

INFO-H-415 : Advanced Database

NoSQL Databases (2)

Lecturer : Esteban Zimanyi
Teaching-Assistants : François Picalausa, Serge Boucher
<http://cs.ulb.ac.be/public/teaching/infoh415>

Academic Year 2010–2011

Map-Reduce

Following is a database structure for a message board:

```
users: [ {
  nickname: string,
  email: string
} ]
threads: [ {
  topic: string,
  board: string,
  size: integer,
  messages: [ {
    author: string (refers to an user email),
    content: string
  } ]
} ]
```

Write the following queries using the map reduce paradigm:

- Give for each board its number of threads and its number of messages.
- Give the authors, along with the number of messages that the author has posted.
- Give for each board the number of distinct authors that have posted in those boards (hint: use two successive map reduce operations).

Solutions

```
var silent = false;

function myprint(ex, query) {
  if (silent) return;

  print("\nResultat de l'exercice " + ex + "\n");
  if (query.forEach) {
    query.forEach(function (q) {
      print(tojson(q));
    });
  }
  else print(tojson(query));
}

function mapreduce1() {
  var map = function () {
    emit(this.board, {board: this.board, msgs: this.size, threads: 1});
  };
  var reduce = function(key, values) {
    var result = { board: key, msgs: 0, threads: 0 };

    values.forEach(function (value) {
      result.msgs += value.msgs;
      result.threads += value.threads;
    });

    return result;
  };
  return db.threads.mapReduce(map, reduce);
}

function mapreduce2() {
  var map = function () {
    this.messages.forEach( function (m) {
      emit(m.author, {author: m.author, count: 1});
    })
  };
  var reduce = function(key, values) {
    var result = {author: key, count: 0};

    values.forEach( function(value) {
      result.count += value.count;
    });

    return result;
  };
  return db.threads.mapReduce(map, reduce);
}

function mapreduce3() {
  /* Find unique (board, author) pairs */
  var map1 = function() {
    var vboard = this.board;
```

```

        this.messages.forEach( function(m) {
            emit({ board: vboard, author: m.author },
                { board: vboard, author: m.author });
        });
    };
    var reduce1 = function(key, values) {
        return key;
    };

    /* Count authors per board */
    var map2 = function () {
        emit(this.value.board, {board: this.value.board, count: 1});
    };
    var reduce2 = function(key, values) {
        var result = {board: key, count: 0};

        values.forEach( function(value) {
            result.count += value.count;
        });

        return result;
    };

    return db[db.threads.mapReduce(map1, reduce1).result].mapReduce(map2, reduce2);
}

myprint(1, db[mapreduce1().result].find());
myprint(2, db[mapreduce2().result].find());
myprint(3, db[mapreduce3().result].find());

```