



Comment comparer 2 intergiciels de Grilles/de Cloud sur Grid'5000

Découvrir Grid'5000 à travers une expérience imaginée

David Margery



Concevoir son expérience

- Grid'5000 est un instrument scientifique qui peut être utilisé pour mesurer beaucoup de choses
 - Quels sont les paramètres à faire varier ?
 - Quels sont les conditions expérimentales à mesurer ?
 - Quels sont les éléments à figer durant la campagne de mesure
- La plonger dans Grid'5000
 - Parcourir la doc ou utiliser la référence API pour découvrir les ressources utiles pour l'expérience



L'API de référence

- Une description versionnée de Grid'5000 pour programmes
 - Connaître la description de Grid'5000 à une date donnée est possible
 - Une décalage entre la description et la réalité reste possible
- Une API REST
 - Utilise HTTP comme couche applicative
 - Codes d'erreurs normalisés
 - Serveurs robustes, caches, proxies..
 - Utilise les verbes HTTP (GET, PUT, POST, DELETE)
 - Toutes les interfaces de Grid'5000 écrites utilisant ce paradigme
 - Assez indépendante du langage utilisé pour l'accès. Il suffit que ce langage dispose d'une bibliothèque parlant le HTTP..



Expérience imaginée

- Utilisation de la structuration en clusters de Grid'5000
 - Un nœud du cluster sera élu nœud de service
 - Le reste constituera les ressources (nœuds)
- On pourrait avoir plus de clusters pour l'expérience, et moins de nœuds par clusters
 - Les ressources des clusters seront simulées, mais les liens entre clusters en réseau local ?
- Les intergiciels comparés
 - Une autorité de certification
 - Un serveur par cluster utilisé
 - Un script pour déclarer les nœuds au serveur existe



Déployer l'intergiciel (1)

- Choix d'une stratégie

- Une image kadeploy pour autorité +serveur
- Une image kadeploy pour les noeuds

- Installation

- Il faut **développer** chaque image
 - Installation d'une image de référence sur deux nœuds
 - Accès à cette image en tant que root
 - Installation de la pile logicielle (autorité +serveur | noeud)
 - Sauvegarde de l'image installée
 - Test en déploiement sur deux autres nœuds
- C'est un processus assez coûteux pour certains intergiciels qui n'ont pas été prévu pour cela (la mode du Cloud pourrait nous aider)
- Certaines phases de l'installation supposeront que le nom des serveurs, et leurs IP ne vont pas changer
 - En particulier la gestion de certificats, mais pas seulement!



2 stratégies pour la certification

- En général, la chaîne de confiance va supposer que l'IP et le nom DNS des machines ne va pas changer
 - Scripter toute phase d'installation avec une dépendance au nom de machine ou à son IP
 - Attention à écrire des scripts passant à l'échelle d'un déploiement de plusieurs centaines de machines
 - Fait en janvier 2008 pour une image globus où chaque nœud devait faire signer son certificat
 - Fixer l'IP en utilisant un pool d'IP pour la virtualisation
 - Il faudra alors traduire les noms DNS des noeuds utilisés au sens Grid'5000 en nom dans le domaine `virtual.site.grid5000.fr` pour l'expérience
 - Il faut changer l'IP des machines à la volée lors du boot



Réserver les ressources

- Oargrid
 - Pour obtenir une topologie bien définie
- Disco
 - Pour obtenir le maximum des ressources d'un certain type disponibles à un instant donné, pour une période donnée.
- Monitoring API
 - Pour écrire soit même son algorithme de découverte de ressources
- Au final, sur un site, c'est OAR qui travaille
 - Une API est en cours de déploiement pour éviter la ligne de commande
- A venir
 - Obtenir les mêmes ressources que la dernière fois ?
- Scripter le déploiement



Simplified OAR2.4_deploy.sh

- SSH_OPTIONS='-o StrictHostKeyChecking=no '
- #elect the frontend
- OAR_MASTER=`head -1 \$OAR_FILE_NODES`
- ...
- kadeploy3 -a lenny2.6.26-OAR2.4_Serv-Front.env -m \$OAR_MASTER -k ~/.ssh/id_rsa.pub --output-ok-nodes \$OAR_MASTER_FILE &
- kadeploy3 -a lenny2.6.26-OAR2.4_Node.env -f \$OAR_SLAVES_FILE -k ~/.ssh/id_rsa.pub --output-ok-nodes \$DEPLOYED_SLAVES_FILE &
- wait
- wait
- ... (do some checks)
- scp \$SSH_OPTIONS \$DEPLOYED_SLAVES_FILE root@\$OAR_MASTER:/tmp/slave_nodes
- ssh \$SSH_OPTIONS root@\$OAR_MASTER 'yes yes | oar_resources_init /tmp/slave_nodes'
- ssh \$SSH_OPTIONS root@\$OAR_MASTER source /tmp/oar_resources_init.txt



Réaliser l'expérience

- Qualifier les noeuds obtenus
 - Une étape trop souvent oubliée
 - Un disque ou un processeur ralenti, un lien réseau au débit mal négocié peuvent-ils affecter la validité de vos résultats ?
- Lancer les expériences
- Collecter les données, et les conditions expérimentales
 - État des liens réseau intersites pendant l'expérience
 - Metroflux (à annoncer)
 - Métriques ganglia
 - Les noeuds peuvent publier des métriques
 - Une API permet de récolter les données
 - À venir : une vue globale, par réservation.
- Conclure, et publier ?



Conclusion

- Grid'5000 est un instrument scientifique
 - Qui permet la comparaison de logiciels de Grilles, de clouds, etc...
 - C'est une capacité assez unique
 - Qui permet de mener des expériences reproductibles
 - Pourvu que les noeuds ne soient pas en panne, ou partis à la retraite
- Mener des expériences solides est possible
 - Nécessite une bonne méthodologie générale
 - Nécessite d'investir pour mettre au point le déploiement de ces logiciels
 - Automatiser le déploiement de systèmes distribués nécessite de l'expertise et un investissement initial
 - Partager cet investissement serait un plus, mais a aussi un coût
- Grid'5000 devrait être l'outil idéal pour la recherche en grilles
 - Je suis aussi ici pour comprendre les freins.