Journées des Développeurs JDEV2017

du 4 au 7 juin 2017 à Marseille

T2 – Ingénierie et web des données

La Plateforme

« The NetWork of Researchers»

Pour moissonner, analyser, visualiser et valoriser les données de la production scientifique sous forme de graphe

Version : 1.1 (05/07/2017)

A-Phat LY (CNRS/LLL) & Yvan STROPPA (CNRS/Paul Painlevé)

Table des matières

1 SITUATION

1.1 Objectif

L'objectif du projet « The NetWork of Researchers » est l'exploitation des données structurées ou non du Big Data en s'appuyant sur les sources OpenData telles que Hal-SHS, ArXiv, Google Scholar …afin de reconstituer, de visualiser et de caractériser les réseaux des chercheurs.
Pour atteindre cet objectif, il est nécessaire de passer par les différentes étapes :

- Moissonner et récolter (OAI/PMH ou via du scraping) les informations des données structurées ou non dans les domaines de la recherche scientifique.
- Exploiter et traiter les données s'appuyant sur une infrastructure distribuée (flexible et scalable) robuste. Appliquer des méthodes de classifications et des statistiques.
- Permettre d'exploiter et de visualiser sous différentes représentations graphiques les résultats à partir d'un navigateur.

1.2 Description

La plateforme « The NetWork of Researchers » permet de visualiser et de valoriser les données de la production scientifiques issues du Web (publications) comme HAL-SHS et ArXiv (actuellement environ **3 000 000** publications). Elle permet de formaliser à partir de publications/Conférences/books … des graphes/networks illustrant les collaborations temporelles entre les chercheurs en associant ou pas les réseaux de citations. De ce fait, à partir des productions scientifiques contenues dans les différentes archives collectées, la plateforme va reconstruire les réseaux ainsi constitués par les coauteurs.

Le portail permet de distinguer les productions scientifiques en tenant compte de leur statut (Journal, Working Paper, conférences…) et de mesurer le degré d'interactions entre les chercheurs. On pourra agréger ces relations entre chercheurs pour les porter au niveau de leur structure d'affiliation (d'appartenance), à la discipline et autre regroupement de type institutionnel. On pourra dans la mesure où l'on dispose d'informations complémentaires représenter dans ces graphes les relations interdisciplinaires.

La plateforme sera accompagnée de traitements statistiques, de modélisation et l'analyse des données, s'appuyant sur une infrastructure performante et robuste, permet de préparer les données afin de répondre aux requêtes et d'exécuter les calculs distribués (clusters) et de fournir des caractéristiques statistiques calculées sur les différents graphes. Les résultats obtenus couplés avec la plateforme vont permettre de visualiser en 2D le réseau des chercheurs et de leurs interactions avec les métadonnées qui les caractérisent (nom, prénom, affiliation, publication, co-auteur,…).

2 INFRASTRUCTURE DISTRIBUEE DE MONGODB

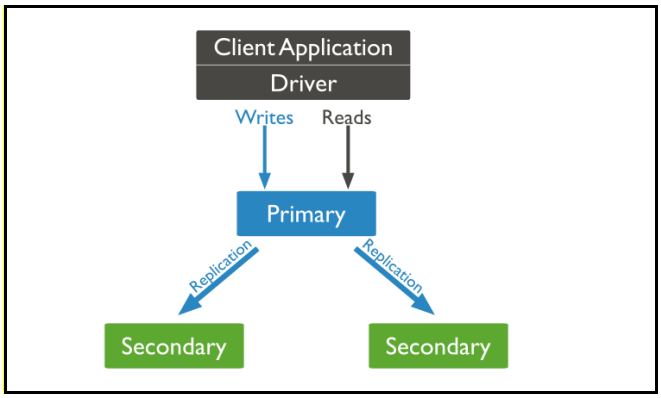
2.1 La réplication de MongoDB

Un cluster Mongo est composé au minimum quatre machines (4 VM) avec 1 maître (Primary) et trois esclaves (secondary). Si le maître tombe en panne, une élection est provoquée entre les slaves pour choisir celui qui deviendra le master de façon transparente.

Le cluster garantit la réplication des données en son sein. Pour activer la réplication, il suffit de mettre dans le fichier de configuration de mongodb dans la ligne *replication* et au-dessous «*replSetName: Nom_Replication* »

replication :
replSetName : clusterA

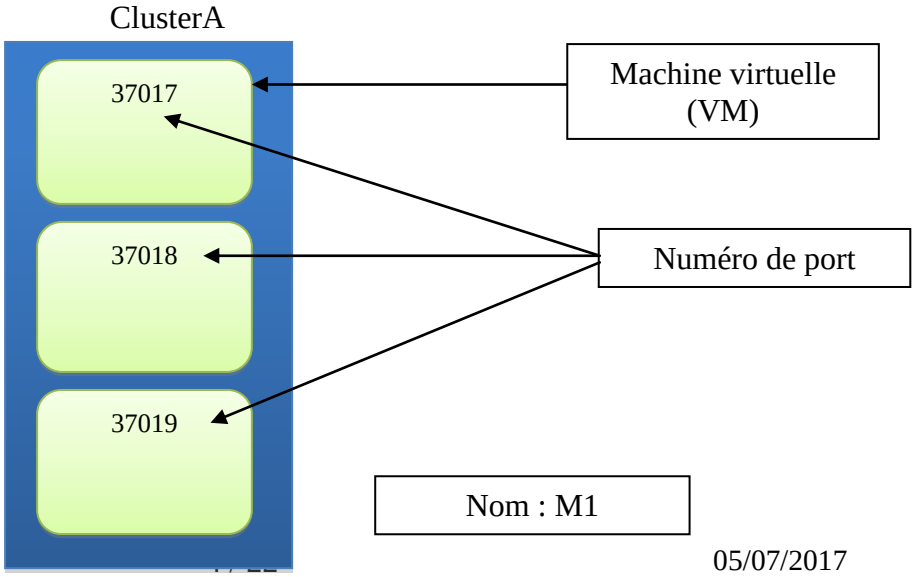
(voir exemple page 5 : **Instance_A1.conf**)



Réplication

2.2 Notion de Cluster

Montage d'un cluster



Tous les caractéristiques (numéro de port, la réplication, l'interface web, API Rest,…) de chaque instances de mongodb sont enregistrées dans un fichier de configuration dans un répertoire dédié (voir exemple page 5 : **Instance_A1.conf**).

Ensuite, il suffit de lancer la commande « mongo » pour lancer chaque instance de mongo comme ci-dessous :

Lancer les instances de mongodb:

```
mongod --f /home/jdev/infrastructure/clusterA/instanceA_1/mongod_instA1.conf
```

On effectuera la même procédure en copiant le fichier de configuration de **Instance_A1.conf en** l'InstanceA_2.conf et InstanceA_3.conf avec le bon numéro de port et mettre le bon chemin sur les répertoires. Ensuite il suffit de lancer les deux commandes suivantes :

```
mongod --f /home/jdev/infrastructure/clusterA/instanceA_2/mongod_instA2.conf
mongod --f /home/jdev/infrastructure/clusterA/instanceA_3/mongod_instA2.conf
```

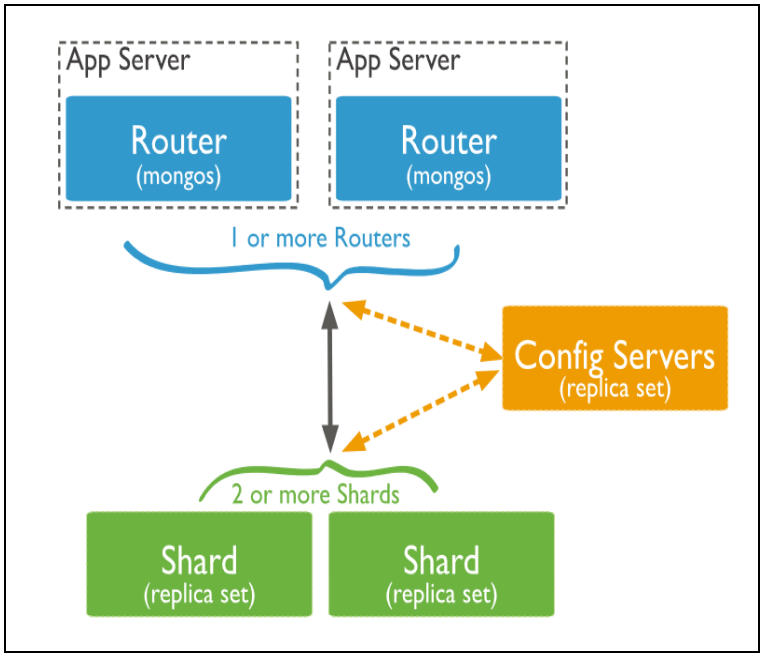
Instance_A1.conf

```
# where to write logging data.
systemLog:
  destination: file
  logAppend: true
  path: /home/jdev/clusterA/instanceA_1/journal/mongod.log

# Where and how to store data.
storage:
  dbPath: /home/jdev/clusterA/instanceA_1/data
  journal:
    enabled: true

# how the process runs
processManagement:
  fork: true
  pidFilePath: /home/jdev/clusterA/instanceA_1/mongod.pid  #

# network interfaces
net:
  port: 37017

http:
    enabled: true
     JSONPEnabled: true
     RESTInterfaceEnabled: true

replication:
  replSetName: "clusterA
```

Une fois que les 3 instances de mongo client sont lancées, on va les assembler pour former un cluster. M1 est le nom du VM. Si vous travailler sur plusieurs machines ou VM, il suffit de remplacer par le nom de votre machine (VM).

```
#mongo --port 37017
>rs.initiate()
>rs.add("M1:37018")
>rs.add("M1:37019")
>rs.status()
```

La commande rs.status() permet de voir que l'instance 37017 est devenu le maître et que les autres instances avec les ports 37018 et 37019 sont devenus « secondaires ».On fera de même pour un second cluster.



L'infrastructure de MongoDB se compose des éléments suivants:

- **Fragments** (Shards): chaque morceau contient un sous-ensemble des données tranchées. Chaque fragment peut être déployé en tant que jeu de répliques.

- **Mongos**: le mongo agit comme un routeur de requête, fournissant une interface entre les applications clientes et le cluster fragmenté.

- **Serveurs de configuration**: les serveurs de configuration stockent les métadonnées et les paramètres de configuration du cluster. Depuis MongoDB 3.4, les serveurs de configuration doivent être déployés en tant que jeu de réplications.

- **Fichiers de configurations : l**e fichier de configuration contient tous les métadonnées des clusters

2.3 Notion de Collection

Le terme collection est issu du monde NoSQL (pour Not Only SQL) et correspond par analogie à la notion de table dans les bases de données relationnelles (MySQL, PosgreSQL, Microsoft Access, Oracle, etc.). Une collection permet de stocker des documents, notion analogue à celle de ligne.

En comparant le schéma de structure d'une base de MongoDB à celui d'une base de données relationnelle comme SQL, on peut faire le parallèle suivant:

Base de données relationnelle (SQL)	Base de données NoSQL (MongoDB)
Table	Collection
Ligne	Document
Index	Index
Jointures	Données embarquées dans le document

Exemple de 2 documents au format JSON (key, valeur):

{_id: "Her", acteurs : [{nom:"Johansson", prenom:"Scarlett"}, {nom:"Phoenix", prenom:"Joaquim"}]}

{_id: "Avengers", acteurs : [{nom:"Johansson", prenom:"Scarlett"}]}

Dans ces exemples, on peut voir que les acteurs sont dupliqués. Dans le monde NoSQL, on n'hésite pas à « dénormaliser » le schéma de la base de données pour favoriser les performances à la lecture.

2.4 Notion de SHARDING

Après la réplication, un autre avantage des systèmes distribués est de pouvoir répartir les données entre les différents serveurs. C'est ce qu'on appelle le *partitionnement* ou le **Sharding** dans le cadre de MongoDB.

Si l'on a une grande collection de données, il peut être préférable de répartir cette collection sur plusieurs serveurs. Ainsi, lorsque l'on interrogera la base, les serveurs pourront effectuer la même requête en parallèle sur une quantité de données réduite.

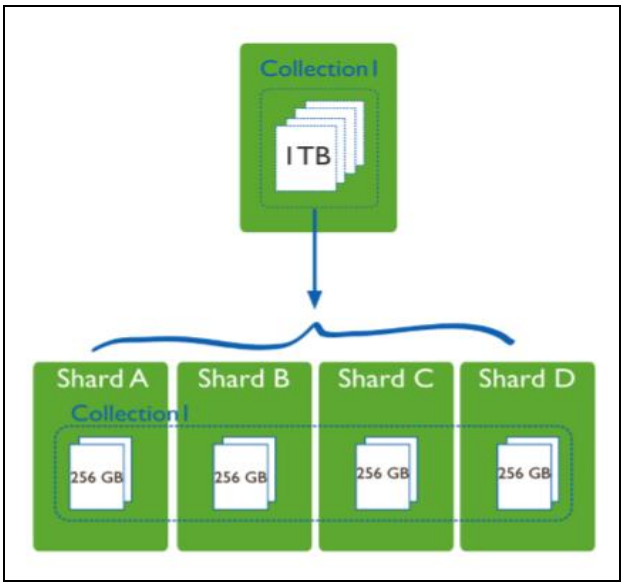


Schéma sur le Sharding ou le partitionnement

2.5 Les Outils

RockMongo: il s'agit d'une application PHP disponible sur le site http://www.rockmongo.com et qui nécessite un serveur Apache/PHP. De plus, il est nécessaire d'installer l'extension PHP de connexion à MongoDB.

RoboMongo: il s'agit d'un client graphique disponible pour toutes les plate-formes sur le site robomongo.org. L'installation est très simple, l'outil semble assez complet et en évolution rapide. L'interface RoboMongo montre l'interface en action

On va utiliser RoboMongo.

3 EXPLOITATION DES DONNEES DES COLLECTIONS

3.1 Les sources de données

- HAL-SHS
- ArXiv

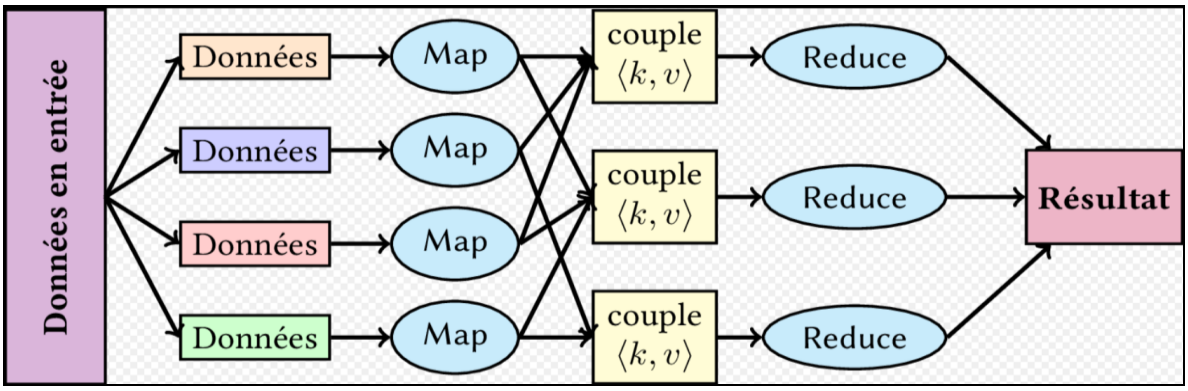
Les données sont récupérées par le protocole OAI-PMH pour environ ***3 000 000*** *de publications.*

3.2 Introduction sur le MapReduce

Définition :
MapReduce est un patron d'architecture inventé par Google pour effectuer des calculs distribués sur des volumes de données très volumineux.

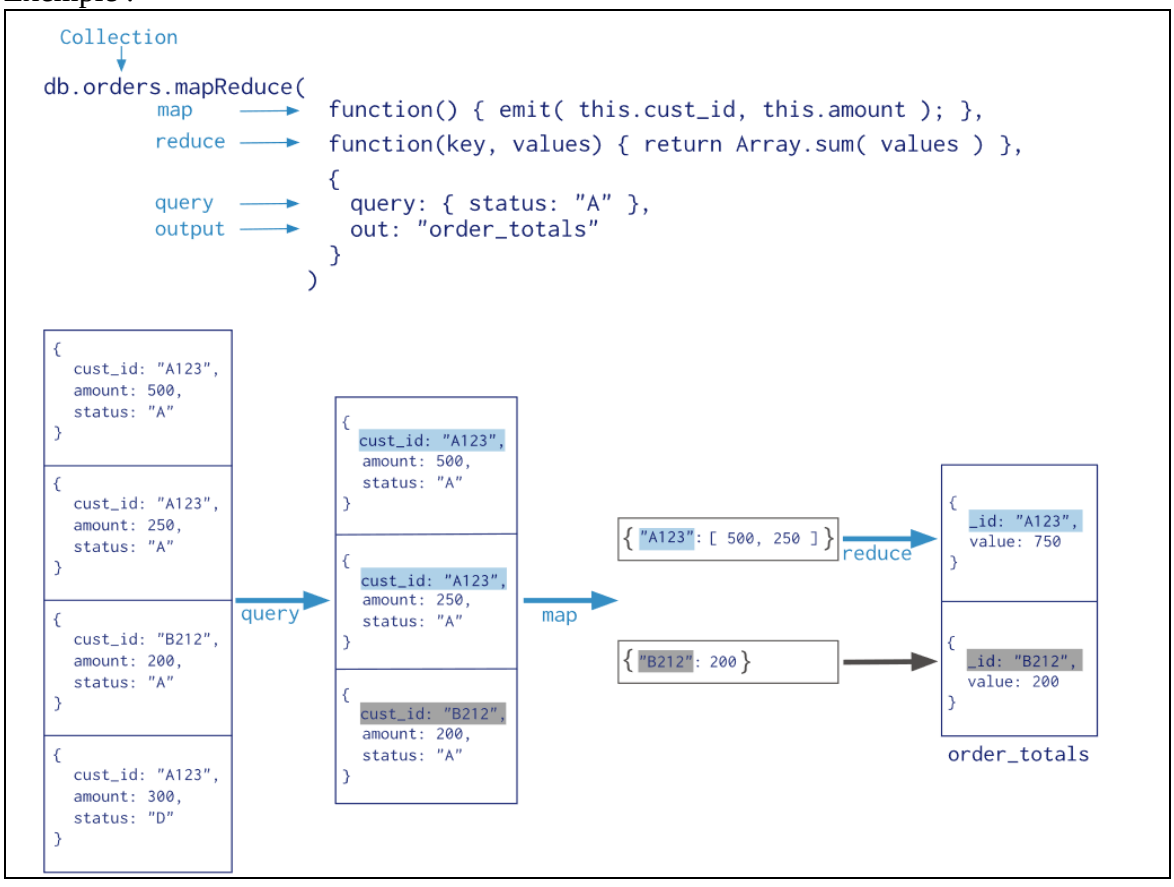
MapReduce permet de manipuler de grandes quantités de données en les distribuant dans un cluster de machine pour être traitées.

Principe de Map et de Reduce



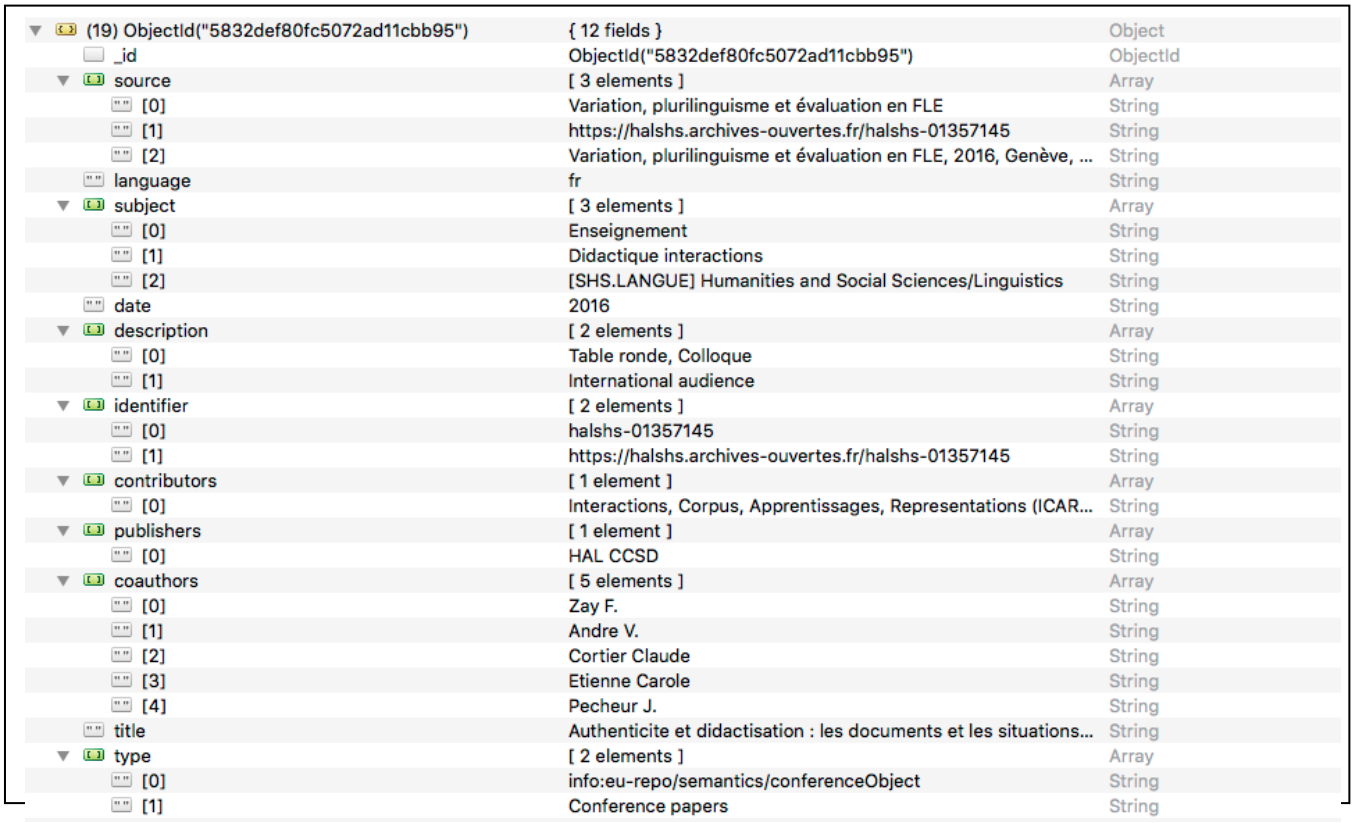
On voit bien sur le schéma ci-dessus que les données initiales sont « splittées » (coupées) en morceaux. Ensuite, on applique la fonction « Map » sur les données coupées sous forme de couple (Key, value) : Key est la clé, Value est la valeur. On applique par la suite la fonction « Reduce » pour effectuer des calculs pour une clé donnée. Tous les résultats de chaque « Reduce » vont être agrégés et regrouper dans un seul résultat final.

Exemple :



3.3 Structure des données dans MongoDB

Soit la structure de données suivante stockée dans une instance de mongodb.



(19) ObjectId("5832def80fc5072ad11cbb95")	{ 12 fields }	Object
_id	ObjectId("5832def80fc5072ad11cbb95")	ObjectId
source	[3 elements]	Array
[0]	Variation, plurilinguisme et évaluation en FLE	String
[1]	https://halshs.archives-ouvertes.fr/halshs-01357145	String
[2]	Variation, plurilinguisme et évaluation en FLE, 2016, Genève, ...	String
language	fr	String
subject	[3 elements]	Array
[0]	Enseignement	String
[1]	Didactique interactions	String
[2]	[SHS.LANGUE] Humanities and Social Sciences/Linguistics	String
date	2016	String
description	[2 elements]	Array
[0]	Table ronde, Colloque	String
[1]	International audience	String
identifier	[2 elements]	Array
[0]	halshs-01357145	String
[1]	https://halshs.archives-ouvertes.fr/halshs-01357145	String
contributors	[1 element]	Array
[0]	Interactions, Corpus, Apprentissages, Representations (ICAR...	String
publishers	[1 element]	Array
[0]	HAL CCSD	String
coauthors	[5 elements]	Array
[0]	Zay F.	String
[1]	Andre V.	String
[2]	Cortier Claude	String
[3]	Etienne Carole	String
[4]	Pecheur J.	String
title	Authenticite et didactisation : les documents et les situations...	String
type	[2 elements]	Array
[0]	info:eu-repo/semantics/conferenceObject	String
[1]	Conference papers	String

3.4 Etude de cas par les exemples

Nous allons détailler les différentes possibilités qu'offrent mongodb dans la cadre du Map-Reduce

Types de cas :

Cas N°1	MAP	retourne une Key et Value
Cas N°2	MAP	retourne des Keys et Value
Cas N°3	MAP	retourne une Key et des Values
Cas N°4	MAP	retourne une Keys et Values
Cas N°5	MAP	

3.4.1 Cas 1 : Je souhaite calculer le nombre de publications par langue

(KEY, VALUE)

Effectuez un calcul du nombre de publications par langue

Fonction maplanguage

```
function() {
        var langue="Undefined";
        if (this.language) {
                langue=this.language;
        }
        emit(langue,{count:1});
}
```

Fonction reducedate

```
function(key,values) {
        var res={count:0};
        values.forEach (function (v) {res.count+=v.count;});
        return res;
        }
```

Pour vérifier au préalable dans la collection que le champs language est existant, sinon lors de l'exécution de la fonction map l'évaluation du champ ne peut se faire et la fonction mapReduce n'aboutit pas.

```
db.reduit.find({"language":{$exists: false}},{}).count();
```

```
Pour publications
        "timeMillis" : 5806,
        "counts" : {
                "input" : 336341,
                "emit" : 336341,
                "reduce" : 4347,
                "output" : 80
         },
Pour reduits
        "timeMillis" : 201,
        "counts" : {
                "input" : 10094,
                "emit" : 10094,
                "reduce" : 273,
                "output" : 29
        },
```

```
"results" : [
                {
                        "_id" : "/",
                        "value" : {
                                "count" : 1530
                        }
                },
                {
                        "_id" : "ar",
                        "value" : {
                                "count" : 5
                        }
                },
                {
                        "_id" : "az",
                        "value" : {
                                "count" : 1
                        }
                },
                {
                        "_id" : "ca",
                        "value" : {
                                "count" : 5
                        }
                },
etc ….
```

3.4.2 Cas 2 : Je souhaite calculer le nombre de publications par langue et par année

(KEYS, VALUE)

Effectuez un calcul du nombre de publications par langue et par année

Fonction map

```
function() {
        var nouv_date="";
        nouv_date=this.date.substr(0,4);
        emit({lang:this.language,date:nouv_date},{count:1});
}
```

Fonction reduce

```
function(key,values) {
        var res={count:0};
        values.forEach (function (v) {res.count+=v.count;});
        return res;
        }
```

Premier résultat par année et par langue

Pour reduits

```
"timeMillis" : 265,
        "counts" : {
                "input" : 10094,
                "emit" : 10094,
                "reduce" : 458,
                "output" : 249
        },
          "ok" : 1
```

Pour publications

```
        "timeMillis" : 8169,
        "counts" : {
                "input" : 336341,
                "emit" : 336341,
                "reduce" : 9388,
                "output" :  1064
        },
          "ok" : 1
```

```
                {
                        "_id" : {
                                "lang" : "ar",
                                "date" : "2015"
                        },
                        "value" : {
                                "count" : 1
                        }
                },
                {
                        "_id" : {
                                "lang" : "ar",
                                "date" : "2016"
                        },
                        "value" : {
                                "count" : 1
                        }
                },
```

3.4.3 Cas 3 : Je souhaite récupérer le nombre de publication par thématique

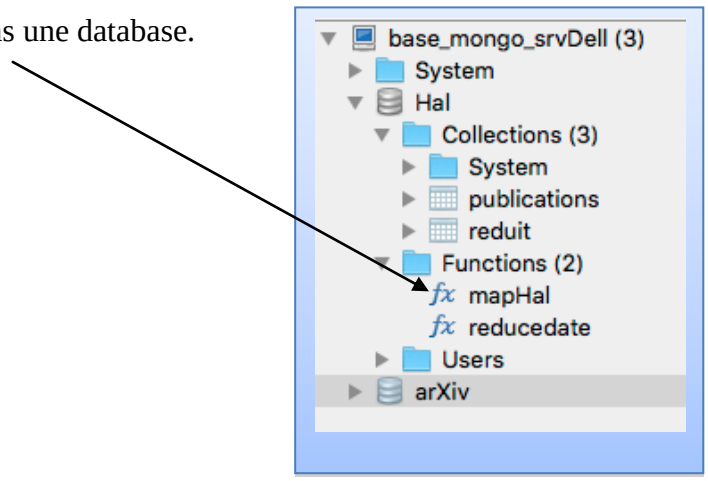
On souhaite récupérer le nombre de documents par thématique (SHS….)
Définition de la fonction mapHal

```
function() {
        var thematic="NotDefined";
        var autresThe=""; // permet de vérifier la présence du field  subject
        if(this.subject) {
                var premier=true;
                for (var i=0 ;i< this.subject.length; i++) {
                        if (this.subject[i].indexOf("SHS.") != -1) {
                                if (premier) {
                                        thematic=this.subject[i];
                                        premier= false;
                                } else {
                                        autresThe +=this.subject[i]+ "  ";
                                }
                        }
                }
        }
        emit(thematic,{count:1, divers:autresThe});
}
```

Définition de la fonction reduce: reducedate

```
function(key,values) {
        var res={count:0};
        values.forEach (function (v) {res.count+=v.count;});
        return res;
        }
```

On définit les fonctions dans une database.



Utilisation de ces deux fonctions dans une instance de mongo.

```
use Hal
db.reduit.mapReduce(db.eval("mapHal"), db.eval("reducedate"),{out: {inline:1}})
```

Pour vérifier s'il existe des documents sans subject, il suffit d'exécuter la commande suivante

```
db.reduit.find({"subject":{$exists: false}},{}).count();
```

3.4.4 Cas 4 : plusieurs clés et plusieurs values

(KEYS, VALUES)

On va travailler sur les deux axes en même temps, plusieurs clés en conservant Date et Language et plusieurs values en prenant en compte le nombre de documents et la moyenne du nombre de coauteurs.

Fonction map4

```
function() {
        var nouv_date="";
        nouv_date=this.date.substr(0,4);
        var nbauteurs=0;
        if(this.coauthors) {
                nbauteurs  = this.coauthors.length;
        }
        emit({lang:this.language,date:nouv_date},{count:1, nbauthors:nbauteurs});
}
```

Fonction reduce4 :

```
function(key,values) {
        var res={count:0, moy:0};
        var nbtot = 0;
        values.forEach (function (v) {res.count+=v.count;});
        values.forEach (function (v) {nbtot+=v.nbauthors;});
        res.moy = nbtot/res.count;
         return res;
        }
```

Exemple à ne pas faire : car ne respecte pas les règles de la fonction reduce. Le système nous donne comme résultat un NaN

Il faut procéder de la manière suivante :

Définir la fonction map4

```
function() {
        var nouv_date="";
        var valeurs={count:0, nbauteurs:0};
        var nbauteurs=1;

        nouv_date=this.date.substr(0,4);
        valeurs.nbauteurs  = this.coauthors.length;
        valeurs.count =1;
        // retour de la key, values
        emit({language:this.language,date:nouv_date},valeurs);
 }
```

Définir la fonction reduce4

```
function(key,values) {
        // Attention au renvoi de la fonction car comme
        // le traitement de map-reduce est recursif
        var res={count:0, nbauteurs:0};
        for (var idx = 0; idx < values.length; idx++) {
                res.count += values[idx].count;
                if (values[idx]. nbauteurs)
                        res. nbauteurs += values[idx]. nbauteurs;
        }
        return res;
}
```

Et enfin pour le calcul de la moyenne

```
function (key, reducedVal) {
        reducedVal.avg = reducedVal. nbauteurs /reducedVal.count;
        return reducedVal;
}
```

Pour appeler le traitement :

```
db.publications.mapReduce(
        db.eval("map4"),
        db.eval("reduce4"),
        {       out:{inline:1},
                finalize:db.eval("finalize4")
        }
)
```

Le résultat obtenu pour la collection publications avec 336341 documents


```
"timeMillis" : 9868,
"counts" : {
        "input" : 336341,
        "emit" : 336341,
        "reduce" : 10448,
        "output" : 1064
  },
```

```
        {
                "_id" : {
                        "language" : "zh",
                        "date" : "2015"
                },
                "value" : {
                        "count" : 6,
                        "nbauteurs" : 10,
                        "avg" : 1.6666666666666667
                }
        },
        {
                "_id" : {
                        "language" : "zh",
                        "date" : "2016"
                },
                "value" : {
                        "count" : 2,
                        "nbauteurs" : 3,
                        "avg" : 1.5
                }
        }
```


3.4.5 Cas 5 : Je veux connaitre, pour un auteur donné, le nombre de coauteurs avec qui le chercheur travaille

Deux fonctions ont été créées mapauthors et reduceauthors.

mapauthors

```
function() {

  // onparcourt le tableau des coauteurs

  for (var i =0; i<this.coauthors.length;i++) {

     emit(this.coauthors[i],{count:1});

  }
```

reduceauthors

```
function(key,values) {

  var res={count:0};

  values.forEach( function(v)

  {

     res.count+=v.count;

  });
```

```
use Hal
db.publications.mapReduce(db.eval("mapauthors"),db.eval("reduceauthors"),{out:{inline:1},
query:{"coauthors":{$regex:"bergounioux gabriel",$options:"i"}}})
:1}})
```

3.5 Méthode Agrégat de MongoDB

Pour l'exemple 1

```
db.reduit.aggregate(
   { $group: { _id: { date: "$date"},
               nb: { $sum: 1 } } },
    {$sort:{_id: -1}}
)
```

Pour l'exemple 2:

```
db.reduit.aggregate({$group:{_id:{date:{$substr:["$date",0,4]},day:"$la
nguage"},nb:{$sum:1}}},{$sort:{_id: -1}} )
```

Pour l'exemple 3:

Exemple d'utilisation d'agrégat avec la syntaxe suivante :

```
db.reduit.aggregate([
        {$unwind : "$subject"},
        {$match: {"subject":{$regex: {"SHS.}}},
        {$group: { _id:{"theme":"$subject"},nb:{$sum:1}}}
])
```

La fonction $unwind a pour objectif d'éclater le document en plusieurs documents temporaire selon un champ donnée. Ce qui permet ensuite de traiter les éléments séparément.

Exemple pour le document suivant :


```
{ "_id" : ObjectId("5832def80fc5072ad11cbb8e"), "source" : [ "L'expérience du cinéma",
"https://halshs.archives-ouvertes.fr/halshs-01357187", "L'expérience du cinéma, Hermann,
2015, cahier textuel, 9782705691189" ], "language" : "fr", "subject" : [ "cinéma", " Eastwood", "
critique cinématographique", "[SHS:ART] Humanities and Social Sciences/Art and art history" ],
"date" : "2015", "description" : [ "International audience" ], "identifier" : [ "ISBN :
9782705691189", "halshs-01357187", "https://halshs.archives-ouvertes.fr/halshs-01357187" ],
"contributors" : [ "Centre D'Etude et de Recherche Interdisciplinaire de l'UFR LAC (CERILAC) ;
Universite Paris Diderot - Paris 7 (UP7)" ], "publishers" : [ "Hermann", "HAL CCSD" ],
"coauthors" : [ "Toulza Pierre-Olivier" ], "title" : "Retour a l'inspecteur. Trois moments dans la
reception critique de la carriere de Clint Eastwood", "type" : [ "info:eu-repo/semantics/bookPart",
```


Résultat de la requête suivante :

```
db.reduit.aggregate([{$unwind:"$subject"},{$match: {"title":{$regex:"Retour a l'inspecteur."}}}]);
```


```
{ "_id" : ObjectId("5832def80fc5072ad11cbb8e"), "source" : [
"L'expérience du cinéma", "https://halshs.archives-ouvertes.fr/halshs-
01357187", "L'expérience du cinéma, Hermann, 2015, cahier textuel,
9782705691189" ], "language" : "fr", "subject" : "cinéma", "date" : "2015",
"description" : [ "International audience" ], "identifier" : [ "ISBN :
9782705691189", "halshs-01357187", "https://halshs.archives-
ouvertes.fr/halshs-01357187" ], "contributors" : [ "Centre D'Etude et de
Recherche Interdisciplinaire de l'UFR LAC (CERILAC) ; Universite Paris
Diderot - Paris 7 (UP7)" ], "publishers" : [ "Hermann", "HAL CCSD" ],
"coauthors" : [ "Toulza Pierre-Olivier" ], "title" : "Retour a l'inspecteur.
Trois moments dans la reception critique de la carriere de Clint
Eastwood", "type" : [ "info:eu-repo/semantics/bookPart", "Book section" ] }

{ "_id" : ObjectId("5832def80fc5072ad11cbb8e"), "source" : [
"L'expérience du cinéma", "https://halshs.archives-ouvertes.fr/halshs-
01357187", "L'expérience du cinéma, Hermann, 2015, cahier textuel,
9782705691189" ], "language" : "fr", "subject" : " Eastwood", "date" :
"2015", "description" : [ "International audience" ], "identifier" : [ "ISBN :
9782705691189", "halshs-01357187", "https://halshs.archives-
ouvertes.fr/halshs-01357187" ], "contributors" : [ "Centre D'Etude et de
Recherche Interdisciplinaire de l'UFR LAC (CERILAC) ; Universite Paris
Diderot - Paris 7 (UP7)" ], "publishers" : [ "Hermann", "HAL CCSD" ],
"coauthors" : [ "Toulza Pierre-Olivier" ], "title" : "Retour a l'inspecteur.
Trois moments dans la reception critique de la carriere de Clint
Eastwood", "type" : [ "info:eu-repo/semantics/bookPart", "Book section" ] }

{ "_id" : ObjectId("5832def80fc5072ad11cbb8e"), "source" : [
"L'expérience du cinéma", "https://halshs.archives-ouvertes.fr/halshs-
01357187", "L'expérience du cinéma, Hermann, 2015, cahier textuel,
9782705691189" ], "language" : "fr", "subject" : " critique
cinématographique", "date" : "2015", "description" : [ "International
audience" ], "identifier" : [ "ISBN : 9782705691189", "halshs-01357187",
"https://halshs.archives-ouvertes.fr/halshs-01357187" ], "contributors" : [
"Centre D'Etude et de Recherche Interdisciplinaire de l'UFR LAC
(CERILAC) ; Universite Paris Diderot - Paris 7 (UP7)" ], "publishers" : [
```

Pour l'exemple 4 :

On va travailler sur les deux axes en même temps, plusieurs clés en conservant Date et Language et plusieurs values en prenant en compte le nombre de documents et la moyenne du nombre de coauteurs.

Requête

```
db.publications.aggregate(
        {$group:{
                _id:{date:{$substr:["$date",0,4]},
                day:"$language"
                },
                nb:{$sum:1},
                nbauthors:{$sum:{$size:"$coauthors"}},
                avg:{$avg:{$size:"$coauthors"}}}
        },
        {$sort:{_id: -1}}
)
```

Partie des résultats sur la collection publications

```
{ "_id" : { "date" : "2015", "day" : "fr" }, "nb" : 10863, "nbauthors" : 16472, "avg" :
1.5163398692810457 }
{ "_id" : { "date" : "2015", "day" : "fi" }, "nb" : 1, "nbauthors" : 1, "avg" : 1 }
{ "_id" : { "date" : "2015", "day" : "fa" }, "nb" : 1, "nbauthors" : 1, "avg" : 1 }
{ "_id" : { "date" : "2015", "day" : "eu" }, "nb" : 2, "nbauthors" : 2, "avg" : 1 }
{ "_id" : { "date" : "2015", "day" : "es" }, "nb" : 204, "nbauthors" : 274, "avg" :
1.3431372549019607 }
{ "_id" : { "date" : "2015", "day" : "eo" }, "nb" : 1, "nbauthors" : 1, "avg" : 1 }
{ "_id" : { "date" : "2015", "day" : "en" }, "nb" : 5760, "nbauthors" : 12899, "avg" :
2.2394097222222222 }
```

Détail sur la commande MapReduce

Syntaxe de la commande (https://docs.mongodb.com/v3.2/reference/command/mapReduce/)

```
db.runCommand(
          {
            mapReduce: <collection>,
            map: <function>,
            reduce: <function>,
            finalize: <function>,
            out: <output>,
            query: <document>,
            sort: <document>,
            limit: <number>,
            scope: <document>,
            jsMode: <boolean>,
            verbose: <boolean>,
            bypassDocumentValidation:
<boolean>
          }
         )
```

Explications complémentaires sur la fonction Reduce :

- Pas d'accès de la fonction reduce à la base de données même en lecture.
- La fonction reduce ne doit pas modifier le système.
- MongoDB n'appelera pas la fonction reduce pour une clé avec une seule valeur. La variable "values" est un tableau dans lequel se trouve des objets mapped selon la clé.
- MongoDB peut invoquer la fonction reduce plus d'une fois avec la même clé Key. Dans ce cas, les valeurs de sortie des précédent reduce pour cette clé deviendront des entrées pour la nouvelle fonction reduce.
- La fonction reduce peut accèder à des variables définies dans le paramètre scope.
- Les entrées de reduce doivent etre pas plus grand que la moitié d'un MongoDB's maximum BSON document size. Cette recommandation peut-être transgressée quand de grands documents sont retournés et joins ensemble dans une sous étape de reduce.

Parce-qu'il est possible d'invoquer plus d'une fois la fonction reduce pour une même clé, les propriétés suivantes sont nécessairement vraies:

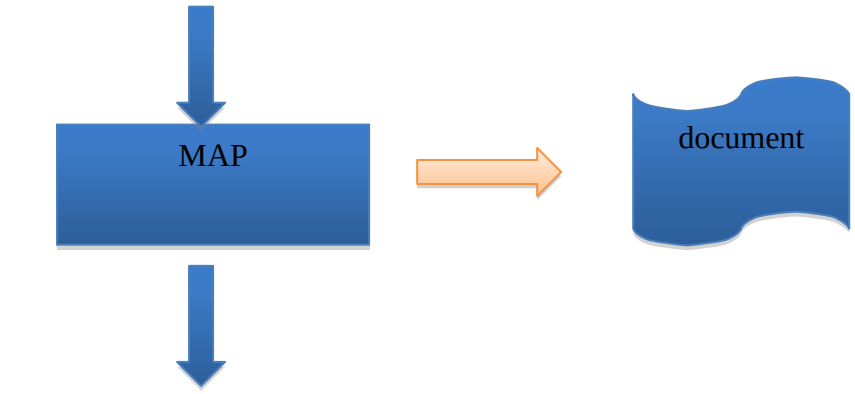
- Le type l'objet de retour doit être identique au type de valeur émis par la fonction map.
- La fonction reduce doit être *associative*. Ce qui signifie les propriétés suivantes : reduce(key, [C, reduce(key, [A, B])]) == reduce(key, [C, A, B])

- La fonction reduce doit être *idempotent*. Ce qui entraine :
  reduce(key, [reduce(key, valuesArray)]) == reduce(key, valuesArray)

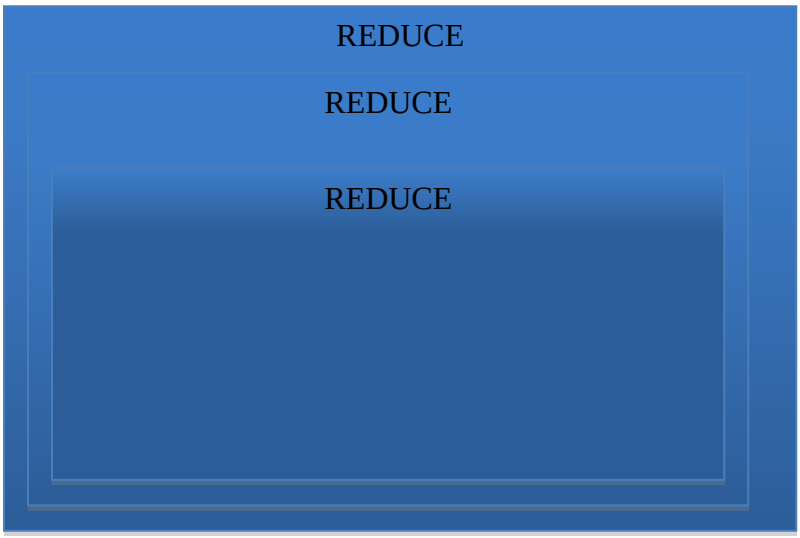
- La fonction reduce doit être *commutative*: ce qui signifie que l'ordre dans les éléments de valuesArray ne doit pas affecté la sortie de la fonction reduce., ce qui entraine la propriétés suivantes :
  reduce(key, [A, B]) == reduce(key, [B, A])

Principe de traitement Map-Reduce dans MongoDB

D'où la solution :



Pour une même clé:



4 PROBLÉMATIQUES SPÉCIFIQUES

4.1 Description

On a constaté qu'il existe des anomalies au niveau de l'écriture dans Hal qui ne sont pas strictes notamment au niveau des noms et prénoms des coauteurs mais également pour les titres. Cela facilite au niveau des saisis mais cela va engendrer des erreurs au moment de l'extraction des données et donc une mauvaise interprétation du graphe dans sa visualisation.

5 ANOMALIES DES DONNEES CONSTATEES

La plateforme « Network of Researchers", permet de visualiser le réseau du chercheur avec l'auteur et de ses liens avec les co-auteurs . Pour cela il suffit de faire une recherche par nom et prénom dans la base de données MongoDB. On s'est aperçu que la recherche par nom, prénom fait remonter des nom et prénom qui ne correspondent pas à la personne recherchée. Le souci provient du saisi du nom et prénom au niveau de Hal.

Exemple :

Martin Sébastien
Sébastien Martin
M. Sébastien
Sébastien M.

Hurlin Christophe
Hurlin C.
C. Hurlin
Christophe H.

Le souci posé est que je recherche « Martin Sébastien », je récupère également « Sébastien M. » qui peut être « Sébastien Michel » et pas forcément la personne recherchée. Ce qui induit en erreur dans la visualisation du graphe du chercheur. On peut les discriminer en comparant le titre de la Publication sous réserve que le titre soit identique.

5.1 Redondance

Après des tests, il est arrivé de trouver dans les publications dont les titres sont pratiquement identiques.

Exemple ?: