IOSystem

General Information

Since Hama 0.4.0 we provide a input and output system for BSP Jobs.

TODO: Some blahblah about key value and stuff What's in case when no input is configured? and stuff like that should be documented here..

Input

Configuring Input

When setting up a BSPJob, you can provide a InputFormat and a Path where to find the input.

```
     BSPJob job = new BSPJob();
     // detail stuff omitted
     job.setInputPath(new Path("/tmp/test.seq");
     job.setInputFormat(org.apache.hama.bsp.SequenceFileInputFormat.class);
```

Another way to add input paths is following:

```
   SequenceFileInputFormat.addInputPath(job, new Path("/tmp/test.seq"));
```

You can also add multiple paths by using this method:

```
   SequenceFileInputFormat.addInputPaths(job, "/tmp/test.seq,/tmp/test2.seq,/tmp/test3.seq");
```

Note that these paths must be separated by a comma.

In case of a `SequenceFileInputFormat` the key and value pair are parsed from the header.

When you use want to read a basic textfile with `TextInputFormat` the key is always `LongWritable` which contains how much bytes have been read and `Text` which contains a line of your input.

Using Input

You can now read the input from each of the functions in `BSP` class which has `BSPPeer` as parameter. (e.G. setup / bsp / cleanup)

In this case we read a normal text file:

```
@Override
 public final void bsp(
     BSPPeer<LongWritable, Text, KEYOUT, VALUEOUT> peer)
     throws IOException, InterruptedException, SyncException {

     // this method reads the next key value record from file
     KeyValuePair<LongWritable, Text> pair = peer.readNext();

     // the following lines do the same:
     LongWritable key = new LongWritable();
     Text value = new Text();
     peer.readNext(key, value);
 }
```

Consult the docs for more detail on events like end of file.

There is also a function which allows you to re-read the input from the beginning.

This snippet reads the input five times:

```
 for(int i = 0; i < 5; i++){
   LongWritable key = new LongWritable();
   Text value = new Text();
   while (peer.readNext(key, value)) {
      // read everything
   }
   // reopens the input
   peer.reopenInput()
 }
```

Custom Inputformat

You can implement your own inputformat blabla

Output

Configuring Output

Using Input

Custom Outputformat

Implementation notes

Internal implementation details

BSPJobClient

1. Create the splits for the job 2. writeNewSplits() 3. job.set("bsp.job.split.file", submitSplitFile.toString()); 4. Sets the number of peers to split.lenth

JobInProgress

1. Receives splitFile 2. Add split argument to TaskInProgress constructor

Task

1. Gets his split from Groom 2. Initializes everything in BSPPeerImpl