## Example 1 (hadoop – MapReduce con combiner)

public static void main(String[] args) throws Exception {

Configuration conf = new Configuration();

Job job = Job.getInstance(conf, "word count");

job.setJarByClass(WordCount.class);

job.setMapperClass(TokenizerMapper.class);

job.setCombinerClass(IntSumReducer.class);

job.setReducerClass(IntSumReducer.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

FileInputFormat.addInputPath(job, new Path(args[0]));

FileOutputFormat.setOutputPath(job, new Path(args[1]));

System.exit(job.waitForCompletion(true) ? 0 : 1);

}

}

### OUTPUT: console

===== Righe notevoli rilevate =================

Input: 1

Mapper: 1

Chain nodes: 0

Combiners: 1

Partitioners: 0

Reducers: 1

NumReduceTask: 0

Outputs: 1

NamedOutputs: 0

===== Riconosciuta struttura hadoop ==============

--- INPUT --------

aggiunta risorsa IO: IO\_newPath(args[0])

aggiunto input: job.INPUT\_newPath(args[0])

aggiunto link: "IO\_newPath(args[0])"->job.INPUT\_newPath(args[0])

--- MAPPER --------

aggiunto mapper: job.MAPPER\_TokenizerMapper

aggiunto link:job.INPUT\_newPath(args[0])->job.MAPPER\_TokenizerMapper

--- CHAIN ---------

--- COMBINER ------

aggiunto combiner:job.COMBINER\_IntSumReducer

aggiunto link:job.MAPPER\_TokenizerMapper->job.COMBINER\_IntSumReducer

sto aggiungendojob.COMBINER\_IntSumReducer

--- PARTITIONER ---

--- REDUCER -------

aggiunto reducer:job.REDUCER\_IntSumReducer

aggiunto link:job.COMBINER\_IntSumReducer->job.REDUCER\_IntSumReducer

--- OUTPUT --------

aggiunta risorsa IO: IO\_newPath(args[1])

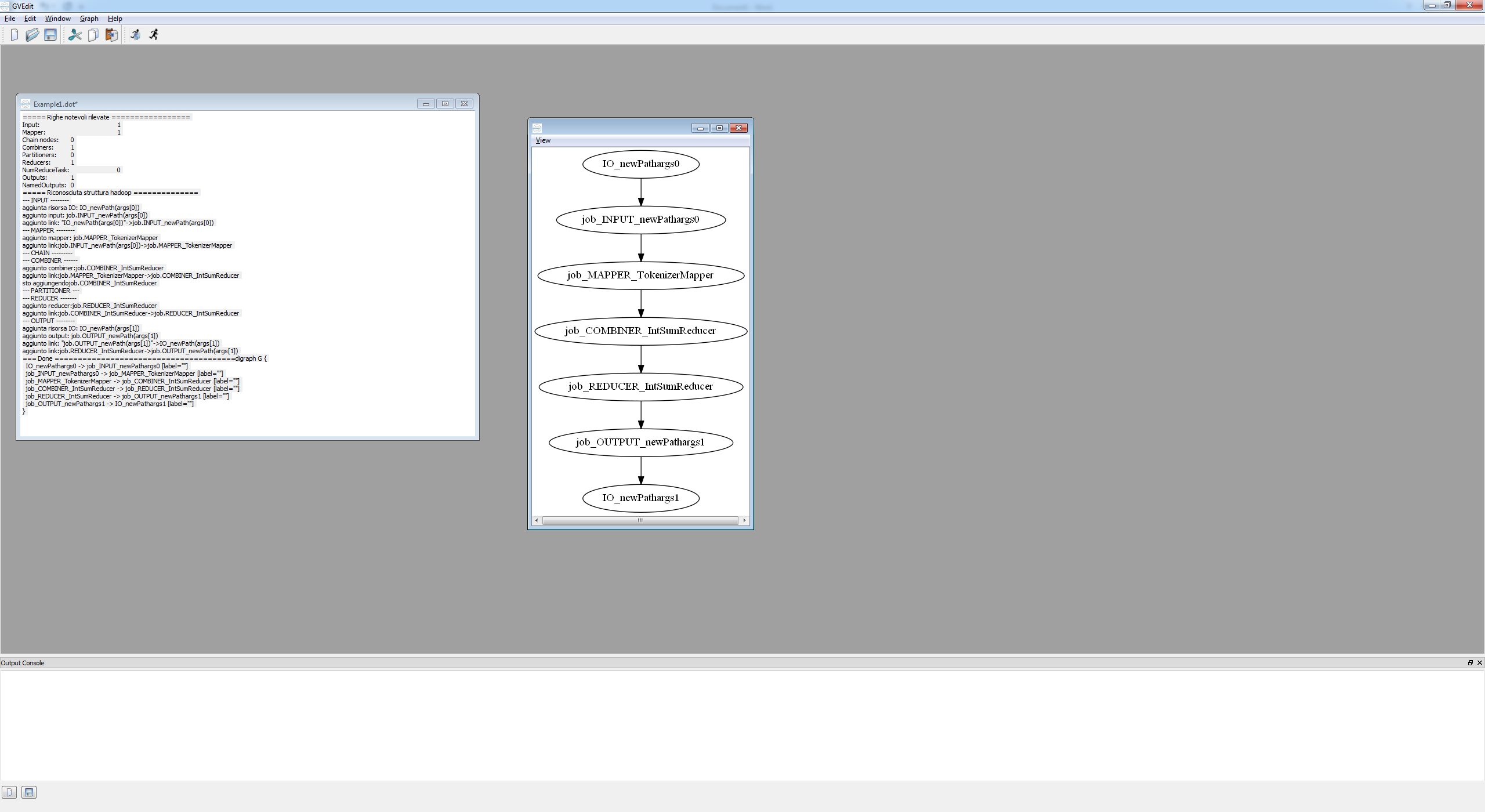
aggiunto output: job.OUTPUT\_newPath(args[1])

aggiunto link: "job.OUTPUT\_newPath(args[1])"->IO\_newPath(args[1])

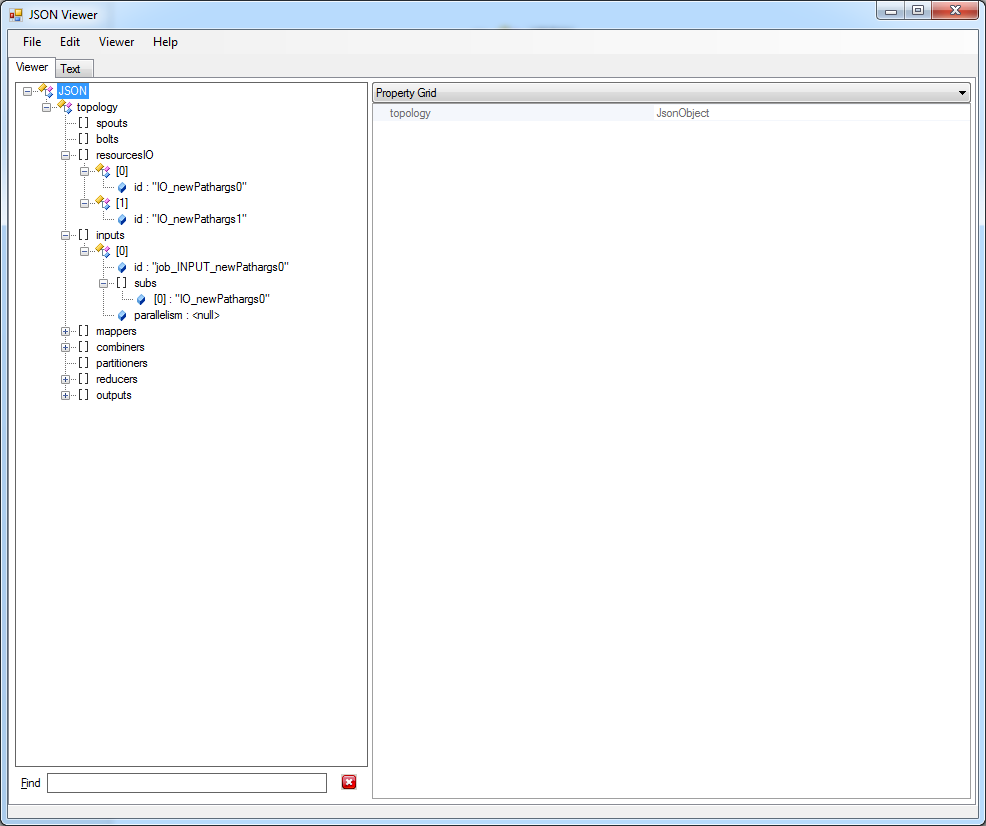
aggiunto link:job.REDUCER\_IntSumReducer->job.OUTPUT\_newPath(args[1])

=== Done =======================================

### OUTPUT: file example1.dot



### OUTPUT: file Example1.json



## Example 4 (hadoop – Collegamento tra job)

public class WordCountSorting {

// here the word keys shall be sorted

//let us write the wordcount logic first

public static void main(String[] args)throws IOException,InterruptedException,ClassNotFoundException {

//THE DRIVER CODE FOR MR CHAIN

Configuration conf1=new Configuration();

Job j1=Job.getInstance(conf1);

j1.setJarByClass(WordCountSorting.class);

j1.setMapperClass(MyMapper.class);

j1.setReducerClass(MyReducer.class);

j1.setMapOutputKeyClass(Text.class);

j1.setMapOutputValueClass(IntWritable.class);

j1.setOutputKeyClass(LongWritable.class);

j1.setOutputValueClass(Text.class);

Path outputPath=new Path("FirstMapper");

FileInputFormat.addInputPath(j1,new PathArgs0);

FileOutputFormat.setOutputPath(j1,outputPath);

outputPath.getFileSystem(conf1).delete(outputPath);

j1.waitForCompletion(true);

Configuration conf2=new Configuration();

Job j2=Job.getInstance(conf2);

j2.setJarByClass(WordCountSorting.class);

j2.setMapperClass(MyMapper2.class);

j2.setNumReduceTasks(0);

j2.setOutputKeyClass(Text.class);

j2.setOutputValueClass(IntWritable.class);

Path outputPath1=new Path(args[1]);

FileInputFormat.addInputPath(j2, outputPath);

FileOutputFormat.setOutputPath(j2, outputPath1);

outputPath1.getFileSystem(conf2).delete(outputPath1, true);

System.exit(j2.waitForCompletion(true)?0:1);

}

}

### OUTPUT: console

===== Righe notevoli rilevate =================

Input: 2

Mapper: 2

Chain nodes: 0

Combiners: 0

Partitioners: 0

Reducers: 1

NumReduceTask: 1

Outputs: 2

NamedOutputs: 0

===== Riconosciuta struttura hadoop ==============

--- INPUT --------

aggiunta risorsa IO: IO\_newPathArgs0

aggiunto input: j1.INPUT\_newPathArgs0

aggiunto link: "IO\_newPathArgs0"->j1.INPUT\_newPathArgs0

aggiunta risorsa IO: IO\_outputPath

aggiunto input: j2.INPUT\_outputPath

aggiunto link: "IO\_outputPath"->j2.INPUT\_outputPath

--- MAPPER --------

aggiunto mapper: j1.MAPPER\_MyMapper

aggiunto link:j1.INPUT\_newPathArgs0->j1.MAPPER\_MyMapper

aggiunto mapper: j2.MAPPER\_MyMapper2

aggiunto link:j2.INPUT\_outputPath->j2.MAPPER\_MyMapper2

--- CHAIN ---------

--- COMBINER ------

--- PARTITIONER ---

--- REDUCER -------

aggiunto reducer:j1.REDUCER\_MyReducer

aggiunto link:j1.MAPPER\_MyMapper->j1.REDUCER\_MyReducer

--- OUTPUT --------

aggiunta risorsa IO: IO\_outputPath1

aggiunto output: j1.OUTPUT\_outputPath

aggiunto link: "j1.OUTPUT\_outputPath"->IO\_outputPath

aggiunto link:j1.REDUCER\_MyReducer->j1.OUTPUT\_outputPath

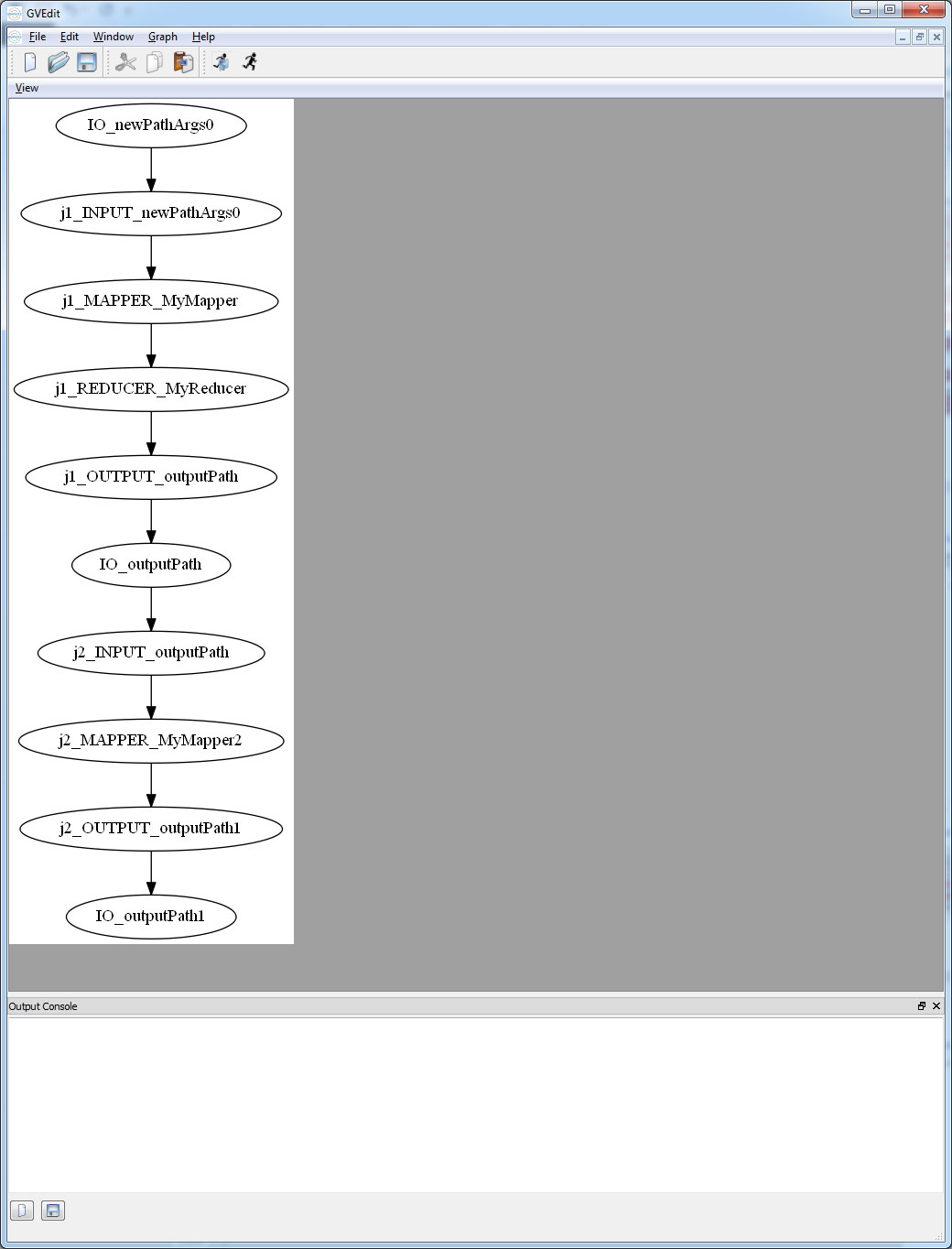
aggiunto output: j2.OUTPUT\_outputPath1

aggiunto link: "j2.OUTPUT\_outputPath1"->IO\_outputPath1

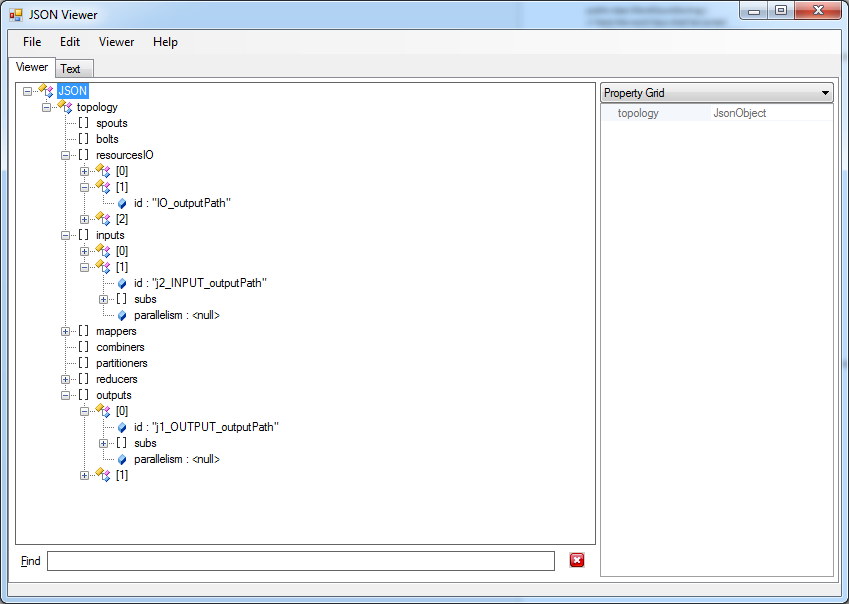
aggiunto link:j2.MAPPER\_MyMapper2->j2.OUTPUT\_outputPath1

=== Done =======================================

### OUTPUT: file Example4.dot



### OUTPUT: file Example4.json



## FocusedCrawler (storm)

public static StormTopology createTopology(XMLConfiguration config) {

…

// Create topology

TopologyBuilder builder = new TopologyBuilder();

builder.setSpout("wpSpout", wpSpout, 1);

builder.setBolt("WpDeserializer", wpDeserializer, 4)

.shuffleGrouping("wpSpout");

builder.setBolt("expander", urlExpander, 8)

.shuffleGrouping("WpDeserializer");

builder.setBolt("articleExtraction", articleExtraction, 1)

.shuffleGrouping("expander", "webpage");

builder.setBolt("mediaExtraction", mediaExtraction, 1)

.shuffleGrouping("expander", "media");

builder.setBolt("webPageUpdater", webPageUpdater, 4)

.shuffleGrouping("articleExtraction", "webpage")

.shuffleGrouping("mediaExtraction", "webpage");

builder.setBolt("textIndexer", textIndexer, 1).shuffleGrouping("articleExtraction", "webpage");

job.setMapperClass(TokenizerMapper.class);

builder.setBolt("mediaupdater", mediaUpdater, 1)

.shuffleGrouping("articleExtraction", "media")

.shuffleGrouping("mediaExtraction", "media");

job.setReducerClass(IntSumReducer.class);

builder.setBolt("mediatextindexer", mediaTextIndexer, 1)

.shuffleGrouping("articleExtraction", "media")

.shuffleGrouping("mediaExtraction", "media");

StormTopology topology = builder.createTopology();

return topology;

}

}

### OUTPUT: console

===== Righe notevoli rilevate =================

Spouts 1

Bolts: 8

=== Riconosciuta struttura storm ================

setSpout("wpSpout",wpSpout,1)

aggiunto spout:wpSpout

aggiunto bolt:WpDeserializer

aggiunto bolt:expander

aggiunto bolt:articleExtraction

aggiunto bolt:mediaExtraction

aggiunto bolt:webPageUpdater

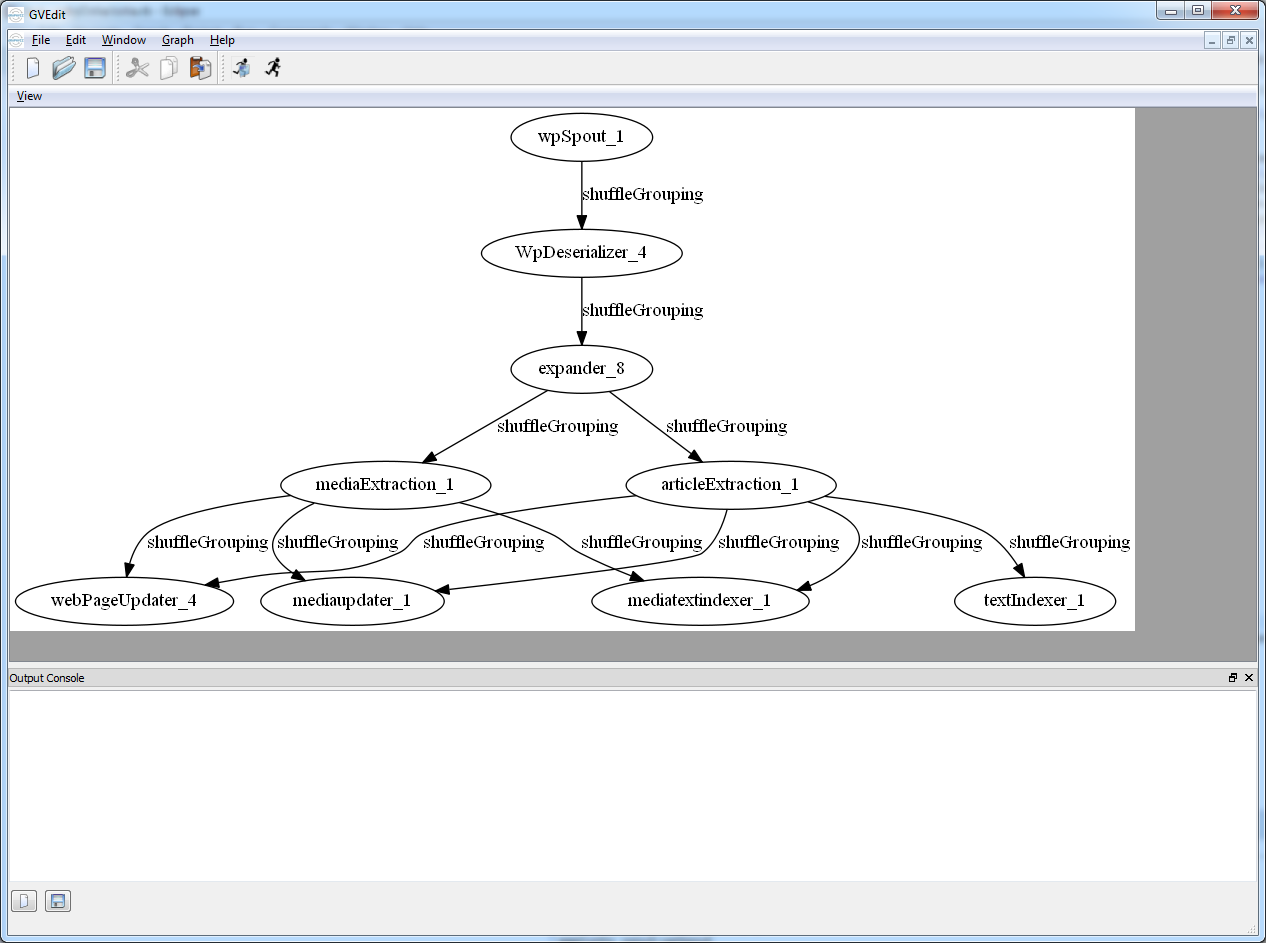
aggiunto bolt:textIndexer

aggiunto bolt:mediaupdater

aggiunto bolt:mediatextindexer

=== Done =======================================

### OUTPUT: file FocusedCrawler.dot



### OUTPUT: file FocusedCrawler.json

