MAPREDUCE

Dr. Padraig Corcoran

Big Data

- Modern data-mining applications require processing of large amounts of data quickly.
  - Ranking of web pages by importance.
  - Searching friends in social networking sites.
- Modern computing systems use *computing clusters* as opposed to *supercomputers*.
- Recall, a computer cluster consists of a collection of compute nodes.

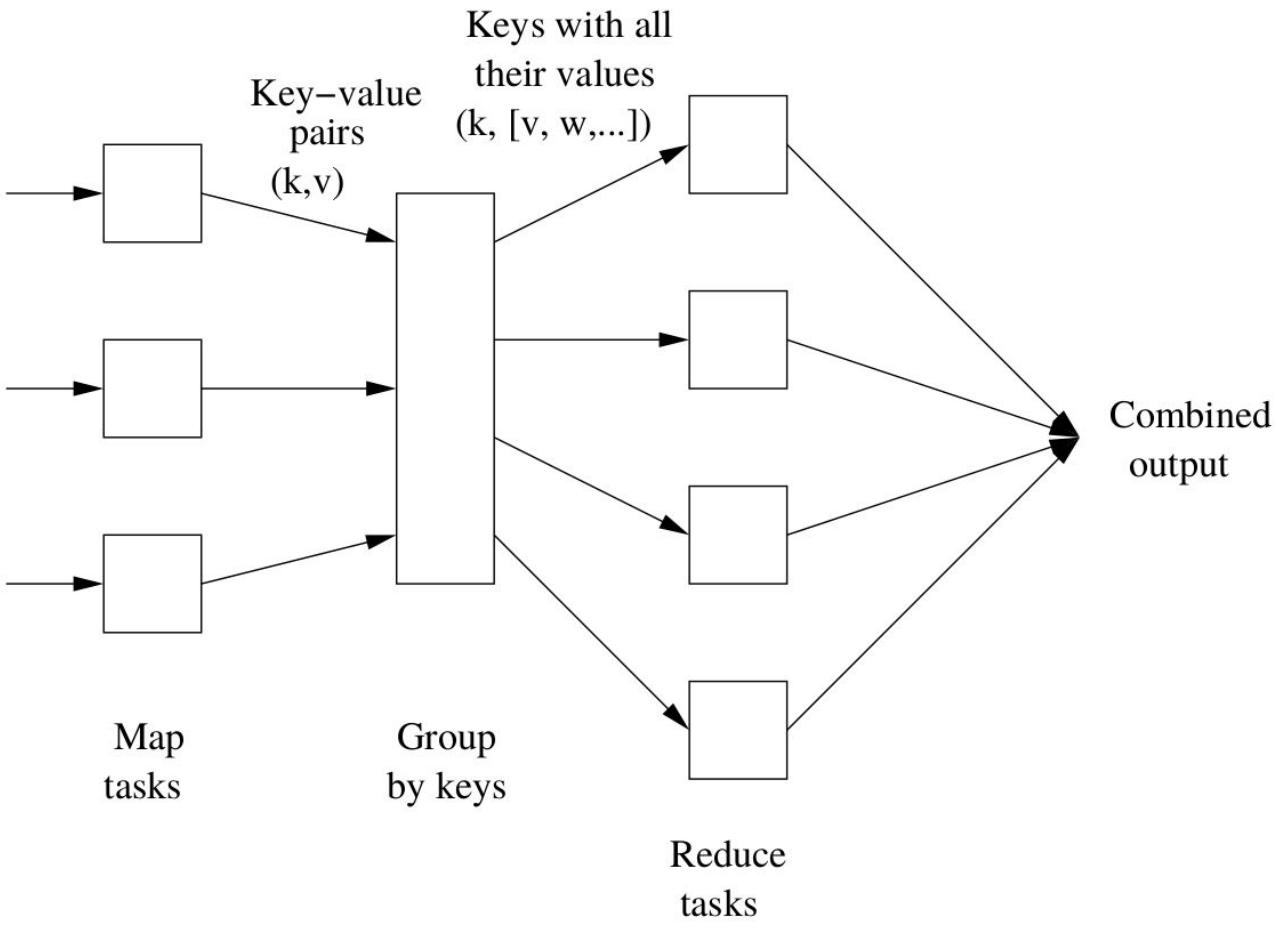
MapReduce

- A *programming model* for processing large datasets that is tolerant to computation failures.
- MapReduce programs run on a computer cluster.
- Developed by J. Dean and S. Ghemawat of Google (see paper entitled “MapReduce: Simplified Data Processing on Large Clusters”).
- Hadoop is an open-source implementation developed by the Apache Foundation.

- Many computations are conceptual straightforward.
- When data is large computation must be distributed.
- This leads to issues of how to parallelize computation, distribute data, and handle failures.
- MapReduce allows us to express these simple computations but hides the messy details of:
  - parallelization
  - fault-tolerance
  - data distribution
  - load balancing

Programming Model

- User specifies the *map* and *reduce* functions.
- Map function applied to each logical "record" in the input.
- Returns set of intermediate key/value pairs e.g. (hello, 1).
- All intermediate values associated with the same intermediate key are grouped.
- Reduce function applied to all the values that share the same key.
- Returns set of key/value pairs e.g. (hello, 4).



Schematic of a MapReduce computation

Example

- Problem of counting the number of occurrences of each word in a large collection of documents.
- **Pseudocode** for map and reduce functions in solution.

```
function map(String name, String document):
        // name: document name
        // document: document contents
        for each word w in document:
                emit (w, 1)

function reduce(String word, Iterator partialCounts):
        // word: a word
        // partialCounts: a list of aggregated partial counts
        sum = 0
        for each pc in partialCounts:
                sum += pc
        emit (word, sum)
```

- Consider the 4 documents each containing words.
  - Document 1 – hello
  - Document 2 – goodbye
  - Document 3 – tree house
  - Document 4 – hello
- Map applied to each document; the following key/value pairs are produced
  - Document 1 – (hello, 1)
  - Document 2 – (goodbye, 1)
  - Document 3 – (tree, 1), (house, 1)
  - Document 4 – (hello, 1)

- Grouping by key produces
  - (hello, [1,1])
  - (goodbye, [1])
  - (tree, [1])
  - (house, [1])

- Reduce applied to each of these groups produces
  - (hello, 2)
  - (goodbye, 1)
  - (tree, 1)
  - (house, 1)

MapReduce in Python with MrJob

- Write Python programs that run on Hadoop.
- Test your code locally without installing Hadoop.

Counting word occurrences in document

- Here is the python code (MRWordCounter.py):

```
from mrjob.job import MRJob

class MRWordCounter(MRJob):
        def mapper(self, key, line):
                for word in line.split():
                        yield word, 1

        def reducer(self, word, occurrences):
                yield word, sum(occurrences)

if __name__ == '__main__':
        MRWordCounter.run()
```

- To run the program enter the command

  >>python MRWordCounter.py document.txt

Today's lab

- Anonymous web browsing data from Microsoft.
- Contains information in CSV (comma-separated values).
- “This dataset records which areas (Vroots) of www.microsoft.com each user visited in a one-week timeframe in February 1998.”
- Each line contains a different record type.

Attribute lines:

- Begin with an A and describe each web site (Vroots).

- 'A,1277,1,"NetShow for PowerPoint","/stream"'

- 'A' marks this as an attribute line.
- '1277' is the attribute ID number for the Vroot.
- '1' may be ignored.
- '"NetShow for PowerPoint"' is the title of the Vroot.
- '"/stream"' is the URL relative to "http://www.microsoft.com".

Case and Vote Lines

- For each user (specified by an ID), there is a case line followed by zero or more vote lines.

- C,"10164",10164
  V,1277,1
  V,1009,1
  V,1052,1

- C,"10164",10164
  V,1277,1
  V,1009,1
  V,1052,1

- 'C' marks this as a case line, '10164' is the case ID number of a user.
- 'V' marks the vote lines for this case.
- '1277', 1009', 1052' are the attribute ID's of Vroots that a user visited.
- '1' may be ignored.

References

- Dean, Jeffrey, and Sanjay Ghemawat. MapReduce: simplified data processing on large clusters. Communications of the ACM 51.1 (2008): 107-113.
- MrJob Python documentation https://mrjob.readthedocs.io/en/latest/