ΠΡΟΧΩΡΗΜΕΝΑ ΘΕΜΑΤΑ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

Εξαμηνιαία Εργασία

Εισαγωγή στο MapReduce

Παρακάτω ακολουθεί ψευδοκώδικας-mapreduce για κάθε κομμάτι της εργασίας

1)α) map(key, value): // key:
Start_Hour; value: Duration
emit(Start_Hour, (Duration, 1))

reduce(Start_Hour, List<(Duration, 1)>): For ever e in List

totalDuration += e.Duration Counter += 1
**emit(Start_Hour,
totalDuration/Counter)**

β) map(key, value): // key: route_id;
value: vendor_id,cost
emit(vendor_id, cost)

reduce(vendor_id, List(cost): Max_c = Max(List<cost>)
emit(vendor_id, Max_c) (ο κάθε reducor παράγει ένα μέγιστο και κρατιέται το μεγαλύτερο όλων)

3) map1(key, value): // key: Node;

```
value: OutBoundLink_Node emit(Node,
OutBoundLink_Node)

reduce1(Node,
List<OutBoundLink_Node>): emit(Node,
List<OutBoundLink_Node>)


map2(Node, List<OutBoundLink_Node>, rank):
For every out_bound_node in List
    emit( (one of the outbound)node,
    contribs)

reduce2(node , List<contribs>):
For every e in List
    totalContribs += e.contribs
emit(node, totalContribs)
```

4)

```
mapA(key, value): // key:
line; value: column,value
emit(column, (line,value))

reduceA(column, List<line,value>):
emit(column, List<line,value>)

mapB(key, value): // key: line;
value: column,value emit(line,
(column,value))

reduceB(line, List<column,value>):
emit(line, List<column,value>)
```

```
mapC(key,value): //key: lineB(or ColumnA) ,
value((lineA,valueA),(columnB,valueB))
emit((lineA,columnB),value1*value2)

reduceC((i,j),
List<value>): For every e
in List
 Pij += e.value emit((i,j), Pij)
          ( or ( i ,j, Pij ) )
```