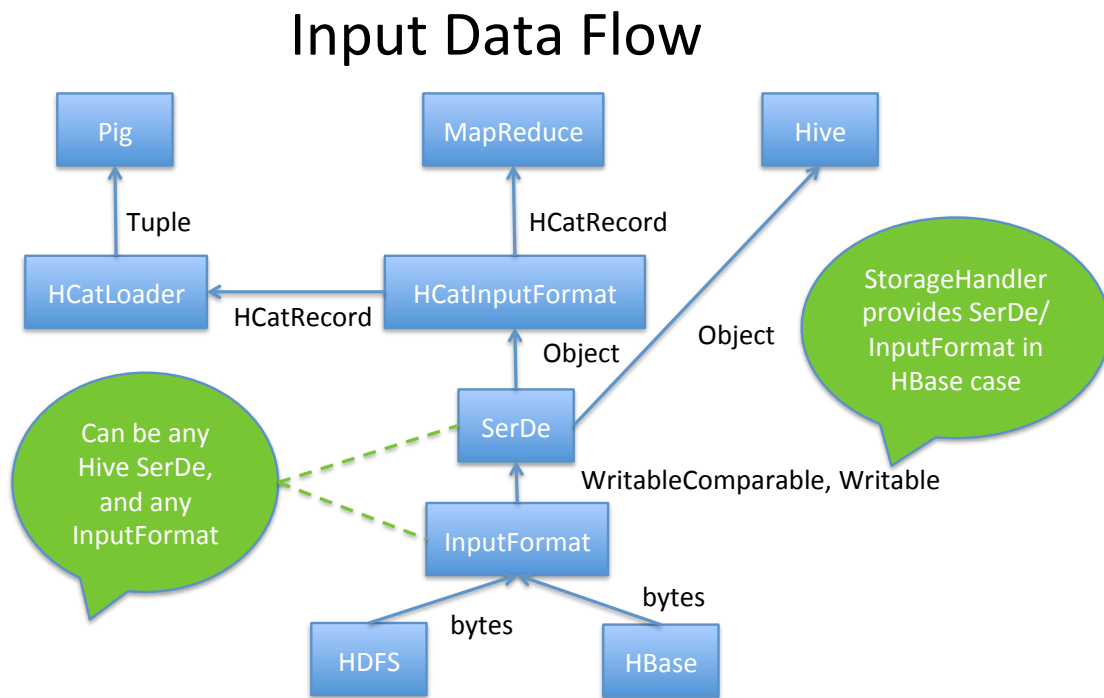


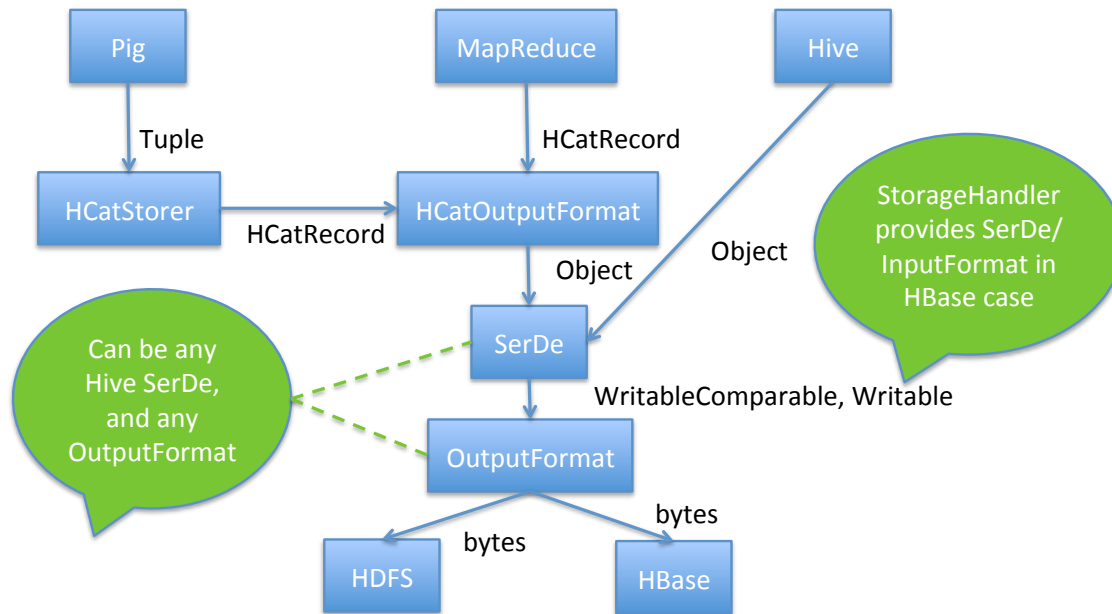
HCat StorageDriver -> SerDe conversion design notes:

Data flow in input case:



Data flow in output case:

# Output Data Flow



Page 3

## Rewrite of HCatInputFormat/HCatOutputFormat

Currently HCatInputFormat and HCatOutputFormat use a storage driver to convert from/to the native InputFormat/OutputFormat. This will change to them interacting with a SerDe to create HCatRecords. They will continue to set up job information for the metastore.

As far as I can tell neither HCIF nor HCOF are authorizing operations at the moment. They are just relying on the underlying opens to fail. This won't fail on the client side which we would like it to. Changes to this need to be integrated with Enis' changes to move HCat security to an implementation of Hive security.

Changes to InputJobInfo.create - remove serverUri and serverKerberosPrincipal as arguments. Add JobConf reference. These should be picked up in the conf file.

Change to HCatInputFormat/HCatBaseInputFormat -

1. createRecordReader changes to return an HCat specific RecordReader that takes an Object + ObjectInspector and wraps it in an HCatRecord. It will still need to call the underlying RecordReader, but it will need to push the resulting value through SerDe.deserialize. The HCatRecord will then be wrapped around the resulting Object + the ObjectInspector returned from that SerDe.

2. HCatInputFormat will need to change to take wrap the old `mapred.InputFormat` class, rather than the `mapreduce.InputFormat` class it currently wraps.

Changes to `OutputJobInfo.create` - remove `serverUri` and `serverKerberosPrincipal` as arguments. Add `JobConf` reference. These should be picked up in the conf file.

Changes to `HCatOutputFormat/HCatBaseOutputFormat` -

1. `getRecordWriter` changes to return an HCat specific `RecordWriter` that takes an `HCatRecord`. It will then need to fetch the appropriate `SerDe` push the `HCatRecord` through `SerDe.serialize()`, providing the `HCatRecord.getObjectInspector` as the `ObjectInspector`.
2. `HCatOutputFormat` will need to change to take wrap the old `mapred.OutputFormat` class, rather than the `mapreduce.OutputFormat` class it currently wraps.

Changes to `HCatOutputCommitter`: The `OutputCommitter` will have check whether the underlying `mapred.OutputFormat` has registered an `OutputCommitter` in the `JobConf` or not. If so, it must be invoked. This implies a limitation that HCat cannot write dynamic partitions across different output formats. It will also cause problems for Pig and Hive if users try to store to tables with different output formats in one MR job.

Update of `StorageHandler` to include any HCat specific functionality that is required

1. add `getAuthorizationProvider`
2. do we need `getAuthenticationProvider`? I don't think so, but we need to figure this out.
3. Hive needs to add a property in its call to `configureTableProperties` that indicates whether this operation is read or write

`HCatRecord` needs to be completely rewritten. Inbound records need to wrap an `Object` + `ObjectInspector` from a `SerDe`. Outbound records need to store data in a list (as today) and provide an `ObjectInspector` for writing data out. A `getObjectInspector` method will need to be added to `HCatRecord`. If the user tries to write to an inbound record `HcatRecord` will serialize the added value into the object using the `ObjectInspector`.

Currently `HCatalog` handles JSON by using the `PigLoader` -> `StorageDriver` converter. We will need to write a JSON `serde`.

CLI changes: The command line interface needs to be changed to remove checks for `ISD` or `OSDs`.

I have not yet defined a `SerDe` simplifier class. I'll leave that as an exercise for later.