

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 utilise le nom d'une méthode en tant qu'identifiant de tâche l'utilisateur ne pourrait pas surcharger les méthodes. Car deux méthodes avec le même nom mais des paramètres différents représenteraient que un seul nœud

2. Donner les grandes lignes de votre algorithme de la méthode exécutée. 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 exécuté les méthodes racine et stock
7:     <do stuff>                                             ▷ et si elle a des dépendances linkFrom on la stock dans Mtab
8:     if nodeParam is in C or nodeNbParam == 0 then
9:       result ← resultat
10:    end if
11:  end for
12:  for <m in Mtab> do
13:    <invoke m en récupérant les paramètres dans G et C>
14:  end for
15:  return result
16: end procedure
```

3. Décrire comment vous avez assuré le parallélisme des tâches indépendantes.

Nous avons considéré chaque nœud du graphe et lui avons dédié un thread, il commence par chercher la méthode qu'il doit appeler, puis pour chacun de ces arguments essaye de le récupérer, si ce dernier n'existe pas encore il attend une fois que il a fini de récupérer la valeur de ces arguments il invoque la fonction et met le résultat dans la map

1. Décrire votre protocole de communication

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

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

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

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

2. Comment le maître affecte ses tâches équitablement sur les esclaves ?
3. Comment les esclaves sont assurés de respecter leur borne de tâche courante ?
4. Comment gérez-vous la communication d'un résultat entre deux tâches dépendantes ? Donner les avantages et les inconvénients de votre solution.
5. (niveau 2) Comment gérez-vous le fait d'avoir plusieurs jobs en cours d'exécution sur le cluster ?
6. (niveau 3) Décrire le mécanisme qui permet de détecter la panne d'un esclave. Comment gérez-vous la réaffectation des tâches perdues ?