Strata CONFERENCE + HADOOP WORLD

Tools and Techniques That Make Data Work

cloudera

Ask Bigger Questions

Unifying Your Data Management with Hadoop

Strata + Hadoop World 2013

Jayant Shekhar

Sr. Solutions Architect

Machine Data

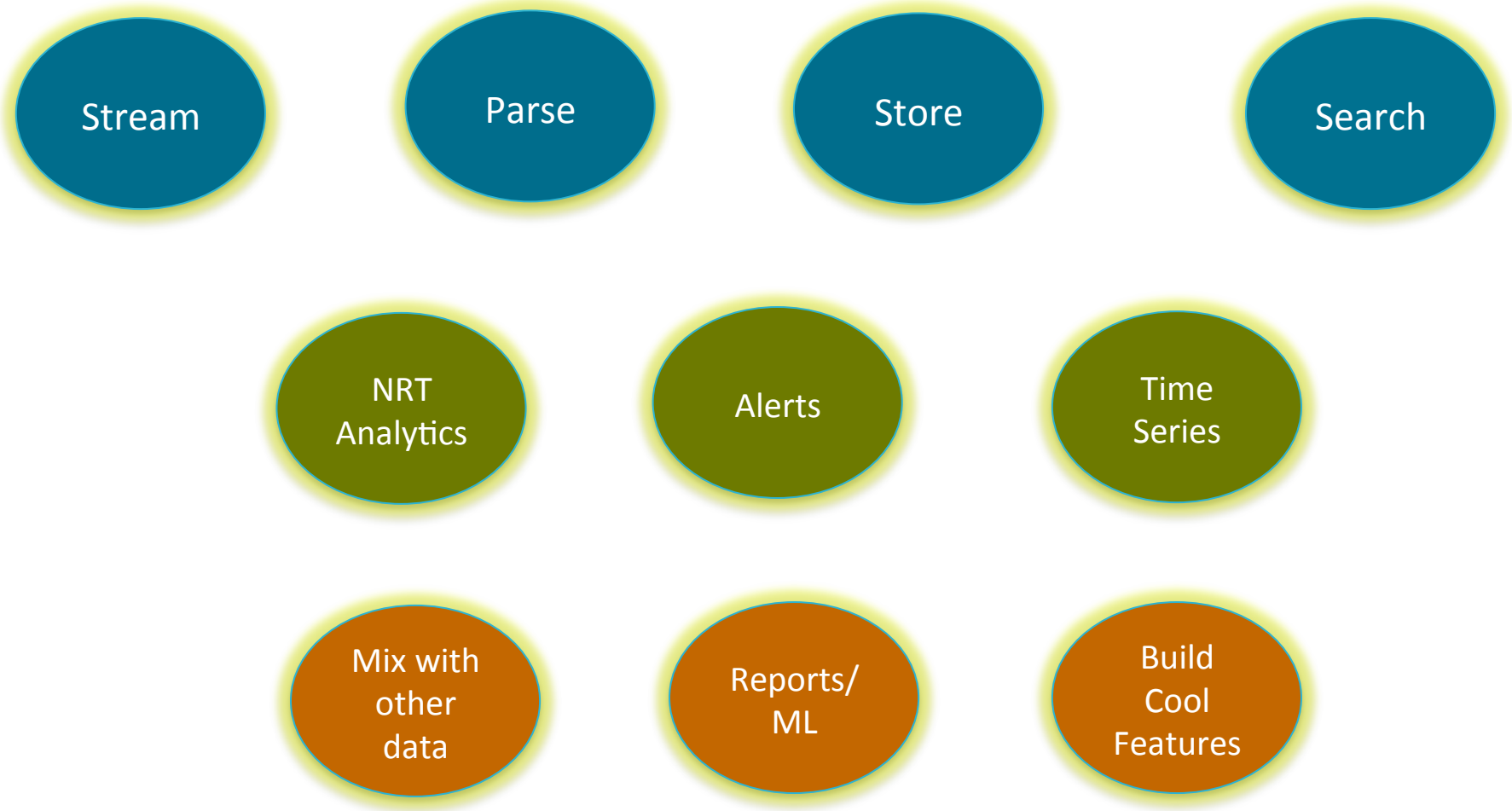
Machine/Streaming Data

- Logs
- Diagnostic Bundles
- Utility Data
- Machine Monitoring Data
- User Activity

- Machine data is a critical piece with highest volume and is fast moving
- Systems are hardest to build and scale for it
- The rest of the data fall naturally into the design.

- In a hospital various reading are taken – heart beat, blood pressure, breathing rate
- Water companies measure the acidity of the water in their reservoirs
- Racing cars : companies want to know every aspect of how their car is performing
- Utility meters
- Web server
- linux/firewall/router : syslogs

What you want to do with Machine Data?



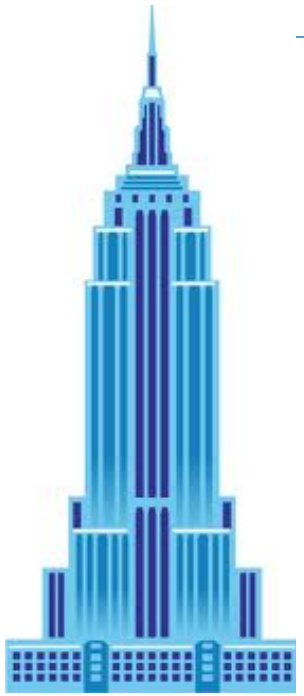
So what's the CHALLENGE?

Huge

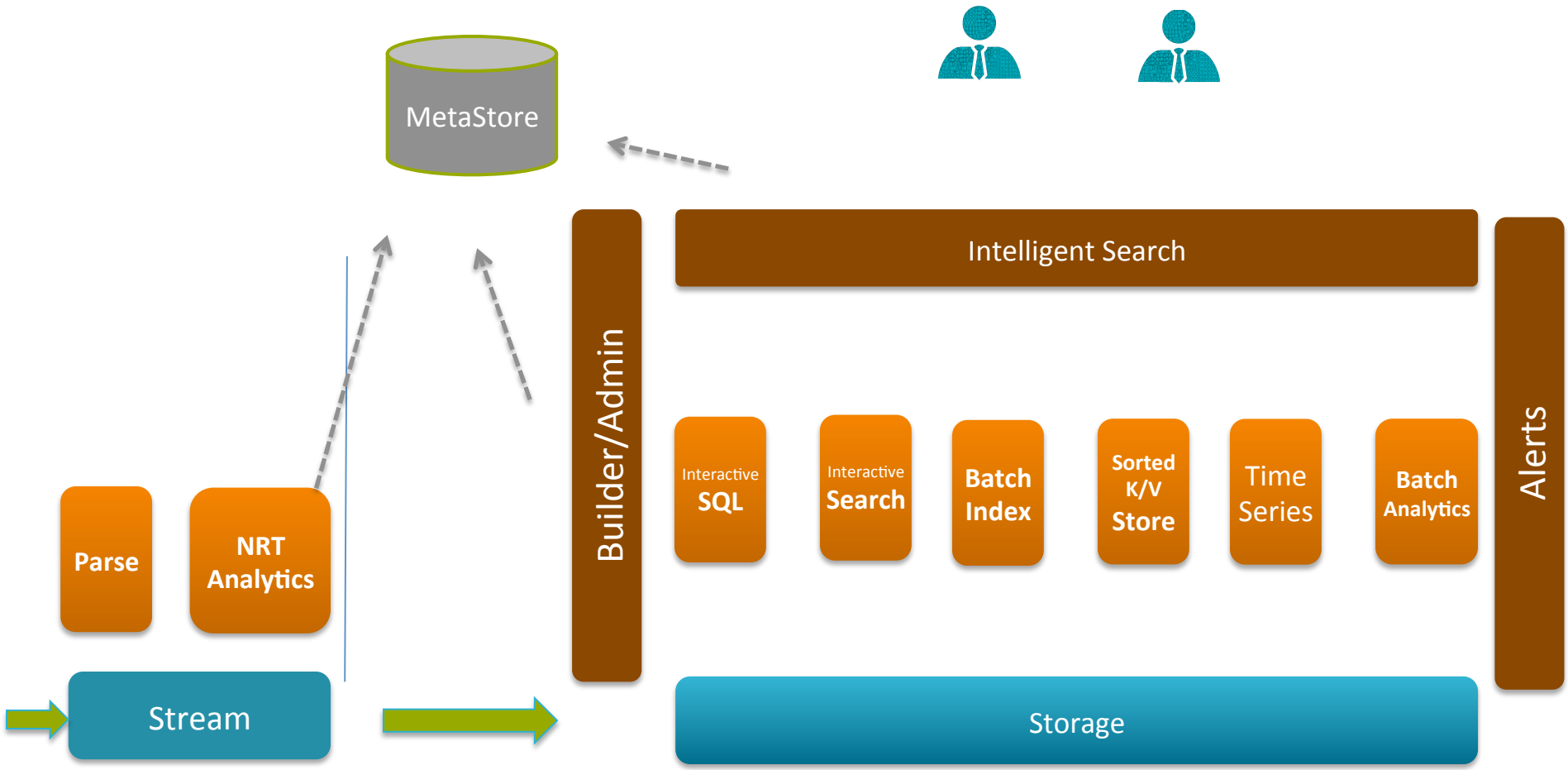
Fast Moving

1TBx1000=1PB

The whole story changes with Scale.



Overall Architecture



Some Log data…

Redhat Linux Server

Mar 7 04:02:08 avas syslogd 1.4.1: restart.
Mar 7 04:02:16 avas clamd[11165]: /var/amavis/amavis-20040307T033734-10329/parts/part-00003: Worm.Mydoom.F FOUND
Mar 7 04:05:55 avas clamd[11240]: /var/amavis/amavis-20040307T035901-10615/parts/part-00002: Worm.SomeFool.Gen-1 FOUND
Mar 7 04:11:15 avas dccifd[11335]: write(MTA socket,4): Broken pipe

Apache

64.242.88.10 - - [07/Mar/2004:16:05:49 -0800] "GET /twiki/bin/edit/Main/Double_bounce_sender?topicparent=Main.ConfigurationVariables HTTP/1.1" 401 12846
64.242.88.10 - - [07/Mar/2004:16:06:51 -0800] "GET /twiki/bin/rdiff/TWiki/NewUserTemplate?rev1=1.3&rev2=1.2 HTTP/1.1" 200 4523
64.242.88.10 - - [07/Mar/2004:16:10:02 -0800] "GET /mailman/listinfo/hsdivision HTTP/1.1" 200 6291

Log4j

0 [main] DEBUG com.vaannila.report.SampleReport - Sample debug message
0 [main] INFO com.vaannila.report.SampleReport - Sample info message
0 [main] WARN com.vaannila.report.SampleReport - Sample warn message

ERROR [2009-09-13 09:56:01,760] [main] (RDFDefaultErrorHandler.java:44) http://www.xfront.com/owl/ontologies/camera/#(line 1 column 1): Content is not allowed in prolog.
[DEBUG] 55:17 (LogExample.java:main:11) Here is some DEBUG

[INFO] 55:17 (LogExample.java:main:12) Here is some INFO

Cisco PIX Logs

Mar 29 2004 09:54:18: %PIX-6-302005: Built UDP connection for faddr 198.207.223.240/53337 gaddr 10.0.0.187/53 laddr 192.168.0.2/53
Mar 29 2004 09:54:19: %PIX-6-302005: Built UDP connection for faddr 198.207.223.240/3842 gaddr 10.0.0.187/53 laddr 192.168.0.2/53
Mar 29 2004 09:54:19: %PIX-6-302005: Built UDP connection for faddr 198.207.223.240/36205 gaddr 10.0.0.187/53 laddr 192.168.0.2/53

cloudera®
Ask Bigger Questions

Search

Storing/Indexing data into the right Stores

Impala/HDFS/HIVE	All data goes in here and accessible with SQL queries Supports very high write throughputs and very fast scans All Data
SolrCloud	Data needed for real-time complex search Last X Days
HBase	Data needed for real-time serving and searching/scanning based on key Supports very high write/read throughputs. Supports filtering. Row Key is indexed and sorted. Limit scanning of huge sets. Result Sets, Configuration
OpenTSDB	Time Series Data for Monitoring/Alerting Metrics

Searching Solr

Searches are done via HTTP GET on the select URL with the query string in the q parameter.

- q=video&fl=name,id (return only name and id fields)
- q=video&fl=name,id,score (return relevancy score as well)
- q=video&fl=*,score (return all stored fields, as well as relevancy score)
- q=video&sort=price desc&fl=name,id,price (add sort specification: sort by price descending)
- q=video&wt=json (return response in JSON format)

Use the "sort' parameter to specify "field direction" pairs, separated by commas if there's more than one sort field:

- q=video&sort=price desc
- q=video&sort=price asc
- q=video&sort=inStock asc, price desc
- "score" can also be used as a field name when specifying a sort

Searching Solr – Faceted Search

Faceted search allows users who're running searches to see a high-level breakdown of their search results based upon one or more aspects (facets) of their documents, allowing them to select filters to drill into those search results.

http://localhost:8983/solr/select?q=*:*&facet=true&facet.field=tags

▶Restaurant Type	▶State	▶Price Range	▶City
Fast Food (10073)	New York (4020)	< $5 (4000)	New York, NY (2021)
Sit-down Chain (2530)	California (3459)	$5 - $10 (7000)	San Francisco, CA (1499)
Local Diner (998)	Illinois (2450)	$10 - $20 (1007)	Chicago, IL (850)
Up-scale (400)	Georgia (1620)	$20 - $50 (1300)	Atlanta, GA (620)
	Texas (1501)	$50+ (550)	Austin, TX (501)
	...	...	...

Impala

- Raises the bar for query performance. Does extensive query optimization.
- Uses the same metadata, SQL syntax (Hive SQL), ODBC driver and user interface (Hue Beeswax) as Apache Hive
- Impala circumvents MapReduce to directly access the data through a specialized distributed query engine
- Queries that require multiple MapReduce phases in Hive or require reduce-side joins will see a higher speedup than, say, simple single-table aggregation queries

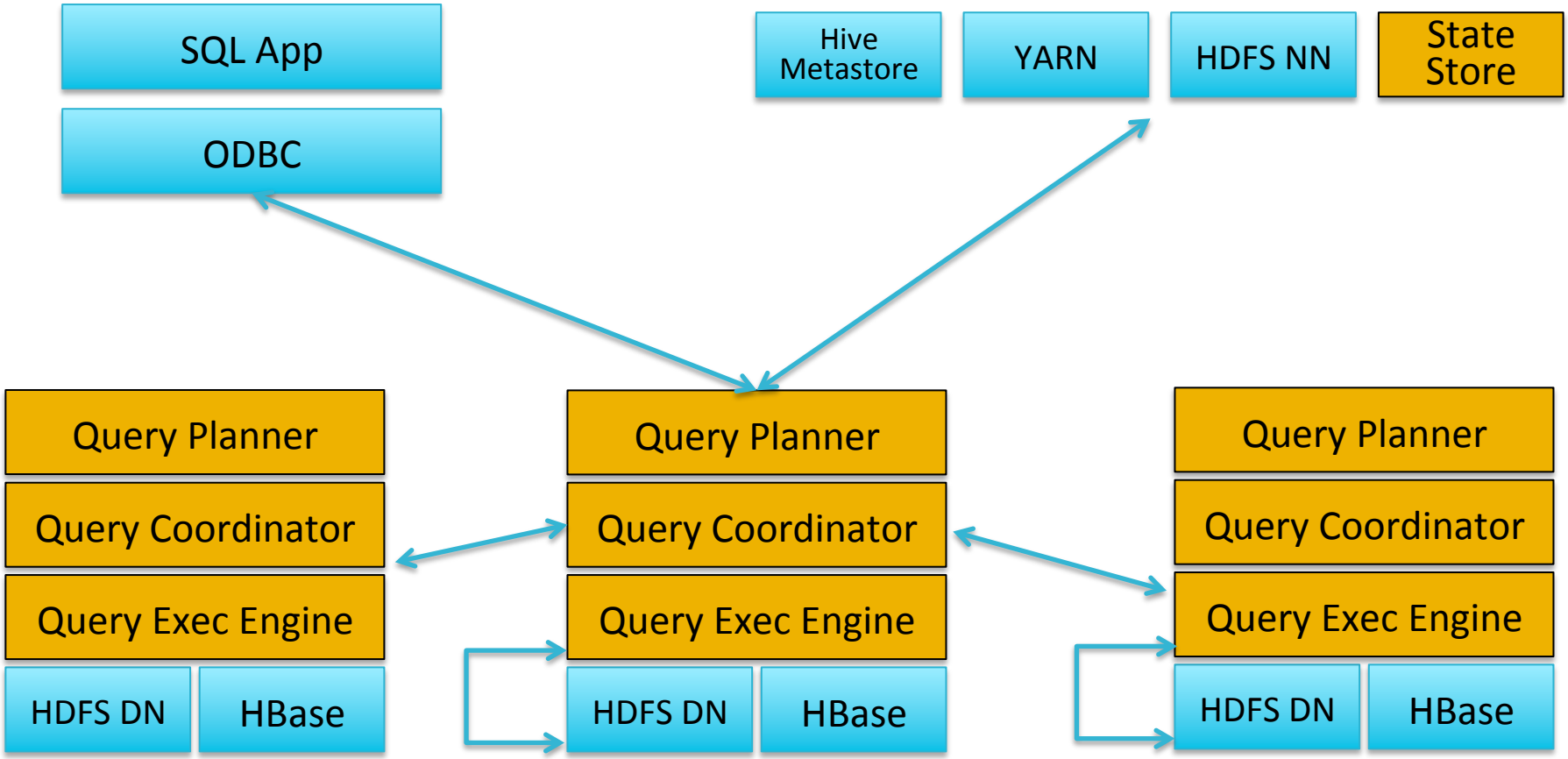
Searching with Impala

- Perfect for large data scans
- Partitioning will reduce the amount of data scanned
- Impala caches the Metadata
- Define SQL statements for searching from Impala/HIVE. Use Regex for defining new fields during search time.

Partitions

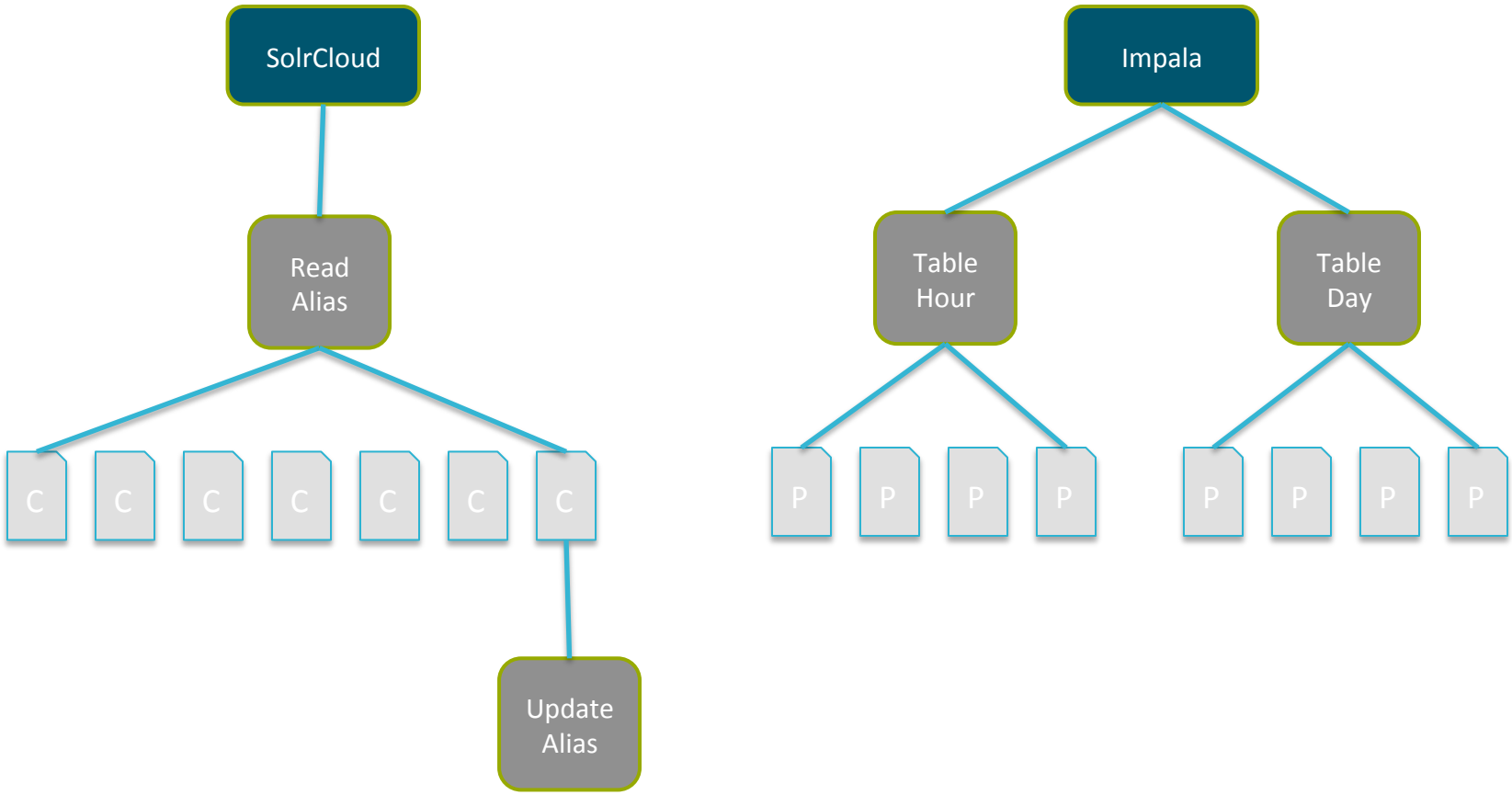
- Partition by day: 365
- Partition by hour: 8760
- Partition by minute: 525600

Impala



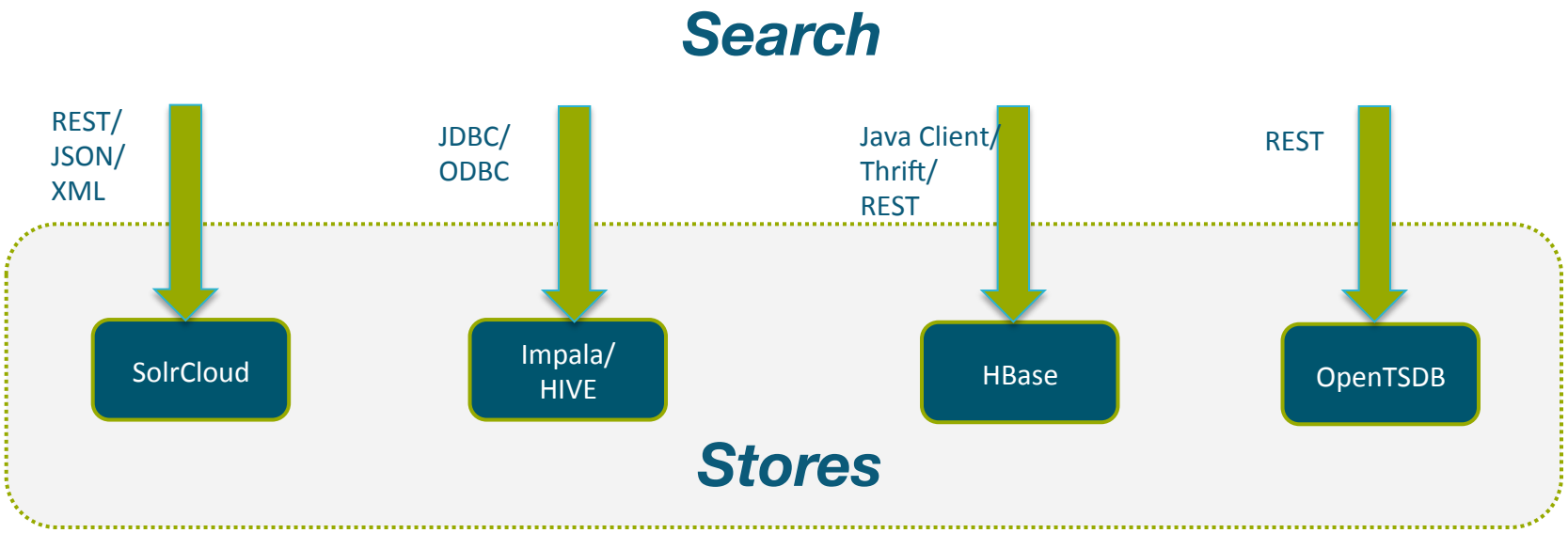
Store/Search

More Optimizations = Faster Performance

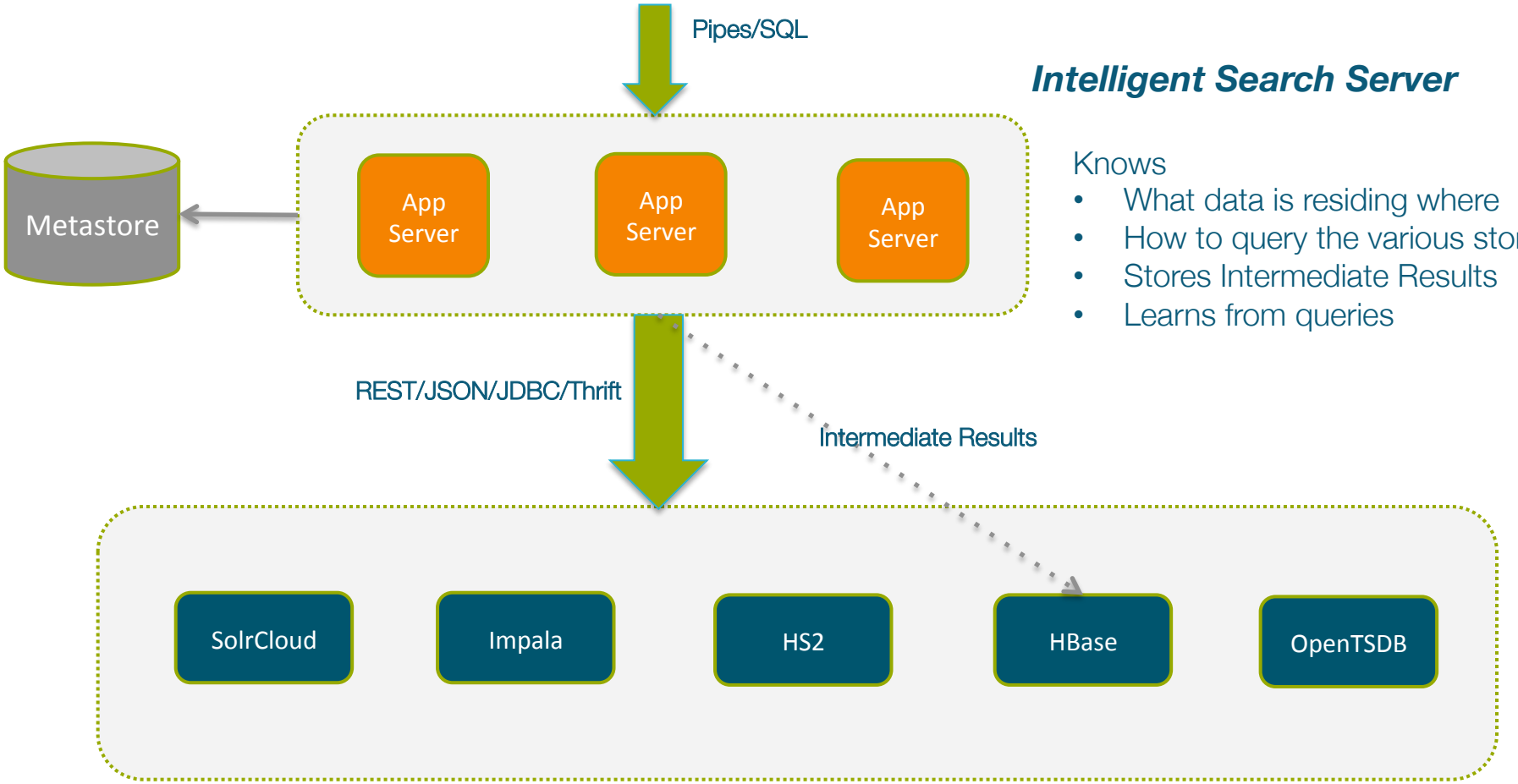


http://localhost:8983/solr/admin/collections?action=CREATEALIAS&name=readalias&collection=C2,C3

Store/Search



Unified Data Access



Intelligent Search Server

Knows

- What data is residing where
- How to query the various stores
- Stores Intermediate Results
- Learns from queries

Builder/Admin

Performs all the Background Management & Admin tasks

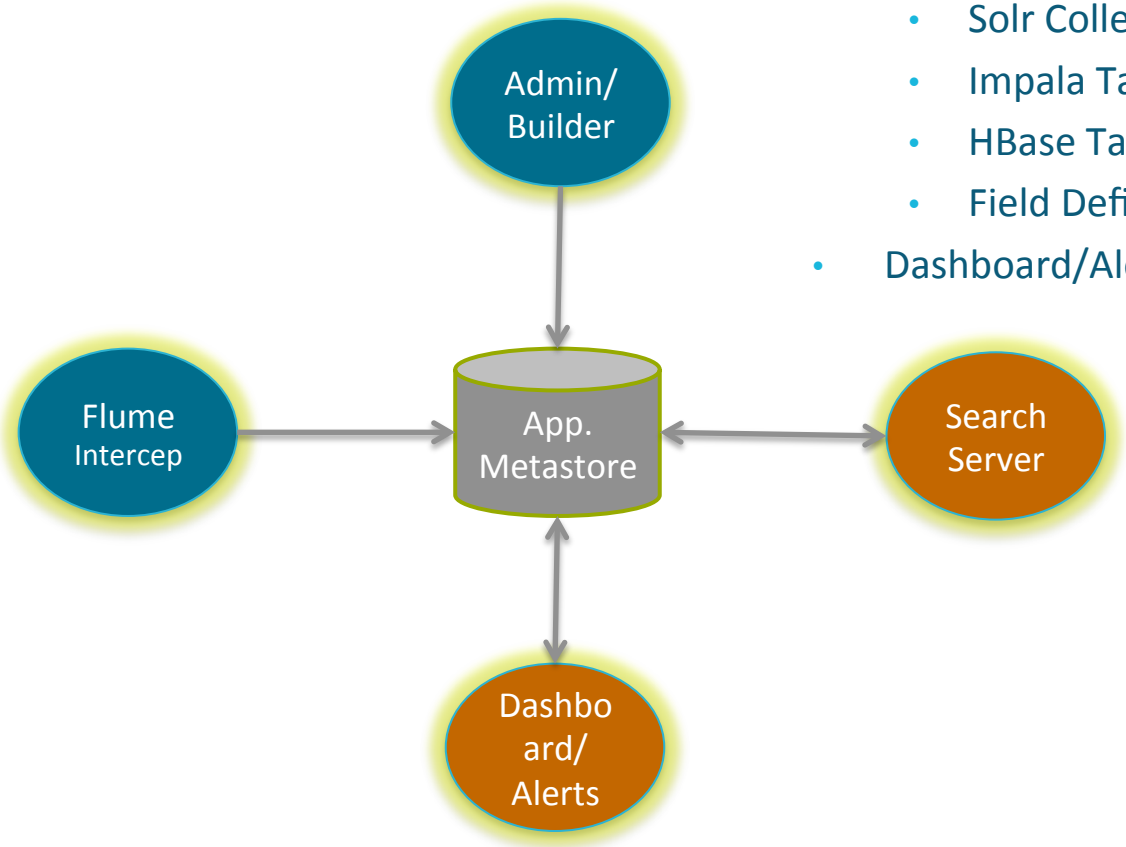
- Create new DataSets
- Manage Schemas
- Manage Collections : Store last X days of data in Solr. Use Aliases to map to collections/day.
- Regenerate Solr Index when needed or requested by the Admin
- Manage the Impala Partitions. Last X days vs last Y months vs last Z years

Intelligent Search Server

- Parse Query Requests
- Get DataSet definition from the metadata store
- Generate the Query Plan
  - Should I fetch from SolrCloud/Impala?
  - Is there intermediate result stored that I can use?
  - If not a power-user, would this query be very long running?
- Execute the Query
- Store results in HBase if applicable
- Support expressions, aggregate functions, expressions, normal functions

Metastore

Impala/HIVE
Tables & Partitions



- User Definitions
  - Normal/Power user/Admin
- DataSets Definitions
  - Solr Collections
  - Impala Tables
  - HBase Tables
  - Field Definitions
- Dashboard/Alerts Definitions

Searching/Querying

- SLA requirements
  - Be able to search last X days of logs within seconds
  - Be able to search last Y weeks of logs within minutes
  - Be able to search last Z months of logs within 15 minutes
- Searching consists of:
  - Specifying a dataset and time period
  - Searching for a regular expression in the dataset
  - Displaying the matching records
  - Displaying the count of keywords in various facets (hostname, city, IP)
  - Further filtering by various facet selections
  - Allow users to define new fields

Sample Queries

How many times did someone view a page on the website?

```
dataset=logs1 method=GET | stats count AS Views
```

How many resulted in purchases?

```
dataset=logs1 method=GET | stats count AS Views, count(eval(action="purchase")) as Purchases
```

What was purchased and how much was made?

```
dataset=logs1 * action=purchase | stats count AS "# Purchased", values(price) AS Price, sum(price) AS Total by product_name
```

Which items were purchased most?

```
dataset=logs1 action=purchase | top category_id
```

::You can also save your searches under some label::

Drill Downs, More Queries & Subsearch

Click on ‘Tablets’ from the Top Purchases

This kicks off a new search. Search is updated to include the filter for the field/value pair category=flowers

How many different customers purchased tablets?

dataset=logs1 action=purchase category_id=tablets| stats uniqueuecount(clientip)

How many tablets did each customer buy?

dataset=logs1 action=purchase category_id=tablets| stats count BY clientip

The customer who bought the most items yesterday and what he or she bought?

dataset=logs1 action=purchase [search dataset=logs1 action=purchase | top limit=1 clientip | table clientip] | stats count, values(product_id) by clientip

Querying with SQL & Pipes

Query	SQL
Top 25: business with most of the reviews	SELECT name, review_count FROM business ORDER BY review_count DESC LIMIT 25 | chart ...
Top 25: coolest restaurants	SELECT r.business_id, name, SUM(cool) AS coolness FROM review r JOIN business b ON (r.business_id = b.business_id) WHERE categories LIKE '%Restaurants%' GROUP BY r.business_id, name ORDER BY coolness DESC LIMIT 25

Index Fields

- Index Time
  - Solr : Will slow down with more indexes
  - Impala : Relies on Partitioning, Bucketing and Filtering
  - Define additional indexed fields through the Builder
- Search Time field Extraction
  - Does not affect the index size
  - Size of data processed gets larger
  - Storing of results helps

Adding New Fields & Updating the Index

Adding new field

- Update the Morphines to parse the new field
- Update the Solr Schema.
- Update the Impala/HIVE table definitions

Indexing Options

- Re-index all data on HDFS
  - Also used when say an index is lost
  - Can also be run on the data in HBase
- Support new fields for new data only

Speeding Search…

- Save the Search Results (HBase)
- Search Results can be shared
- Searches are speeded by saving previous result and then running an incremental search.

Dashboards & Charts

- Create New Dashboards and populate them
- Add a search you have just run to a new or existing dashboard

Chart of purchases and views for each product

```
dataset=ecommerce method=GET | chart count AS views, count(eval(action="purchase")) AS purchases
by category_id”
```

- Top Items Sold
- Total Number of Exceptions
- Total Number of Visits
- Map of Visitor Locations
- Pages/Visit

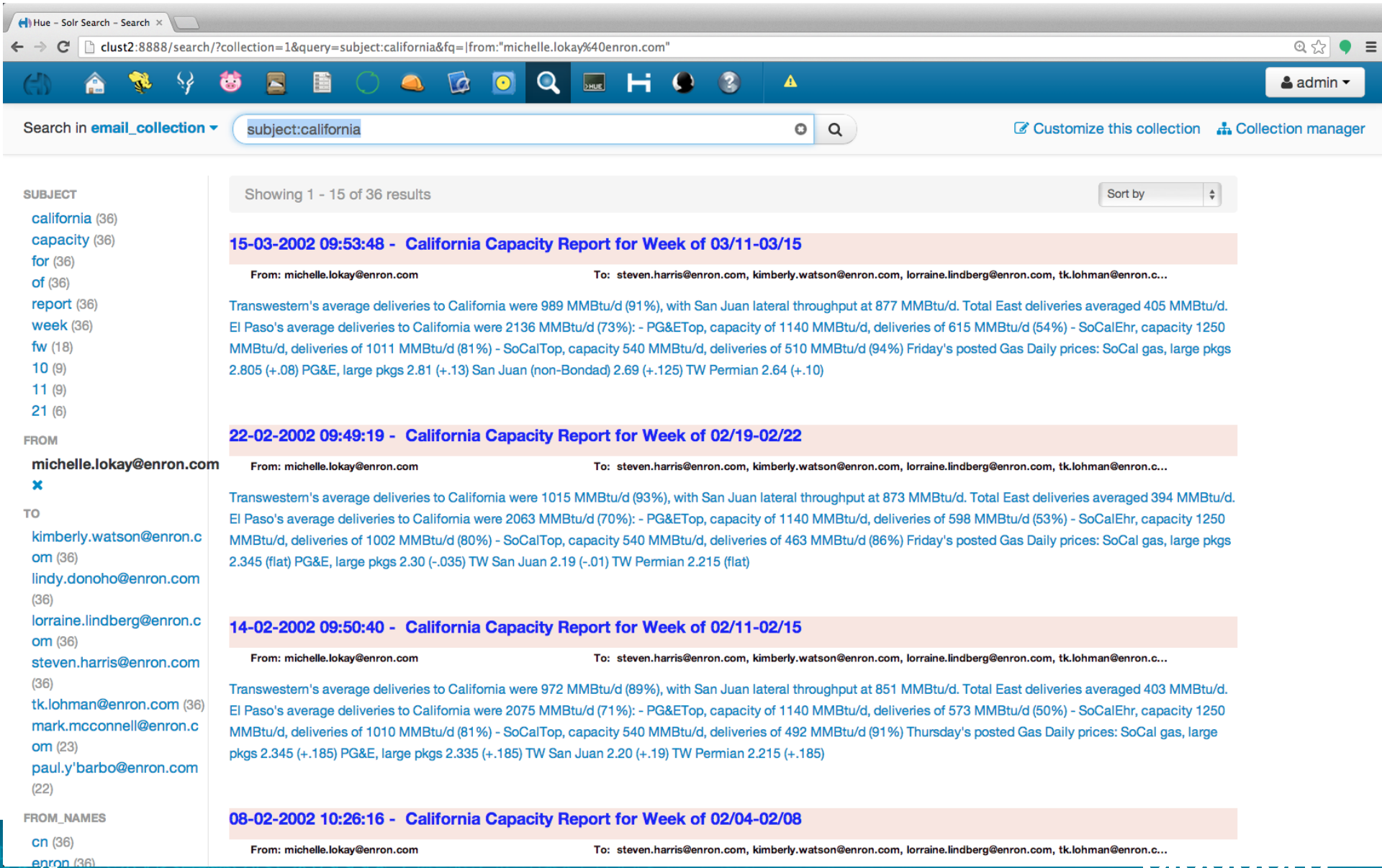
Define the Schema for Incoming Data

- Log data comes in different formats : apache logs, syslog, log4j etc.
- Define the fields to be extracted from each : Timestamp, IP, Host, Message…

- Define Solr Schema
  - Can create separate collections for different datasets and time ranges

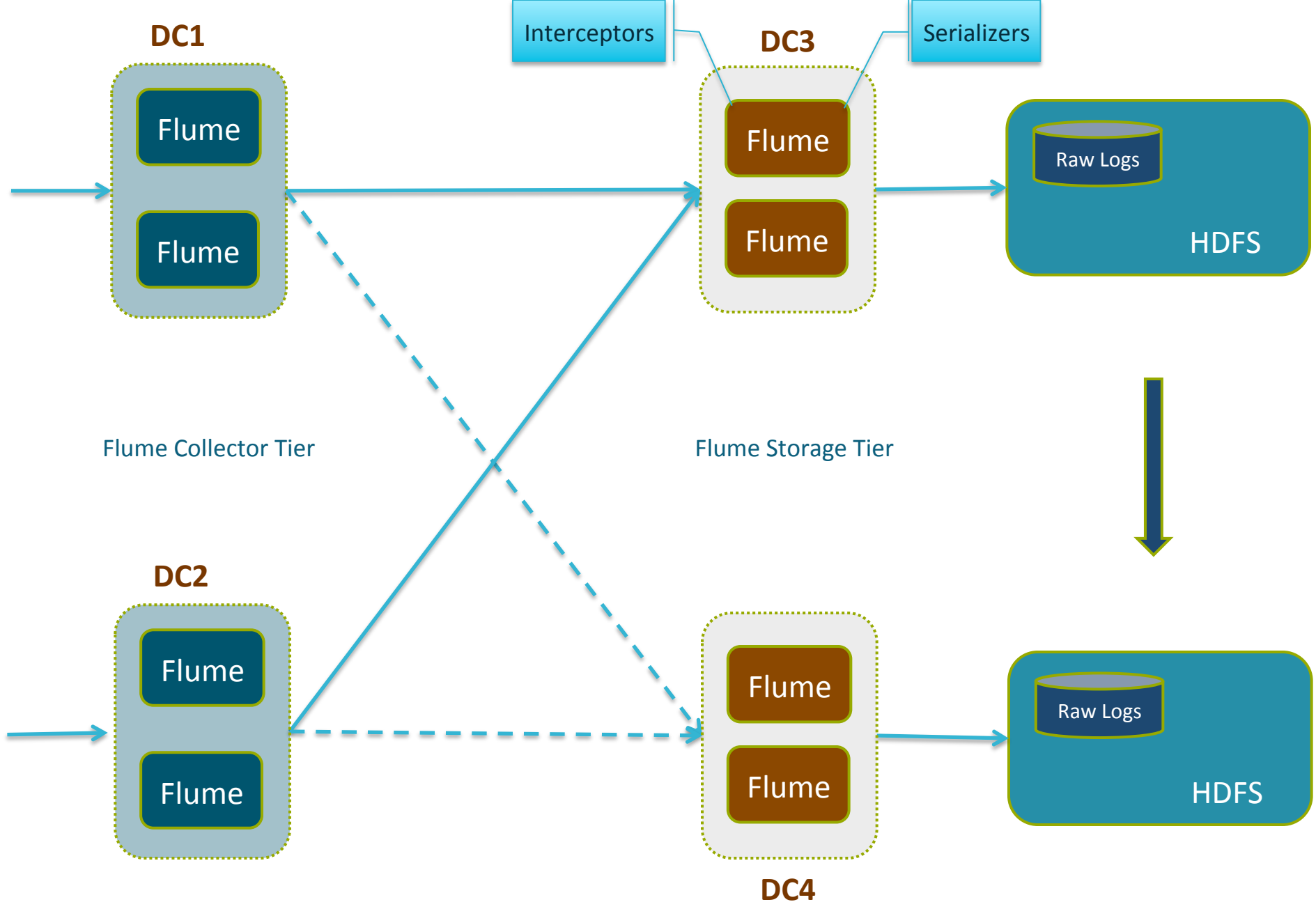
- Define Tables for Impala & HIVE
  - Partition things by date. If needed partition some stuff by hour.
  - Impala performs great on partitioned data for NRT queries
- Define Schema for HBase Tables (Need to optimize for writes and for reads)
  - Composite Key : DataSet, Application, Component, Some Prefix, Timestamp
  - Application, User ID, Timestamp

Unified Data Access with Hue

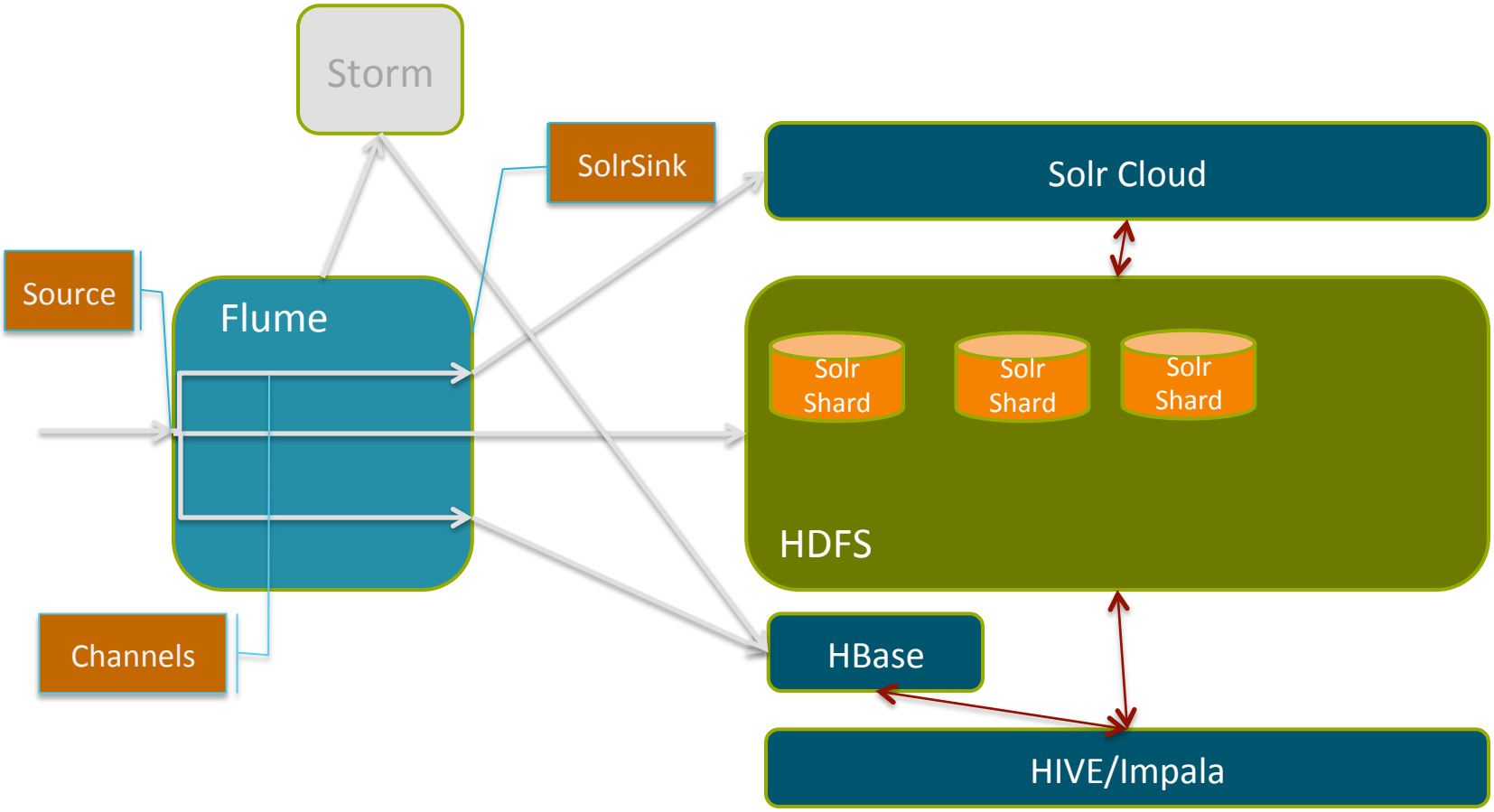


cloudera
Ask Bigger Questions

Streaming, Parsing, Indexing, NRT Analytics, Alerts



Streaming in data into HDFS/HBase/SolrCloud



Parsing with Morphlines

Can be embedded into any Application…

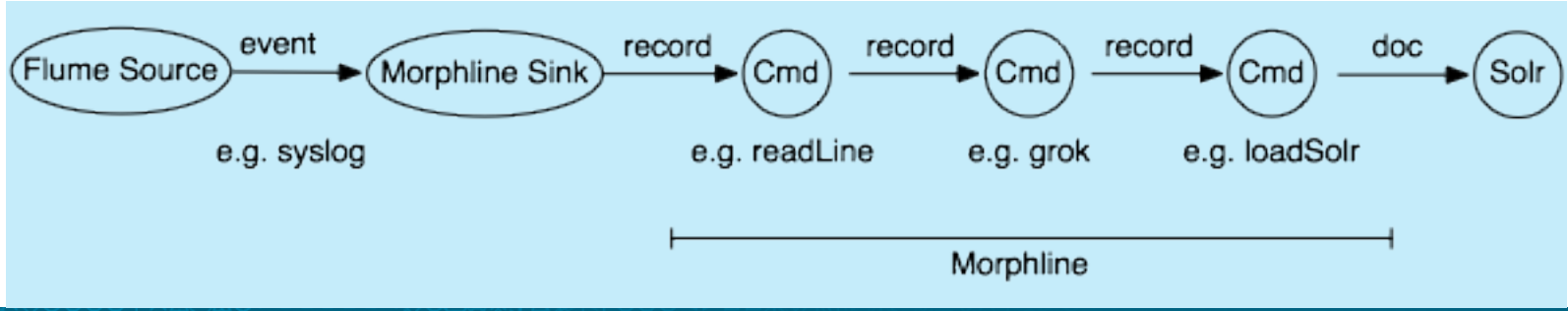
From various processes

- Flume
- MapReduceIndexerTool
- My Application

Morphline

ETL into various Stores

- SolrCloud
- HBase
- HDFS
- …



Parsing

```
morphlines : [
    morphline1
    commands : [
        readMultiLine
        Break up the message text into SOLR fields
        Geneate Unique ID
        Convert the timestamp field to "yyyy-MM-dd'T'HH:mm:ss.SSSZ"
        Sanitize record fields that are unknown to Solr schema.xml
        load the record into a SolrServer
    ]
]
```

http://cloudera.github.io/cdk/docs/current/cdk-morphlines/morphlinesReferenceGuide.html

https://github.com/cloudera/cdk/tree/master/cdk-morphlines/cdk-morphlines-core/src/test/resources/test-morphlines

```
SOLR_COLLECTION : "collection1"
SOLR_COLLECTION : ${?ENV_SOLR_COLLECTION}

ZK_HOST : "127.0.0.1:2181/solr"
ZK_HOST : ${?ENV_ZK_HOST}

SOLR_HOME_DIR : "example/solr/collection1"
SOLR_HOME_DIR : ${?ENV_SOLR_HOME_DIR}

SOLR_LOCATOR : {
  collection : ${SOLR_COLLECTION}
  zkHost : ${ZK_HOST}
  solrHomeDir : ${SOLR_HOME_DIR}
  # batchSize : 1000
}
SOLR_LOCATOR : ${?ENV_SOLR_LOCATOR}

morphlines : [
  {
    id : morphline1
    importCommands : ["com.cloudera.**", "org.apache.solr.**"]

    commands : [
      {
        sanitizeUnknownSolrFields {
          solrLocator : ${SOLR_LOCATOR}
        }
      }

      {
        loadSolr {
          solrLocator : ${SOLR_LOCATOR}
          boosts : {
            id : 1.0
          }
        }
      }

      { logDebug { format : "output record: {}", args : ["@{}"] } }
    ]
  }
]
```

Using Java Code

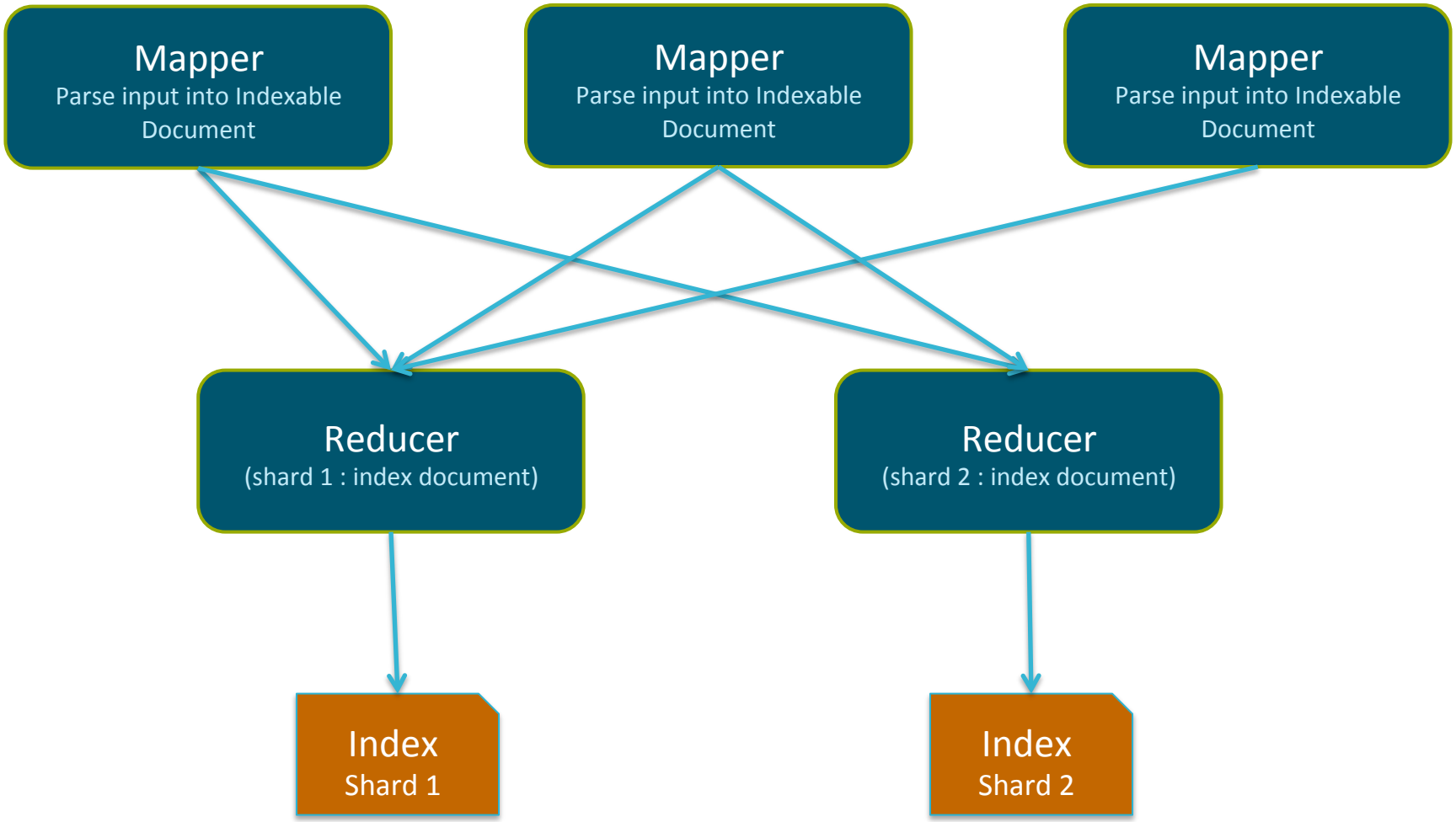
```
morphlines : [
    {
        id : morphline1
        importCommands : ["com.cloudera.**", "org.apache.solr.**"]
        commands : [
            { java
                {
                    code: """
                        List tags = record.get("tags");
                        if (!tags.contains("hello"))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

Real Time Indexing into Solr

agent.sinks.solrSink.type=org.apache.flume.sink.solr.morphline.MorphlineSolrSink
agent.sinks.solrSink.channel=solrChannel
agent.sinks.solrSink.morphlineFile=/tmp/morphline.conf

Morphline file, which encodes the transformation logic, is exactly identical in both Real Time and Batch Indexing examples

Batch Indexing with MapReduceIndexTool



Scalable way to create the indexes on HDFS

Batch Indexing with MapReduceIndexTool

MapReduceIndexerTool is a MapReduce batch job driver that creates a set of Solr index shards from a set of input files and writes the indexes into HDFS

MR Indexer

- Read the HDFS directory
- Pass them through the Morphline
- Merge the indexes into live SOLR servers

 # hadoop jar /usr/lib/solr/contrib/mr/search-mr--job.jar*
org.apache.solr.hadoop.MapReduceIndexerTool --morphline-file <morphline file>
--output-dir <hdfs URI for indexes> --go-live --zk-host clust2:2181/solr –collection logs_collection
<HDFS URI with the files to index>

Tagging Data at Source

```
agent.sources.messages-source.interceptors = hostname timestamp zinfo
agent.sources.messages-source.interceptors.hostname.type = host
agent.sources.messages-source.interceptors.timestamp.type = timestamp
agent.sources.messages-source.interceptors.zinfo.type = com.xyz.flume.interceptor.MyInterceptor$Builder
agent.sources.messages-source.interceptors.zinfo.headers.store = ecommerce
agent.sources.messages-source.interceptors.zinfo.headers.source = /var/log/messages
agent.sources.messages-source.interceptors.zinfo.headers.sourcetype = syslog_messages
```

```
public class MyInterceptor implements Interceptor {
   Map<String,String> staticHeaders;

   private MyInterceptor(Map<String,String> staticHeaders) {
      this.staticHeaders = staticHeaders;
   }
   public Event intercept(Event event) {
      Map<String,String> headers = event.getHeaders();
      headers.putAll(staticHeaders);
      return event;
   }
```

cloudera
Ask Bigger Questions

Monitoring/Alerts & Time Series

NRT Use Cases

- Analytics/Aggregations
  - Total number of page-views of a URL in a given time-period
  - Reach : Number of unique people exposed to a URL
  - Generate analytic metrics like Sum,Distinct,Count,Top K etc.
- Alert when the number of HTTP Error 500 in the last 60 sec > 2
- Get real-time state information about infrastructure and services.
- Understand outages or how complex systems interact together.
- Real time intrusion detection
- Measure SLAs (availability, latency, etc.)
- Tune applications and databases for maximum performance
- Do capacity planning

Monitoring /Alerting Use Cases

- Counting: real-time counting analytics such as how many requests per day, how many sign-ups, how many purchases, etc.
- Correlation: near-real-time analytics such as desktop vs. mobile users, which devices fail at the same time, etc.
- Research: more in-depth analytics that run in batch mode on the historical data such as detecting sentiments, etc.

NRT Alerts & Aggregations Implementation

- Rule based alerts in Flume
- Aggregations in Flume/HBase
- Time-series data in HBase/OpenTSDB

HBase

- Counters : avoids need to lock a row, read the value, increment it, write it back, and eventually unlock the row

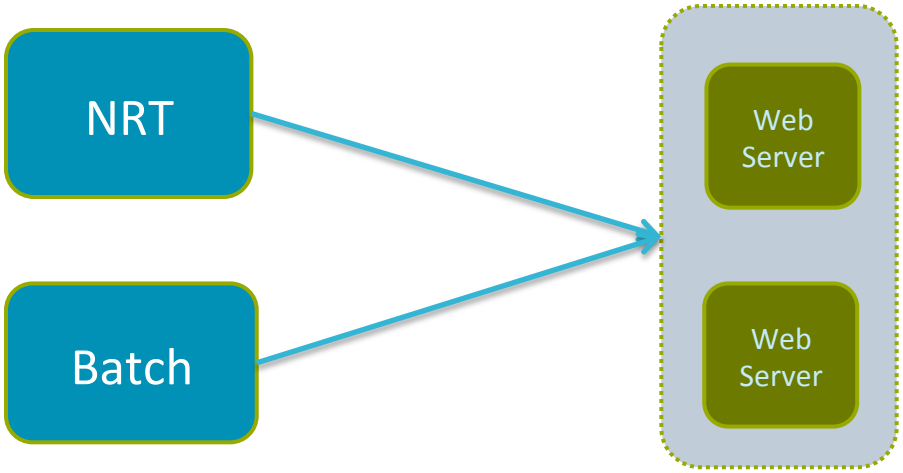
```
hbase(main):001:0> create 'counters', 'daily', 'weekly', 'monthly'
0 row(s) in 1.1930 seconds

hbase(main):002:0> incr 'counters', '20110101', 'daily:hits', 1
COUNTER VALUE = 1

hbase(main):003:0> incr 'counters', '20110101', 'daily:hits', 1
COUNTER VALUE = 2

hbase(main):04:0> get_counter 'counters', '20110101', 'daily:hits'
COUNTER VALUE = 2
```

```
Increment increment1 = new Increment(Bytes.toBytes("20110101"));
increment1.addColumn(Bytes.toBytes("daily"), Bytes.toBytes("clicks"), 1);
increment1.addColumn(Bytes.toBytes("daily"), Bytes.toBytes("hits"), 1);
increment1.addColumn(Bytes.toBytes("weekly"), Bytes.toBytes("clicks"), 10);
increment1.addColumn(Bytes.toBytes("weekly"), Bytes.toBytes("hits"), 10);
Result result1 = table.increment(increment1);
```

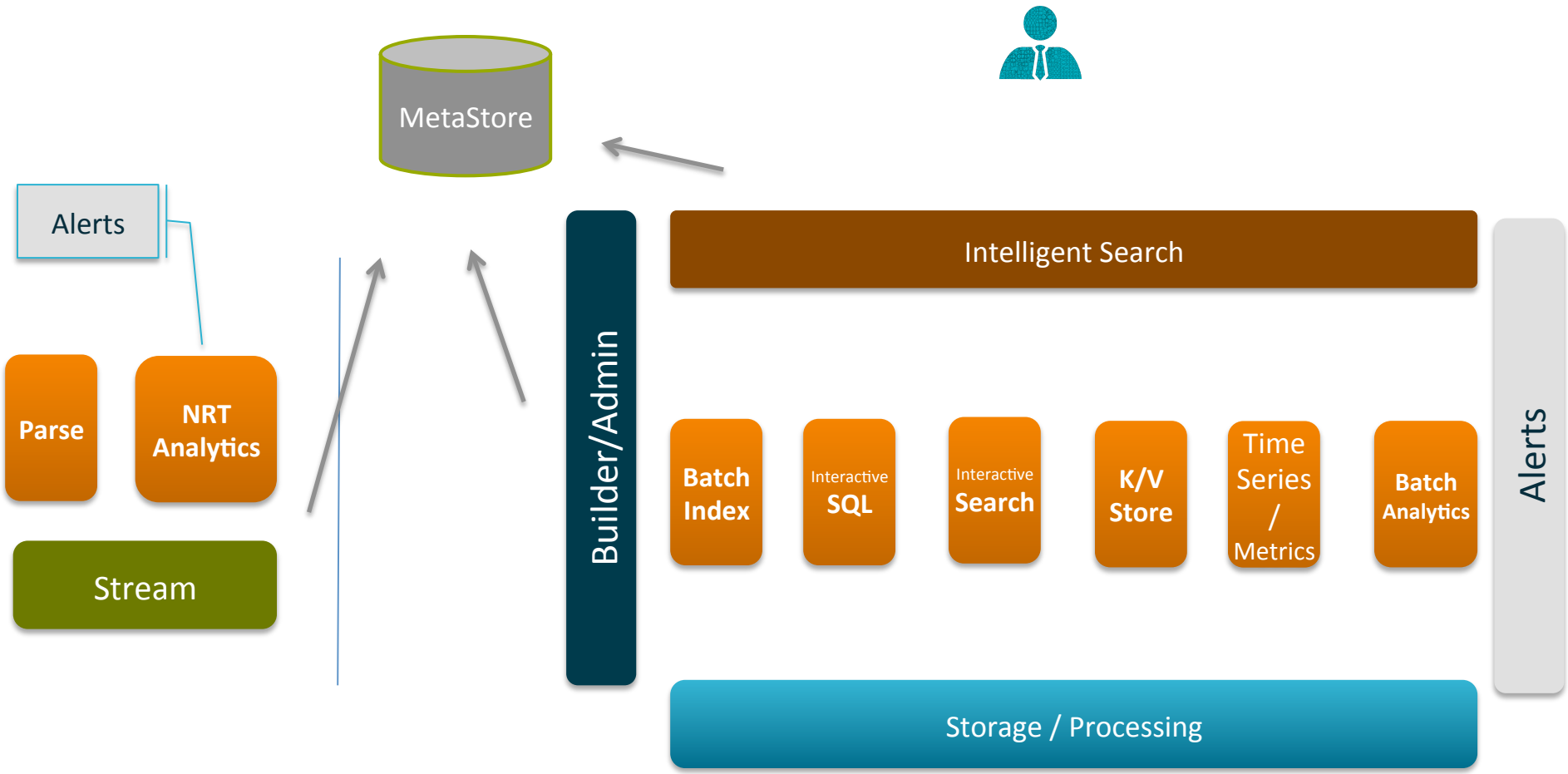
- Allows alerting logic to be at one place in Flume Interceptors
- You write your code in simple interfaces in Flume
- Backed up by HBase
- Easy to define rules over here

(NRT + Batch) Analytics

- Batch Workflow
  - Does incremental computes of the data and loads the result into say HBase
  - Is too slow for the needs in many cases. Also the views are out of date
- Compensating for last few hours of data is done in Flume
  - Applications query both real-time view and batch view and merge the results



Alerts



Time Series with OpenTSDB

Time series is a series of data-points of some particular metric over time.

- OpenTSDB is a time series database.
- It is also a data plotting system.
- Runs on HBase
- Each TSD can handle 2000+ new data points per sec per core

Interacting with OpenTSDB

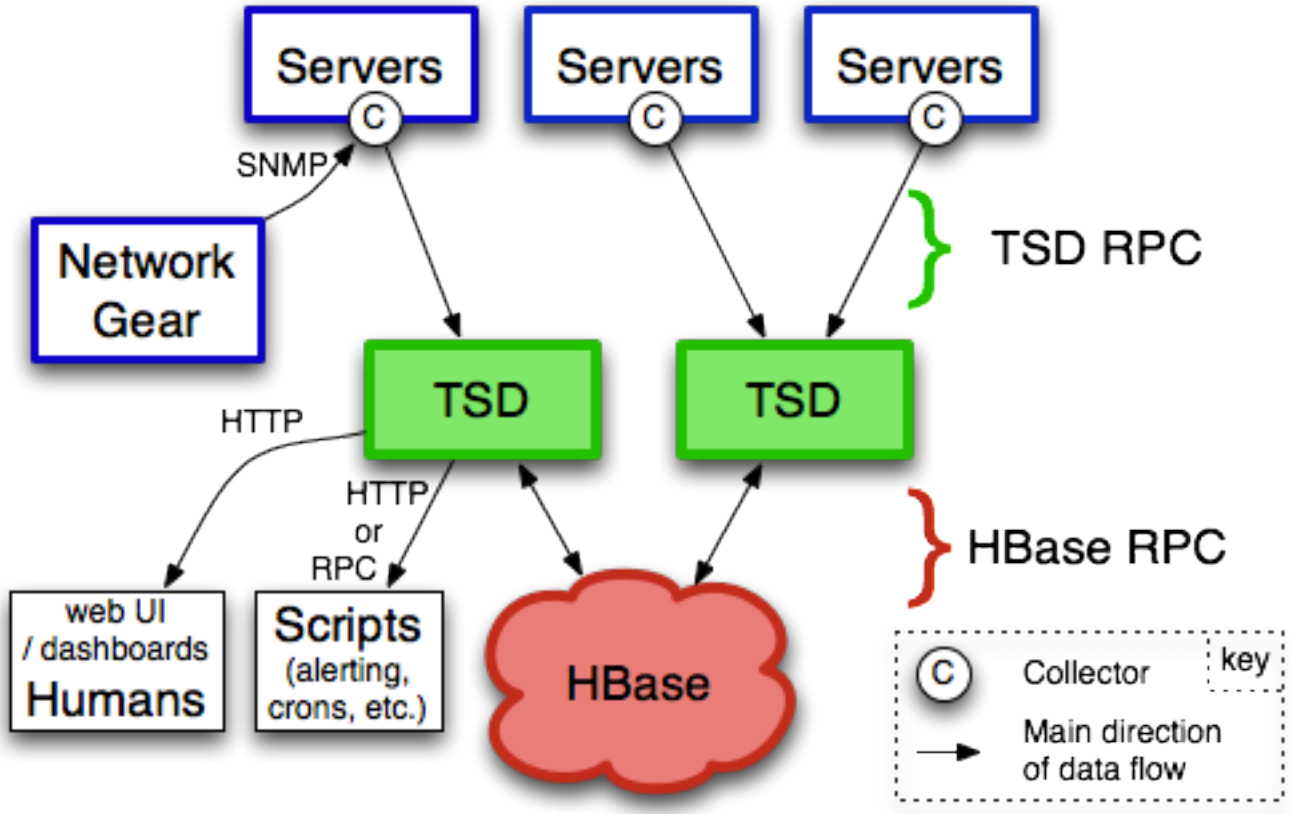
You can communicate with the TSD via a simple telnet-style protocol, and via HTTP

```
put proc.loadavg.1m 1288946927 0.36 host=foo
put proc.loadavg.5m 1288946927 0.62 host=foo
put proc.loadavg.1m 1288946942 0.43 host=foo
put proc.loadavg.5m 1288946942 0.62 host=foo
```

In OpenTSDB, a data point is made of:

- A metric name : (http.hits)
- A UNIX timestamp
- A value (64 bit integer or double-precision floating point value).
- A set of tags (key-value pairs) that annotate this data point : (to store for all the places where a metric exists) : eg. hostname, customer

OpenTSDB Deployment

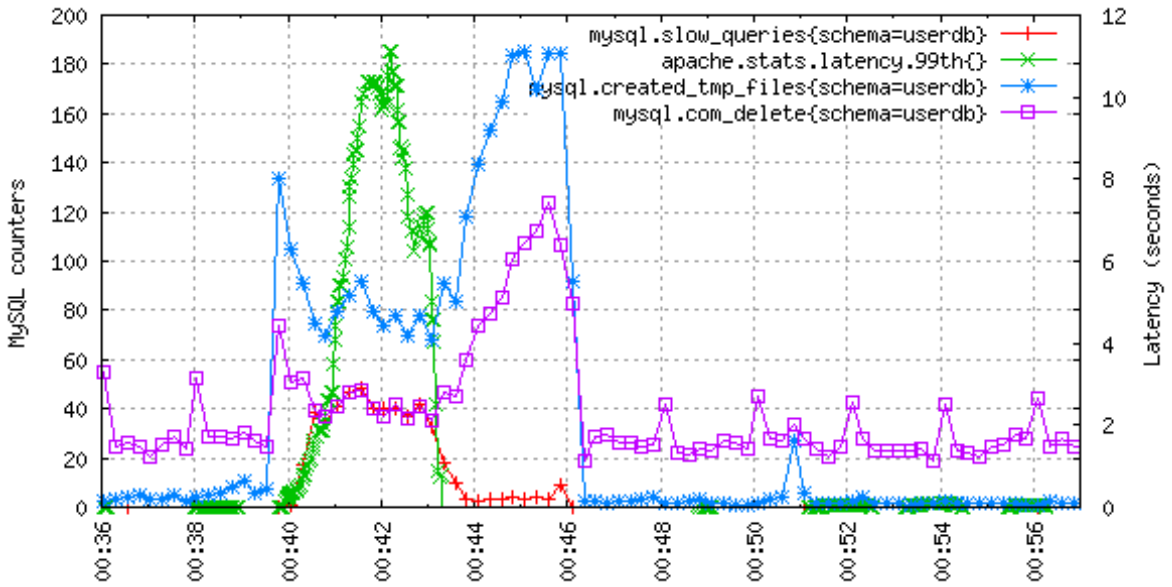


OpenTSDB Query

http://localhost:4242/api/query?start=1h-ago&m=sum:rate:proc.stat.cpu{host=foo,type=idle}
http://localhost:4242/api/query?start=1h-ago&tsuid=sum:000001000002000042,000001000002000043

Parameter	Date Type	Required	Description	Example
Start Time	String or Integer	Yes	Starting time for the query. This may be an absolute or relative time. See *Dates and Times* for details	24h-ago
End Time	String or Integer	No	An end time for the query. If the end time is not supplied, the current time on the TSD will be used. See *Dates and Times* for details.	1h-ago
Metric	String	Yes	The full name of a metric in the system. Must be the complete name. Case sensitive	sys.cpu.user
Aggregation Function	String	Yes	A mathematical function to use in combining multiple time series	sum
Tags	String	No	An optional set of tags for filtering or grouping	host=*,dc=lax
Downsampler	String	No	An optional interval and function to reduce the number of data points returned	1h-avg
Rate	String	No	An optional flag to calculate the rate of change for the result	rate

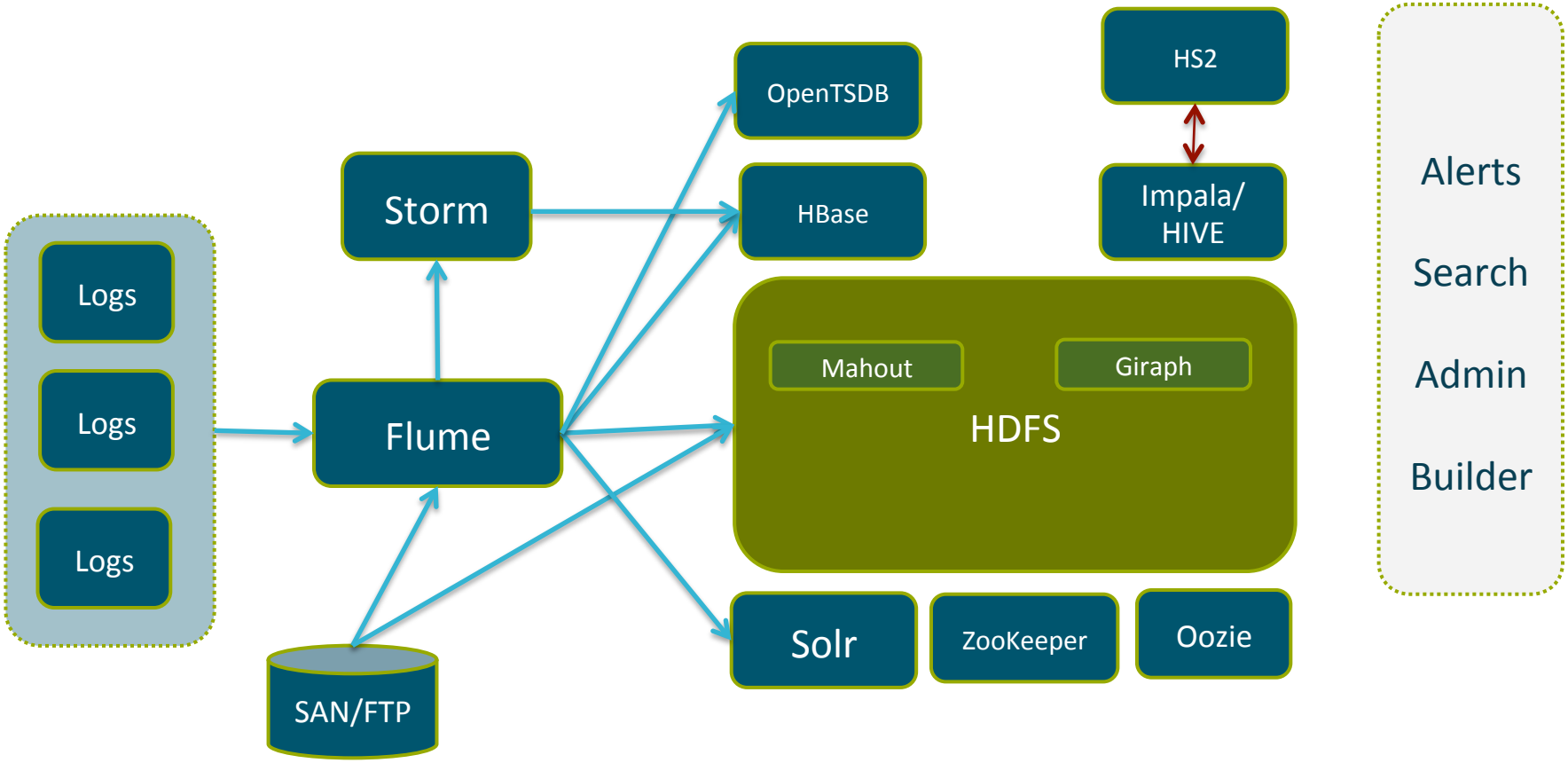
OpenTSDB Graphs & Alerts



check_tsd!-d 60 -m rate:apache.stats.hits -t status=500 -w 1 -c 2

(look back upto 60 seconds, warning threshold is 1, critical threshold is 2)

Overall Architecture… Another View…



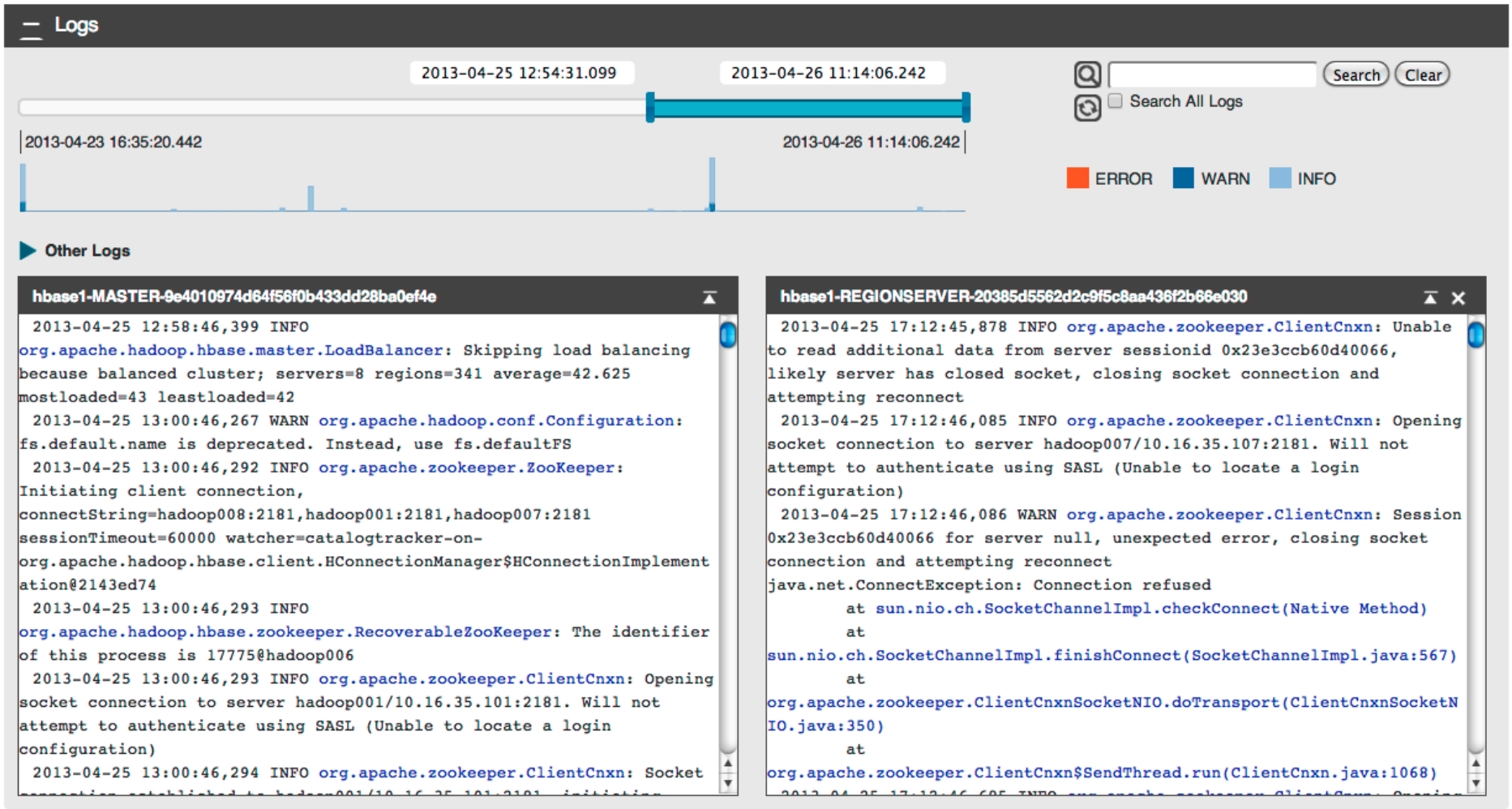
cloudera
Ask Bigger Questions

CSI at Cloudera

CSI - Cloudera Support Interface

- Components Used
  - HBase, Solr, Impala, MR
- Features
  - Enables searching & analytics for data from different sources in a single UI
- Data Collected
  - Customer Diagnostics
    - Hadoop Daemon Logs
    - Hadoop Daemon Configurations
    - Host hardware info
    - Host OS settings and configurations
  - Support Cases, Public Apache Jiras, Public Mailing Lists, and Salesforce Account Data

CSI – Log Visualization within Customer Dashboard



CSI – Ad-hoc Data Analytics

Which collections have impala installed:

```
SELECT customerKey, clusterName, collectionTS, count(*) AS totalMachines
FROM machine_sysstats
WHERE rpmqa LIKE '%impala%'
GROUP BY customerKey, clusterName, collectionTS;
```

How frequently do we see different java versions:

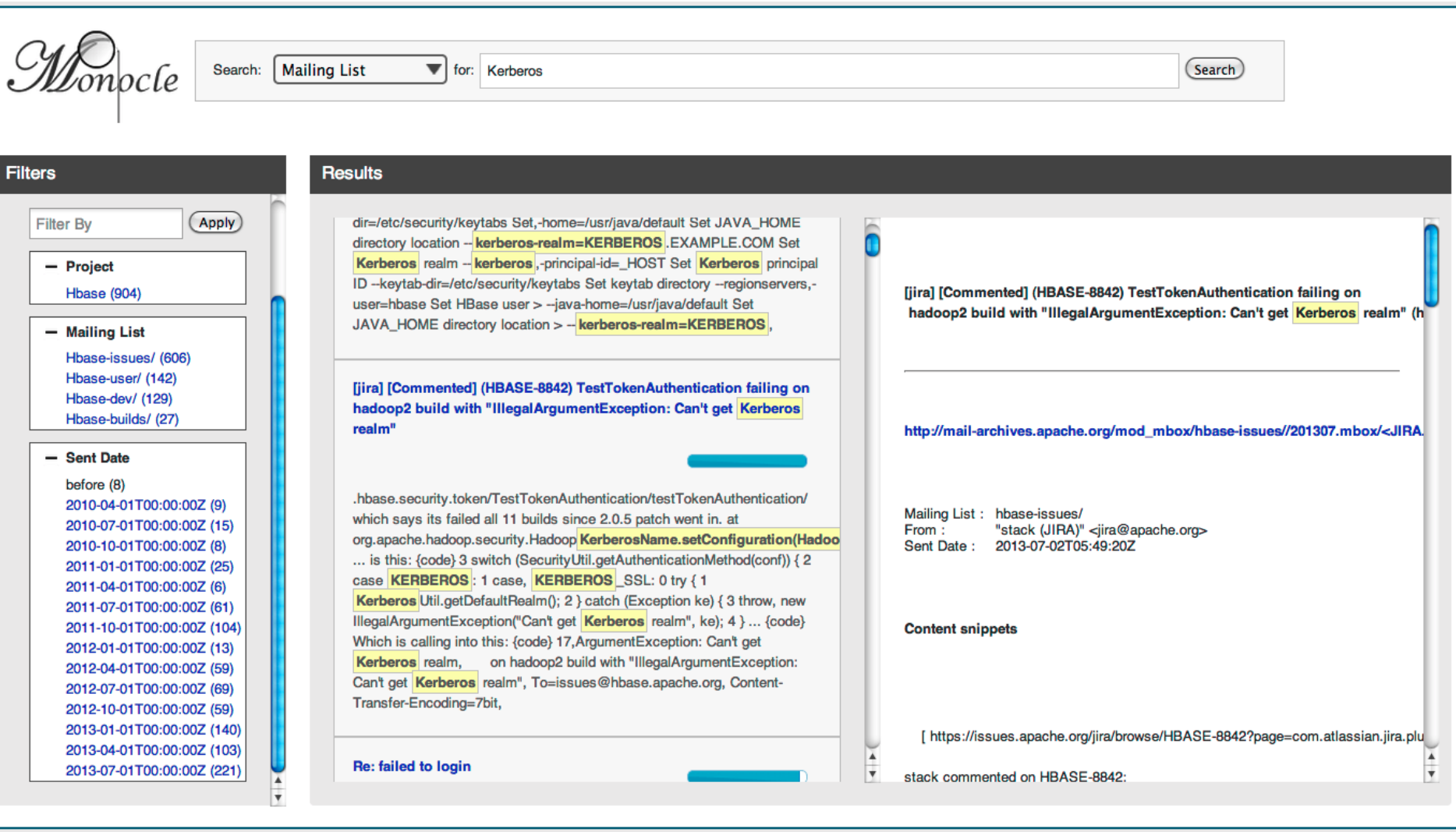
```
SELECT javaVersion, count(*)
FROM machine_sysstats
GROUP BY javaVersion;
```

Who are our free/trial customers (or at least, what are their hostnames)?

```
SELECT machinename, cm.free, cm.trial FROM machine_sysstats ms INNER JOIN cluster_metadata cm
ON ms.customerKey = cm.customerKey AND ms.clusterName = cm.clusterName AND ms.collectionTS = cm.mostRecentCollect
WHERE cm.free = "true" OR cm.trial = "true"
ORDER BY machinename LIMIT 1000
```

What are the largest clusters we have seen (a nested query!)?

```
SELECT q.customerKey, q.customerName, q.clusterName, max(q.cnt) AS cnt FROM
  (select sys.customerKey, cm.customerName, sys.collectionts, sys.clusterName, count(sys.machineName) AS cnt FROM
    INNER JOIN customer_metadata cm ON sys.customerKey = cm.customerKey GROUP BY sys.customerKey, cm.customerName
  GROUP BY customerKey, customerName, clusterName
  ORDER BY cnt DESC LIMIT 10
```



Queries Supported

- What are the most commonly encountered errors?
- How many IOExceptions have we recorded from datanodes in a certain month?
- What is the distribution of workloads across Impala, Apache Hive, and HBase?
- Which OS versions are most commonly used?
- What are the mean and variance of hardware configurations?
- How many types of hardware configuration are there at a single customer site?
- Does anyone use a specific parameter that we want to deprecate?

Summary

The beauty of this system is it solves a number of core and difficult use cases. It is also open to integrating with various other systems and making the overall solution much better.

- Scalability
- Flexibility in building new features & products
- Low Cost of Ownership
- Ease of Managing Big Data

Thank you!

Jayant Shekhar, Sr. Solutions Architect, Cloudera @jshekhar

Additional Questions & Discussions : Cloudera Booth : 1:00-1:30pm