

# Travaux pratiques: implémentation de Hadoop MapReduce “from scratch” en Java.

Etape 1: faire un programme séquentiel non parallélisé qui compte le nombre d'occurrences des mots dans un fichier.

**Prérequis pour les questions suivantes:**

- programmer en Java  
<https://perso.telecom-paristech.fr/bellot/CoursJava/>
- manipuler l'environnement de développement Eclipse  
<https://perso.telecom-paristech.fr/bellot/CoursJava/tps/>
- lire et écrire dans un fichier en java  
<https://perso.telecom-paristech.fr/bellot/CoursJava/JavalO.html>
- 

## **1 Premier comptage en séquentiel pur**

Implémentez un logiciel en java qui compte le nombre d'occurrences des mots d'un fichier d'entrée de manière non parallélisée (monothread, une seul thread), en utilisant un seul processeur.

Quelle structure de donnée est la plus pertinente pour stocker les résultats: List, HashMap ou HashSet ou une autre ? Pour quelle raison ?

Testez votre programme avec un fichier d'entrée *input.txt* avec comme contenu:

```
Deer Beer River
Car Car River
Deer Car Beer
```

Résultat:

```
Deer 2
Beer 2
River 2
Car 3
```

## **2 Premier tri en séquentiel pur**

Modifiez votre programme pour trier par nombre d'occurrences:

Résultat:

```
Car 3
Deer 2
```

Beer 2  
River 2

### 3 Deuxième tri alphabétique en séquentiel pur

Modifiez le programme pour trier alphabétiquement pour les mots à égalité du nombre d'occurrences:

Résultat:

Car 3  
Beer 2  
Deer 2  
River 2

### 4 Test du programme séquentiel sur le code forestier de Mayotte

Testez ensuite votre programme avec le code forestier de Mayotte disponible sur github *forestier\_mayotte.txt* :

<https://github.com/legifrance/Les-codes-en-vigueur>

Votre programme a-t-il fonctionné du premier coup ?

Vérifiez en ouvrant le fichier texte qu'il contient bien du texte et non du code HTML.

Ne perdez pas de temps à corriger les éventuelles erreurs dues aux caractères spéciaux ou à des mots suspects ou illisibles (de toutes façons par la suite il y aura du chinois dans le texte).

### 7 Les 50 mots du code de la déontologie de la police nationale

Testez votre programme avec le code de déontologie de la police nationale disponible sur github *deontologie\_police\_nationale.txt* : <https://github.com/legifrance/Les-codes-en-vigueur>

De même ne perdez pas de temps à filtrer les caractères spéciaux ou autres mots bizarres. Pourquoi ? Car nous travaillerons ensuite sur des textes en chinois, japonais, arabe et d'autres langues. Si vous implémentez une étape de filtrage ici en français elle ne servira à rien par la suite. Quels sont les 5 premiers mots (qui ressemblent à des mots) parmi les 50 premiers de la liste triée résultat ?

### 8 Les 50 mots du code du domaine public fluvial

Testez votre programme avec le code du domaine public fluvial *domaine\_public\_fluvial.txt*. Quels sont les 5 premiers mots (qui ressemblent à des mots) parmi les 50 premiers de la liste triée résultat ?

### 9 Les 50 mots du code de la santé publique

Testez votre programme avec le code de la santé publique *sante\_publicue.txt*.

Quels sont les 5 premiers mots (qui ressemblent à des mots) parmi les 50 premiers de la liste triée résultat ?

### 10 Chronométrage du programme séquentiel

Chronométrer votre programme sur le code de la santé publique.

Chronométrage possible avec:

```
long startTime = System.currentTimeMillis();
```

...

```
long endTime = System.currentTimeMillis();  
long totalTime = endTime - startTime;
```

Combien de temps faut-il pour chacune des étapes:

- Compter le nombre d'occurrences
- Tri (par nombre d'occurrences et alphabétique)

## 11 Travailler sur des plus gros fichiers

Testez votre programme sur un cas réel: un extrait de toutes les pages internet transformées au format texte brut (format WET). Toutes les pages sur internet au format texte sont disponibles sur <http://commoncrawl.org/the-data/get-started/> : chaque mois, environ 3 milliards de pages web, soit 250 To de données sont stockées. Ces données sont disponibles par tranche de moins d'1Go environ, vous travaillerez sur une tranche de 380Mo.

J'ai choisi une tranche en particulier pour avoir une comparaison entre nous (vous pouvez tester sur d'autres tranches si vous voulez). Téléchargez cette tranche ici:

<https://commoncrawl.s3.amazonaws.com/crawl-data/CC-MAIN-2017-13/segments/1490218189495.77/wet/CC-MAIN-20170322212949-00140-ip-10-233-31-227.ec2.internal.warc.wet.gz>

Décompressez et obtenez le fichier

```
CC-MAIN-20170322212949-00140-ip-10-233-31-227.ec2.internal.warc.wet
```

Il s'agit d'une tranche contenant un ensemble de sites internet au format texte brut (WET). Testez votre programme avec ce fichier en entrée. Chronométrez-le.

## Etape 2: travailler avec plusieurs ordinateurs en réseau.

**Prérequis et documentation pour les questions suivantes :**

- avoir un compte et pouvoir se connecter aux machines de l'école (contacter la DSI de l'école dans le cas contraire)
- lancer un interpréteur de commande (console linux , shell) pour taper des commandes:  
[https://fr.wikipedia.org/wiki/Interpréteur\\_de\\_commandes](https://fr.wikipedia.org/wiki/Interpréteur_de_commandes)  
[https://fr.wikipedia.org/wiki/Shell\\_Unix](https://fr.wikipedia.org/wiki/Shell_Unix)
- connaître quelques commandes de base sous Linux  
[https://fr.wikipedia.org/wiki/Commandes\\_Unix](https://fr.wikipedia.org/wiki/Commandes_Unix)
- <https://prod-edx-mktg-edit.edx.org/course/linux-basics-the-command-line-interface>
- savoir qu'un ordinateur a un nom d'hôte (hostname) et plusieurs adresses IP:  
<https://en.wikipedia.org/wiki/Hostname>  
[https://fr.wikipedia.org/wiki/Adresse\\_IP](https://fr.wikipedia.org/wiki/Adresse_IP)
- savoir qu'un ordinateur peut faire partie d'un domaine (comme à l'école, le domaine enst.fr ou le domaine telecom-paristech.fr )  
[https://fr.wikipedia.org/wiki/Nom\\_de\\_domaine](https://fr.wikipedia.org/wiki/Nom_de_domaine)

- savoir qu'un nom peut être transformé en adresse IP (et inversement) par un serveur qui gère le système du nom de domaine (DNS, Domain Name System)  
[https://fr.wikipedia.org/wiki/Domain\\_Name\\_System](https://fr.wikipedia.org/wiki/Domain_Name_System)
- se connecter à distance à un ordinateur avec SSH en ligne de commande  
[https://fr.wikipedia.org/wiki/Secure\\_Shell](https://fr.wikipedia.org/wiki/Secure_Shell)  
<http://www.commentcamarche.net/faq/74-se-connecter-a-distance-avec-ssh-linux>

## **12 Nom court, nom long**

quel est le nom COURT de votre ordinateur (le nom simple sans le domaine) ? quel est le nom LONG de votre ordinateur (le nom avec le domaine) ? Comment les connaître en ligne de commande ? Sur les ordinateurs de l'école, est-il possible d'obtenir ces noms autrement qu'en ligne de commande ?

## **13 Adresse ip**

Comment connaître les adresses (plusieurs) IP de votre ordinateur en ligne de commande ? Autrement (en passant par un site internet par exemple) ?

## **14 Du nom vers l'IP**

Comment à partir du nom d'un ordinateur, obtenir les adresses IP en ligne de commande ?

## **15 De l'IP vers le nom**

Comment, à partir d'une adresse IP, obtenir les noms associés en ligne de commande ?

## **16 Ping pong à l'intérieur!**

Testez la communication avec d'autres ordinateurs (pas le vôtre) depuis le réseau de l'école en utilisant la commande ping (pour arrêter le ping faire CTRL + C). suivi du nom court, du nom long, de l'IP. Les trois méthodes fonctionnent-elles ?

## **17 Ping pong à l'extérieur**

Si vous effectuez le ping depuis un réseau différent, il est possible que celui ne fonctionne pas (filtrage des accès vers le réseau de l'école depuis un réseau extérieur), contactez la DSI pour mettre en place une connection VPN / OpenVPN afin d'être sur le même réseau que les machines en salle de TP.

## **18 Calculer en ligne de commande sur l'ordinateur local**

Comment lancer un calcul en ligne de commande sur votre ordinateur (par exemple 2 + 3) ? Parmi les multiples réponses possibles, lesquelles permettent de lancer le calcul et d'obtenir le résultat en appuyant une seule fois sur la touche <Entrée> ?

## **19 Calculer en ligne de commande sur un ordinateur distant**

Comment lancer un calcul (par exemple 2 + 3) en ligne de commande sur un autre ordinateur (à distance) ? Il faudra certainement vous authentifier avec un mot de passe. Comment obtenir le résultat du calcul immédiatement après avoir tapé son mot de passe ?

## 20 Calculer à distance sans mot de passe

Comment lancer un calcul à distance en utilisant SSH sans taper le mot de passe et en une seule ligne de commande (c'est à dire qu'on appuie sur <Entrée> et on a le résultat directement)?

Attention: en utilisant vos ordinateurs personnels, il faut parfois vérifier le "fingerprint" avec le message:

```
The authenticity of host 'mint.phcomp.co.uk (78.32.209.33)' can't be established.  
RSA key fingerprint is 6a:de:e0:af:56:f8:0c:04:11:5b:ef:4d:49:ad:09:23.  
Are you sure you want to continue connecting (yes/no)? yes
```

Pour ne plus avoir à rentrer quoi que ce soit dans le terminal, il faut donc désactiver la vérification de ce "fingerprint", sinon vous aurez l'erreur

*Host key verification failed*

Suivez ce guide pour désactiver la vérification de "fingerprint":

<https://www.shellhacks.com/disable-ssh-host-key-checking/>

Etape 3: travailler avec des fichiers locaux ou sur un serveur NFS.

**Prérequis et documentation pour les questions suivantes :**

- pouvoir transférer un fichier d'un ordinateur à un autre en utilisant la commande SCP (Secure Copy): [https://fr.wikipedia.org/wiki/Secure\\_copy](https://fr.wikipedia.org/wiki/Secure_copy)
- connaître l'architecture d'un système de fichier NFS (Network File System) : [https://fr.wikipedia.org/wiki/Network\\_File\\_System](https://fr.wikipedia.org/wiki/Network_File_System)

## 21 Chemin absolu

Quel est le chemin absolu de votre répertoire personnel, votre *home directory* ?  
(commandes "cd" puis "pwd")

## 22 Un fichier dans le répertoire personnel

Créez un fichier fperso.txt contenant le texte "bonjour" dans votre répertoire personnel (sur un ordinateur de l'école).

Vérifiez le contenu du fichier avec cette commande exactement:

```
cat ~/fperso.txt
```

## 23 Ou se trouve le fichier dans le répertoire personnel

Ce fichier est-il sur le disque dur de l'ordinateur ou autre part ? Comment savoir où est stocké physiquement ce fichier, à l'aide de quelle commande ?

## 24 Un dossier et un fichier dans le répertoire temporaire

Créez un dossier /tmp/<votre nom d'utilisateur> en remplaçant <votre nom d'utilisateur>.

Créez un fichier ftemp.txt dans le répertoire /tmp/<votre nom d'utilisateur> .

Vérifiez le contenu du fichier avec cette commande **exactement**:

```
cat /tmp/<votre nom d'utilisateur>/ftemp.txt
```

Ce dossier et ce fichier sont-ils sur le disque dur de l'ordinateur ou autre part ? Comment savoir où sont stockés physiquement ces éléments, à l'aide de quelle commande ?

### **25 Trois ordinateurs A B C. On commence avec A. Utilisation du serveur NFS.**

Pour les questions suivantes, utilisez trois ordinateurs: A, B C.

Connectez vous physiquement (avec un clavier, une souris et un écran) sur l'ordinateur A. Sur A, créez un fichier text.txt contenant le texte "mon texte sur NFS" dans votre répertoire personnel.

Vérifiez que le fichier existe et vérifiez son contenu. Pour cela, sur A, utilisez la commande :

```
cat ~/text.txt
```

### **26 Trois ordinateurs A B C. On continue sur B et sur C. Utilisation du serveur NFS.**

Connectez-vous à B (physiquement ou à distance) et vérifiez que le fichier text.txt est également présent dans votre répertoire personnel. Pour cela, sur B, utilisez la commande :

```
cat ~/text.txt
```

De même, connectez-vous à C et vérifiez que text.txt est aussi présent.

Remarquez que vous n'avez pas copié le fichier mais qu'il est présent sur A, B et C grâce au serveur NFS.

### **27 Trois ordinateurs A B C. On commence avec A. Utilisation des disques locaux.**

Déconnectez vous de B et de C et revenez sur l'ordinateur A.

Sur A, créez un dossier /tmp/<votre nom d'utilisateur> et un fichier local.txt contenant le texte "mon texte sur disque local" dans ce dossier /tmp/<votre nom d'utilisateur>.

Vérifiez que le fichier existe et vérifiez son contenu. Pour cela, sur A, utilisez la commande :

```
cat /tmp/<votre nom d'utilisateur>/local.txt
```

### **28 Trois ordinateurs A B C. On continue sur B et sur C. Utilisation des disques locaux.**

Connectez-vous à B et C (physiquement ou à distance) et vérifiez que le dossier <votre nom d'utilisateur> ainsi que le fichier local.txt **ne sont pas** présent dans /tmp . Pour cela vérifiez avec la commande:

```
ls /tmp
```

### **29 Depuis A, copier de A vers B avec les disques locaux.**

Comment, à partir de A, transférer le fichier /tmp/local.txt **sur B** (dans /tmp/<votre nom d'utilisateur>/local.txt) en utilisant scp ? Vérifiez que le fichier est bien présent sur B. Attention: si vous avez une erreur "no such file or directory" (ou l'équivalent français), vous devez d'abord créer le répertoire /tmp/<votre nom d'utilisateur>/ avec la commande `mkdir -p` associée à un ssh pour l'ordinateur distant.

### **30 Depuis A, copier de B vers C avec les disques locaux.**

Comment, à partir de A, transférer le fichier de B (depuis /tmp/<votre nom d'utilisateur>/local.txt) **vers C** (dans /tmp/<votre nom d'utilisateur>/local.txt) ? Vérifiez que le fichier est bien présent sur C. De même que la question précédentes, vous devez créer les répertoires /tmp/<votre nom d'utilisateur>/ correspondants.

## Etape 4: lancer des programmes java à distance manuellement.

### Prérequis et documentation pour les questions suivantes :

- exporter un .JAR exécutable (Runnable JAR File)
- utiliser les packages et savoir lancer un programme Java en ligne de commande

### 31 Un premier programme SLAVE sous Eclipse

Faire un programme Java nommé "SLAVE" qui calcule 3+5, affiche le résultat, sous Eclipse (Pour lancer Eclipse: Menu applications>développement), lancer ce programme dans Eclipse (flèche verte "exécuter")

### 32 Exportation en JAR

Exporter le programme Java en slave.jar exécutable (Java ARchive dite Runnable) sous Eclipse. Attention de bien vérifier que le JAR est de type "Runnable"/"exécutable".

### 33 Exécution sur disque dur local

Créez le répertoire `/tmp/<votre nom d'utilisateur>/`

Copiez slave.jar exécutable dans le répertoire `/tmp/<votre nom d'utilisateur>/`

Testez et Lancer le slave.jar en ligne de commande sur votre ordinateur local.

### 34 Copie du JAR et exécution distante

Depuis la machine A contenant `/tmp/<votre nom d'utilisateur>/slave.jar`

Créez à distance sur la machine B (s'il n'existe pas) un répertoire `/tmp/<votre nom d'utilisateur>/`

Copiez slave.jar sur la machine B dans le répertoire `/tmp/<votre nom d'utilisateur>/`

Exécutez à distance (depuis A sur la machine B) le slave.jar.

Quelle est la commande tapée pour effectuer cette dernière action ?

## Etape 5: lancer des programmes en ligne de commande depuis java et afficher la sortie standard et la sortie d'erreur.

### Prérequis et documentation pour les questions suivantes :

- utiliser le processBuilder en java  
<http://docs.oracle.com/javase/7/docs/api/java/lang/ProcessBuilder.html> .
- lancer un exécutable (ou une commande) en ligne de commande depuis un programme écrit en java
- connaître les sorties standard et les sorties d'erreurs

### 35 Un programme MASTER java qui lance un autre programme en ligne de commande!

Ecrire un programme java nommé "MASTER" qui lance la commande suivante en utilisant ProcessBuilder:

```
ls -al /tmp
```

(vous pouvez aussi tester cette commande dans un terminal avant)

Récupérer et afficher la sortie de cette commande.

Vous devez utiliser ProcessBuilder de cette façon:

```
ProcessBuilder pb = new ProcessBuilder("ls", "-al", "/tmp");
```

### **36 Un programme MASTER java qui gère les erreurs de lancement d'un autre programme en ligne de commande.**

Modifiez votre programme "MASTER" pour qu'il affiche la sortie d'erreur en cas d'erreur lors de l'exécution de la commande. Testez la sortie d'erreur avec une commande qui échoue avec un sortie d'erreur. Essayez par exemple d'exécuter la commande "ls /jesuisunhero". Explications: si on tape la commande "ls /jesuisunhero", le dossier /jesuisunhero n'existant pas, on aura une erreur de type "impossible d'accéder à /jesuisunhero: aucun fichier ou dossier de ce type." qui s'affiche dans la sortie d'erreur. En effet, il y a deux sorties: les sorties standards (sans erreur) et les sorties d'erreurs.

Vous devez utiliser ProcessBuilder de cette façon:

```
ProcessBuilder pb = new ProcessBuilder("ls", "/jesuisunhero");
```

Sur pb, vous pouvez récupérer le flux de la sortie standard et le flux de la sortie d'erreur.

### **37 Un programme MASTER java qui lance un slave.jar en ligne de commande.**

Modifiez votre programme "MASTER" pour qu'il lance "SLAVE", c'est à dire slave.jar situé sur la même machine que "MASTER" dans le dossier

```
/tmp/<votre nom d'utilisateur>/slave.jar
```

## **Etape 6: gérer les timeout du MASTER.**

### **38 Un SLAVE qui simule un calcul de 10 secondes.**

Modifiez votre programme SLAVE pour qu'il simule une attente de 10 secondes avant d'afficher le résultat du calcul 3+5. Pour cela utilisez

```
Thread.sleep(10000);
```

Vérifiez le bon fonctionnement du SLAVE et constatez qu'il y a 10 secondes entre le démarrage du SLAVE et l'affichage du résultat. Attention de ne rien afficher avant les 10 secondes pour que la question suivante fonctionne correctement.

Générez de nouveau le slave.jar. Copiez-le en écrasant le slave.jar dans le dossier

```
/tmp/<votre nom d'utilisateur>/slave.jar
```

Testez le lancement à partir de MASTER.

### **39 Gérer les timeout au niveau du MASTER.**

Modifier le MASTER pour qu'il attende que quelque chose soit écrit dans la sortie standard (sans erreur) ou dans la sortie d'erreurs du SLAVE pendant un certain temps maximum. Au bout du temps imparti le MASTER considère un timeout.

Il arrête les éventuels threads (si vous utilisez des threads - non obligatoire) s'occupant des sorties et/ou stoppe le processus créé avec ProcessBuilder.

TEST1 : Testez le bon fonctionnement du timeout en lançant le SLAVE avec un timeout de 2 secondes sur les sorties (standard et d'erreur). Le timeout étant plus court (au niveau du



MASTER 2 secondes) que le temps de calcul du SLAVE (10 secondes), le MASTER doit arrêter les éventuels threads (si vous en utilisez) et le processus.

TEST 2: Testez ensuite avec un timeout de 15 secondes, il ne devrait pas y avoir de timeout.

TEST 3: Testez ensuite en changeant le SLAVE pour qu'il écrive non plus dans la sortie standard (sans erreur) mais dans la sortie d'erreur. Pour cela, remplacez dans le Slave les `System.out.print...` par `System.err.print...`

#### Aide:

SOLUTION 1 (difficile) : Vous utilisez deux threads (processus légers) pour chaque processus lourd (lancé avec ProcessBuilder). Un thread s'occupe de lire la sortie standard et l'autre la sortie d'erreur. Une solution pour gérer les timeout consiste à utiliser une structure *ArrayBlockingQueue* (taille fixe) ou *LinkedBlockingQueue* (taille dynamique) donnée en paramètre de chaque thread s'occupant de lire la sortie standard (sans erreur) ou la sortie d'erreur. Chaque thread va écrire ce qui est lu depuis les sorties dans cette structure en utilisant la méthode *put*. Le timeout de 2 secondes peut alors être paramétré lors de la récupération des éléments insérés dans la structure en utilisant la méthode *poll* sur l'*ArrayBlockingQueue* ou la *LinkedBlockingQueue* de cette manière:

```
poll(2, TimeUnit.SECONDS);
```

SOLUTION 2 (facile) : Une solution consiste à 1) rediriger la sortie d'erreur dans la sortie standard à l'aide de la méthode *inheritIO()* du ProcessBuilder avec *pb.inheritIO()*; et 2) attendre la fin du process avec un *waitFor* sur le process qui génère un timeout en utilisant *boolean b = p.waitFor(3, TimeUnit.SECONDS); //si b vaut false il y a un timeout*

Attention: dans le cas de l'utilisation de *inheritIO()*, vous ne pourrez plus récupérer et/ou traiter la sortie standard ou la sortie d'erreur du process: par exemple, `p.getInputStream()` renverra un flux vide.

SOLUTION 3 (difficile): il s'agit d'utiliser la solution 1 avec un seul thread et de rediriger la sortie d'erreur dans la sortie standard avec *pb.redirectErrorStream(true)* sur le *processBuilder*.

## Etape 7: déployer automatiquement le programme SLAVE sur un ensemble de machines.

### **40 Un programme DEPLOY : Test de connection SSH multiple**

Créer un fichier texte à la main contenant : les adresses IP et/ou les noms des machines que nous voulons utiliser pour notre système réparti (par exemple toutes les machines de cette salle de TP), avec un nom ou une IP par ligne dans le fichier.

Créer un nouveau programme java DEPLOY qui lit ce fichier ligne par ligne et teste si la connection SSH fonctionne bien sur chacune des machines. Attention, il s'agit bien d'un nouveau programme qui est séparé de MASTER ou SLAVE, vous ne l'exécuterez qu'en cas de mise à jour du SLAVE. Ceci permet de vérifier qu'une machine n'est pas éteinte ou qu'il y a un problème de connection (par exemple).

Pour vérifier que la connection fonctionne bien, vous pouvez faire afficher le nom de la machine distante (en exécutant la commande `hostname` à distance) et vérifier que l’affichage a effectivement lieu, sans erreurs. Réutilisez des parties de codes de la cinquième étape.

Votre programme DEPLOY lance-t-il les connections de manière séquentielle (les unes après les autres) ou de manière parallèle?

#### **41 Un programme DEPLOY : copie de slave.jar multiple**

Modifier votre programme DEPLOY pour qu’il copie le `slave.jar` dans `/tmp/<votre nom d’utilisateur>/` sur les ordinateurs dont la connection SSH fonctionne.

Pour cela, vous devez utiliser la commande `mkdir -p` pour créer les répertoires dans `/tmp` s’ils n’existent pas déjà, attendre que le `mkdir` se termine puis copier avec `scp` le fichier `slave.jar`. Comment faites-vous pour attendre que le `mkdir` se termine correctement?

Vérifiez ensuite manuellement que le fichier a bien été copié sur les ordinateurs distants.

Attention de bien attendre la fin du `mkdir` avant de lancer le `scp` (on ne veut pas avoir de copie avant que le dossier soit effectivement créé).

Lors des copies, faites attention au caractère “ / ” à la fin d’un chemin :

`/tmp/toto` est un chemin vers un fichier nommé `toto`

`/tmp/toto/` est un chemin vers un dossier nommé `toto`.

Votre programme DEPLOY lance-t-il les copies de manière séquentielle (les unes après les autres) ou de manière parallèle?

Etape 8: lancer le programme SLAVE sur un ordinateur à distance.

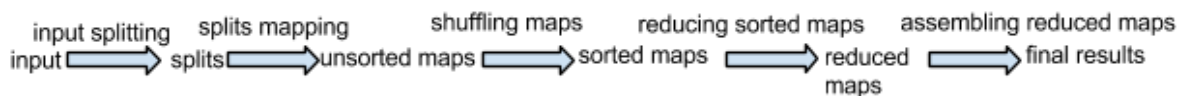
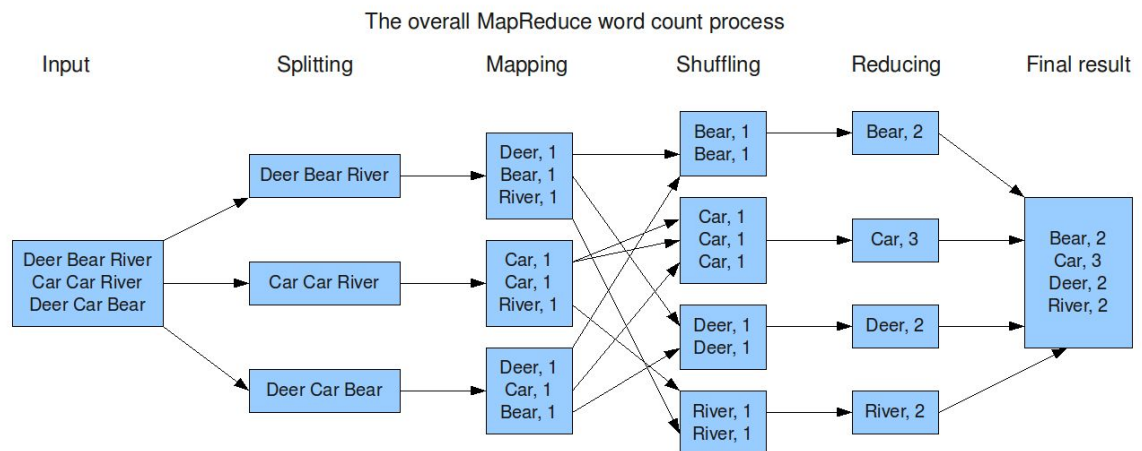
#### **42 Master lançant Slave à distance.**

Modifier votre programme “MASTER” pour qu’il lance à distance le programme “SLAVE” déjà déployé avec le programme “DEPLOY” sur une machine distante.

Etape 9: MapReduce - SPLIT et MAP

**Prérequis et documentation pour les questions suivantes :**

- connaître l’[architecture de MapReduce](#)
- Connaître la convention des nom des fichiers pour notre TP



- connaître le vocabulaire: **S UM SM RM** (Splits, Unsorted Maps, Sorted Maps, Reduced Maps)

### 43 Un MASTER qui déploie les splits

Créez trois fichiers correspondants à des "splits" dans le répertoire temporaire. Dans un premier temps, créez ces fichiers manuellement.

```
/tmp/<votre nom d'utilisateur>/splits
```

S0.txt S1.txt S2.txt.

S0.txt contient:

Deer Beer River

S1.txt contient:

Car Car River

S2.txt contient:

Deer Car Beer

Modifiez votre MASTER pour qu'il copie les fichiers de splits dans 3 ordinateurs différents.

Attention, le répertoire `/tmp/<votre nom d'utilisateur>/splits` doit être créé sur les 3 ordinateurs s'il n'existe pas. Cette création peut se faire de manière automatique (en créant de manière programmatique ces répertoires). Attention de faire attention de bien attendre que la création des dossiers soit bien effectuée avant de lancer la copie des splits. Comment attendez-vous que la création des dossiers soit bien effectuée avant de copier véritablement les fichiers?

De la même manière que le programme DEPLOY, le MASTER va copier ces splits vers 3 ordinateurs dont la connexion SSH fonctionne.

Votre programme MASTER lance-t-il les copies de manière séquentielle (les unes après les autres) ou de manière parallèle?

#### 44 Un SLAVE qui fait la phase de map

Modifiez le SLAVE pour qu'il calcule un map à partir d'un split.

Pour cela, il prend un mode de fonctionnement en argument: 0 correspond au calcul du map à partir d'un split, puis un nom de fichier "Sx.txt" en entrée depuis le dossier `splits` et calcule un fichier "UMx.txt" en sortie dans le dossier `maps`, avec x variant (ici de 1 à 3). De la même manière que précédemment, le dossier `maps` doit être créé avant de pouvoir écrire des fichiers dedans. Vous devez attendre que le dossier `maps` soit créé avant de travailler avec. Comment attendez-vous que le dossier `maps` soit bien créé avant de travailler avec? Le nom du fichier sera donné comme argument args du main:

```
public static void main (String[] args)
```

Testez dans un terminal le `slave.jar` comme suit:

```
cd /tmp/<votre nom d'utilisateur>/  
java -jar slave.jar 0 /tmp/<votre nom d'utilisateur>/splits/S0.txt
```

Le fichier `/tmp/<votre nom d'utilisateur>/maps/UM0.txt` doit être créé contenant

```
Dear 1  
Beer 1  
River 1
```

Testez le fonctionnement de votre SLAVE avec le fichier `S1.txt` contenant  
Car Car River  
Testez dans un terminal le JAR comme suit:

```
cd tmp/<votre nom d'utilisateur>/  
java -jar slave.jar 0 /tmp/<votre nom d'utilisateur>/splits/S1.txt
```

Le fichier `/tmp/<votre nom d'utilisateur>/maps/UM1.txt` doit être créé contenant

```
Car 1  
Car 1  
River 1
```

Comme votre SLAVE est modifié, utilisez le DEPLOY pour déployer la nouvelle version.

Posez-vous la question: pourquoi retrouvons-nous deux lignes

Car 1

Car 1

Au lieu d'une seule ligne

Car 2 ?

Un indice: la phase de reduce qui arrivera plus tard, fera une addition des valeurs. Dans ce cas, la "fonction" de reduce est très simple: c'est une grande addition. Imaginez une

fonction de reduce beaucoup plus complexe qui effectue un algorithme complexe appliqué sur toutes les valeurs...

#### 45 Un MASTER qui lance les SLAVE pour la phase de map.

Modifiez le MASTER pour qu'il lance la phase de map et affiche le dictionnaire "UMx - machines". Attention, n'affichez que le nom du fichier UMx et son emplacement. Le contenu des UMx reste sur les slaves et n'est pas envoyé au MASTER.

Exemple d'affichage:

UM1 - c129-02

UM2 - c129-06

etc ...

Pour bien synchroniser le MASTER, veillez à afficher ce dictionnaire qu'une fois les fichiers UMx créés par les slaves. Le MASTER doit donc attendre que SLAVES se terminent correctement.

Votre programme MASTER lance-t-il les SLAVE de manière séquentielle (les uns après les autres) ou de manière parallèle?

#### 46 Un SLAVE qui affiche les clés en sortie

Modifiez le SLAVE pour qu'il affiche uniquement les clés sur la sortie standard lors du calcul  $S_x \rightarrow UM_x$  (ne modifiez pas la création du fichier UMx). Cela permet d'envoyer au MASTER uniquement les clés sans les valeurs. Testez le fonctionnement de votre SLAVE avec un fichier S1.txt contenant

Car Car River

Testez dans un terminal le JAR comme suit:

```
cd tmp/<votre nom d'utilisateur>/  
java -jar slave.jar 0 /tmp/<votre nom d'utilisateur>/splits/S1.txt
```

L'affichage doit être

Car

River

Le fichier UM1 doit toujours être créé, comme évoqué précédemment.

Comme votre SLAVE est modifié, utilisez le DEPLOY pour déployer la nouvelle version.

#### 47 Un MASTER qui affiche les clés

Modifiez le MASTER pour qu'il reçoive les clés issues de la phase de map. Pour recevoir les clés, vous utiliserez l'affichage dans la sortie standard du SLAVE (récupéré grâce à l'inputstream du process). Testez en affichant le dictionnaire "clés - listes des UMx".

Exemple d'une entrée dans le dictionnaire:

clé Car - liste d'UMs: <UM1, UM2>

(pour une clé, on a plusieurs UMx)

#### 48 Un MASTER attend la fin de la phase de map.

Modifiez le MASTER pour qu'il attende la fin de la phase de map. Testez avec un affichage final "phase de MAP terminée", cet affichage doit s'effectuer uniquement quand tout est fini pour la phase de map (création des fichiers UMx et création complète des dictionnaires "UMx-machines" et "clés-UMx").

## Etape 10: MapReduce - SHUFFLE et REDUCE

#### 49 Un SLAVE qui effectue le shuffle

Modifiez le SLAVE pour qu'il prenne plusieurs fichiers UMx en entrée et calcule un fichier SMx en sortie. Attention, votre SLAVE doit toujours être compatible avec le calcul de Sx -> UMx (mode 0). Le mode de fonctionnement doit donc être donné en argument args du main, en rappel: mode 0 pour "calcul de Sx -> UMx" (phase de map) ou mode 1 pour "calcul de UMx -> SMx" (phase de shuffle).

En fonction du mode, le SLAVE sait s'il reçoit un Sx ou des UMx comme arguments (args) du main.

De plus, la clé et le nom du fichier de sortie SMx doivent également être passés en arguments du main.

Testez le fonctionnement de votre SLAVE SHAVADOOP avec deux fichiers UM1.txt et UM2.txt

UM1.txt contenant:

```
Car 1
Car 1
River 1
```

UM2.txt contenant:

```
Dear 1
Car 1
Bear 1
```

Testez dans un terminal le JAR comme suit, l'argument " 1 " correspond au mode de fonctionnement du slave, ici le mode " 1 " est celui qui calcule un SM1 en output à partir de plusieurs UMx en input, c'est à dire le shuffle (attention, le numéro x dans SMx correspond au numéro de clé, il est incrémenté pour chaque clé: SM2, SM3 etc. et ne doit pas être confondu avec le mode - 0 ou 1 - N'oubliez pas de garder le mode "0" par ailleurs, comme évoqué précédemment.

```
cd tmp/<votre nom d'utilisateur>/
java -jar slave.jar 1 Car /tmp/<votre nom
d'utilisateur>/maps/SM1.txt /tmp/<votre nom
d'utilisateur>/maps/UM1.txt /tmp/<votre nom
d'utilisateur>/maps/UM2.txt
```

Le fichier /tmp/<votre nom d'utilisateur>/maps/SM1.txt doit être créé avec comme contenu:

Car 1  
Car 1  
Car 1

Comme votre SLAVE est modifié, utilisez le DEPLOY pour déployer la nouvelle version.

### 50 Un SLAVE qui effectue le reduce

Modifiez le SLAVE pour un nouveau mode qui calcule un RMx. C'est le mode 2: phase de reduce. Gardez les modes précédents: 0 pour la phase de map, 1 pour la phase de shuffle.

Le dossier `reduces` doit être créé s'il n'existe pas déjà.

Testez dans un terminal le JAR comme suit:

```
cd tmp/<votre nom d'utilisateur>/  
java -jar slave.jar 2 Car /tmp/<votre nom  
d'utilisateur>/maps/SM1.txt /tmp/<votre nom  
d'utilisateur>/reduces/RM1.txt
```

Le fichier `/tmp/<votre nom d'utilisateur>/reduces/RM1.txt` doit être créé contenant

Car 3

Attention: on travaille maintenant dans le dossier "reduces"

Comme votre SLAVE est modifié, utilisez le DEPLOY pour déployer la nouvelle version.

### 51 Un MASTER qui prépare la phase de shuffle

Modifiez le MASTER pour qu'il copie les UMx au bon endroit pour la phase de reduce. Pour cela, il doit utiliser les deux dictionnaires "UMx-machines" et "clés-UMx". Pour chaque clé, on a plusieurs UMx. Ces UMx peuvent être sur des machines différentes. Il faut rapatrier certains UMx (ceux pour une clé particulière) sur une seule machine "slave" qui va calculer le reduce (pour cette clé particulière).

Modifiez le MASTER pour qu'il copie, pour chaque clé, les UMx nécessaires uniquement des différentes machines sur une seule (pas tous les UMx! seulement ceux nécessaires).

Attention: si une machine traite plusieurs clés en parallèle, il est possible que ces clés partagent plusieurs UMx. Dans ce cas, il ne faut pas copier plusieurs fois les UMx correspondants mais seulement une seule fois! Ceci permet d'éviter des situations où l'on copie plusieurs fois en même temps (en parallèle) un même fichier au même endroit (comportement indéterministe).

Votre programme MASTER prépare-t-il la phase de shuffle de manière séquentielle (les actions sont effectuées les unes après les autres) ou de manière parallèle?

Votre programme MASTER prépare-t-il la phase de shuffle de manière plutôt répartie (les machines envoient les fichiers sans que le contenu des fichiers passe par le master) ou de manière centralisée (le contenu des fichiers copiés passent systématiquement par le MASTER)?

## **52 Un MASTER qui lance le reduce**

Modifiez le MASTER pour qu'il lance la phase de reduce une fois la phase de shuffle terminée. Comment attendez-vous la fin de la phase de shuffle?

Vous pouvez tester en affichant temporairement le dictionnaire "RMx - machines".

Votre programme MASTER lance-t-il les SLAVE de manière séquentielle (les uns après les autres) ou de manière parallèle?

## **53 Un MASTER qui affiche le résultat final**

Modifiez le MASTER pour qu'il affiche le résultat final.

## **54 Un projet Hadoop-Mapreduce complet, utilisable en environnement "BigData"**

Copiez votre projet tel qu'il est à la question précédente. Ceci vous permet de garder une version de votre projet qui fonctionne avec l'exemple de démo tel que décrit lors des questions précédentes (petit fichier input.txt de démo avec 3 lignes).

Améliorez la copie de votre projet pour qu'il fonctionne avec des plus gros fichiers en input.

## **55 Des décisions optimisées pour le transfert des fichiers**

De la même manière que précédemment, créez une copie de votre projet sur laquelle vous travaillez sur l'optimisation du nombre de transfert de fichiers lors de la phase de shuffle.

## **56 Des décisions optimisées pour la répartition des calculs**

De la même manière que précédemment, créez une copie de votre projet sur laquelle vous travaillez sur l'optimisation de la répartition des calculs sur les machines: nombre de calcul de map, nombre de calcul de reduce réparti et équitable entre les machines.

## **57 Une prise en compte des pannes**

De la même manière que précédemment, créez une copie de votre projet sur laquelle vous travaillez sur la robustesse: faites en sorte que votre programme fonctionne malgré des pannes inopinées de certaines machines. Pour tester, vous pouvez éteindre une ou plusieurs machines au milieu de calculs. Vous pouvez inventer d'autres pannes vous-même et présenter vos résultats. Comparez avec la méthode de robustesse utilisée par Google :

<https://research.google.com/archive/mapreduce-osdi04.pdf>