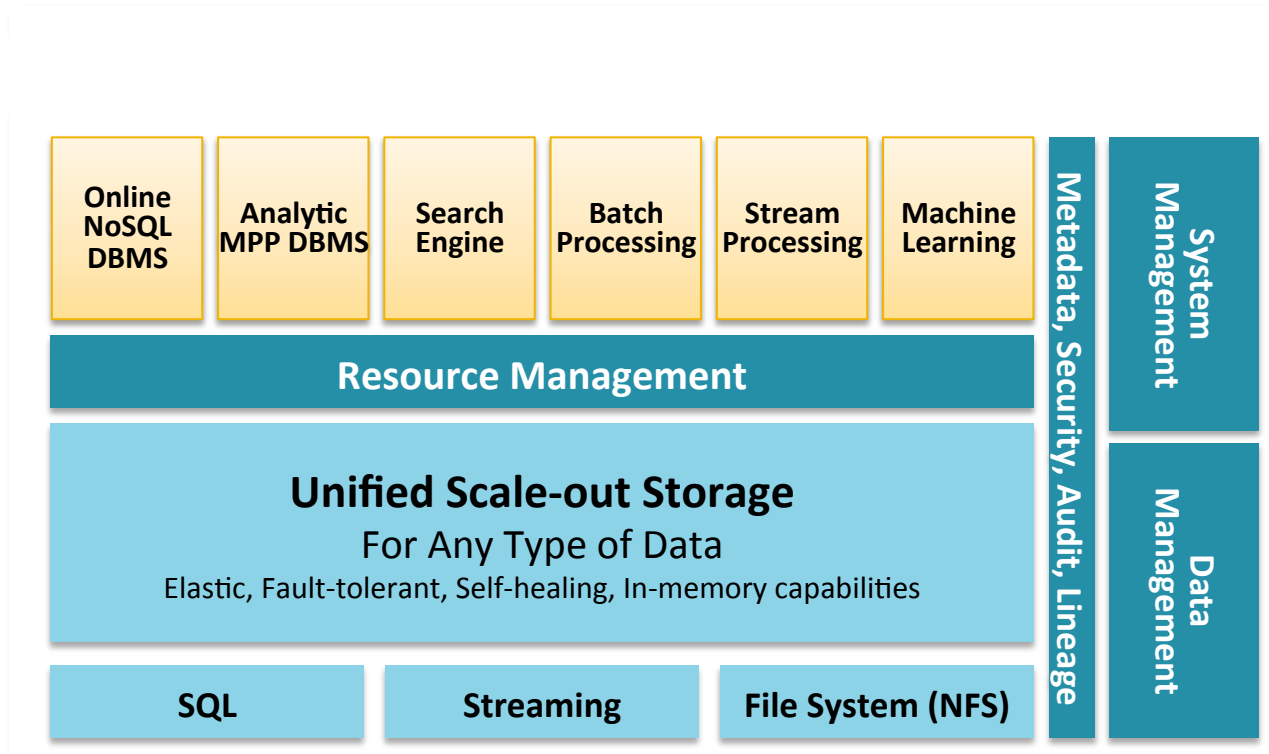


# Adding Search as a First Class Citizen to Hadoop

---

Wolfgang Hoschek ([whoschek@cloudera.com](mailto:whoschek@cloudera.com))  
Sr. Software Engineer @ Cloudera Search & Hadoop Platform Team  
CERN Seminar, Geneva

# The Enterprise Data Hub



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

# Search Simplifies Interaction - to Everyone!

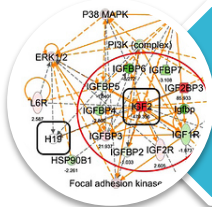
---



Explore



Navigate



Correlate

Experts know MapReduce. Savvy people know SQL.

**Everyone knows Search!**

# What is Cloudera Search?

---

## Interactive search for Hadoop

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

## Apache Solr integrated with CDH

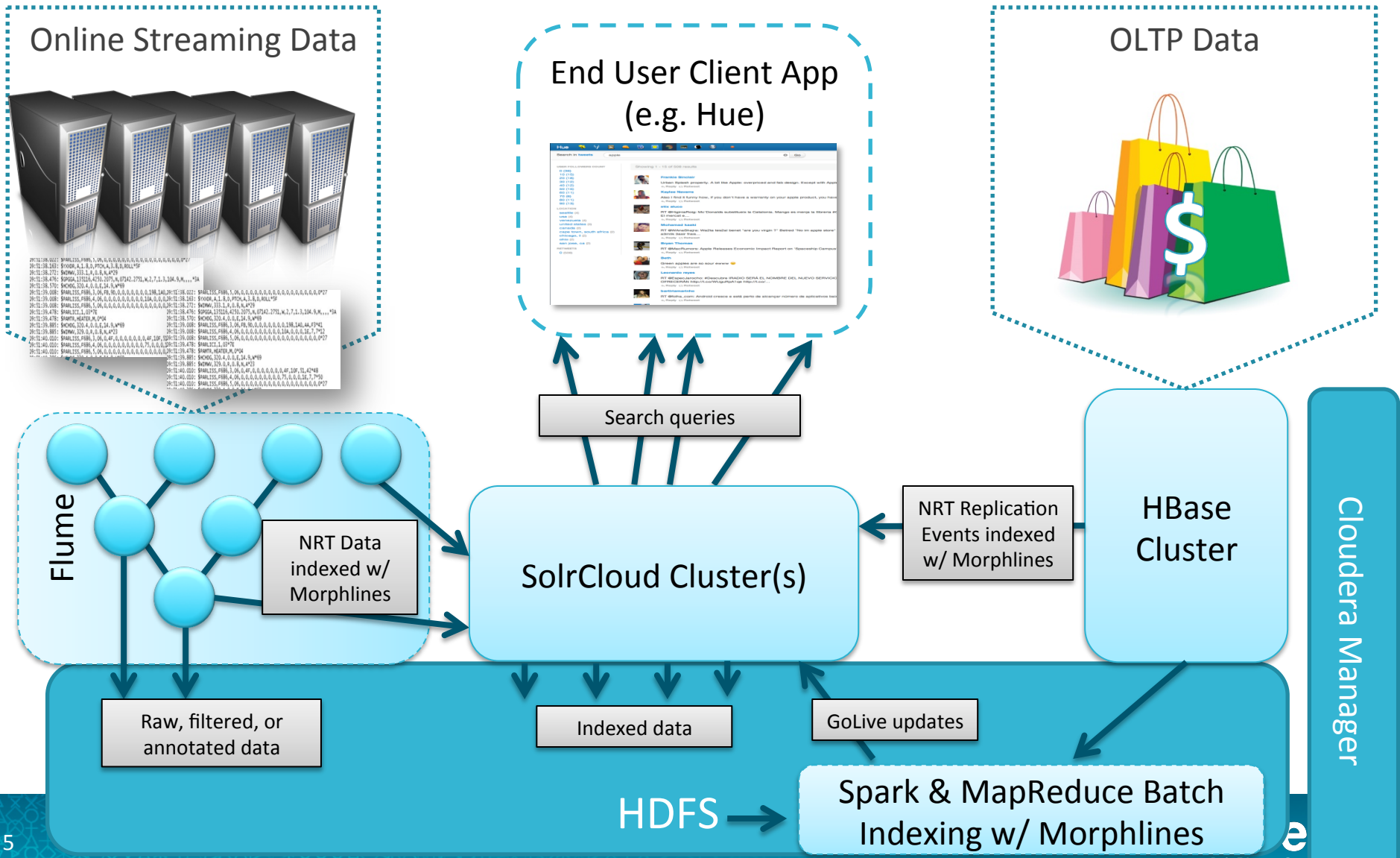
- Established, mature search with vibrant community
- Incorporated as part of the Hadoop ecosystem
  - Apache Flume, Apache HBase
  - Apache MapReduce, Kite Morphlines
  - Apache Spark, Apache Crunch

## Open Source

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



# Cloudera Search Architecture Overview



# Challenges

---

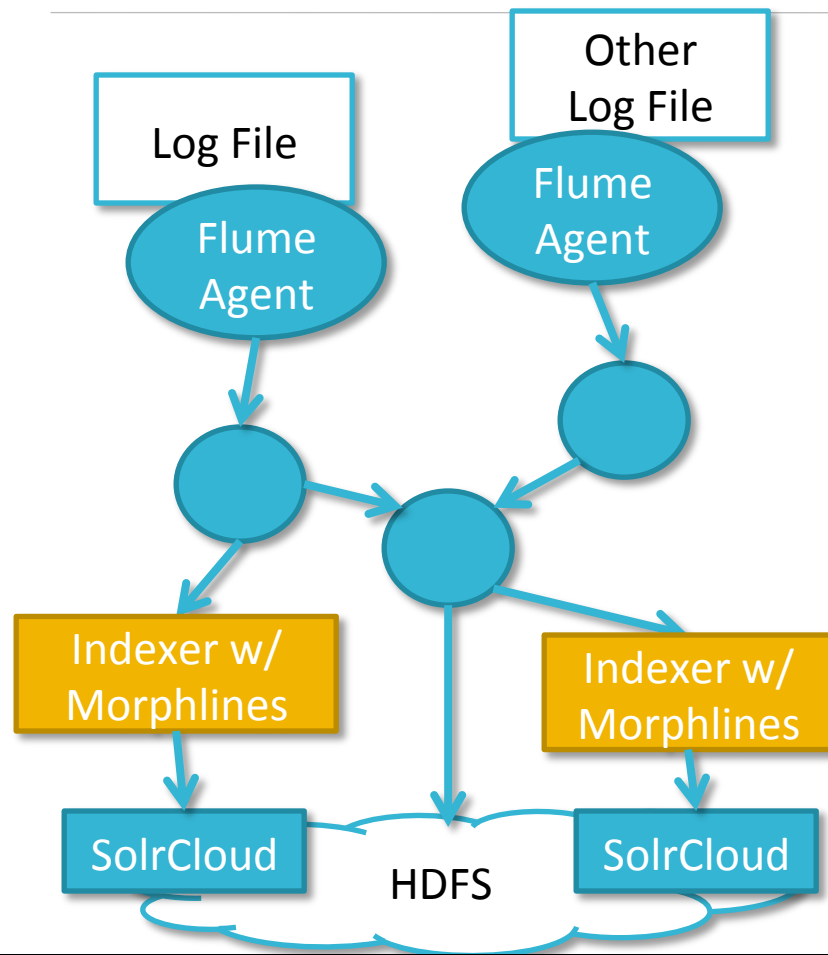
- Scalable/Reliable Index Storage
- Near Real Time (NRT) indexing
- Scalable Batch Indexing
- Usability

# Apache Lucene/Solr

---

- Lucene - full text search library
- Solr – search service on Lucene
- SolrCloud – distributed search
  - Partitioned and replicated inverted index
  - Low latency, scalable, reliable, HA, secure
  - Uses Zookeeper
- Solr on HDFS
  - Scalable, cost-efficient index storage
  - Higher availability
  - Search *and* process data in *one* platform

# Near Real Time Indexing with Apache Flume



## Solr and Flume

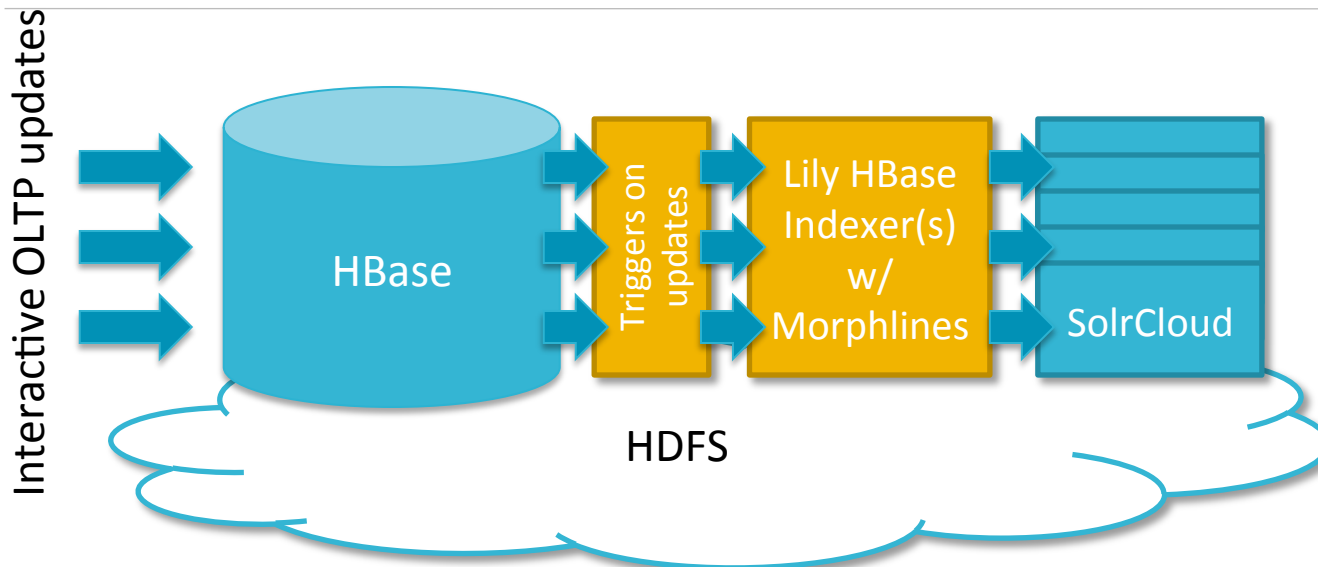
- Reliable streaming data ingestion at scale
- Indexing at data ingest
- Flexible ETL via Morphlines
- Packaged as “Flume Morphline Solr Sink”

Flume.conf

```
agent.sinks.solrSink.type = org.apache.flume.sink.solr.morphline.MorphlineSolrSink
agent.sinks.solrSink.morphlineFile = /etc/flume-ng/conf/morphline.conf
```



# Searchable Real-Time Data – Indexing HBase

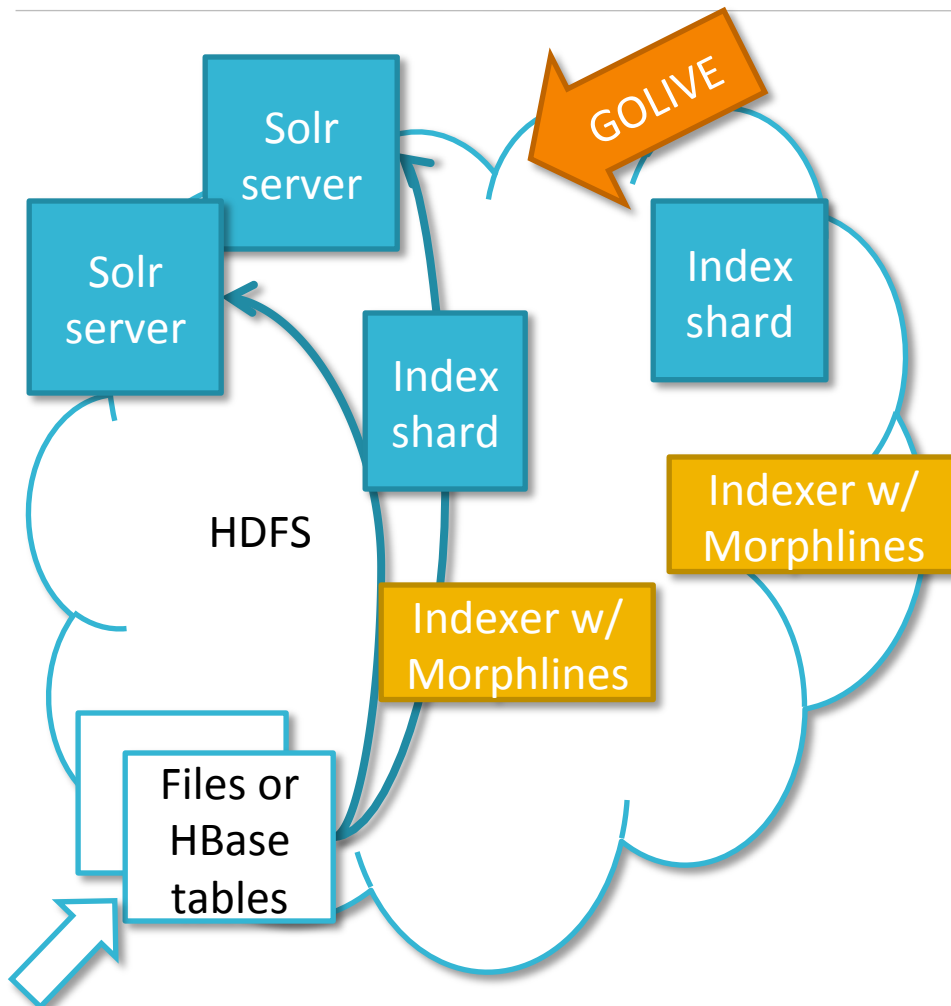


- Updates Solr index immediately on HBase cell update
- Service that acts as a HBase replication listener
- Non-intrusive, Flexible, Scalable, Reliable
- Partitioned and Replicated Indexing w/ Failover, HA
- <http://github.com/ngdata/hbase-indexer>

`hbase-indexer.conf`

```
<indexer table="myHBaseTable"  
mapper="com.ngdata.hbaseindexer.morphline.MorphlineResultToSolrMapper">  
  <param name="morphlineFile" value="/path/to/extractHBaseCell.conf"/>
```

# 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”

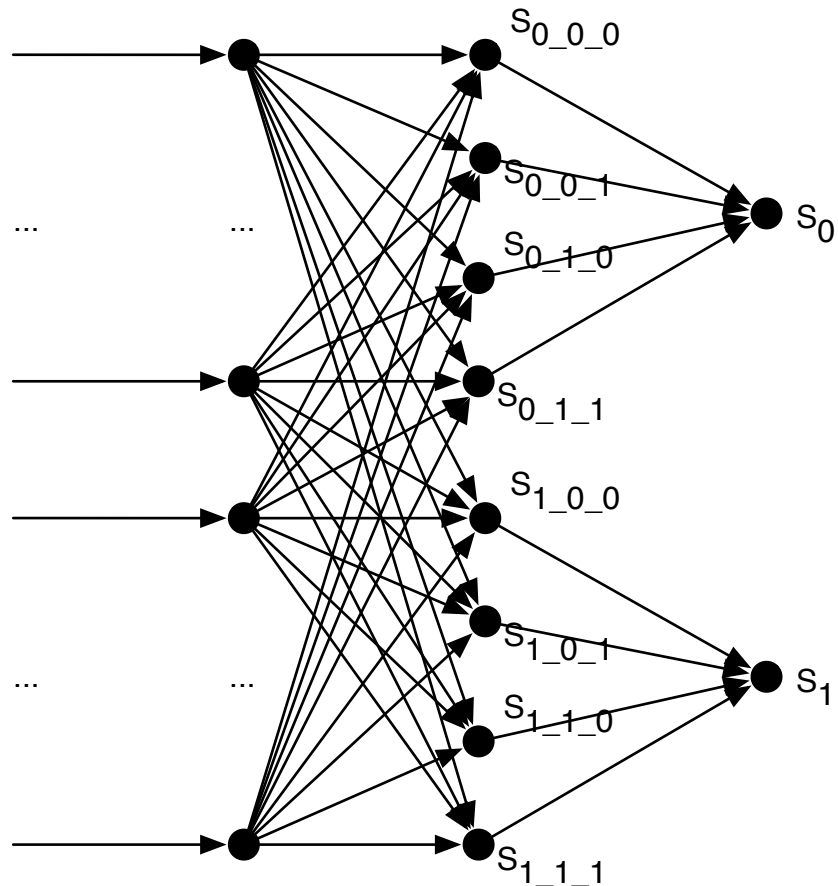
## Solr and Spark

- “CrunchIndexerTool on Spark”

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

# Scalable Batch Indexing

Input Files	Extractors (Mappers)	Leaf Shards (Reducers)	Root Shards (Mappers)
-------------	----------------------	------------------------	-----------------------

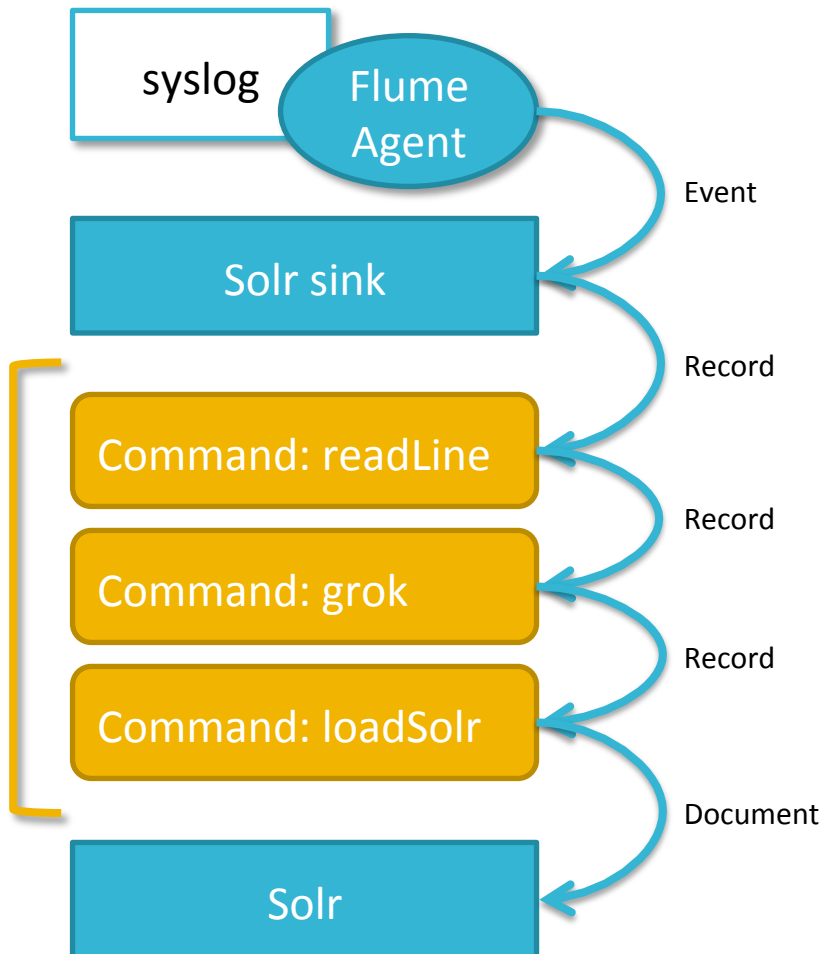


- Morphline runs inside Mapper
- Can exploit all reducer slots even if #reducers >> #solrShards



```
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.

# 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);  
          """  
        }  
      }  
    ]  
  }  
]
```

# 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 Plugin Commands

---

- Easy to add new I/O & transformation cmds
- Integrate existing functionality and third party systems
- Implement Java interface `Command` or subclass `AbstractCommand`
- Add it to Java classpath
- No registration or other administrative action required



# Morphline Performance and Scaling

---

- The runtime compiles morphline on the fly
- The runtime processes all commands of a given morphline in the same thread
- Piping a record from one command to another is fast
  - just a cheap Java method call
  - no queues, no handoffs among threads, no context switches, and no serialization between commands
- For scalability, deploy many morphline instances on a cluster in many Flume agents and MapReduce tasks and Spark tasks

# Security

---

## Cloudera Search + Sentry

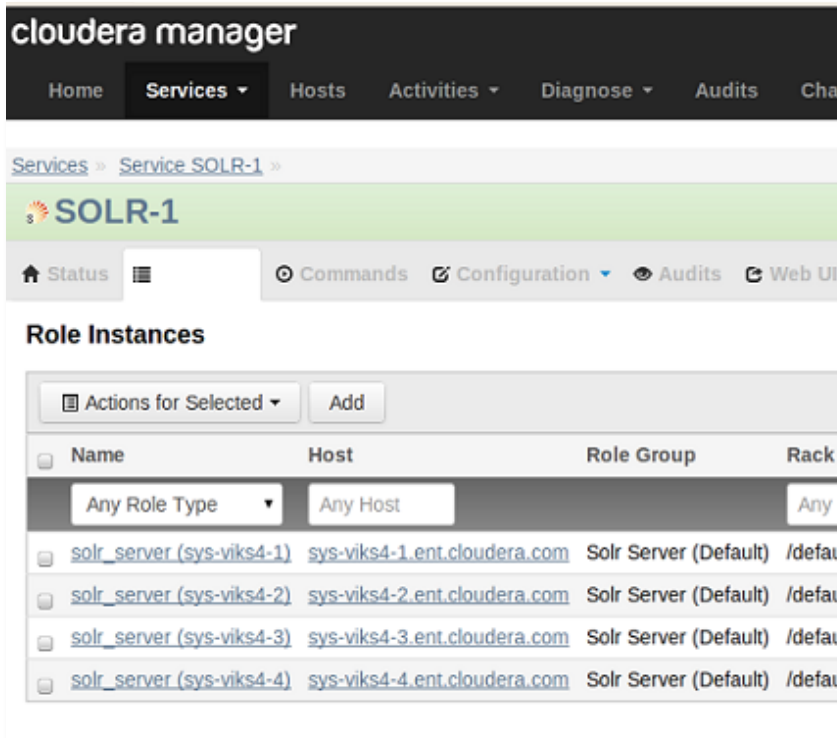
- Kerberos
- Encryption on the wire and at rest
- Cluster level access control
- Index level access control
- Document level access control
- Field level access control

# Customizable Hue UI

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



# Simplified Management



## Cloudera Manager

- Install, configure, deploy Solr services on the cluster
- Unified management and monitoring
- Resource management

# Simplified Configuration

The screenshot shows the Cloudera Manager interface for configuring a Flume-NG Solr Sink. The top navigation bar includes 'Services', 'Hosts', 'Activities', 'Diagnose', 'Audits', 'Charts', 'Reports', and 'Administration'. The breadcrumb trail is 'Services > Service flume1 > flume1'. A 'Concerning Health' indicator is present in the top right. The main navigation includes 'Status', 'Instances', 'Commands', 'Configuration', 'Audits', and 'Charts Library'. The configuration page has a search bar, a 'Notes' field, and a 'Save Changes' button.

Category	Property	Value	Description
Default	Morphlines File	<pre>SOLR_LOCATOR : {   # Name of solr collection   collection : collection3    # ZooKeeper ensemble   zkHost : "\$ZK_HOST"    # The maximum number of documents to send to Solr   per network batch (throughput knob)   # batchSize : 100 }</pre> <pre>morphlines : [   {     id : morphline1     importCommands : ["com.cloudera.**", "org.apache.solr.**"]     commands : [ </pre>	Text that goes into morphlines.conf file used by the Flume-NG Solr sink. The text goes verbatim into the config file except that \$ZK_HOST is replaced by the ZooKeeper quorum of the Solr service.

# Conclusions

---

- Experts know MapReduce. Savvy people know SQL. Everyone knows Search!
- Integrated Search seamlessly into Hadoop
- Cloudera Search available for CDH 4 & CDH5
  - Free [Download](#)



**cloudera<sup>®</sup>**  
Ask Bigger Questions

©2014 Cloudera, Inc. All rights reserved.