

Solution générique de calcul GRID
exploitant des messageries instantanées
(Java / Python, XML, XMPP / IRC)

Joffrey Hérard

Responsable : Olivier Flauzac

2016-2017

Table des matières

1	Introduction	4
2	Les Acteurs	5
3	Les Échanges	6
4	Les Problèmes	7
5	Les Erreurs	8
5.1	Les Problèmes d'exécution	8
5.2	Les Problèmes réseaux	9
5.3	Gestions des erreurs	10
5.3.1	Gestions des erreurs sur l'exécution	10
5.3.2	Gestions des erreurs sur le réseaux	10
6	Modélisation	11
6.1	Connexions	11
6.1.1	Connexions du/des Provider(s)	11
6.1.2	Connexions du/des Worker(s)	11
6.2	Représentation des JOBS	12
6.3	Représentation des fichiers de lignes de commande	13
6.4	Les différentes fonctions principales	14
6.4.1	Contraintes	14
6.4.2	Split	14
6.4.3	Exec	14
6.4.4	Build	14
6.5	Description d'une exécution quelconque	15
6.6	Gestions des erreurs	16
7	Conclusion	17

8	Annexes	18
8.1	Organisation du Projet	18
8.1.1	Outils et langages	18
8.1.2	Versionnage	18
8.2	Execution	19
8.3	Code	20

Chapitre 1

Introduction

Sujet : Solution générique de calcul GRID exploitant des messageries instantanées (Java / Python, XML, XMPP / IRC) Durant ce TER, la mise en place d'un système de calcul repartie entre plusieurs machine avec l'évaluation de possibilité d'exécutions ou non par la machine cible, il fallait aussi évaluer quels échanges allaient être réalisés par les acteurs durant une exécution type et ceci en avec le protocole XMPP ou IRC .

Chapitre 2

Les Acteurs

Nous avons donc deux genres d'acteur pour chaque travail différent disponibles

- Fournisseur de travail/Provider, unique pour chaque travail.
- Des travailleurs/Workers, de 1 à n , n définit par le problèmes.

Chapitre 3

Les Échanges

Voici la liste des différents message qui transitent a travers une exécution type.

1. Nous avons en premier le message de type "ENVOI JOB" il contient :
 - l'identifiant du problème,
 - Le code des contraintes,
 - Le code a exécuter,
 - La ligne de commande pour l'exécuter.
2. Ensuite il y a le message ou le workers signale qu'il est prêt il contiens juste un message pour signale dans une chaîne de caractère " Je suis prêt".
3. Il y a enfin le message qui renvoi le résultat "REPONSE JOB" il contient :
 - L'identifiant pour savoir si le code a pu être exécuté.
 - L'identifiant du problème.
 - La valeur du retour de l'exécution.
 - Code de contraintes, si on a pas pu exécuter .
 - Code exécutable, si on a pas pu exécuter .
 - Ligne de commande associe, si on a pas pu exécuter .

Voici la liste des fichiers schéma XML associe ainsi que leurs locations au sein du projet :

- "ENVOI_JOB" = ../Schema_XML/ENVOI_RECEPTION.xsd.
- "READY"= ../Schema_XML/ENVOI_RECEPTION.xsd
- "REPONSE_JOB"= ../Schema_XML/ENVOI_RECEPTION.xsd.

Tout les codes du projet sont présenter en annexe.

Chapitre 4

Les Problèmes

Nous allons dans cette partie du rapport montrer la représentation nécessaire et désirer pour représenter un travail. Donc un problèmes c'est quoi ?

- Identifiant d'un problème, un entier de 0 a n.
- Code de contraintes, ce code est forcement un code Perl avec un code de retour bien particulier 0 pour non exécutable et 3 pour exécutable et donc que l'on peut exécuter.
- Type du fichier par exemple ".c , .cpp, .cc, .java , .pl etc.."
- Code d'exécution, peut importe son langage. on peut l'exécuter si le code contraintes l'a valider
- La ligne de commande pour exécuter le code par exemple "perl monfichier.pl"

Fichier du schéma XML correspondant → = ../Schema_XML/DB_JOB.xsd. Tout les codes du projet sont présenter en annexe.

Chapitre 5

Les Erreurs

5.1 Les Problèmes d'exécution

Les problèmes qui peuvent opérer à travers le système, sont d'abord pour la partie XMPP :

1. Mauvais nom de domaine
2. Problème de Chatroom déjà existante
3. Problème d'exécution : aucun worker peut exécuter le code, comment le détecter ?
- 4.

Les problèmes qui peuvent opérer à travers le système, sont d'abord pour la partie IRC :
Nous avons exactement les mêmes soucis

5.2 Les Problèmes réseaux

Nous avons plusieurs problèmes lie au réseaux quelque soit le protocole utilise :

1. Latence/Impossible a établir une connexion a la Chatroom
2. Latence/Impossible a envoyer un message d'un Provider vers un Worker
3. Latence/Impossible a envoyer un message d'un Worker vers un Provider
4. Un Worker est déconnecte en plein milieu de sa tache
5. Un Provider est déconnecte durant l'attente d'une réponse sur un JOB

5.3 Gestions des erreurs

5.3.1 Gestions des erreurs sur l'exécution

1. Mauvais nom de domaine → Redemander le nom de domaine jusqu'à validation .
2. Problème de Chatroom déjà existante → Message d'erreur un problème exactement identique est en cours d'exécution .
3. Problème d'exécution : aucun worker peut exécuter le code, comment le détecter ?
→ Mis en place d'un tableau de variable booléenne au départ initialisé à faux, si un worker renvoie avec une impossibilité d'exécution du code dictée par le code contraint, alors on met à vrai et on redistribue. Si aucun est capable on arrête l'exécution.

5.3.2 Gestions des erreurs sur le réseau

1. Latence/Impossible d'établir une connexion à la Chatroom → Message qui explicite le fait d'aller voir un Administrateur Réseaux
2. Latence/Impossible d'envoyer un message d'un Provider vers un Worker → Message qui explicite le fait d'aller voir un Administrateur Réseaux
3. Latence/Impossible d'envoyer un message d'un Worker vers un Provider → Message qui explicite le fait d'aller voir un Administrateur Réseaux
4. Un Worker est déconnecté en plein milieu de sa tâche → Détecteur de présence permis par le protocole XMPP sur une ChatRoom MultiUser
5. Un Provider est déconnecté durant l'attente d'une réponse sur un JOB → Évaluation de présence d'un Provider, si aucun alors arrêter le Job en cours, ou mis en place d'un Timeout.

Chapitre 6

Modélisation

6.1 Connexions

6.1.1 Connexions du/des Provider(s)

6.1.2 Connexions du/des Worker(s)

6.2 Représentation des JOBS

6.3 Représentation des fichiers de lignes de commande

6.4 Les différentes fonctions principales

6.4.1 Contraintes

6.4.2 Split

6.4.3 Exec

6.4.4 Build

6.5 Description d'une exécution quelconque

6.6 Gestions des erreurs

Chapitre 7

Conclusion

Chapitre 8

Annexes

8.1 Organisation du Projet

8.1.1 Outils et langages

8.1.2 Versionnage

8.2 Execution

8.3 Code