Mastering Sqoop for Data Transfer for Big Data

Jarek Jarcec Cecho | Kathleen Ting
Who Are We?

• Jarek Jarcec Cecho
  • Apache Sqoop Committer, PMC Member
  • Software Engineer, Cloudera
  • jarcec@apache.org

• 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
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

• Overlapped/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

Client → Sqoop server

Sqoop server → MR job submission

MR job submission → Hadoop cluster

Hadoop cluster → Data transfers

Data transfers → Meta data queries

Meta data queries → MySQL

MySQL → Data transfers
Sqoop 2 – Design Goals

• Security and Separation of Concerns
  • Role based access and use

• Ease of extension
  • No low-level Hadoop knowledge needed
  • No functional overlap between Connectors

• Ease of Use
  • Uniform functionality
  • Domain specific interactions
Sqoop 2: Connection vs Job metadata

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

Connection (distinct per database)
- --connect
- --username
- --password

Stable, same value reused

Resource throttling

Job (distinct per table)
- --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
Current Status: Sqoop 2

- Primary focus of the Sqoop Community
- First cut: 1.99.1
  - bits and docs: http://sqoop.apache.org/
Demo
SQOOP WANTS YOU