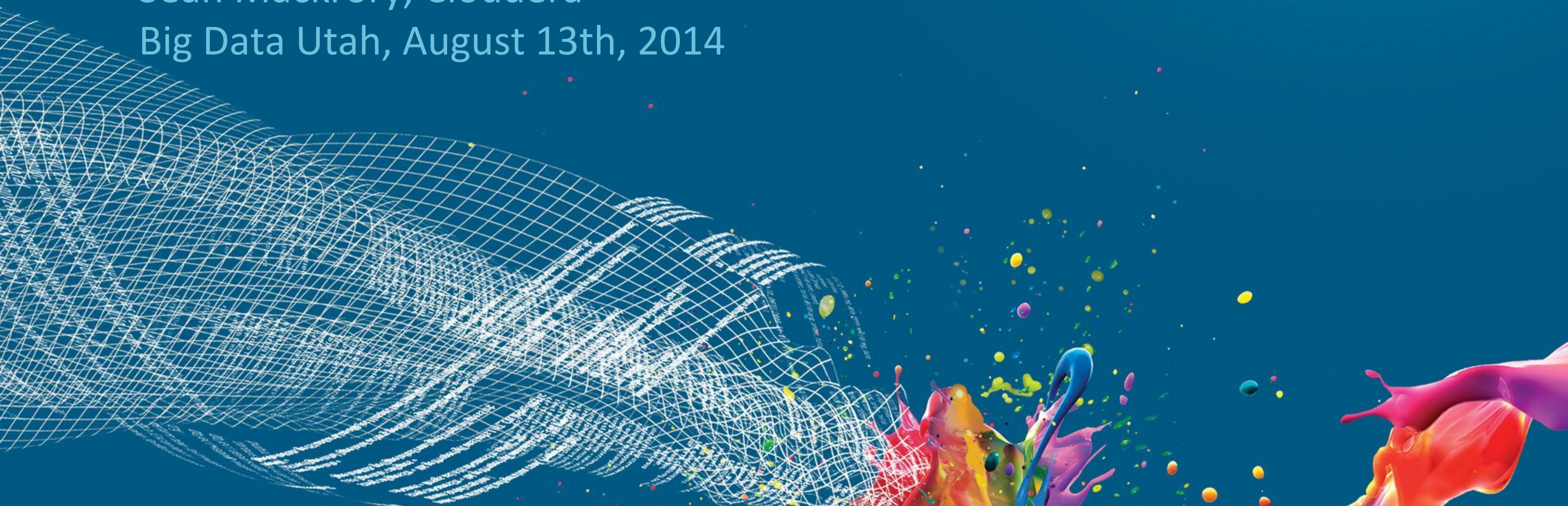
Abstract

Hadoop’s ideal role is best described as the core of a “data hub”: a single repository for all types of data, with many ways to efficiently process the data in-place, efficient ways to transfer data to and from other places when needed, integrated security & management tools, etc. “Search-on-Hadoop”, then, is about much more than just Apache Solr + Apache Hadoop. It’s about an integrated stack of tools to manage every aspect of your searchable data from the time it’s created. We’ll dive into how Solr + Hadoop can be used to build a large-scale, easy-to-use search tool, how to index both text and structured data in batch and near-real-time as it's created, how to manage the security of your searchable data, and more.

Cloudera Search: Apache Solr in the “Data Hub”

Sean Mackrory, Cloudera

Big Data Utah, August 13th, 2014



About Me

- Software Engineer @ Cloudera, Integration team
- Committer @ Apache Bigtop
- PPMC member @ Apache Sentry (incubating)
- Big believer in free-as-in-freedom software
  - Except as noted, everything you see tonight is Apache 2.0
- Based in Colorado (used to work in Provo)
- J Dawgs rules

About the “Enterprise Data Hub”

Hadoop’s ideal role is best described as the core of a “data hub”:

- Flexible storage: structured, unstructured, data formats
- Flexible compute: batch, interactive, SQL, **search**, graph, stream...
- Fast connectors to more specialized systems
- Integrated security and compliance, etc.
- Monitoring & management

Search in the “Enterprise Data Hub”

- SolrCloud: Apache Solr, distributed with Apache ZooKeeper
- Store indices in Apache Hadoop (HDFS)
- Store documents in HDFS or Apache HBase
- Batch indexing with Hadoop / MapReduce
- Near-real-time indexing with Apache Flume and HBase
- Manipulate data with Cloudera Morphlines
- Document-level role-based access control with Sentry
- User experience with Cloudera Hue

Apache Solr

- Stand-alone search server (runs in J2EE container)
- Uses and extends Apache Lucene for text processing
- Full-text, faceted, geospatial search, clustering, etc.
- JSON, XML, HTTP interfaces, very configurable
- Fields can be “stored”, or “indexed”, or both
- SolrCloud adds sharding, replication, fail-over, etc.

Apache ZooKeeper

- “Cluster coordination service”
- You need at least a “quorum” from a ZooKeeper “ensemble”
- Hierarchy of “z-nodes” that can hold both data and children
- z-nodes can be sequential and ephemeral, and have watches
- Used as a reliable data-store, for master election, node registration
- Used by SolrCloud for central configuration, fail-over, etc.

Apache Hadoop (HDFS)

- Breaks large files into blocks, distributes replicas to Data Nodes
  - Replication is configurable
  - Performance & Reliability vs. Capacity
- Name Nodes map files to blocks, and tracks replicas
  - Clients connect directly to Data Nodes when they can
- Files in HDFS are edited by appending or by replacing
- HDFS can store both Solr collections, and their indices

Apache Hadoop (MapReduce)

- Flexible batch processing framework
- “Map” takes each record, maps it to zero or more other records
    - Tries to map each record on the DataNode it lives on
- “Shuffle” groups records by key and sorts
- “Reduce” aggregates the records into results
- Can be used to index documents in HDFS in batch
- YARN separates the management from the framework

Apache HBase

- Implements a “BigTable”-like database on top of HDFS
- Provides random access to read-writable data
- Billions of rows, millions of columns (grouped by family)
- Can be indexed by Solr using NGDATA’s Lily indexer

NGData Lily HBase Indexer

- All HBase updates are “replicated” to the indexer
- Indexer inspects them and prepares them for Solr
- Updates the index for new rows, updates, and deleted rows

Apache Flume

- For ingesting streams of event / log data into Hadoop
- “Sources” ➔ “Channels” ➔ “Sinks”, all on agents
- Sources: Avro, Thrift, HTTP, Twitter, Netcat
- Channels: in-memory, file, JDBC
- Sinks: HDFS, Avro, Thrift, Morphlines/Solr
- Multiplexers, load balancers, routing, etc.

Apache Sentry (incubating)

- FIne-grained role-based access controls for Hive, Impala, Solr
- Uses Hadoop groups to map users to groups
- Uses Sentry policy to map groups to roles
- Sentry policy defines permissions for each role. In Solr:
  - collection=*...*->permission=...
  - Requires Kerberos authentication on the cluster
- “admin” is a pseudo collection for Solr administration
- Requires Kerberos authentication

Cloudera Morphlines

- Now part of the “Kite SDK” for building Hadoop applications
- Used for defining ETL pipelines, like UNIX pipelines w/ sed, etc.
- Commands for reading various types of records, parsing and type conversion, conditionals, logging, metrics, and...
- … commands for storing and indexing commands in Solr!

Cloudera Hue

- “Hadoop User Experience”: UI’s for most services in the stack
- “Search” app allows dashboards and search portals to be built and customized for each of your collections
- The easiest way to interact with a Kerberized cluster (required for using Sentry authorization with Solr)

Tweets Demo

- Load artificial ‘tweets’ into HDFS
- Use MapReduce to index existing corpus of ‘tweets’
- Index and store live ‘tweets’ from Twitter’s “sample” stream
- Index updates to an HBase table of ‘tweets’
- Use Sentry to restrict users access to specific collections
- Create an attractive search UI with Hue

Tweets Demo

Download QuickStart VM 5.1.0-1 for your hypervisor-of-choice:

http://www.cloudera.com/content/support/en/downloads/quickstart_vms.html

The edition of Cloudera Manager in the VM is free (gratis), but it is not distributed under the Apache Software License v2.0, like everything else in the demo.

Clone this project:

http://github.com/mackrorysd/BigDataUtah

Close your eyes!

During the demo we will be streaming live, raw data from Twitter and displaying it. We will search for benign terms, but I am not responsible for any of the content. Please look away during this portion of the demo if you want to ensure you do not see any obscene text or images.

Questions?

Sean Mackrory
@SeanMackrory