.conf2014

YOUR DATA ADVENTURE

Hunk 6.1

Ledion Bitincka

Principal Architect, Splunk

splunk>

Disclaimer

During the course of this presentation, we may make forward-looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC. The forward-looking statements made in the this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward-looking statements we may make. In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only, and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

About Me

- Principal Architect
- 7+ years at Splunk
- Mainly involved in search time stuff:
  - Hunk
  - Key-value pair extraction
  - Scheduler & Alerting
  - Transactions, eventtypes , tags etc
  - MySQLConnect, HadoopConnect
- @ledbit

Agenda

- The problem
- Hunk architecture
- Virtual indexes
- Computation models
- What's new in 6.1

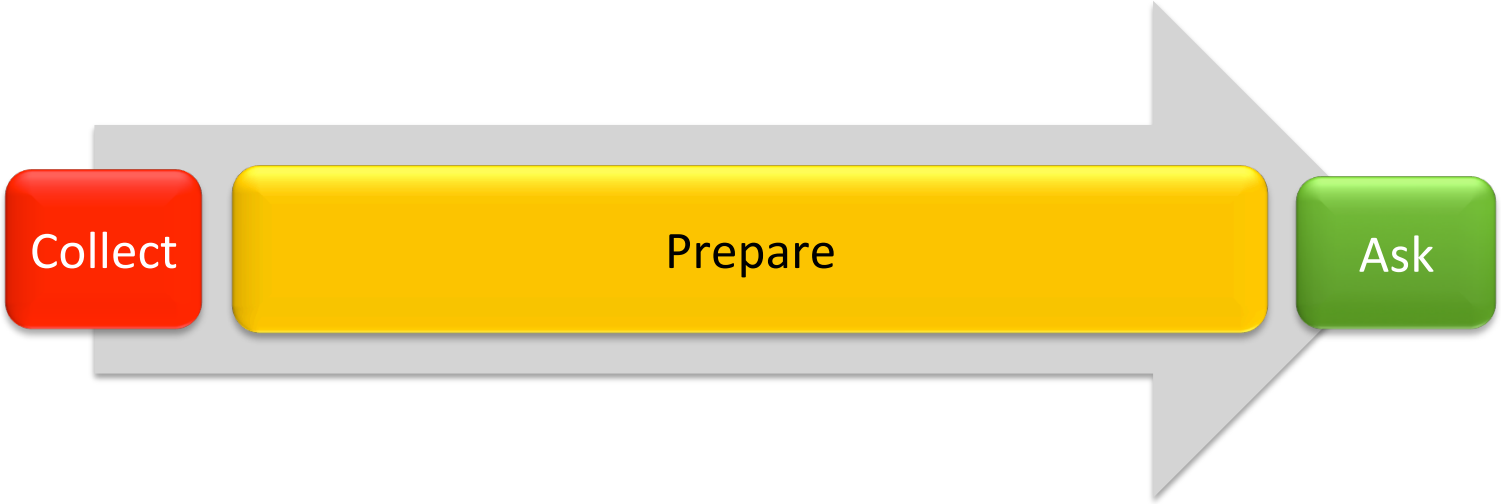
Got Problem?

splunk>

The Problem

- Easy to get data into Hadoop
- Large amounts of data already in Hadoop
- Hard to get value out

Data → Value (Today)



Data → Value (Ideally)

What If?

Hadoop + Splunk =

Hadoop + Splunk = Hunk

Solution Goals

- A viable solution must:
  - Process the data in place
  - Maintain support for Splunk Processing Language (SPL)
  - True schema on read
  - Query previews
  - Ease of setup & use

Support SPL

- Naturally suitable for MapReduce
- Reduces adoption time
- Challenge: Hadoop “apps” written in Java & all SPL code is in C++
- Porting SPL to Java would be a daunting task
- Reuse the C++ code somehow
  - Use “splunkd” (the binary) to process the data
  - JNI is not easy nor stable

Schema on Read

- Apply Splunk's index-time schema at search time
  - Event breaking, time stamping etc
- Anything else would be brittle & maintenance nightmare
- Extremely flexible
- Runtime overhead (manpower >>$ computation)
- Challenge: Hadoop "apps" written in Java & all index-time schema logic is implemented in C++

Intermediate Results

- No one likes to stare at a blank screen!
- Challenge: Hadoop is designed for batch-like jobs

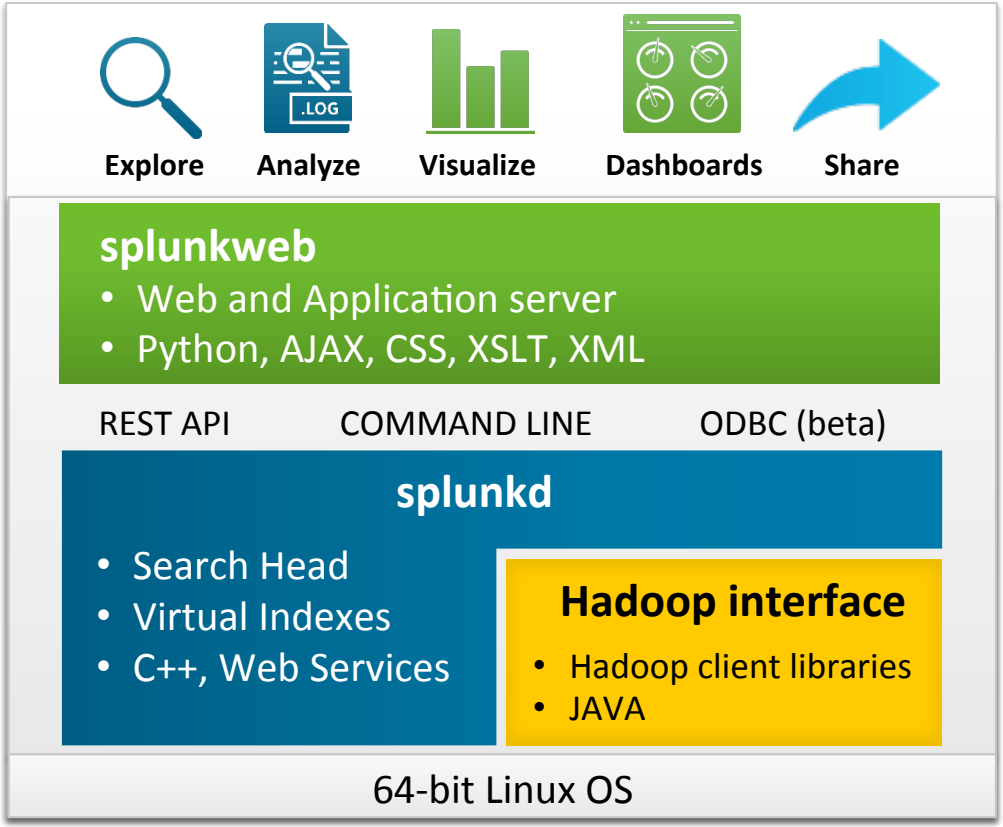
Ease of Setup & Use

- Users should just specify:
  - Hadoop cluster they want to use
  - Data within the cluster they want to process
- Immediately be able to explore & analyze their data

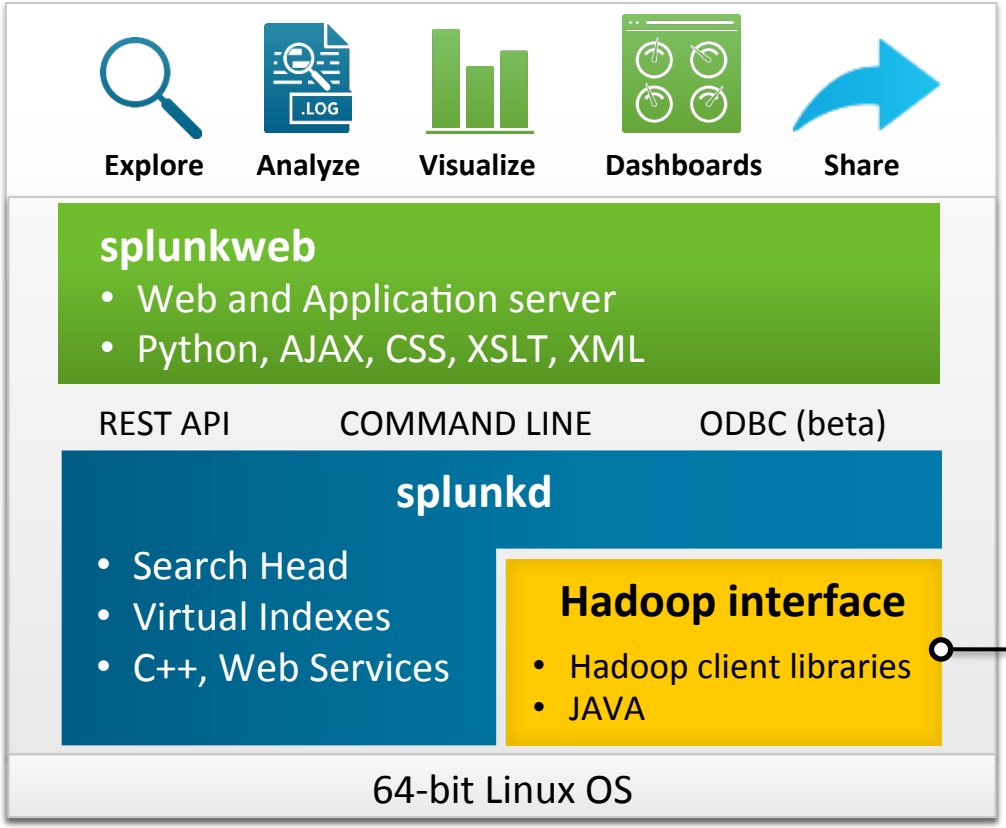
Architecture

splunk>

Hunk Server

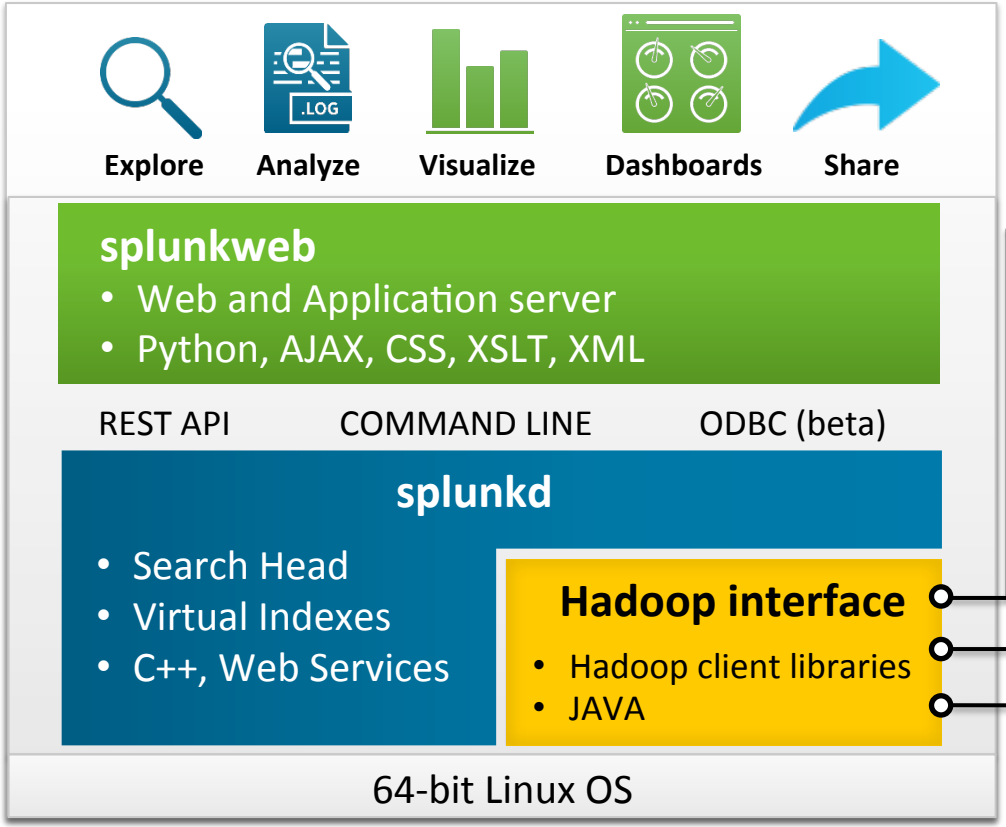

Explore
Analyze
Visualize
Dashboards
Share
splunkweb
• Web and Application server
• Python, AJAX, CSS, XSLT, XML
REST API
COMMAND LINE
ODBC (beta)
splunkd
• Search Head
• Virtual Indexes
• C++, Web Services
Hadoop interface
• Hadoop client libraries
• JAVA
64-bit Linux OS

Connecting to Hadoop



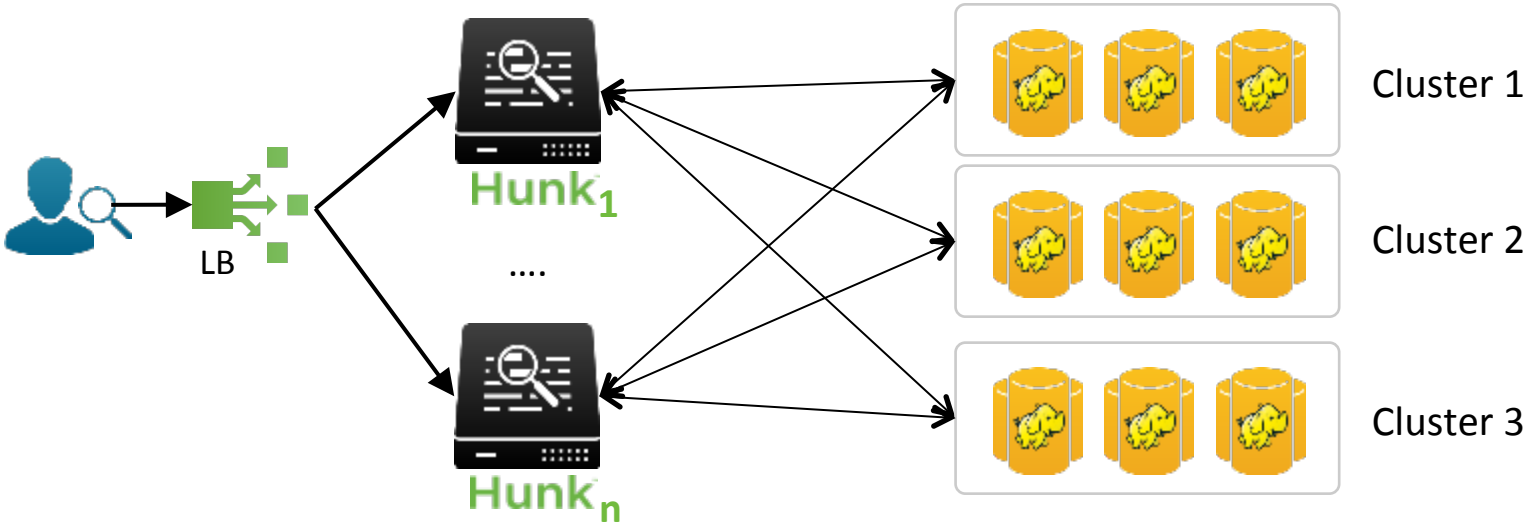
Connect to Apache HDFS and MapReduce or your choice of Hadoop distribution

Multiple Hadoop Clusters



Connect Hunk to multiple Hadoop clusters

Deployment Overview (Advanced)



- Load balance users across
- Hunk Search Head pooling/cluster
- Multiple Hadoop cluster

Virtual Indexes

splunk>

SPL Overview

search index=main | top user | fields - percent

sourcetype	raw	User	<fields...>
syslog	…	…	…
syslog	… ERROR …	user_A	…
other-source	…	…	…
syslog	… ERROR …	user_A	…
syslog	… WARNING …	user_A	…
syslog	… WARNING …	user_A	…
other-source	…	…	…
syslog	… ERROR …	user_B	…
other-source	…	…	…
<events…>	…	…	…

Events fetched from disk

User	count	percent
user_01	22	22
user_02	17	17
…	…	…
user-10	5	5

Summarize into table of top ten users

User	count	percent
user_01	22	~~22~~
user_02	17	~~17~~
…	…	…
user-10	5	~~5~~

Remove "percent" column

User	count
user_01	22
user_02	17
…	…
user-10	5

Final results

`top user` `fields - percent`

SPL Overview

- Search Processing Language = SPL
- Motivated by Unix shell pipes
- First command is always responsible for **event retrieval**
  - Generally, events are retrieved from Splunk's **native indexes**
- Follow-on commands transform events to final results

Native Indexes

Native
Serve as data containers
Access control
Read/writes
Data retention policies
Optimized for keyword searches
Optimized for time range searches

Native Indexes vs. Virtual Indexes

Native	Virtual
Serve as data containers	Serve as data containers
Access control	Access control
Read/writes	Read only
Data retention policies	–
Optimized for keyword searches	–
Optimized for time range searches	Available via regex/pruning

Hunk's Core Technology

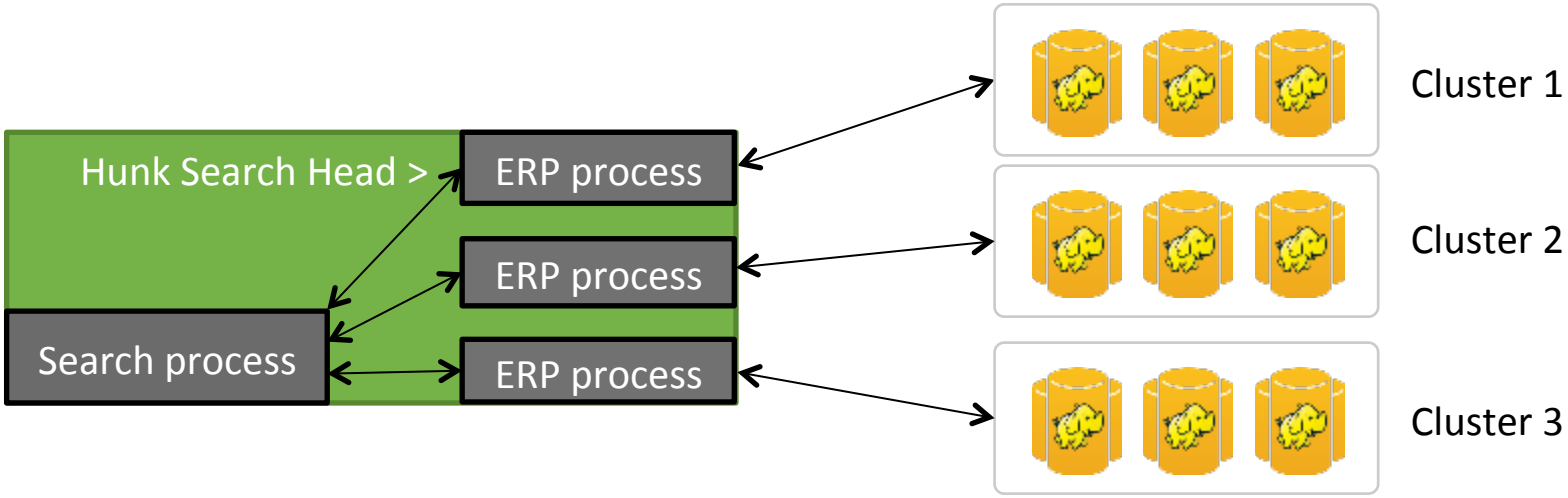
Virtual Indexes (VIX)

External Result Providers (ERPs)

External Result Providers

- Search time helper process responsible for:
  - Access external system
    e.g. Hadoop, Cassandra, RDBMs etc
  - Translate/interpret search request
  - Push computation to external system

External Result Providers (ERPs)



For each Hadoop cluster (or external system) the search process spawns an ERP process which is responsible for executing the (remote part of the) search on that system.

Computation Models

splunk>

Move Data to Computation (Streaming)

- Move data from HDFS to Search Head
- Process it in a streaming fashion
- Visualize the results
- **Problem?**

Move Computation to Data (Reporting)

- Create and start a MapReduce job to do the processing
- Monitor MR job & collect its results
- Merge the results and visualize
- **Problem?**

Search Modes

Streaming	Reporting	
Pull data from HDFS to SH for processing	Push compute down to DN/ TT and consume results	
Low Latency	**High Latency**	
Low Throughput	**High Throughput**	

Low Latency = Interactivity = VALUE
High Throughput = Process larger datasets = VALUE

Search Modes

Streaming	Reporting	Mixed Mode
Pull data from HDFS to SH for processing	Push compute down to DN/ TT and consume results	Start both Streaming and Reporting modes. Show Streaming results until Reporting starts to complete
Low Latency	**High Latency**	**Low Latency**
Low Throughput	**High Throughput**	**High Throughput**

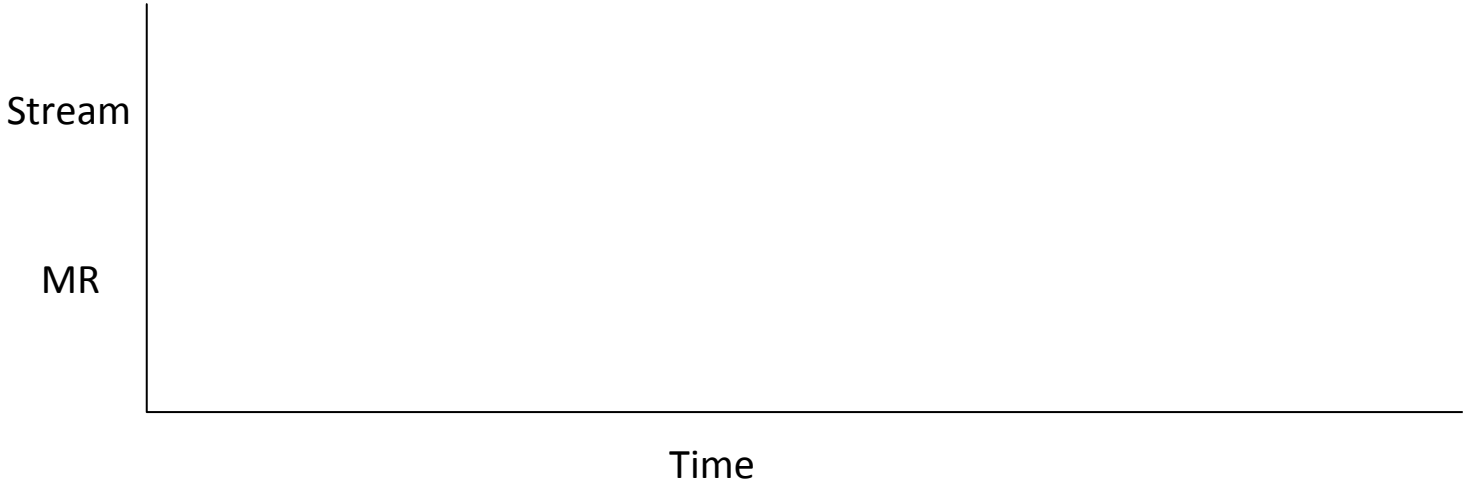
Low Latency = Interactivity = VALUE
High Throughput = Process larger datasets = VALUE

Mixed Mode

- Use **both** computation models concurrently

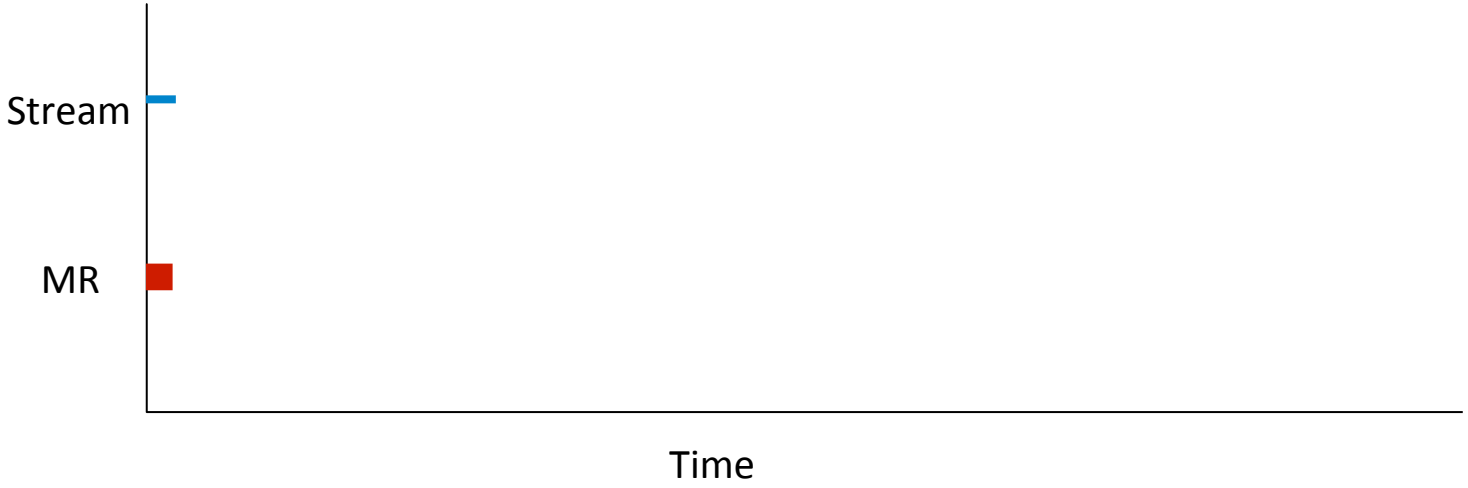
Mixed Mode

- Use **both** computation models concurrently



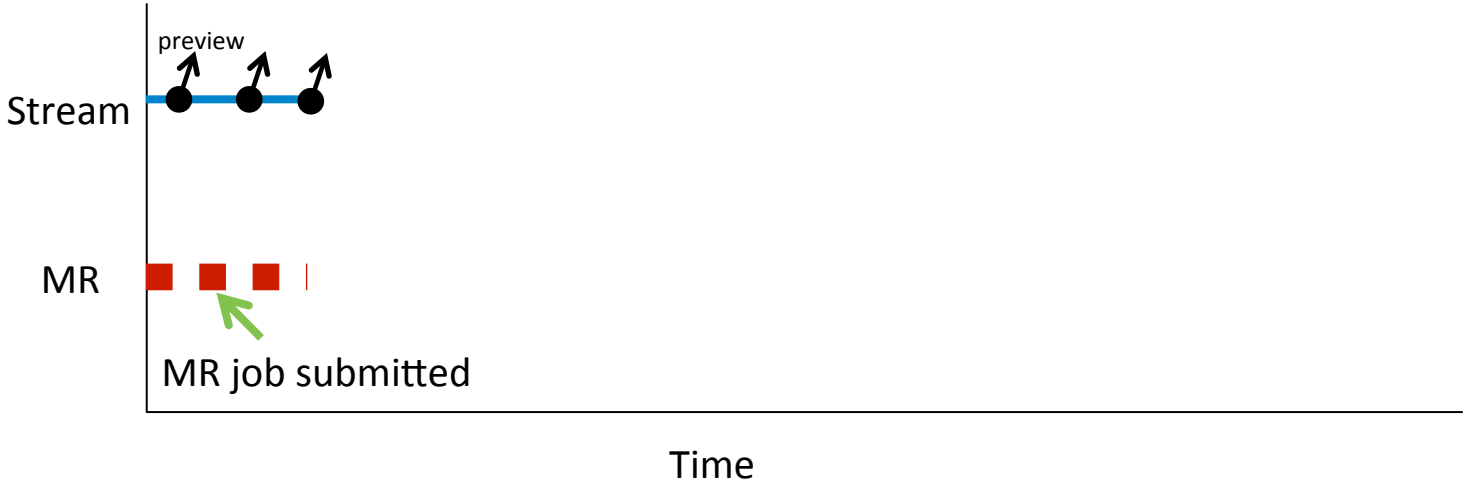
Mixed Mode

- Use **both** computation models concurrently



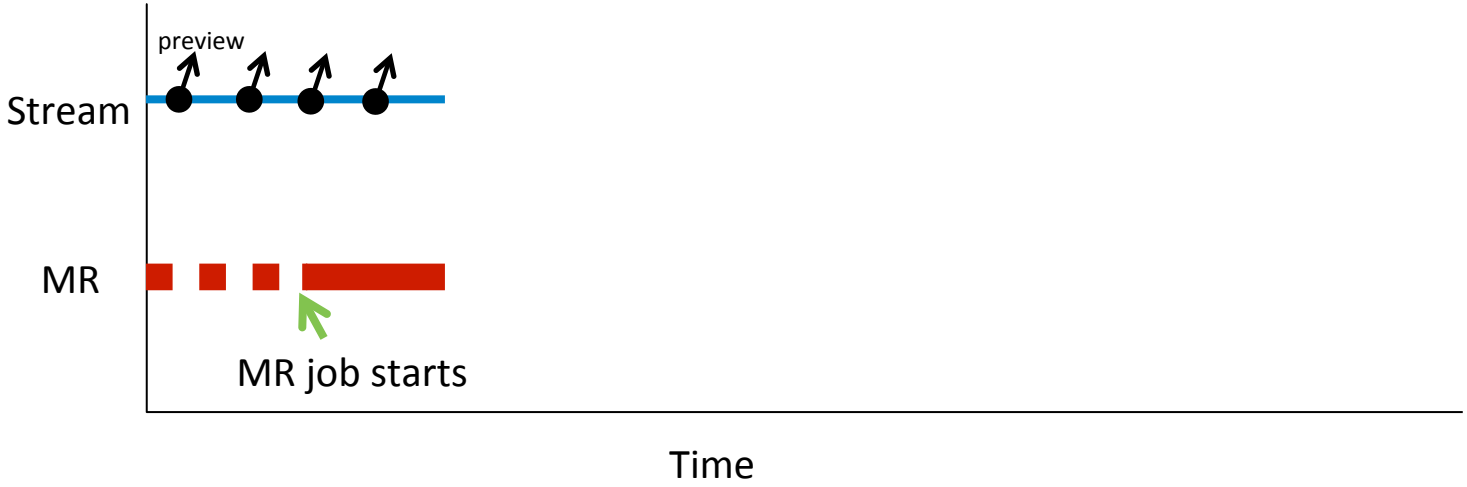
Mixed Mode

- Use **both** computation models concurrently



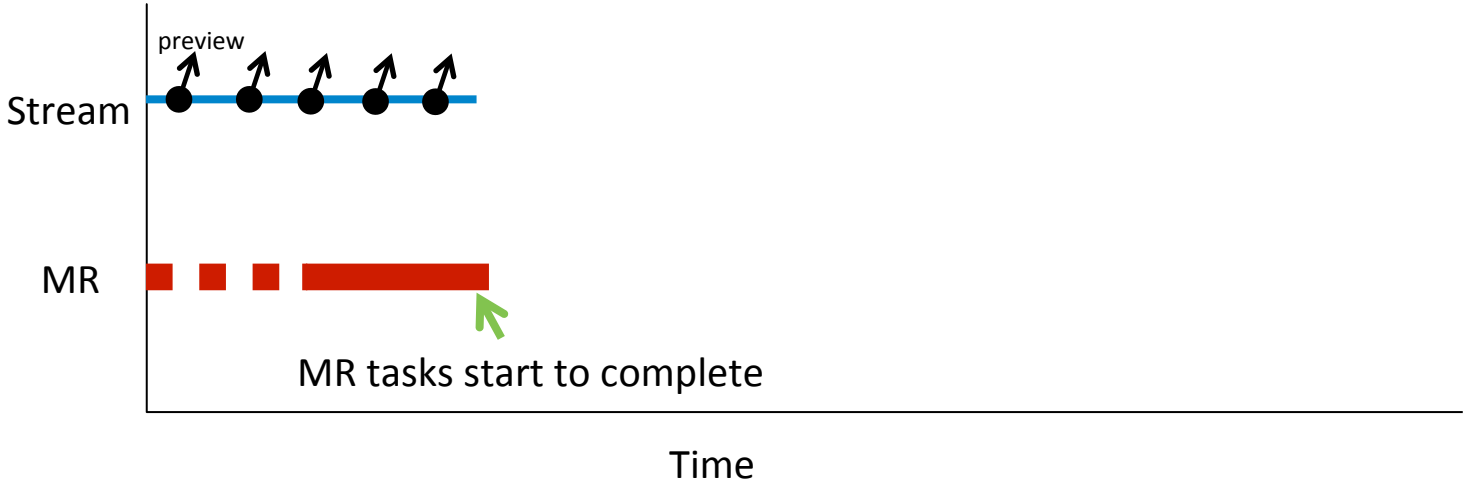
Mixed Mode

- Use **both** computation models concurrently



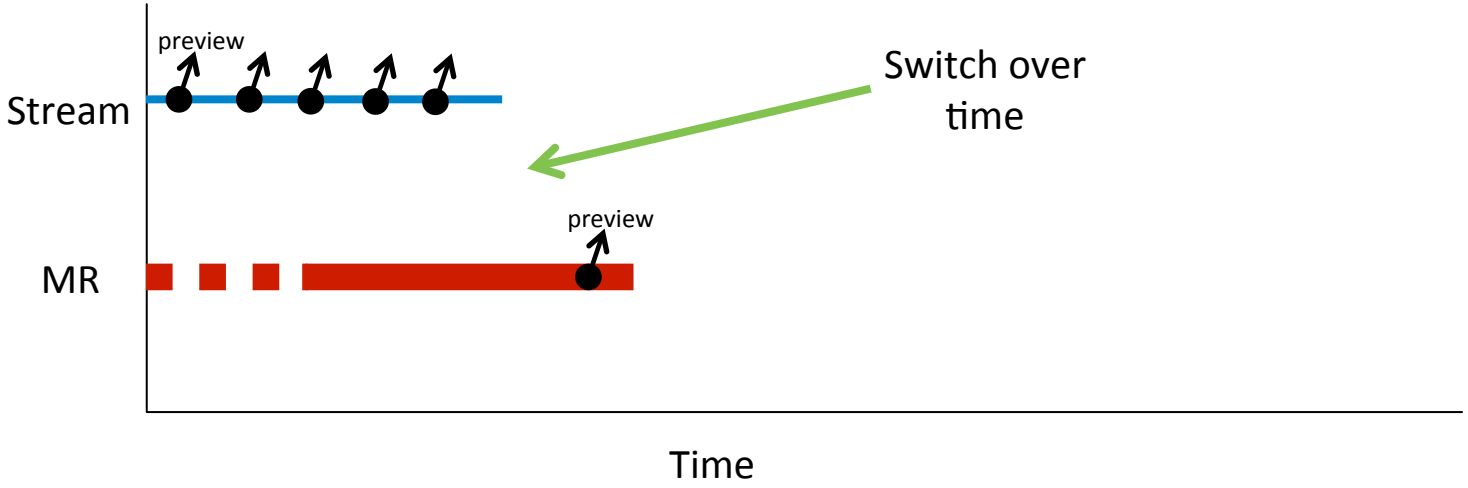
Mixed Mode

- Use **both** computation models concurrently



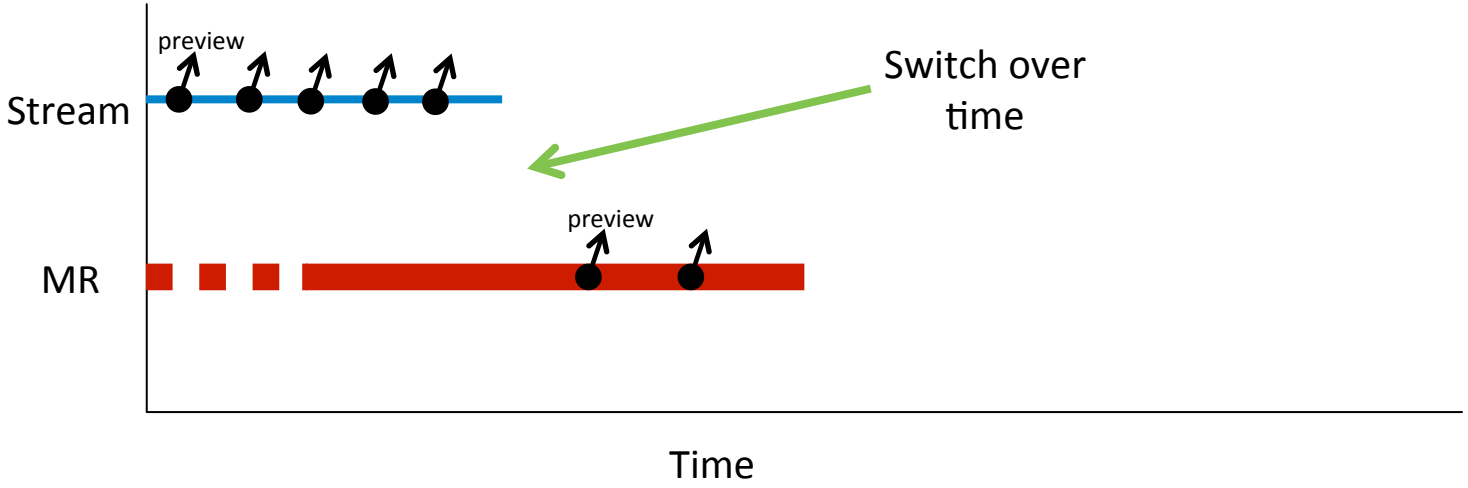
Mixed Mode

- Use **both** computation models concurrently



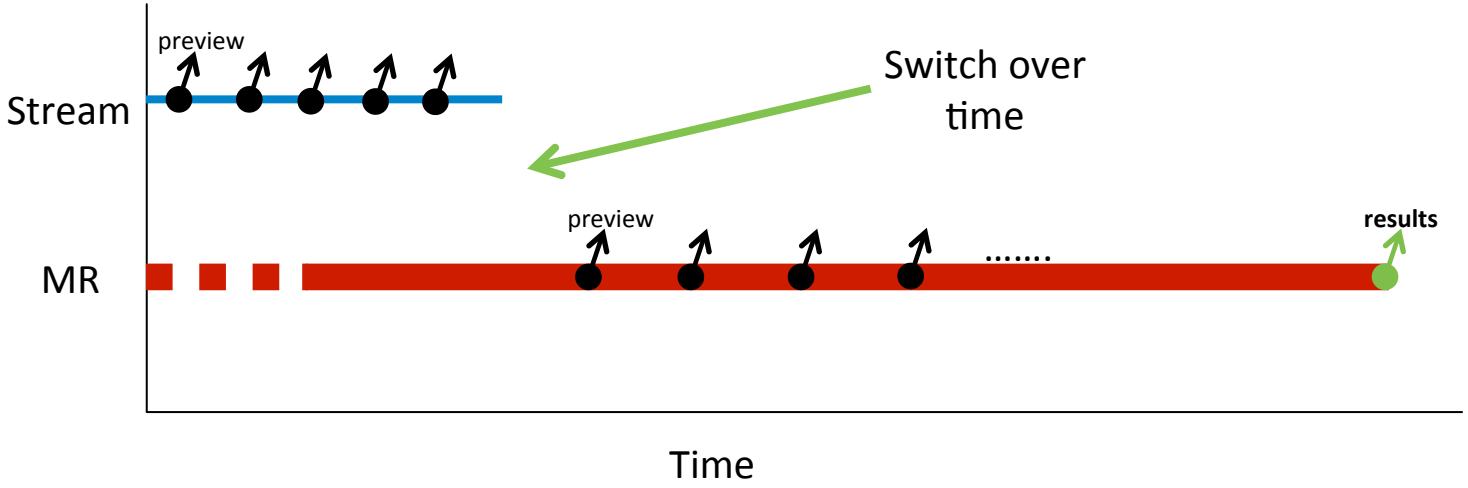
Mixed Mode

- Use **both** computation models concurrently



Mixed Mode

- Use **both** computation models concurrently



New in 6.1

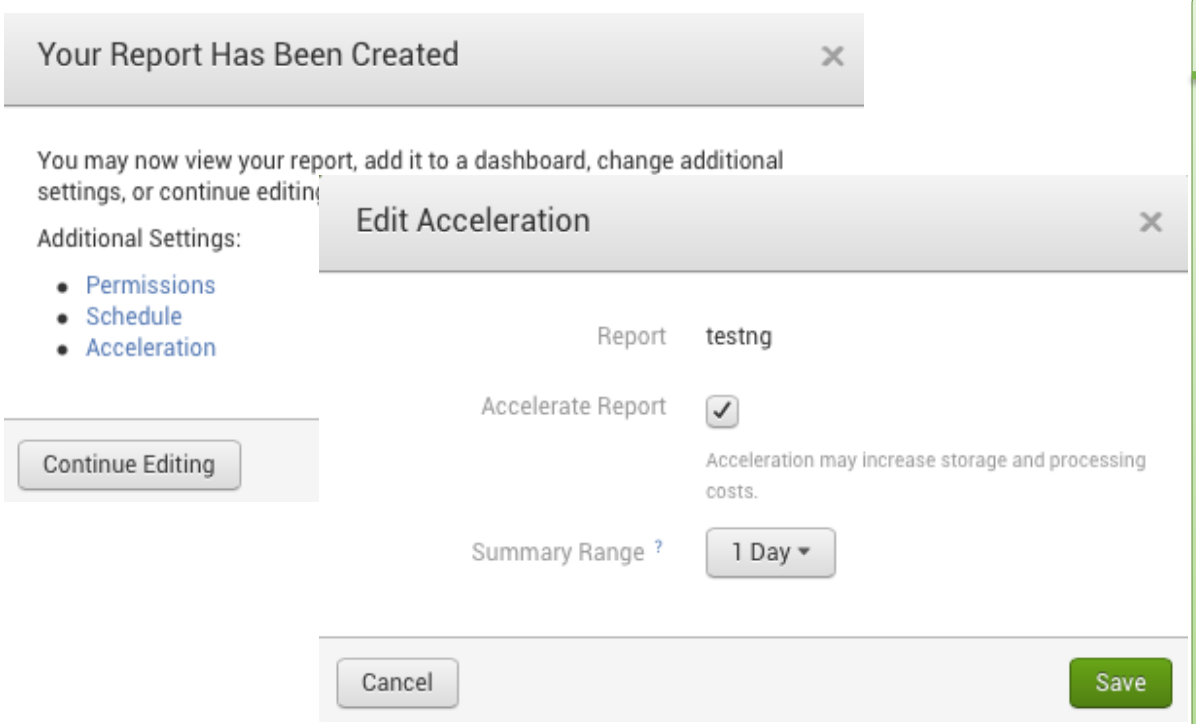
splunk>

More Data …

- Wider support for Hadoop native data formats

Format	Description	Support
Sequence	Key value store	Yes
Avro	Complex objects, with embedded schema	Yes
RC / ORC	Columnar, commonly used by **Hive**	Yes
Parquet	Columnar, commonly used by **Impala**	Yes
Custom	Any other Hadoop file format	Yes

Faster …



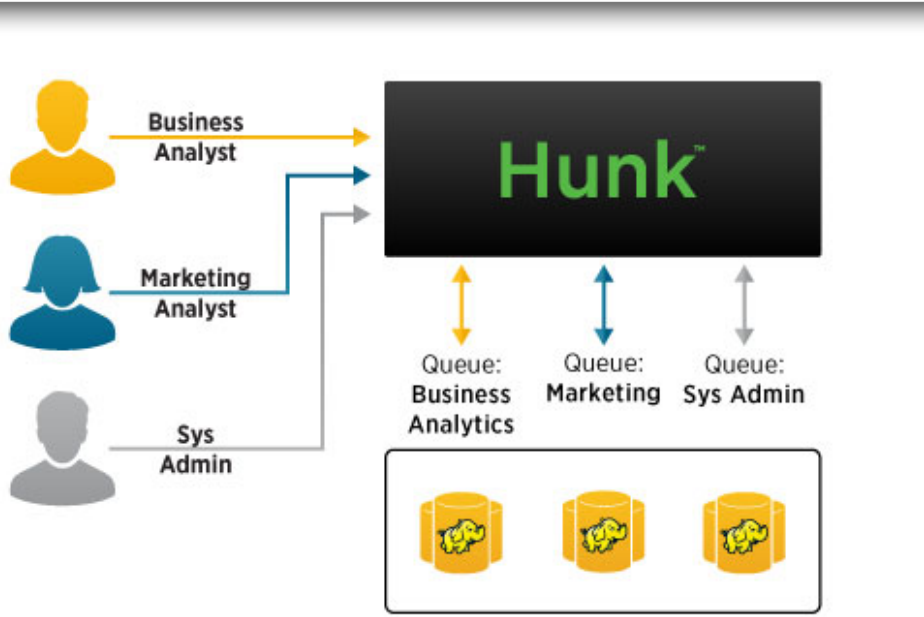
Report Acceleration

- Accelerate searches on virtual indexes served by the Hadoop results provider by reusing Mapper results
- This allows Hunk to accelerate saved searches rather than re-computing the same search
- This feature is identical to Report Acceleration on Splunk Enterprise.

Secure …

Pass-through authentication

- Use LDAP/AD or stand-alone authentication
- Provide role-based security for Hadoop clusters
- Access Hadoop resources under security and compliance
- Integrates with Kerberos for Hadoop security



Open …

Streaming Resource Libraries

mongoDB ACCUMULO™

- Developers stream data for rapid exploration and visualization
- Accumulo/Sqrrl and MongoDB are available on apps.splunk.com

Summary of 6.1

More data …

Faster …

Secure …

Open …

Coming Up in 6.2

splunk>

Helpful resources

- **Download**
  - http://www.splunk.com/hunk
- **Help & Docs**
  - http://docs.splunk.com/Documentation/Hunk/latest/Hunk/MeetHunk
- **Community resource**
  - http://answers.splunk.com