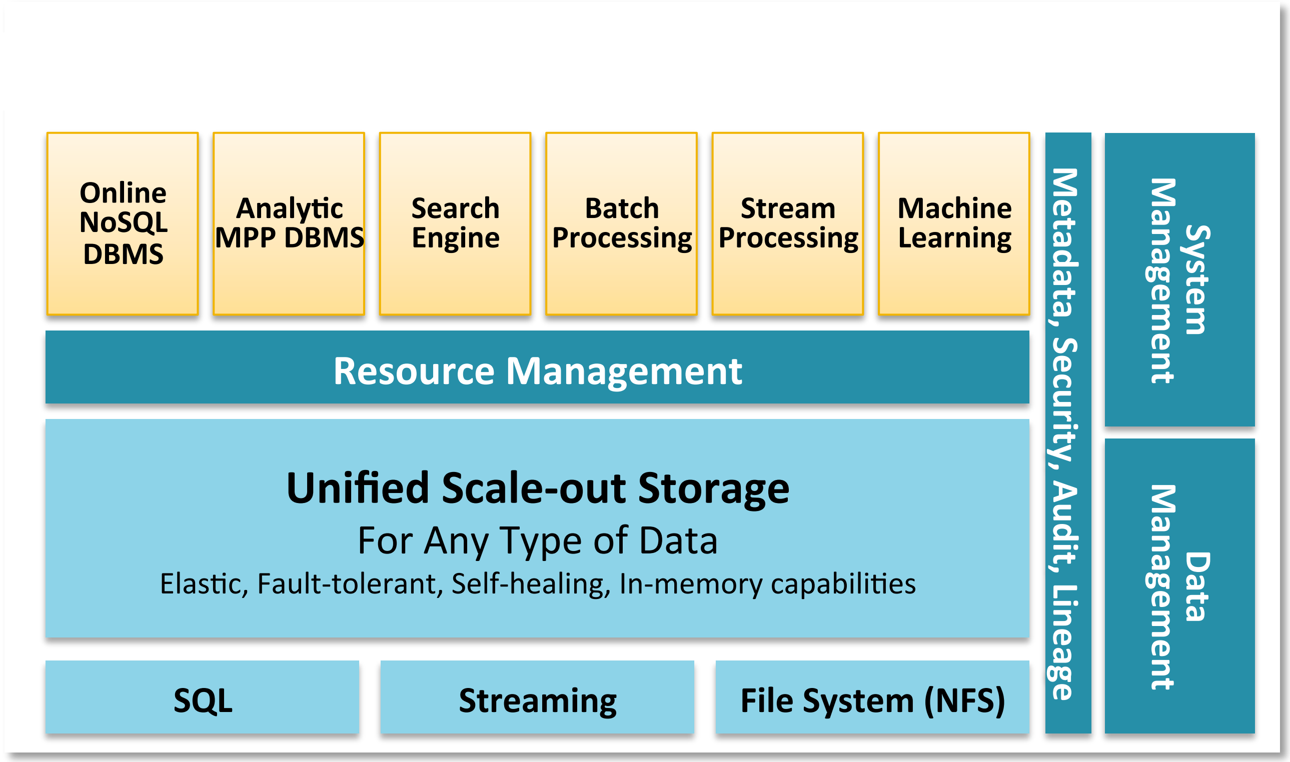
Ingesting HDFS data into Solr using Spark

Wolfgang Hoschek (whoschek@cloudera.com)
Software Engineer @ Cloudera Search
QCon 2015

The Enterprise Data Hub



- Multiple processing frameworks
- One pool of data
- One set of system resources
- One management interface
- One security framework

Apache Spark

- Mission
  - Fast and general engine for large-scale data processing
- Speed
  - Advanced DAG execution engine that supports cyclic data flow and in-memory computing
- Ease of Use
  - Write applications quickly in Java, Scala or Python
- Generality
  - Combine batch, streaming, and complex analytics
- Successor to MapReduce

What is Search on Hadoop?

Interactive search for Hadoop

- Full-text and faceted navigation
- Batch, near real-time, and on-demand indexing

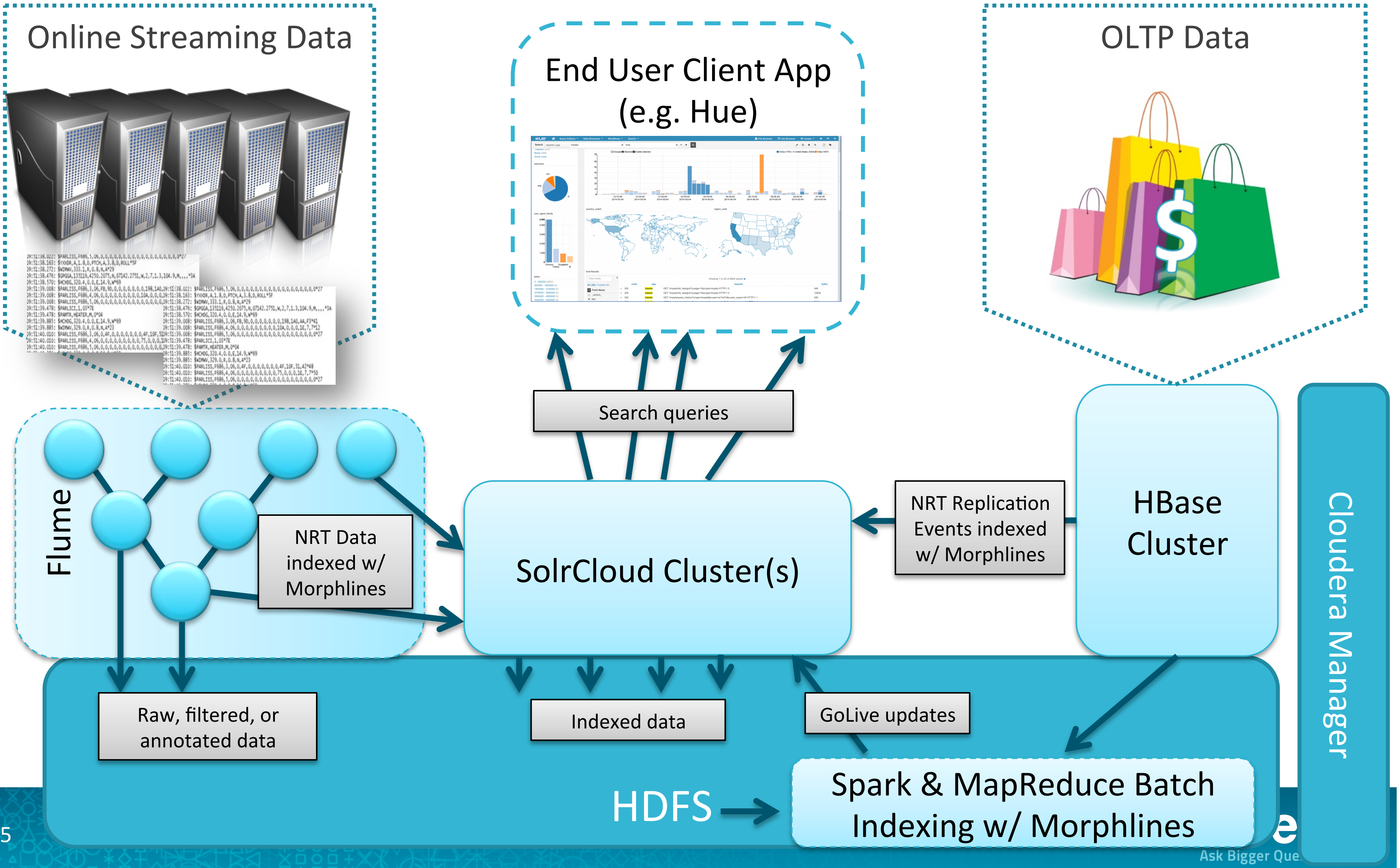
Apache Solr integrated with CDH

- Established, mature search with vibrant community
- Incorporated as part of the Hadoop ecosystem
    - Apache Flume, Apache HBase
    - Apache MapReduce, Kite Morphlines
    - Apache Spark, Apache Crunch

Open Source

- 100% Apache, 100% Solr
- Standard Solr APIs

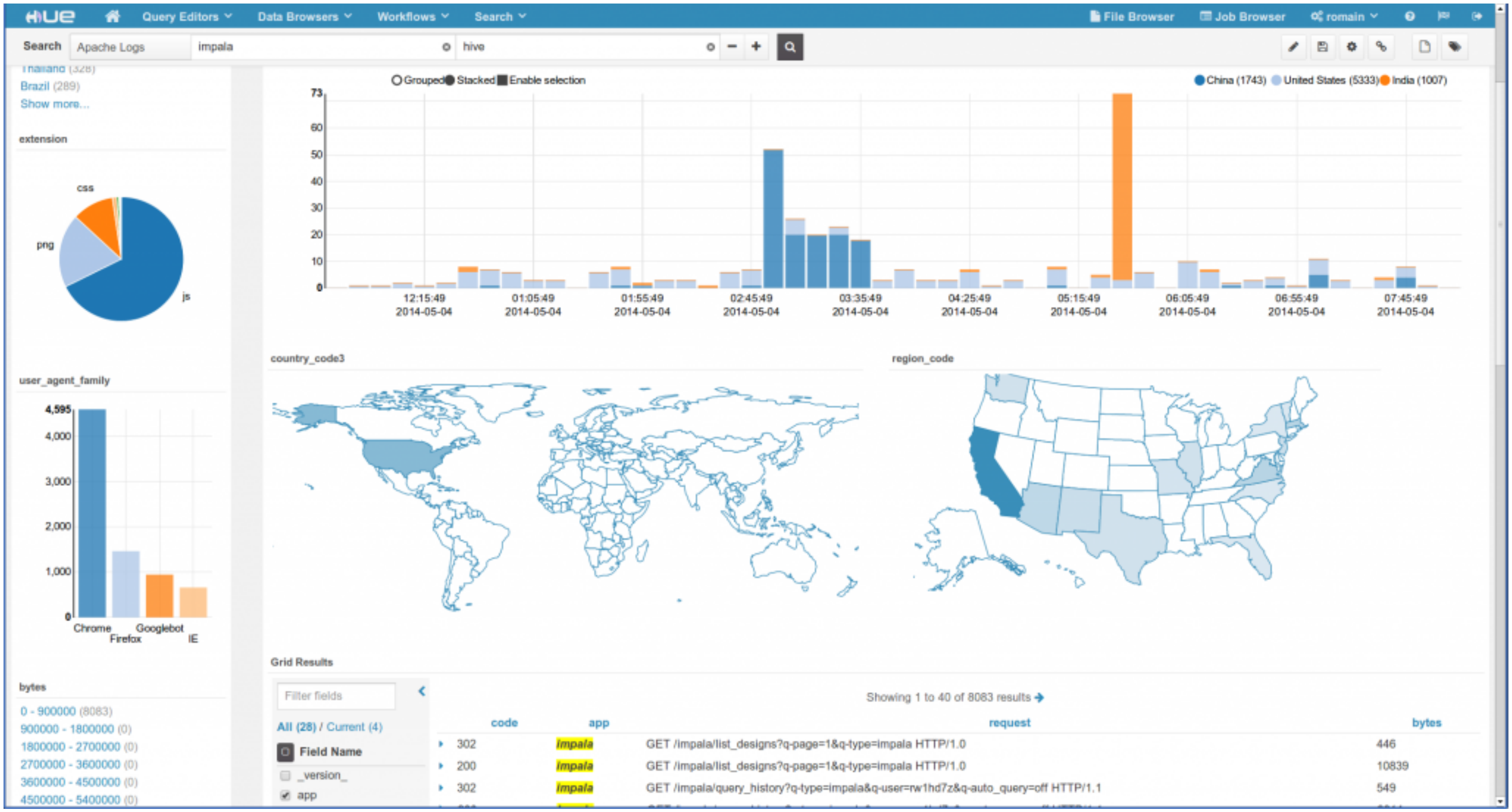
Search on Hadoop - Architecture Overview



Online Streaming Data
End User Client App (e.g. Hue)
OLTP Data
Search queries
Flume
NRT Data indexed w/ Morphlines
SolrCloud Cluster(s)
NRT Replication Events indexed w/ Morphlines
HBase Cluster
Cloudera Manager
Raw, filtered, or annotated data
Indexed data
GoLive updates
HDFS
Spark & MapReduce Batch Indexing w/ Morphlines

Customizable Hue UI

- Navigated, faceted drill down
- Full text search, standard Solr API and query language



0 - 900000 (8083)
900000 - 1800000 (0)
1800000 - 2700000 (0)
2700000 - 3600000 (0)
3600000 - 4500000 (0)
4500000 - 5400000 (0)

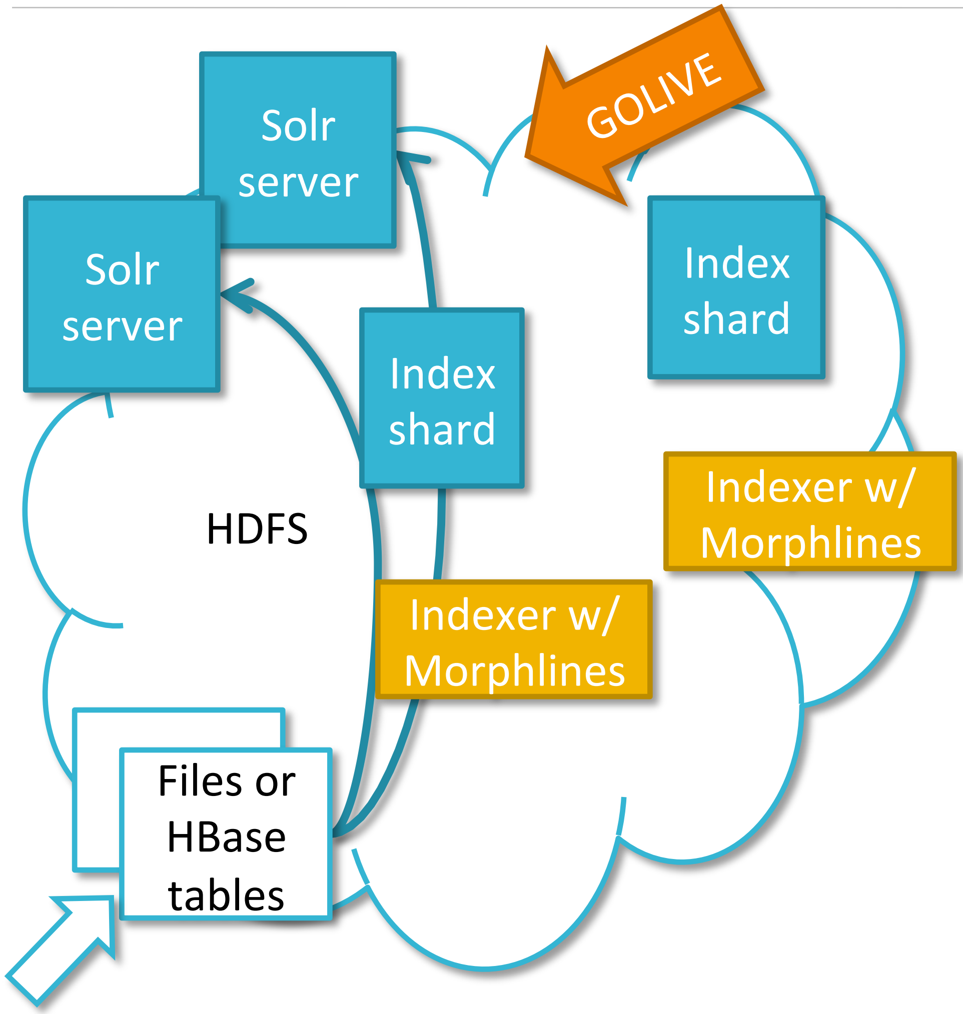
Showing 1 to 40 of 8083 results

code	app	request	bytes
302	impala	GET /impala/list_designs?q-page=1&q-type=impala HTTP/1.0	446
200	impala	GET /impala/list_designs?q-page=1&q-type=impala HTTP/1.0	10839
302	impala	GET /impala/query_history?q-type=impala&q-user=rw1hd7z&q-auto_query=off HTTP/1.1	549

http://gethue.com

cloudera Ask Bigger Questions

Scalable Batch ETL & Indexing



Solr and MapReduce

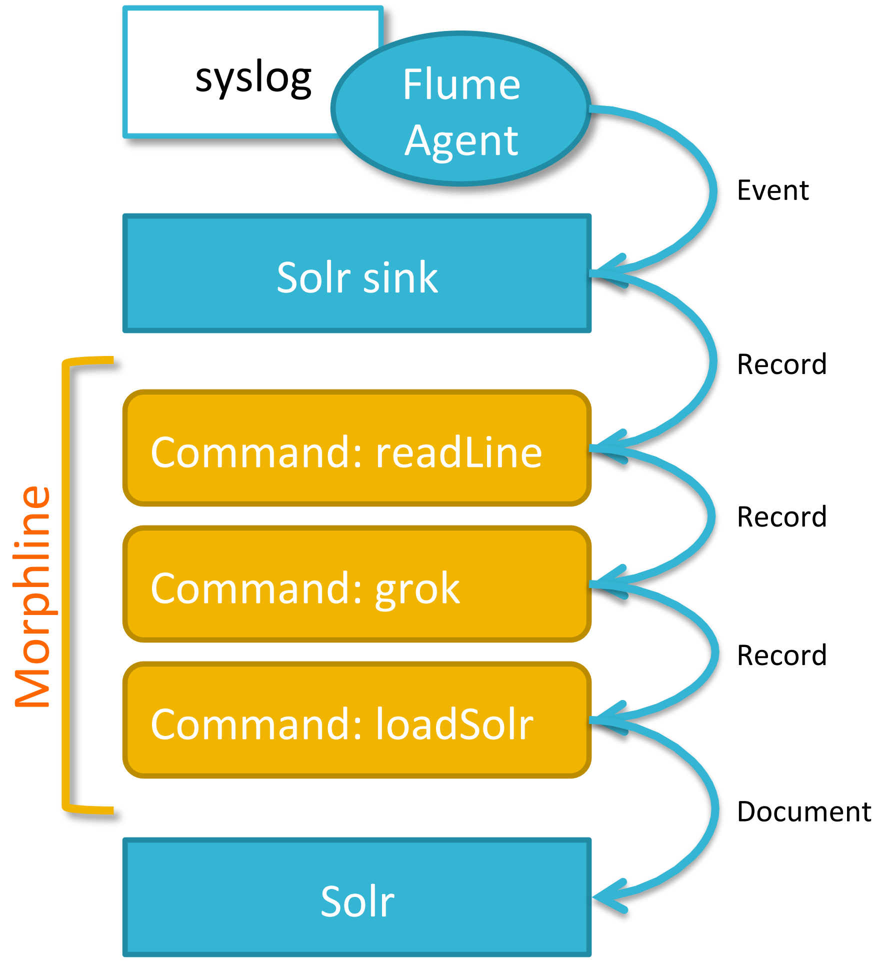
- Flexible, scalable, reliable batch indexing
- On-demand indexing, cost-efficient re-indexing
- Start serving new indices without downtime
- “MapReduceIndexerTool”
- “HBaseMapReduceIndexerTool”
- “CrunchIndexerTool on MR”

Solr and Spark

- “CrunchIndexerTool on Spark”

```
hadoop ... MapReduceIndexerTool --morphline-file morphline.conf ...
```

Streaming ETL (Extract, Transform, Load)



Kite Morphlines

- Consume any kind of data from any kind of data source, process and load into Solr, HDFS, HBase or anything else
- Simple and flexible data transformation
- Extensible set of transf. commands
- Reusable across multiple workloads
- For Batch & Near Real Time
- Configuration over coding
  - reduces time & skills
- ASL licensed on Github https://github.com/kite-sdk/kite

Morphline Example – syslog with grok

```
morphlines : [
 {
   id : morphline1
   importCommands : [”org.kitesdk.**", "org.apache.solr.**"]
   commands : [
     { readLine {} }
     {
       grok {
         dictionaryFiles : [/tmp/grok-dictionaries]
         expressions : {
           message : """<%{POSINT:syslog_pri}>%{SYSLOGTIMESTAMP:syslog_timestamp} %
{SYSLOGHOST:syslog_hostname} %{DATA:syslog_program}(?:\[%{POSINT:syslog_pid}\])?: %
{GREEDYDATA:syslog_message}"""
         }
       }
     }
     { loadSolr {} }
   ]
 }
]
```

Example Input

```
<164>Feb  4 10:46:14 syslog sshd[607]: listening on 0.0.0.0 port 22
```

Output Record

```
syslog_pri:164
syslog_timestamp:Feb  4 10:46:14
syslog_hostname:syslog
syslog_program:sshd
syslog_pid:607
syslog_message:listening on 0.0.0.0 port 22.
```

Current Morphline Command Library

- Supported Data Formats
    - Text: Single-line record, multi-line records, CSV, CLOB
    - Apache Avro, Parquet files
    - Apache Hadoop Sequence Files
    - Apache Hadoop RCFiles
    - JSON
    - XML, XPath, XQuery
    - Via Apache Tika: HTML, PDF, MS-Office, Images, Audio, Video, Email
    - HBase rows/cells
    - Via pluggable commands: Your custom data formats
- Regex based pattern matching and extraction
- Flexible log file analysis
- Integrate with and load data into Apache Solr
- Scripting support for dynamic Java code
- Etc, etc, etc

Morphline Example - Escape to Java Code

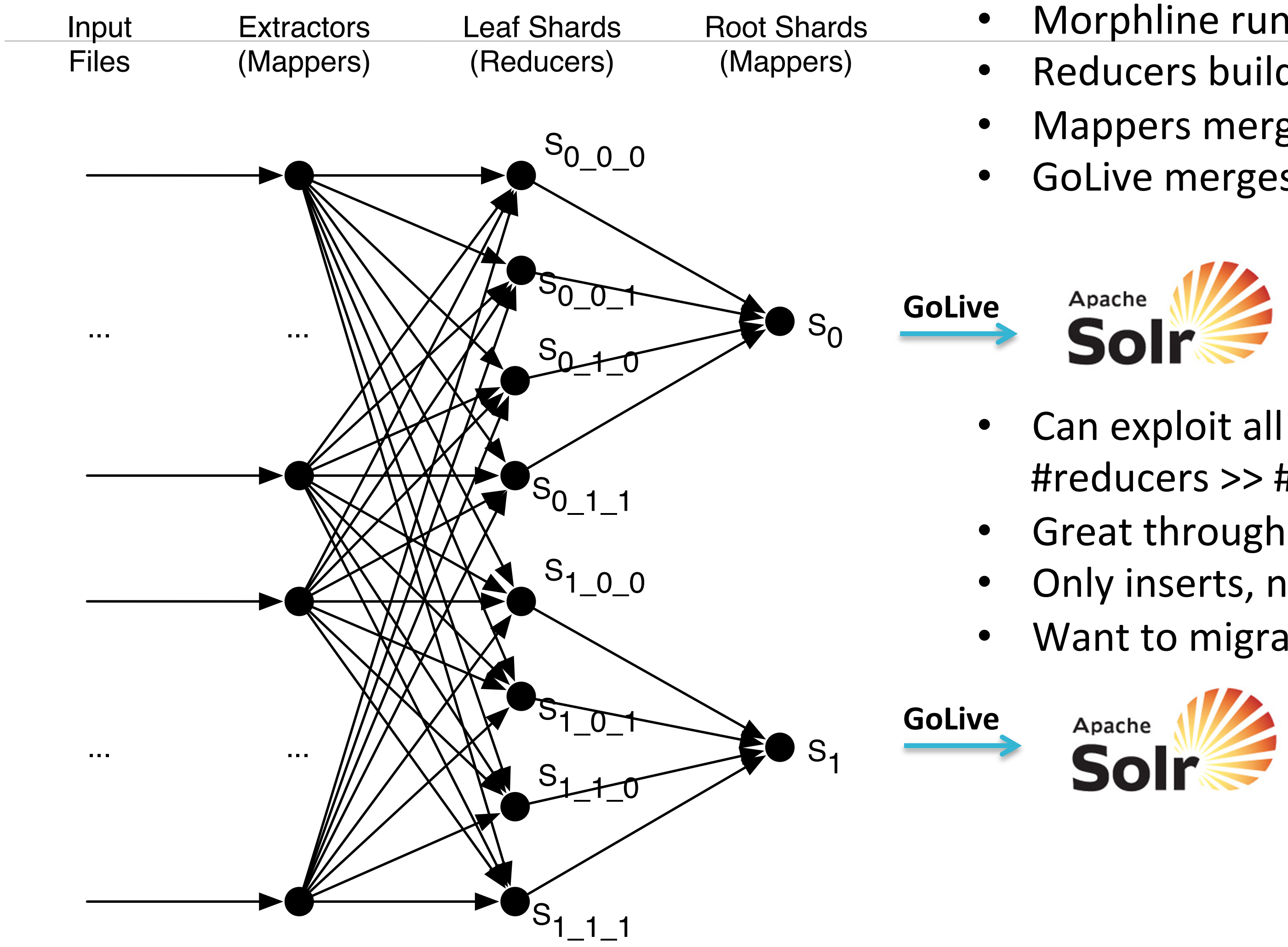
```
morphlines : [
  {
    id : morphline1
    importCommands : [”org.kitesdk.**”]
    commands : [
      { java
        {
          code: """
            List tags = record.get("tags");
            if (!tags.contains("hello")) {
              return false;
            }
            tags.add("world");
            return child.process(record);
                """
        }
      }
    ]
  }
]
```

Example Java Driver Program - Can be wrapped into Spark functions

```
/** Usage: java ... <morphline.conf> <dataFile1> ... <dataFileN> */
public static void main(String[] args) {
  // compile morphline.conf file on the fly
  File conf= new File(args[0]);
  MorphlineContext ctx= new MorphlineContext.Builder().build();
  Command morphline = new Compiler().compile(conf, null, ctx, null);

  // process each input data file
  Notifications.notifyBeginTransaction(morphline);
  for (int i = 1; i < args.length; i++) {
    InputStream in = new FileInputStream(new File(args[i]));
    Record record = new Record();
    record.put(Fields.ATTACHMENT_BODY, in);
    morphline.process(record);
    in.close();
  }
  Notifications.notifyCommitTransaction(morphline);
}
```

Scalable Batch Indexing



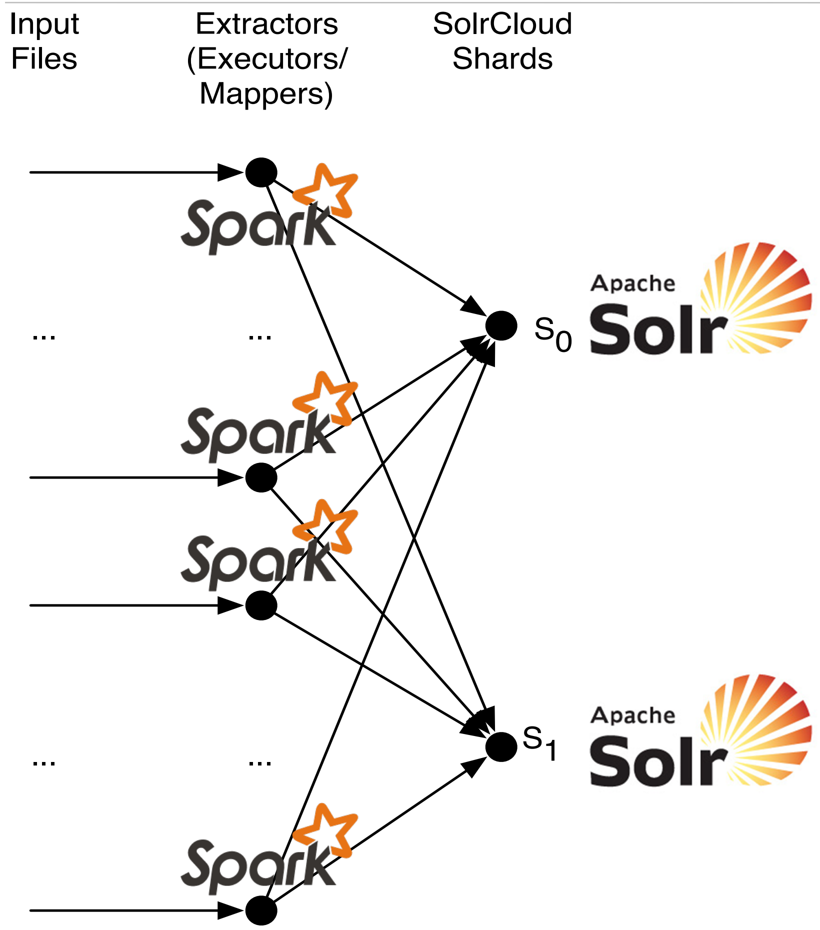
- Morphline runs inside Mapper
- Reducers build local Solr indexes
- Mappers merge microshards
- GoLive merges into live SolrCloud

- Can exploit all reducer slots even if #reducers >> #solrShards
- Great throughput but poor latency
- Only inserts, no updates & deletes!
- Want to migrate from MR to Spark

```
hadoop ... MapReduceIndexerTool --morphline-file morphline.conf ...
```

cloudera
Ask Bigger Questions

Batching Indexing with CrunchIndexerTool



- Morphline runs inside Spark executors
- Morphline sends docs to live SolrCloud
- Good throughput and good latency
- Supports inserts, updates & deletes
- Flag to run on Spark or MapReduce

```
spark-submit ... CrunchIndexerTool --morphline-file morphline.conf ...
```

or

```
hadoop ... CrunchIndexerTool --morphline-file morphline.conf ...
```

More CrunchIndexerTool features (1/2)

- Implemented with Apache Crunch library
  - Eases migration from MapReduce execution engine to Spark execution engine – can run on either engine
- Supported Spark modes
  - Local (for testing)
  - YARN client
  - YARN cluster (for production)
- Efficient batching of Solr updates and deleteById and deleteByQuery
- Efficient locality-aware processing for splittable HDFS files
  - avro, parquet, text lines

More CrunchIndexerTool features (2/2)

- Dry-run mode for rapid prototyping
- Sends commit to Solr on job success
- Inherits Fault tolerance & retry from Spark (and MR)
- Security in progress: Kerberos token delegation, SSL
- ASL licensed on Github
    - https://github.com/cloudera/search/tree/cdh5-1.0.0_5.3.0/search-crunch

Conclusions

- Easy migration from MapReduce to Spark
- Also supports updates & deletes & good latency
- Recommendation
  - Use MapReduceIndexerTool for large scale batch ingestion use cases where updates or deletes of existing documents in Solr are not required
  - Use CrunchIndexerTool for all other use cases

cloudera®
Ask Bigger Questions