HIGH PERFORMANCE COMPUTING OF SEISMIC DATA ON MAPREDUCE

Nikzad Babaii Rizvandi and Prof.Albert Y.Zomaya (Supervisor)
School of Information Technologies
FACULTY OF ENGINEERING & INFORMATION TECHNOLOGIES

AIMS

After an overview of forward/inverse Prestack Kirchhoff Time Migration (PKTM) algorithm, we will explain our proposed approach to fit this algorithm for running on Google's MapReduce framework. Toward the end, we will analyse the relation between MapReduce-based PKTM completion time and the number of map/reduce tasks on pseudo-distributed MapReduce mode.

MOTIVATION

- Seismic imaging is motivated by the business requirement to make best decisions to drill wells during petroleum exploration and production.
- Drilling each oil/gas well takes several tens of millions of dollars, so good seismic imaging notably decreases the risk of drilling a "dry hole".
- A typical seismic imaging job is computationally very expensive which needs to process terabytes of data and requires Gflop-months of computation prior to being interpreted by the experts.
- Pre-Stack Kirchhoff Time Migration (PKTM) is the most popular seismic imaging approach in oil/gas industry.

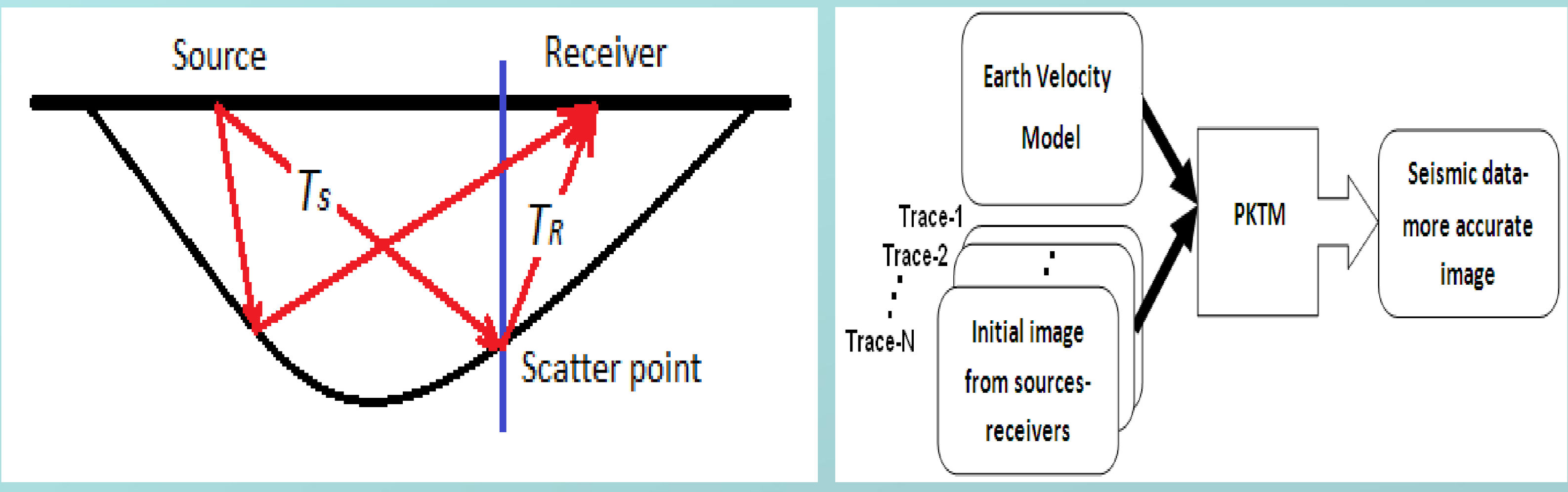

Figure 1. (left) migration curve (right) PKTM in seismic data processing

MAPREDUCE

- A parallel programming model introduced by Google in 2004.
- Widely used in Big Data analysis including social network graphs (e.g., LinkedIn, Facebook), search engine (e.g., Google, Yahoo!), finance (e.g., fraud detection using machine learning techniques).
- Unlimited MapReduce resources on cloud (e.g., Amazon Elastic MapReduce) as pay by hour.
- Highly interested by oil and gas industries to process seismic imaging.
- Challenge: Develop algorithms on MapReduce needs to follow specific programming model

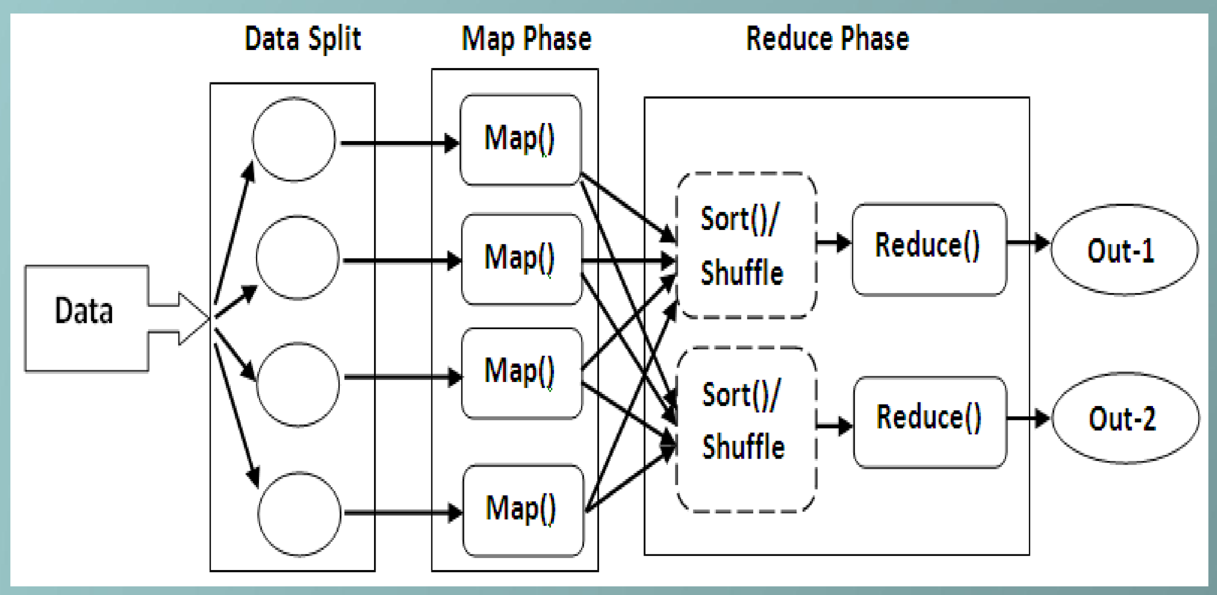

Figure 2. MapReduce programming model

OUR MAPREDUCE IMPLEMENTATION OF PKTM

STEP 1- Transfer data to Hadoop MapReduce File System (HDFS)

STEP 2- Split data in HDFS

- Split data into a set of traces.
- <key,value> is trace index and trace data.

STEP 3- Apply Map function on <keys,values>

- Figure 4 shows Map function in Python
- Results is a set of <intermediate keys, values>

STEP 4- Shuffle <intermediate keys, values> through network

- Each reducer receives all values with the same <intermediate keys>

STEP 5- apply reduce function

- Sum the values of all occurrences of each data (Data[i][j]).
- Then forms the final image for the experiment.

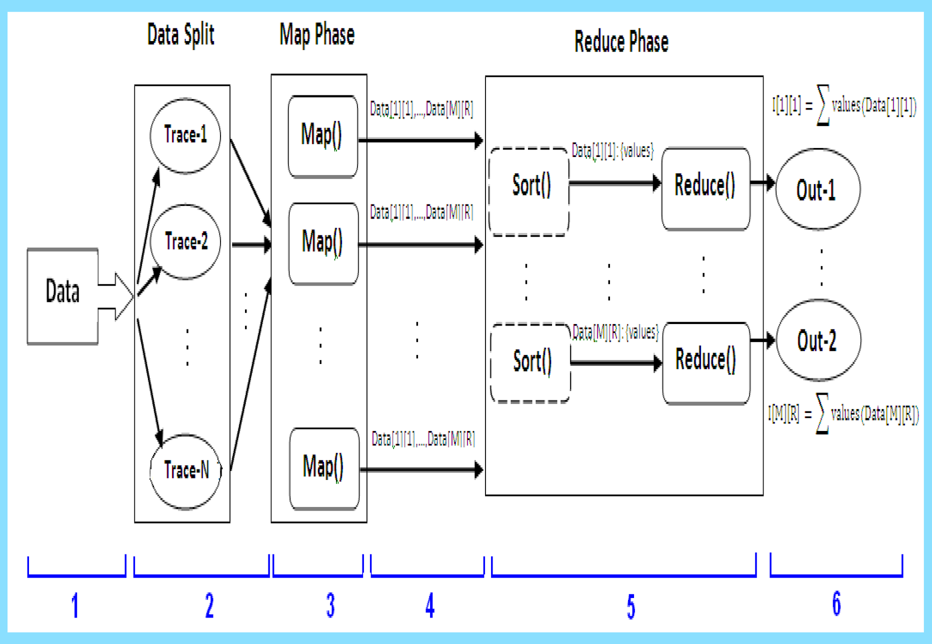

Figure 3. Our MapReduce design for Forward PKTM

```
def Calculate PKTM One Trace():
    # calculate PKTM for one trace
    ix = 0
    while ix<nx:    % # x-axis samples
        x = x0+dx*ix
        iy=0
        while iy<ny:   % # y-axis samples
            y=y0+dy*iy
            iz=0
            hs=(x-y)/velhalf
            while iz<nz:      % # z-axis samples
                z=z0+dz*iz
                t=sqrt(z*z+hs*hs)
                it=floor((t-t0+0.5)/dt)
                if (it<nt):
                    if (adj==0):
                        data[it][iy]=data[it][iy]+model[iz][ix]
                    else:
                        model[iz][ix]=model[iz][ix]+data[it][iy]
                iz=iz+1
             iy=iy+1
        ix=ix+1
```

Figure 4. Map function

**** More information can be found at [2]**

EXPERIMENTAL RESULTS

Experimental setting

- Pseudo-distributed Hadoop MapReduce.
- Run Hadoop version 0.20.2
- Synthetic velocity model → BitMap with 1500*1500 pixels.
- *Velhalf=0.1, dt=8.0, dx=4.0*

Results

- Figure 5 shows relation between number of Map/reduce tasks on completion time.
- Figure 6 shows the result of applying our implementation on synthetic data. 0.20.2

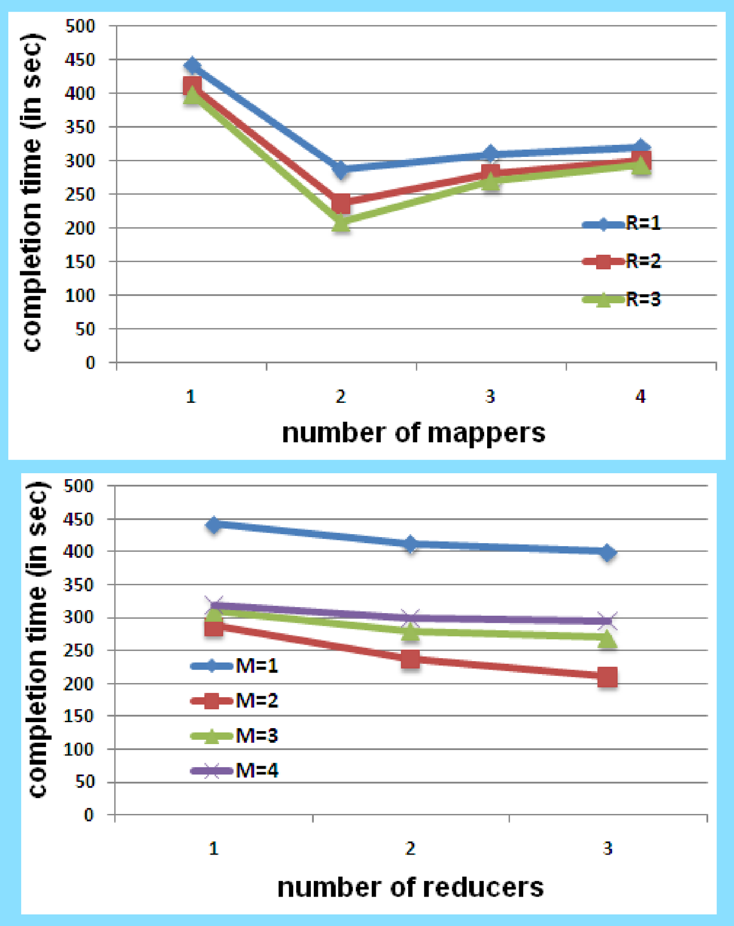

Figure 5. The relation between the completion time and MapReduce configuration parameters

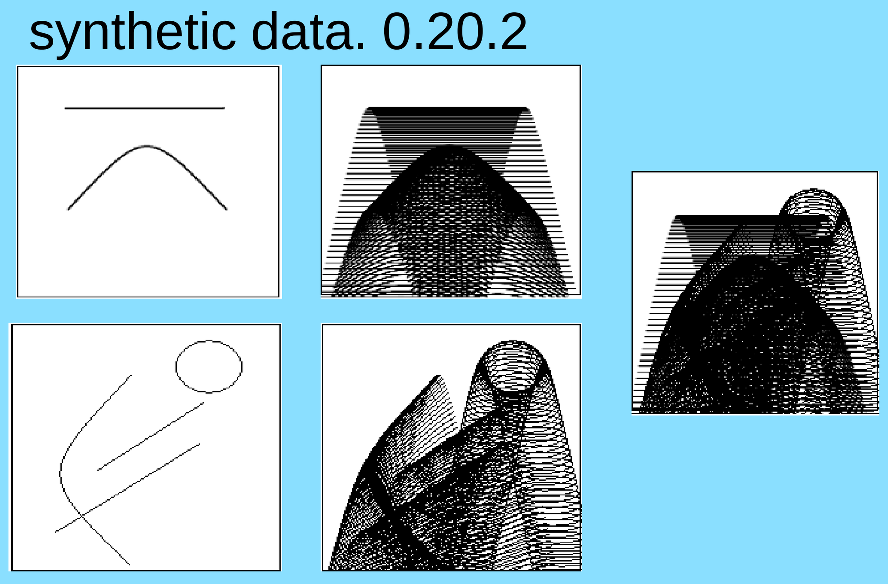

Figure 6. The earth velocity models to test forward PKTM algorithm

CONCLUSION

- The importance of high performance computing on oil/gas industries was explained.
- A MapReduce framework for PKTM has been proposed.
- The influence of choosing MapReduce configuration parameters on completion time of PKTM was studied.

REFERENCES

[1] H. Yan, *et al.*, "DECF: A Coarse-Grained Data-Parallel Programming Framework for Seismic Processing", CSSE 2008
[2] N.B.Rizvandi, et al., "MapReduce Implementation of Prestack Kirchhoff Time Migration (PKTM) on Seismic Data", PDCAT 2011
[3] D. Hengchan, "Parallel Processing of Presteak Kirchhoff Time Migration on a PC Cluster," *Computers & Geosciences,* vol. 31, pp. 891-899, 2005.
[4] Peiyuan Sun and Xiaohua Shi, "An OpenCL Approach of Prestack. Kirchhoff Time Migration Algorithm on General Purpose GPU", PDCAT 2012