



HCatalog

Table Management for Hadoop

Alan F. Gates
@alanfgates



Who Am I?

- HCatalog committer and mentor
- Co-founder of Hortonworks
- Tech lead for Data team at Hortonworks
- Pig committer and PMC Member
- Member of Apache Software Foundation and Incubator PMC
- Author of *Programming Pig* from O'Reilly

Users: Data Sharing is Hard

This is programmer Bob, he uses Pig to crunch data.



Photo Credit: totalAldo via Flickr

Bob, I need today's data

Ok

This is analyst Joe, he uses Hive to build reports and answer ad-hoc queries.



Hmm, is it done yet? Where is it? What format did you use to store it today? Is it compressed? And can you help me load it into Hive, I can never remember all the parameters I have to pass that alter table command.

Dude, we need HCatalog

Pig Example

```
raw      = load '/rawevents/20100819/data' using MyLoader()  
          as (ts:long, user:chararray, url:chararray);  
botless = filter raw by NotABot(user);  
...  
store output into '/processedevents/20100819/data';
```

Processedevents consumers must be manually informed by producer that data is available, or poll on HDFS (this is bad).

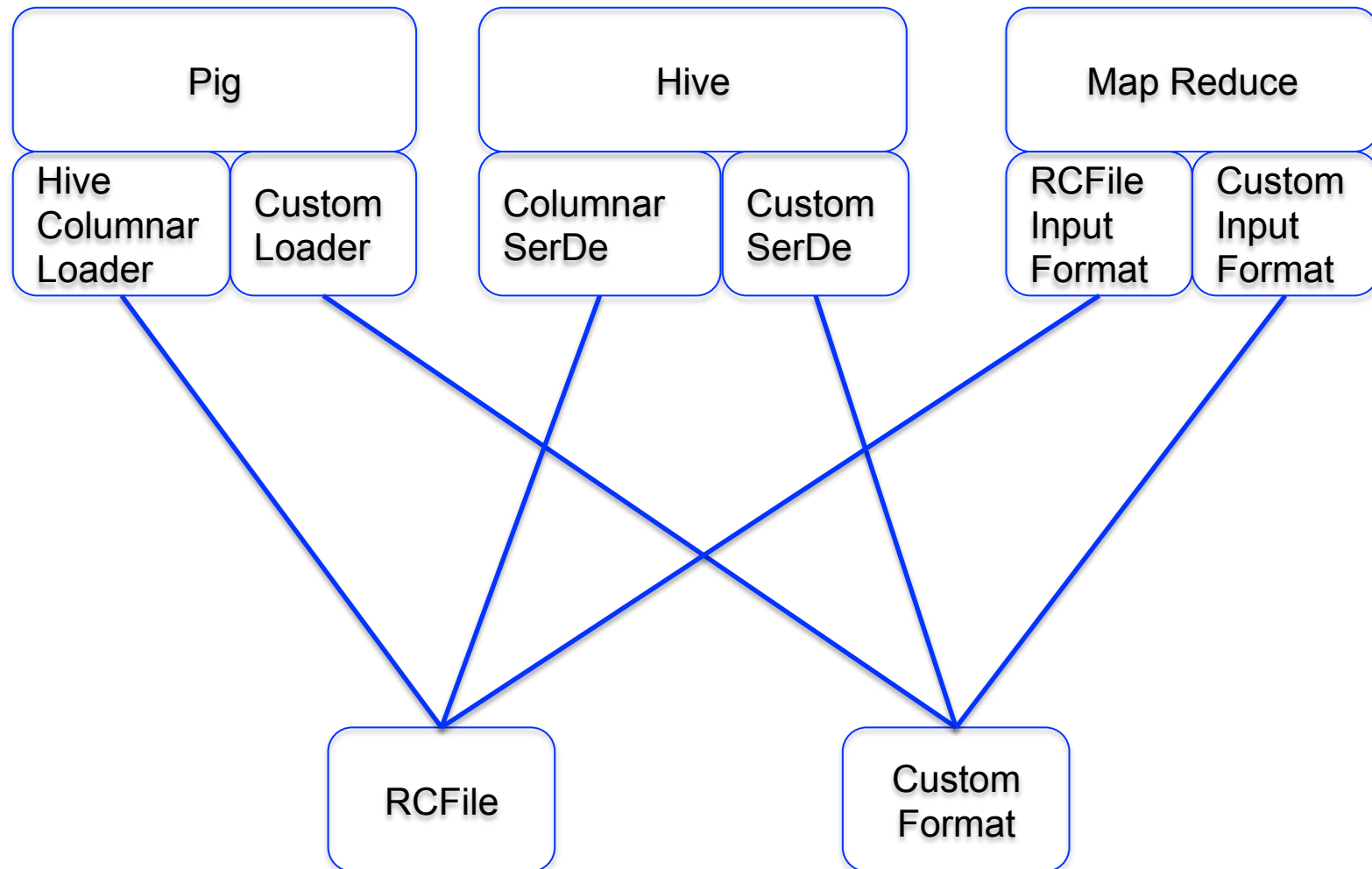


```
raw      = load 'rawevents' using HCatLoader();  
botless = filter raw by date = '20100819' and NotABot(user);  
...  
store output into 'processedevents'  
          using HCatStorer("date=20100819");
```

Processedevents consumers will be notified by HCatalog via JMS that data is available; they can then start their jobs.

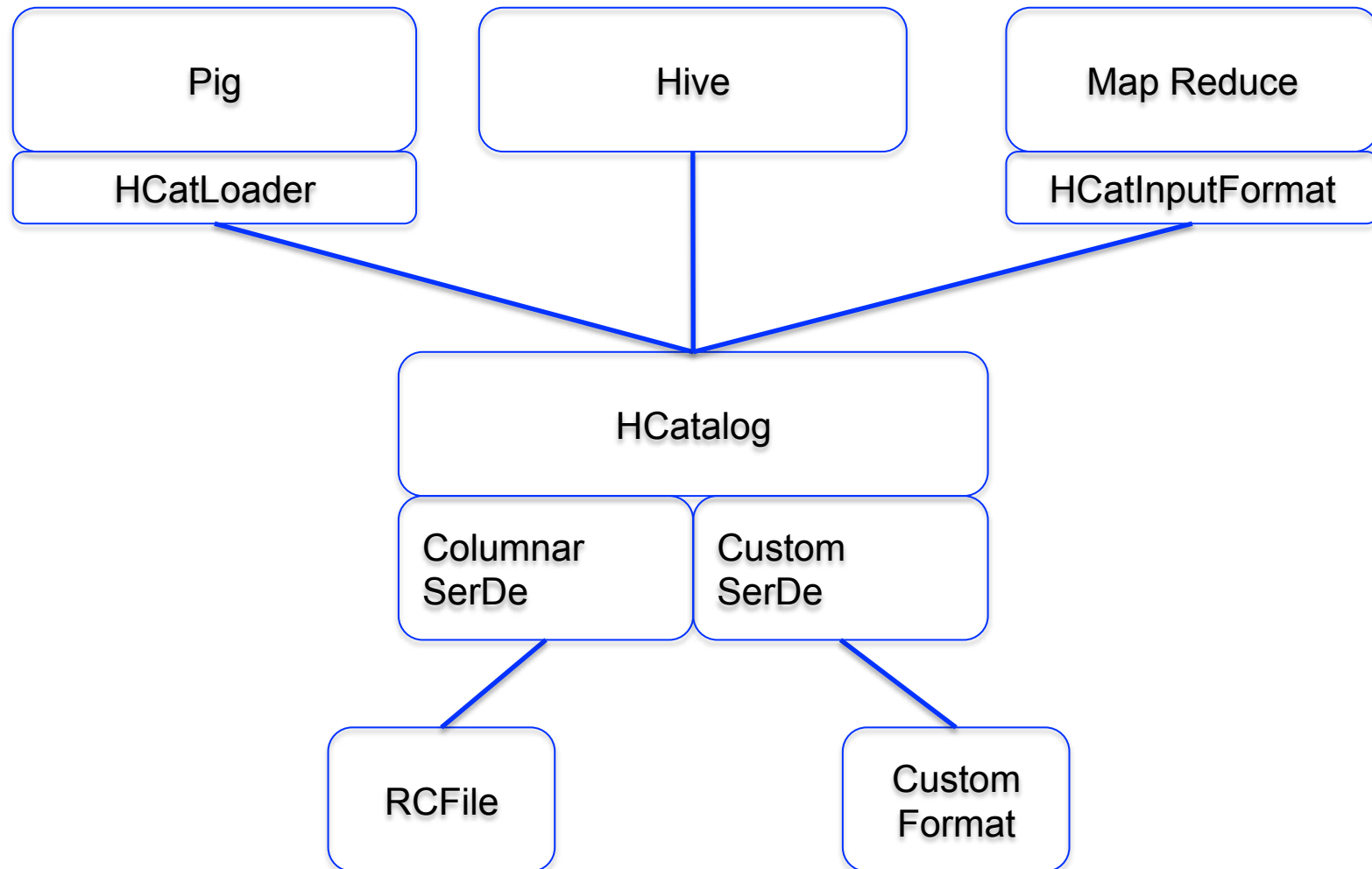
Developers:

Each tool requires its own Translator



Developers:

Each tool requires its own Translator



Ops: Can I Ever Change the Data Format?

- Data format can be changed (e.g. CSV to JSON)
 - no need to reformat existing data
 - new data will be written new format
 - HCatalog can read across the format changes
 - Users programs will not see the difference
- Data location can be changed without affecting user programs
- New columns can be added via `alter table add column`
 - no need to reformat existing data
 - fields not present in old data will get a null when reading old data

Relationship to Hive

- If you are a Hive user, you can transition to HCatalog with no metadata modifications (true starting with version 0.4)
 - Uses Hive's SerDes for data translation (starting in 0.4)
 - Uses Hive SQL for data definition (DDL)
- Provides a different security implementation; security delegated to underlying storage
 - Currently integrated with HDFS
 - Integration with HBase in progress
 - In near future you will be able to choose to use Hive authorization model if you wish

Current Status

- Release 0.4 (expected out next month)
 - Hive/Pig/MapReduce integration
 - Support for any data format with a SerDe (Text, Sequence, RCFile, JSON SerDes included)
 - Notification via JMS
 - Basic HBase integration

Current Development

- Improving HBase integration
 - new HBase security features
 - repeatable read for HBase tables
- REST API for HCatalog
 - Databases, tables, partitions are objects
 - PUT to create or modify objects
 - GET to retrieve information about objects
 - DELETE to drop objects
 - Submitted documents and results encoded in JSON

Future Directions

Storing Semi-/Unstructured Data

Table Users

Name	Zip
Alice	93201
Bob	76331



```
select name, zip  
from users;
```

File Users

```
{"name": "alice", "zip": "93201"}  
{"name": "bob", "zip": "76331"}  
{"name": "cindy"}  
{"zip": "87890"}
```



```
A = load 'Users' as  
    (name:chararray, zip:chararray);  
B = foreach A generate name, zip;
```

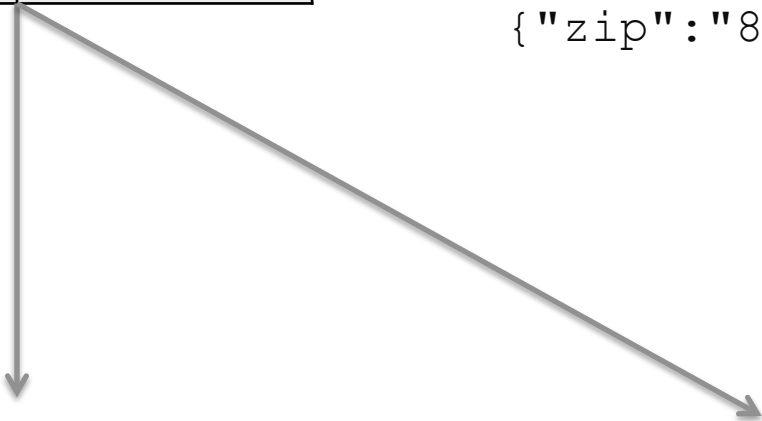
Storing Semi-/Unstructured Data

Table Users

Name	Zip
Alice	93201
Bob	76331

File Users

```
{"name": "alice", "zip": "93201"}  
{"name": "bob", "zip": "76331"}  
{"name": "cindy"}  
{"zip": "87890"}
```



```
select name, zip  
from users;
```

```
A = load 'Users'  
B = foreach A generate name, zip;
```

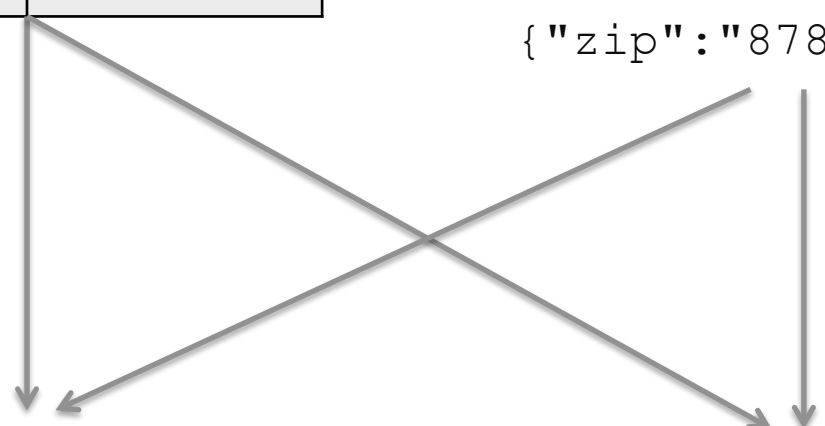
Storing Semi-/Unstructured Data

Table Users

Name	Zip
Alice	93201
Bob	76331

File Users

```
{"name": "alice", "zip": "93201"}  
{"name": "bob", "zip": "76331"}  
{"name": "cindy"}  
{"zip": "87890"}
```



The diagram illustrates the flow of data from two sources to two different query languages. Arrows originate from the 'Table Users' table and the 'File Users' JSON list. One arrow from the table points to the SQL query, and another from the table points to the Pig query. Similarly, one arrow from the file points to the SQL query, and another from the file points to the Pig query.

```
select name, zip  
from users;
```

```
A = load 'Users'  
B = foreach A generate name, zip;
```

Data Lifecycle Management

Cleaning



Archiving



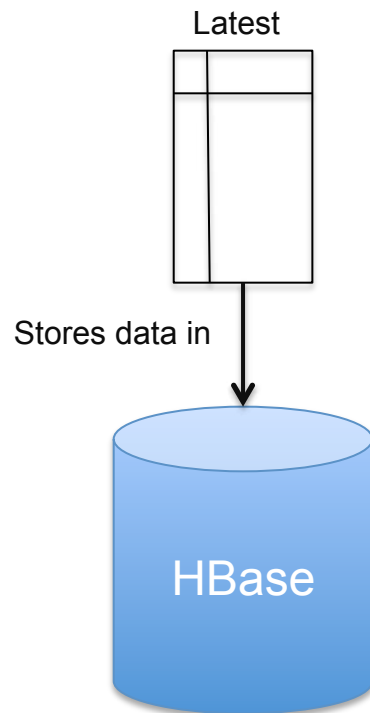
Replication



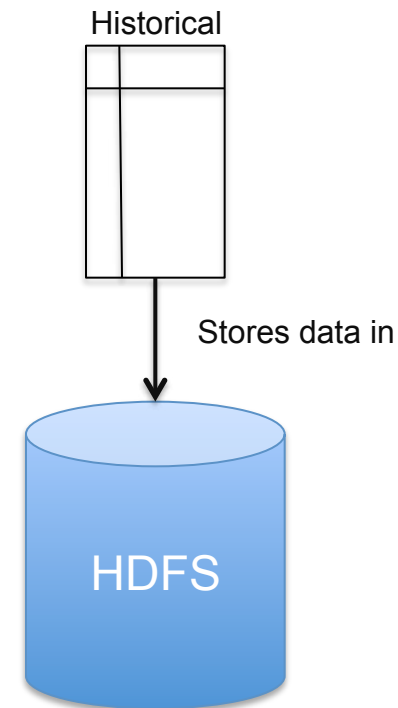
Compaction



Partitions in Different Storage

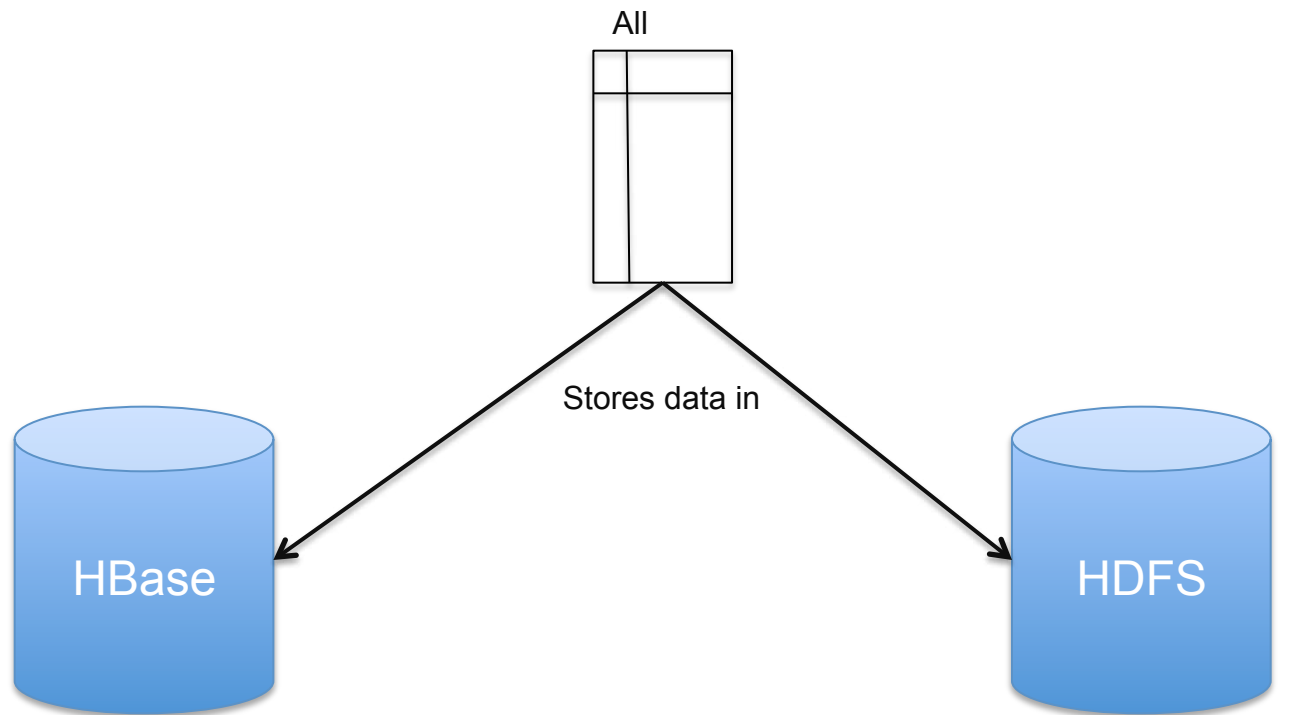


- Data can be streamed in
- available for read almost instantly



- Must load in batches
- scan times 10x HBase

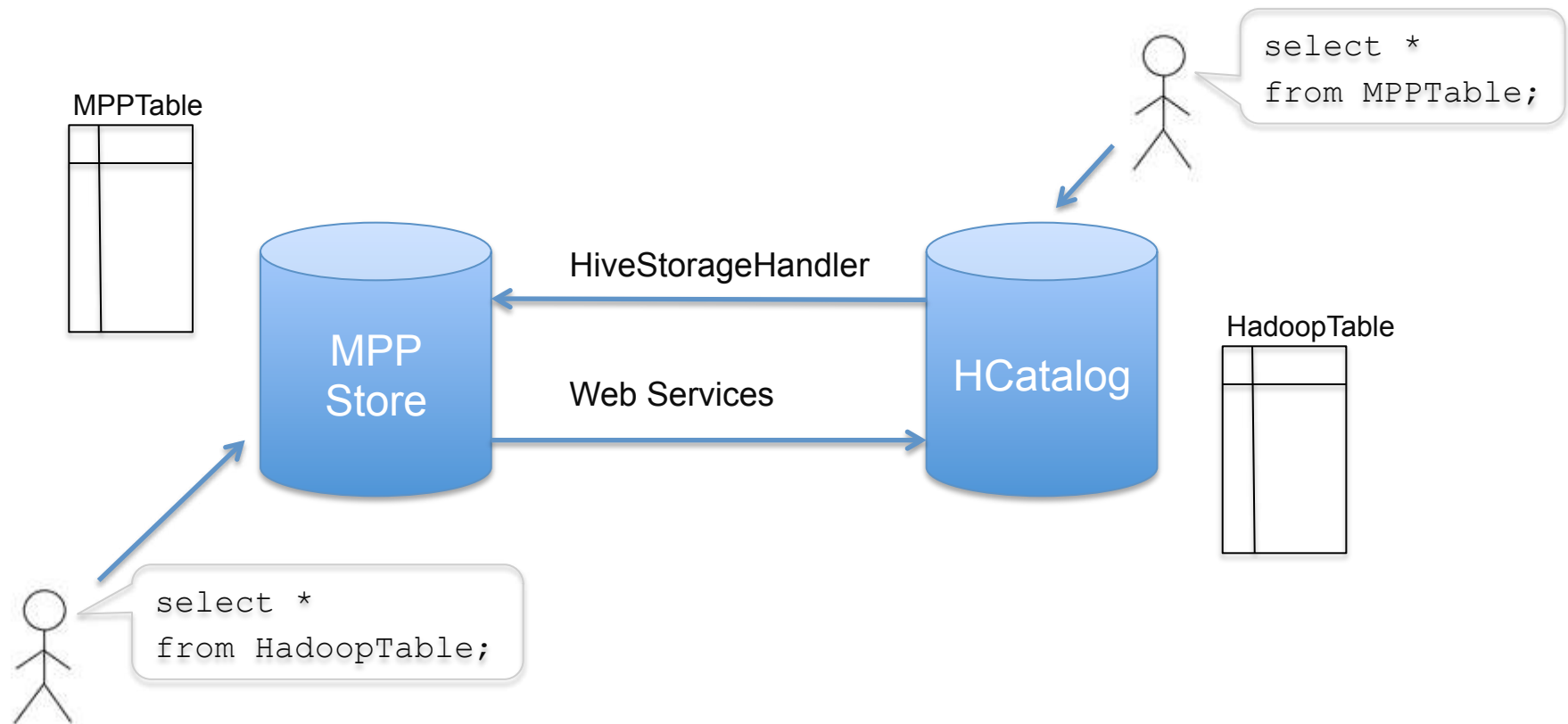
Partitions in Different Storage



- Data can be streamed in
- available for read almost instantly

- Must load in batches
- scan times 10x HBase

Federation with MPP Data Stores



Storing More Metadata

- HCatalog currently stores metadata in RDBMS
 - Most people use MySQL
- Table level statistics can currently be stored
- Would like to store partition statistics
 - But this could overwhelm the RDBMS
- Would like to store user generated metadata for partitions
 - But this could overwhelm the RDBMS
- Could we store this data in HBase instead?
- Could we store all the metadata in HBase instead?



Thank You



Other Resources

- **Next Webcast: Extending Hadoop beyond MapReduce**

- March 7
- 10am Pacific/1pm Eastern
- <http://hortonworks.com/webinars/>

Register Now

- **Hadoop Summit**

- June 13-14
- San Jose, California
- Call for Papers Deadline extended
- Hadoopsummit.org



- **Hadoop Training and Certification**

- Developing Solutions Using Apache Hadoop
- Administering Apache Hadoop
- <http://hortonworks.com/training/>

