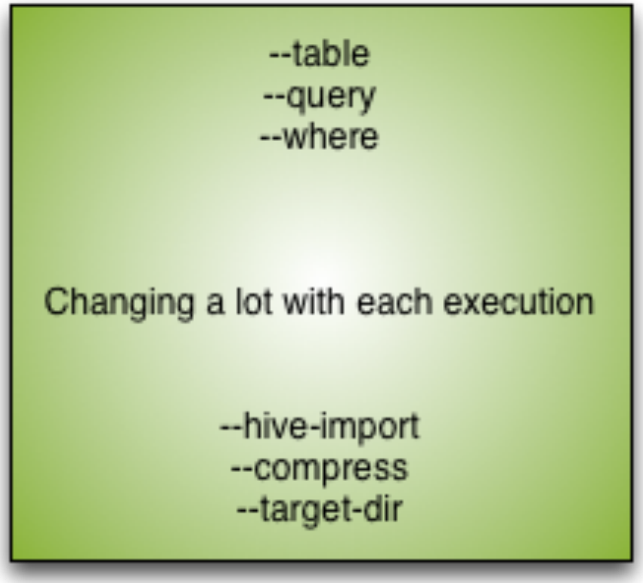
- Connection (distinct per database)
- Job (distinct per table)



--connect
--username
--password

Stable, same value reused

Resource throttling



--table
--query
--where

Changing a lot with each execution

--hive-import
--compress
--target-dir

Sqoop 2: Workings

- Connectors Register Metadata
- Metadata enables creation of Connections and Jobs
- Connections and Jobs stored in Metadata Repository
- Operator runs Jobs that use appropriate connections
- Admins set policy for connection use

Sqoop 2: Security

- Support for secure access to external systems via role-based access to connection objects
  - Administrators create/edit/delete connections
  - Operators use connections

Sqoop 2: Usability & Extensibility

- Connections and Jobs use domain specific inputs (Tables, Operations, etc.)
  - Domain Isolation and thus easy to understand and use
- Connectors work with Intermediate Data Format
- Any downstream functionality needed is provided by Sqoop Framework

Demo

Current Status: Sqoop 2

Primary focus of the Sqoop Community

First cut: 1.99.1

- bits and docs: http://sqoop.apache.org/

SQOOP WANTS YOU