

UNIVERSITE DE NAMUR

FACULTE D'INFORMATIQUE



Déploiement du workflow de test de JHipster 4.8.2 sur IGRIDA

Auteur Armando koko NGUEPI

Superviseur Acher Mathieu

Promoteur Perrouin Gilles

Devroey Xavier

05 Février 2018

1 Introduction

L'exécution des différentes configurations c'est effectué sur la grille de calcul interne à l'IRISA (IGRIDA). Pour atteindre cet objectif, nous avons écrit un certain nombre de script nous permettant de déployer le workflow d'exécution sur cette infrastructure. Le but de ce rapport est de présenter IGRIDA, de donner un aperçu du fonctionnement du script de déploiement ainsi que du lancement et de la récolte des résultats d'exécution.

2 IGRIDA: description et quelques notions utiles

IGRIDA est une grille de calcul mise à la disposition des équipes de recherche de l'IRISA / INRIA, à Rennes. [INR] Il est composé de 138 nœuds de calcul (1700 cœurs) et d'un espace de travail partagé de 3,4 To. Il offre également 5 nœuds GPU avec GPU NVIDIA [INR].

Les tâches sur IGRIDA sont planifiées par le gestionnaire de ressources et de tâches OAR. Il existe sur le site officiel un lien permettant de consulter la charge CPU et les différents jobs actifs sur la grille [INR].

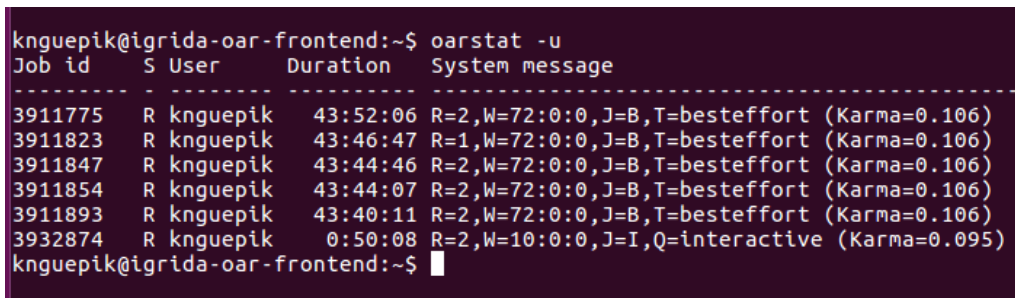
2.1 Les machines virtuelles sur IGRIDA

IGRIDA met à notre disposition des images de système (Ubuntu 16.04 amd64, Ubuntu 17.04 amd64, Debian 8.7 amd64, Debian 9.0.0 amd64, Alpine Linux 3.4.3 amd64 + Docker) et un outil pour charger et démarrer c'est différentes images (veertuosa). Démarrer ubuntu par exemple se ferai en quatre étapes. Supposons qu'on soit déjà connecté au réseau IRISA soit en local ou en VPN

1. se connecter à IGRIDA: «`ssh $user@igrida-oar-frontend`»
2. créer un job (dans ce cas qui utilise deux coeurs d'une machine et qui dure une heure). «`oarsub -I -l core=2,walltime=1`»
3. charger la machine virtuelle Ubuntu. «`module load veertuosa/0.0.1 && veertuosa_launch -name myvm -image /soft/igrida/vms/images/ubuntu-16.04-x86_64.qcow2`»
4. démarrer et se connecter à la machine virtuelle: «`ssh-vm myvm`» pour toute information voire [igrida web site](#)

2.2 Quelques commandes utiles

1. «`oarstat`» permet de visualiser l'état des jobs en cours d'exécution (R=run, W=wait, F=fail) combiné avec l'option «-u» affiche uniquement les jobs appartenant à l'utilisateur courant.



```
knguepik@igrida-oar-frontend:~$ oarstat -u
Job id      S User      Duration  System message
-----
3911775     R knguepik  43:52:06 R=2,W=72:0:0,J=B,T=besteffort (Karma=0.106)
3911823     R knguepik  43:46:47 R=1,W=72:0:0,J=B,T=besteffort (Karma=0.106)
3911847     R knguepik  43:44:46 R=2,W=72:0:0,J=B,T=besteffort (Karma=0.106)
3911854     R knguepik  43:44:07 R=2,W=72:0:0,J=B,T=besteffort (Karma=0.106)
3911893     R knguepik  43:40:11 R=2,W=72:0:0,J=B,T=besteffort (Karma=0.106)
3932874     R knguepik   0:50:08 R=2,W=10:0:0,J=I,Q=interactive (Karma=0.095)
knguepik@igrida-oar-frontend:~$
```

Figure 1: Lister les jobs

2. «`oarsub -C Job-Id`» permet de connecter à la machine sur laquelle s'exécute le job d'identifiant Job-Id.
3. La commande «`oardel Job-Id`» permet de stopper et supprimer le job d'identifiant Job-Id

4. La commande «**rsync -avz /my-local-data login@igrida-oar-frontend:/temp_dd/igrida-fs1/login**», permet de copier le dossier « **my-local-data**» depuis votre répertoire local vers votre répertoire personnel sur IGRIDA. Vous pouvez inversement copier des fichiers depuis IGRIDA vers votre répertoire local
5. Les jobs Array: ils sont utilisés lorsqu'on désire lancer plusieurs instances d'un même programme sur des machines différentes avec éventuellement des paramètres différents. Nous verrons plus bas que ceci est notre cas.

3 Comment déployer le workflow sur IGRIDA ?

Le fichier compressé fourni sur GitHub contient tous les scripts nécessaires au déploiement du workflow. L'image ci-dessous décrit l'organisation du script.

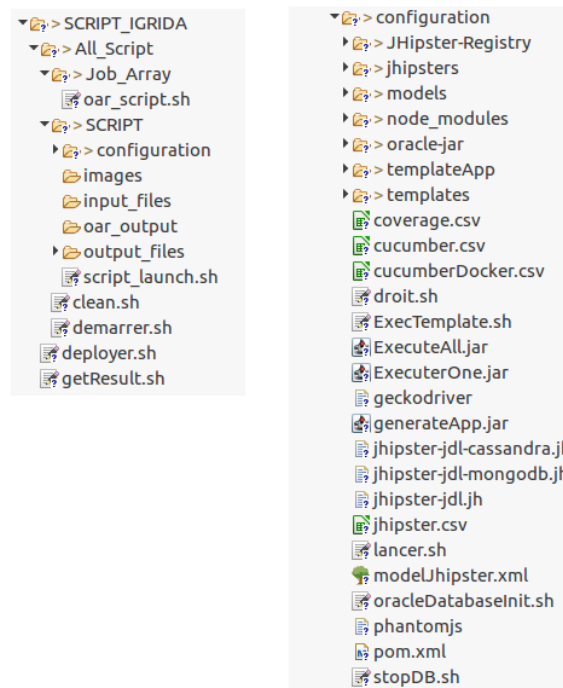


Figure 2: Arborescence du script de déploiement sur IGRIDA

3.1 Fonction des scripts

Le dossier fourni contient un répertoire trois scripts

- Le fichier «**Script_a_Deployer/Deployer.sh**» Il permet de déployer tous les autres scripts dans notre espace personnel
- Le fichier «**Script_a_Deployer/geResult.sh**» Il permet de récolter les résultats de l'exécution depuis IGRIDA et les stocker dans le répertoire «**Script_a_Deployer/RESULTAT**».
- Les configurations sont Générées grâce à l'exécution du script «**generate.sh**»
- La création d'un job array pour l'exécution des tâches de compilation et build se fait grâce au script «**Script_a_Deployer/AllScript/Job_Array/oar_script.sh**»
- Nettoyer les répertoires résultats afin de redéployer à nouveau les scripts sur IGRIDA grâce au script «**Script_a_Deployer/All_Script/clean.sh**»
- Demarrer l'exécution des tâches de compilation et de build sur un nombre de configuration pris en entrée grâce au script «**All_Script/demarrer.sh**»

- Se connecter à la VM afin de lui passer les paramètres d'exécution (nombre de configuration à exécuter) «**Script_a_Deployer/All_Script/SCRIPT/script_launch.sh**»
- Le répertoire «**/SCRIPT/configuration/**» contient les différentes configurations obtenues. Il contient aussi les fichiers utiles à l'exécution du workflow.
- Le repertoire «**Script_a_Deployer/All_Script/SCRIPT/input_files**» Il contient les fichiers passés en entrée au Job Array
- Le repertoire «**Script_a_Deployer/All_Script/SCRIPT/oar_output**» contient les logs relatifs à l'exécution des scripts sur IGRIDA
- Le repertoire «**Script_a_Deployer/All_Script/SCRIPT/output_files**» contient les logs relatifs à l'exécution du workflow sur chaque configuration.

3.2 Deployer les scripts

1. Si vous avez déjà été connecté une fois sur IGRIDA (clé publique est déjà générée) alors passer à la commande suivante sinon: se connecter à IGRIDA voire [igrida web site](#). «**ssh \$user@igrida-oar-frontend**»
2. Ouvrir un terminal et se placer dans le répertoire contenant les scripts. La commande «**sh deployer.sh**» permet de déployer les scripts sur IGRIDA. Notez que vous pouvez générer les configurations en local ou sur IGRIDA (une fois les scripts déployés). Dans le premier cas, vous êtes sûr de les générer toutes, mais cela vous coûtera assez de temps pour générer et ensuite deployer. Dans le dernier cas la génération des configurations peut être interrompue à cause des problèmes liés à la machine distante sur laquelle vous êtes connecté.

```
koko@koko-Lenovo-G50-70:~/Bureau/SCRIPT_IGRIDA$ ./deployer.sh
saisir votre login Igrida :
kngueptk

*****Copie des repertoires de configuration*****
```

Figure 3: Déploiement des scripts sur IGRIDA

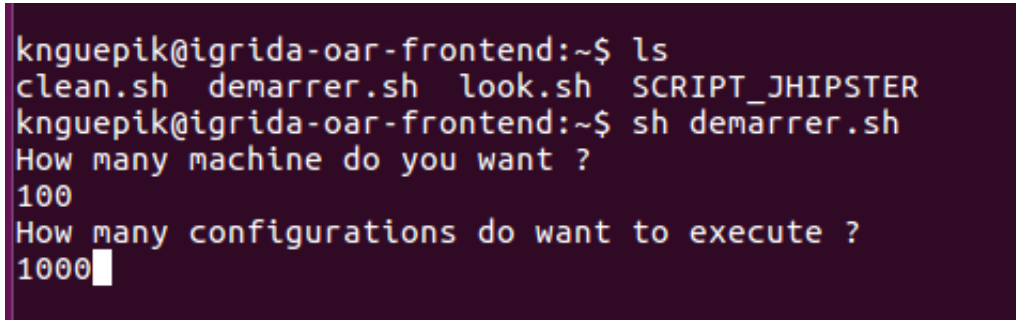
3.3 Preparation de la VM

IGRIDA met à notre disposition un ensemble de système d'exploitation et des outils pour les démarrer comme machine virtuelle. Nous avons choisi dans ce cas d'utiliser le système ubuntu 16 embarqué dans IGRIDA. copie de l'image

1. Se connecter à IGRIDA et copier la VM dans notre répertoire personnel avec cette commande «**cp /soft/igrida/vms/images/ubuntu-16.04-x86_64.qcow2 /temp_dd/igrida-fs1/\$user/SCRIPT_JHIPSTER/images/ubuntu.qcow2**»
2. Créer un job interactif d'une heure ou plus en fonction de si vous désirez générer les configurations sur IGRIDA ou pas. La commande «**oarsub -I -l core=2,walltime=1 -p "virt='YES'"**»
3. Une fois le job crée , si vous avez choisi de générer les configurations sur IGRIDA alors exécuter les commandes si bas, sinon passer à l'étape suivante« **sh generate.sh** »
4. Se connecter à la VM avec les commandes suivantes afin d'y installer les outils nécessaires. «**module load veertuosa/0.0.1 && veertuosa_launch -name myvm -image /temp_dd/igrida-fs1/\$user/SCRIPT_JHIPSTER/images/ubuntu.qcow2 -persistent && ssh-vm myvm**»

4 Demarrer l'exécution des configurations

1. Depuis un terminal local, se connecter à IGRIDA «`ssh $user@igrida-oar-frontend` » et exécuter la commande suivante «`./demarrer.sh` » .Cette derniere vous demandera de fournir le nombre de job (nombre de machine) que vous aimeriez reserver, ensuite le nombre de configuration à exécuter. La repartition des configurations sur les machines se fait de manière proportionnel.



```
knguepik@igrida-oar-frontend:~$ ls
clean.sh demarrer.sh look.sh SCRIPT_JHIPSTER
knguepik@igrida-oar-frontend:~$ sh demarrer.sh
How many machine do you want ?
100
How many configurations do want to execute ?
1000
```

Figure 6: reserver des machines et demarrer des jobs

Remarque: Vous pouvez modifier la durée d'allocation des machines en modifiant le fichier «**SCRIPT_JHIPSTER/oarscript**» se trouvant à racine du votre repertoire personnel.



```
#!/bin/bash
USER="knguepik"
#OAR -t besteffort
#OAR -l core=1,walltime=72:00:00
#OAR -p virt='YES'
#OAR --array-param-file /temp_dd/igrida-fs1/knguepik/SCRIPT_JHIPSTER/input_files/params.txt
#OAR -O /temp_dd/igrida-fs1/knguepik/SCRIPT_JHIPSTER/oar_output/job.%jobid%.output
#OAR -E /temp_dd/igrida-fs1/knguepik/SCRIPT_JHIPSTER/oar_output/job.%jobid%.error

. /etc/profile.d/modules.sh

set -x

module load veertuosa/0.0.1
VM_NAME=vm_${OAR_JOBID}
veertuosa_launch --name ${VM_NAME} --image /temp_dd/igrida-fs1/${USER}/SCRIPT_JHIPSTER/images/ubuntu.qcow2

VM_CMD="cd /mnt/temp_dd/igrida-fs1/${USER}/SCRIPT_JHIPSTER && sh -x -x ./script_launch.sh $* ${OAR_JOBID}"

ssh -vm $VM_NAME "$VM_CMD"
```

Figure 7: modifier la durée de reservation d'une machine

2. Une fois le script exécuté, il se crée un nombre de job égale au nombre de machine demandé. Chaque job correspondant à l'exécution d'un ensemble de configuration sur une machine durant le temps que auriez défini. vous pouvez verifier l'état des jobs crée grâce à la commande «**oarstat -u**»
3. Consulter les erreurs d'exécution liées à IGRIDA dans le repertoire «**/temp_dd/igrida-fs1/knguepik/SCRIPT_JHIPSTER/oar_output/** »
4. Consulter les Logs retournés par les differantes executions dans le repertoire «**/temp_dd/igrida-fs1/knguepik/SCRIPT_JHIPSTER/output_files/**»

5 Comment recolter les resultats ?

Recuperer les resultats d'exécutions consiste à copier le fichier «**.csv**» creer sur la grille ainsi que les fichiers logs relatifs à chaque traitement. Pour le faire, il suffit d'exécuter le script «**getResult.sh**» fournit. Il se créera alors le repertoire «**All_Script/RESULTAT**» contenant le csv et les logs.

Vous pouvez supprimer toutes les exécutions précédentes en exécutant le script «**clean.sh**» depuis votre repertoire personnel. Ceci supprimera la trace des anciennes exécutions, sauf le repertoire contenant les configurations.

References

[INR] INRIA/IRISA. <http://igrida.gforge.inria.fr>.