Giraph formats contrib

Giraph includes a subproject under /trunk that supplies additional vertex input/output formats.

In order to build, the latest giraph-with-dependencies.jar must be installed in your local maven repo. Giraph is not yet hosted in Maven central.

Currently the subproject offers abstract vertex input/output formats to help BSP initialization over graphs stored in Apache HBase and Apache Accumulo.

The current build has been tested against HBase 0.90.5 and Accumulo 1.4.

Users can extend these base classes in order to read/write BasicVertex instances from HBase or Accumulo. Subclasses extending these will be able to build HBase/Accumulo rows into Giraph vertex objects, and vice versa.

These classes wrap their respective delegate input/output formats and offer all the normal functionality those input/output formats provide. For instance, HBase MapReduce jobs normally must set TableInputFormat.INPUT_TABLE and pass that reference to the Configuration instance. Users of HBaseVertexInputFormat can follow the exact same approach.

```
conf.set(TableInputFormat.INPUT_TABLE, TABLE_NAME);
conf.set(TableOutputFormat.OUTPUT_TABLE, TABLE_NAME);

GiraphJob giraphJob = new GiraphJob(conf, "My HBase BSP job");
...
giraphJob.setVertexInputFormatClass(TableEdgeInputFormat.class);
```

In the above example, TableEdgeInputFormat is a subclass of HBaseVertexInputFormat. Since HBaseVertexInputFormat is setup to properly communicate all static helper method calls and configuration from the HBase TableInputFormat, users can pass subclass instances to the Giraph job and be confident the setup routines against the normal HBase classes will transparently take hold.

The same follows for Accumulo, except for one minor ‘gotcha’. Accumulo requires you call the static helper methods on your configuration object before passing that conf to your GiraphJob instance. This has to do with how Giraph delegates to, and not extends Job. Should you forget to do this, the Accumulo format will present an IO exception detailing the cause.

```
Configuration conf = new Configuration();
conf.set(AccumuloVertexOutputFormat.OUTPUT_TABLE, TABLE_NAME);
/*
 * Very important to initialize the formats before sending configuration to the GiraphJob. Otherwise the internally constructed Job in GiraphJob will not have the proper context initialization.
 */
AccumuloInputFormat.setInputInfo(conf, USER, "".getBytes(),
 TABLE_NAME, new Authorizations());
GiraphJob job = new GiraphJob(conf, "My Accumulo BSP job");
```

The following unit test examples help show usage

For HBaseVertex input/output subclass examples, check out: TableEdgeInputFormat.java and TableEdgeOutputFormat.java
These classes are used in the unit test class TestHBaseRootMarkerVertexFormat.java

For AccumuloVertex input/output subclass examples, check out: AccumuloEdgeInputFormat.java and AccumuloEdgeOutputFormat.java
These classes are used in the unit test class TestAccumuloVertexFormat.java