SORBONNE UNIVERSITÉ

# Réponse au question du projet Moteur d'exécution de workflows

Tarik Atlaoui

05 Mai 2020

**1. Pourquoi ne peut-on pas utiliser le nom de la méthode en tant qu'identifiant de tâche ?**

Si on utiliser le nom d'une méthode en tant qu'identifiant de tâche l'utilisateur ne pourrais pas surcharger les méthodes. Car deux méthode avec le meme nom mais des parametre différent représenterais que un seul noeud

**Exercice 3**
**2. Donner les grandes lignes de votre algorithme de la méthode execute. Il est attendu un algorithme décrit de manière synthétique et non un copier/coller de votre code.**

**Algorithm 1** Algorithm Exercice 3

```
1:  procedure EXECUTE(a, b)
2:      G ← Graph < String >
3:      C ← Context < String, Object >
4:      Mtab ← Method[]
5:      result ← Map < String, Object >
6:      for <node in Graph> do                          ▷ ont execute les method racine
7:          <Chercher la méthode qui correspond au node>
8:          if nodeParam is in C or nodeNbParam == 0 then
9:              result ← resultat
10:         else if si il y au moins une dépendance qui n'est pas dans C then
11:             Mtab ← methode                          ▷ la méthode de node
12:         else
13:             <invoke methode de node en recupérant les parametre dans C>
14:             <ou sans parametre>
15:         end if
16:     end for
17:
18:     for <m in Mtab> do          ▷ Ont execute les méthode qui présenter des dépendance
19:         <invoke m en recupérant les parametre dans G et C>
20:     end for
21:     return result
22: end procedure
```

Pour résumer l'algorithme utiliser , au debut en recuper touts les resultat des tache racine tout en gardant les méthode des autre tache qui peuve pas encore etre executer ,puis ont parcour les méthode réstante et en les execute.

**Exercice 4**
**1. Décrire comment vous avez assuré le parallélisme des tâches indépendantes.**

Nous avons considére chaque noeud du graphe et lui avons dédier un thread , il commence par chercher la méthode qu'il doit appeler, puis pour chaqu'un de c'est argument essaye de le recupérer , si ce dernier n'existe pas encore il attend une fois que il a fini de recupérer la valeur de c'est arrgument il invoke la fonction et mets le resultat dans la map

**Exercice 5**

**1. Décrire votre protocole de communication**

Le JobExecutorRemote demande demande le calcule d'un job a une jvm distante puis il peux choisir celle ci lui renvois un tuple contenant le nombre de tache executer ainsi que la Map String , Object calculer

**2. Justifier votre choix de l'API de communication que vous avez utilisée.**

Nous avons choisie Rmi , car elle paraissait etre la plus adapter au problem pas besoin de gérez une base de donner juste l'execution de tache sur une ou plusieur Jvm, aussi elle incluse avec java donc pas besoin d'installation préalable ce qui est un plus.

**3. Quelle hypothèse doit-on faire sur le type des objets du contexte ? Justifiez.**

Ont doit fair l'hypothése que les type utiliser dans "Object" sont sérializable car sinon en peux pas les transmetre d'une machine a une autre.

**4. Quel mécanisme avez-vous utilisé pour la notification d'avancement au client ?**

Dans le host qui fait le calcule il met une variable a false avants le calcule et a vrai apres des que la tache est fini il la place a true (l'idéal c'est de gardez une trace car si deux jobe son demander la notification peux etre pertuber)

**Exercice 6**

**1. Quel est le protocole de communications entre maître-esclaves et éventuellement entre esclave-esclave ?**

1-le Maitre recois la demande a qui il affect un id
2- le client demande sont resultat et ce mets en attente si il existe pas encore
3-il lance un thread qui va chercher des esclave a qui il va demander le travail , il demande a chaqu'un si il a des slot disponible
4-une fois trouver il lui passe le nom des taches qu'il doit executer ainsi que le Job
5-une fois terminer il mette la valeur de retour sur Taskmaster et décrémente un cpt des que celui ci est a 0 le client est libre de recupérér sont Job

**2. Comment le maître affecte ses tâches équitablement sur les esclaves ?**

a chaque demande il déplace l'index du prochain esclave qui va recevoir la ou les tache

**3. Comment les esclaves sont assurés de respecter leur borne de tâche courante ?**

pour chaque tache l'esclave lance un Thread au début il decrémente la variable maxtask et la fin l'incrémente

**4. Comment gérez-vous la communication d'un résultat entre deux tâches dépendantes ? Don- ner les avantages et les inconvénients de votre solution.**

Elle communique par le billet de la map qui va etre retourner a l'utilisateur a chaque fois que une tache ce termine elle mets sont resultat a l'intérieur et ce qui ont besoin d'un de c'est resultat wait dessu jusque leur resultat soit disponible elle facilite la récupération des tache puisque elle sont toute centraliser

**5. (niveau 2) Comment gérez-vous le fait d'avoir plusieurs jobs en cours d'exécution sur le cluster ?**

le master ne fait que recevoir les job le reste est fait par des thread

**6. (niveau 3) Décrire le mécanisme qui permet de détecter la panne d'un esclave. Comment gérez-vous la réaffectation des tâches perdues ?**