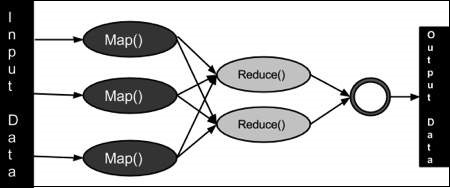
1. Apa itu MapReduce?

MapReduce adalah teknik pemrosesan dan model program untuk komputasi terdistribusi berbasis java. Algoritma MapReduce berisi dua tugas penting, yaitu Map dan Reduce. Peta mengambil satu set data dan mengubahnya menjadi set data lain, di mana elemen individu dipecah menjadi tupel (pasangan kunci / nilai). Kedua, mengurangi tugas, yang mengambil output dari peta sebagai input dan menggabungkan data tupel ke dalam tupel yang lebih kecil. Seperti urutan nama MapReduce menyiratkan, mengurangi tugas selalu dilakukan setelah pekerjaan peta.Keuntungan utama MapReduce adalah mudahnya skala pemrosesan data melalui beberapa node komputasi. Di bawah model MapReduce, primitif pemrosesan data disebut pemetaan dan reduksi. Mengurai aplikasi pemrosesan data ke dalam pemetaan dan reduksi terkadang tidak penting. Tapi, begitu kita menulis aplikasi dalam bentuk MapReduce, menskalakan aplikasi untuk menjalankan lebih dari ratusan, ribuan, atau bahkan puluhan ribu mesin dalam sebuah cluster hanyalah perubahan konfigurasi. Skalabilitas sederhana ini telah menarik banyak programmer untuk menggunakan model MapReduce.  
2. Algoritma

Secara umum paradigma MapReduce didasarkan pada pengiriman komputer ke tempat data berada!Program MapReduce dijalankan dalam tiga tahap, yaitu tahap peta, tahap acak, dan tahap pengurangan.Tahap peta - Pekerjaan peta atau mapper adalah memproses data input. Umumnya data input dalam bentuk file atau direktori dan disimpan dalam sistem file Hadoop (HDFS). File input diteruskan ke fungsi mapper baris demi baris. Mapper memproses data dan membuat beberapa potongan kecil data.Reduce stage - Tahap ini adalah kombinasi dari tahap Shuffle dan tahap Reduce. Tugas Reducer adalah memproses data yang berasal dari mapper. Setelah diproses, ini menghasilkan serangkaian output baru, yang akan disimpan dalam HDFS.Selama pekerjaan MapReduce, Hadoop mengirimkan tugas-tugas Map dan Reduce ke server yang sesuai di cluster.Kerangka kerja ini mengelola semua detail dari penyaluran data seperti menerbitkan tugas, memverifikasi penyelesaian tugas, dan menyalin data di sekitar cluster antara node.  


Contoh Program Diberikan di bawah ini adalah program untuk sampel data menggunakan kerangka kerja MapReduce.

package hadoop;

import java.util.\*;

import java.io.IOException;

import java.io.IOException;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.conf.\*;

import org.apache.hadoop.io.\*;

import org.apache.hadoop.mapred.\*;

import org.apache.hadoop.util.\*;

public class ProcessUnits {

//Mapper class

public static class E\_EMapper extends MapReduceBase implements

Mapper<LongWritable ,/\*Input key Type \*/

Text, /\*Input value Type\*/

Text, /\*Output key Type\*/

IntWritable> /\*Output value Type\*/

{

//Map function

public void map(LongWritable key, Text value,

OutputCollector<Text, IntWritable> output,

Reporter reporter) throws IOException {

String line = value.toString();

String lasttoken = null;

StringTokenizer s = new StringTokenizer(line,"\t");

String year = s.nextToken();

while(s.hasMoreTokens()) {

lasttoken = s.nextToken();

}

int avgprice = Integer.parseInt(lasttoken);

output.collect(new Text(year), new IntWritable(avgprice));

}

}

//Reducer class

public static class E\_EReduce extends MapReduceBase implements Reducer< Text, IntWritable, Text, IntWritable > {

//Reduce function

public void reduce( Text key, Iterator <IntWritable> values,

OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {

int maxavg = 30;

int val = Integer.MIN\_VALUE;

while (values.hasNext()) {

if((val = values.next().get())>maxavg) {

output.collect(key, new IntWritable(val));

}

}

}

}

//Main function

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

JobConf conf = new JobConf(ProcessUnits.class);

conf.setJobName("max\_eletricityunits");

conf.setOutputKeyClass(Text.class);

conf.setOutputValueClass(IntWritable.class);

conf.setMapperClass(E\_EMapper.class);

conf.setCombinerClass(E\_EReduce.class);

conf.setReducerClass(E\_EReduce.class);

conf.setInputFormat(TextInputFormat.class);

conf.setOutputFormat(TextOutputFormat.class);

FileInputFormat.setInputPaths(conf, new Path(args[0]));

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

JobClient.runJob(conf);

}

}

Contoh script: Kompilasi dan Eksekusi Program Unit Proses

Perintah berikut adalah membuat direktori untuk menyimpan kelas java yang dikompilasi.  


Langkah berikut :

Unduh Hadoop-core-1.2.1.jar, yang digunakan untuk mengkompilasi dan menjalankan program MapReduce. Kunjungi tautan berikut ini untuk mengunduh toples. Mari kita asumsikan folder yang diunduh adalah / home / hadoop /.

Langkah 3

Perintah berikut digunakan untuk mengkompilasi program ProcessUnits.java dan membuat toples untuk program tersebut.

