Grep

Grep Example

Grep example extracts matching strings from text files and counts how many time they occured.

To run the example, type the following command:

```
bin/hadoop org.apache.hadoop.examples.Grep <indir> <outdir> <regex> [<group>]
```

The command works different than the Unix `grep` call: it doesn't display the complete matching line, but only the matching string, so in order to display lines matching "foo", use `.*foo.*` as a regular expression.

The program runs two map/reduce jobs in sequence. The first job counts how many times a matching string occured and the second job sorts matching strings by their frequency and stores the output in a single output file.

Each mapper of the first job takes a line as input and matches the user-provided regular expression against the line. It extracts all matching strings and emits (matching string, 1) pairs. Each reducer sums the frequencies of each matching string. The output is sequence files contaning the matching string and count. The reduce phase is optimized by running a combiner that sums the frequency of strings from local map output. As a result it reduces the amount of data that needs to be shipped to a reduce task.

The second job takes the output of the first job as input. The mapper is an inverse map, while the reducer is an indentity reducer. The number of reducers is one, so the output is stored in one file, and it is sorted by the count in a descending order. The output file is text, each line of which contains count and a matching string.

The example also demonstrates how to pass a command-line parameter to a mapper or a reducer. This is done by adding (key, value) pairs to the job's configuration before the job is submitted. Map or reduce tasks are able to access the value by getting it from the job's configuration in the method *configure*.

Grep supports generic options: see DevelopmentCommandLineOptions