DemuxModification

Using the default TimeStamp Parser

By default, Chukwa will use the default TsProcessor.

This parser will try to extract the real log statement from the log entry using the %d{ISO8601} date format. If it fails, it will use the time at which the chunk as been written to disk (collector timestamp).

Your log will be automatically available from the Web Log viewer under the <YourRecordTypeHere> directory

Using a specific Parser

If you want to extract some specific information and perform more processing you need to write your own parser. Like any M/R program, your have to write at least the Map side for your parser. The reduce side is Identity by default.

MAP side of the parser

Your can write your own parser from scratch or extend the AbstractProcessor class that hides all the low level action on the chunk. Then you have to register your parser to the demux (link between the RecordType and the parser)

Parser registration

- Edit ${CHUKWA_HOME}/conf/chukwa-demux-conf.xml and add the following lines
  <property>
  <name><YourRecordType_Here></name>
  <value><org.apache.hadoop.chukwa.extraction.demux.processor.mapper.MyParser></value>
  <description>Parser class for <YourRecordType_Here></description> </property>

(Tips: You can use the same parser for different recordType)

Parser implementation

```
public class MyParser extends AbstractProcessor
{
       protected void parse(String recordEntry,
                                    OutputCollector<ChukwaRecordKey, ChukwaRecord> output,
                                    Reporter reporter)
        {

           // Extract Log4j information, i.e timestamp, logLevel, logger, ...
           SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm");
           // Extract log timestamp & Log4j information
           String dStr = recordEntry.substring(0, 23);
           int start = 24;
           int idx = recordEntry.indexOf(' ', start);
           String logLevel = recordEntry.substring(start, idx);
           start = idx + 1;
           idx = recordEntry.indexOf(' ', start);
           String className = recordEntry.substring(start, idx-1);
           String body = recordEntry.substring(idx + 1);

           Date d = sdf.parse(dStr);
           key = new ChukwaRecordKey();
           record = new ChukwaRecord();

           key = new ChukwaRecordKey();
           key.setKey("<YOUR_KEY_HERE>"));
           key.setReduceType("<YOUR_RECORD_TYPE_HERE>");

           record = new ChukwaRecord();
           record.setTime(d.getTime());

           // Parse your line here and extract useful information
           // Add your {key,value} pairs
           record.add(key1, value1);
           record.add(key2, value2);
           record.add(key3, value3);

           // Output your record
           output.collect(key, record);
        }
}
```

(Tips: see org.apache.hadoop.chukwa.extraction.demux.processor.mapper.Df class, for an example of Parser class)

REDUCE side of the parser

You only need to implement a reduce side if you need to group records together. Here the interface that your need to implement:

The link between the Map side and the reduce is done by setting your reduce class into the reduce type: key.setReduceType("<YourReduceClassHere>");

```
public interface ReduceProcessor
{
           public String getDataType();
           public void process(ChukwaRecordKey key,Iterator<ChukwaRecord> values,
                     OutputCollector<ChukwaRecordKey,
                     ChukwaRecord> output, Reporter reporter);
}
```

(Tips: see org.apache.hadoop.chukwa.extraction.demux.processor.reducer.SystemMetrics class, for an example of Reduce class)

Parser key field

Your data is going to be sorted by RecordType then by the key field. The default implementation use the following grouping for all records:

1. Time partition (Time up to the hour)
2. Machine name (physical input source)
3. Record timestamp

Output directory

The demux process will use the recordType to save similar records together (same recordType) to the same directory:
<Your_Cluster_Information>/<Your_Record_Type>/