Introduction à Grid'5000 pour la réalisation d'expériences

Anne Cadiou[1] Simon Delamare[2] Laurent Pouilloux[1]

[1]Laboratoire de Mécanique des Fluides et d'Acoustique (LMFA), Lyon
CNRS

[2]Laboratoire de l'informatique du parallélisme (LIP), Lyon
CNRS

Utilisation de Grid'5000 pour la réalisation de benchmarks
8 octobre 2020

Plan

Expérimentation ?

La réalisation ≪d'expériences≫ est incontournable en informatique

- pour valider un déploiement, étudier les performances, tester...
- objet d'étude : un logiciel, une configuration, une infrastructure complexe

Expérimentation ?

La réalisation ≪d'expériences≫ est incontournable en informatique

- pour valider un déploiement, étudier les performances, tester...
- objet d'étude : un logiciel, une configuration, une infrastructure complexe

Où ?

- Sur sa machine perso, avec de la virtualisation, etc.

Expérimentation ?

La réalisation ≪d'expériences≫ est incontournable en informatique

- pour valider un déploiement, étudier les performances, tester...
- objet d'étude : un logiciel, une configuration, une infrastructure complexe

Où ?

- Sur sa machine perso, avec de la virtualisation, etc.
  → Ressources insuffisantes pour un déploiement large échelle

Expérimentation ?

La réalisation ≪d'expériences≫ est incontournable en informatique

- pour valider un déploiement, étudier les performances, tester...
- objet d'étude : un logiciel, une configuration, une infrastructure complexe

Où ?

- Sur sa machine perso, avec de la virtualisation, etc.
  → Ressources insuffisantes pour un déploiement large échelle
- Sur le centre de calcul

Expérimentation ?

La réalisation ≪d'expériences≫ est incontournable en informatique

- pour valider un déploiement, étudier les performances, tester...
- objet d'étude : un logiciel, une configuration, une infrastructure complexe

Où ?

- Sur sa machine perso, avec de la virtualisation, etc.
  → Ressources insuffisantes pour un déploiement large échelle
- Sur le centre de calcul
  → Manque de flexibilité pour reconfigurer, pas de droits *root*

Expérimentation ?

La réalisation ≪d'expériences≫ est incontournable en informatique

- pour valider un déploiement, étudier les performances, tester...
- objet d'étude : un logiciel, une configuration, une infrastructure complexe

Où ?

- Sur sa machine perso, avec de la virtualisation, etc.
  - → Ressources insuffisantes pour un déploiement large échelle
- Sur le centre de calcul
  - → Manque de flexibilité pour reconfigurer, pas de droits *root*
- Dans un Cloud

Expérimentation ?

La réalisation ≪d'expériences≫ est incontournable en informatique

- pour valider un déploiement, étudier les performances, tester...
- objet d'étude : un logiciel, une configuration, une infrastructure complexe

Où ?

- Sur sa machine perso, avec de la virtualisation, etc.
  → Ressources insuffisantes pour un déploiement large échelle
- Sur le centre de calcul
  → Manque de flexibilité pour reconfigurer, pas de droits *root*
- Dans un Cloud
  → Virtualisation et containers $\neq$ Hardware

Introduction

Expérimentation ?

La réalisation ≪d'expériences≫ est incontournable en informatique

- pour valider un déploiement, étudier les performances, tester...
- objet d'étude : un logiciel, une configuration, une infrastructure complexe

Où ?

- Sur sa machine perso, avec de la virtualisation, etc.
  - → Ressources insuffisantes pour un déploiement large échelle
- Sur le centre de calcul
  - → Manque de flexibilité pour reconfigurer, pas de droits *root*
- Dans un Cloud
  - → Virtualisation et containers ≠ Hardware

→ Mutualisation ? matériel, outils, bonnes pratiques

Expérimentation ?

La réalisation ≪d'expériences≫ est incontournable en informatique

- pour valider un déploiement, étudier les performances, tester...
- objet d'étude : un logiciel, une configuration, une infrastructure complexe

Où ?

- Sur sa machine perso, avec de la virtualisation, etc.
  - → Ressources insuffisantes pour un déploiement large échelle
- Sur le centre de calcul
  - → Manque de flexibilité pour reconfigurer, pas de droits *root*
- Dans un Cloud
  - → Virtualisation et containers ≠ Hardware

→ Mutualisation ? matériel, outils, bonnes pratiques

Grid'5000

- Une plateforme pour l'expérimentation
  - → À destination de la recherche liée à l'informatique
- Dans tous les domaines de l'informatique distribuée
  - HPC, Cloud Computing, Virtualisation, Systèmes distribués, Réseaux, etc.
  - Informatique ≪lourde≫ : Pas d'embarqué, pas de sans-fil.
  - → SILECS
- Propose à ses utilisateurs :
  - Du matériel varié ...
  - ... complètement reconfigurable selon leurs besoins
  - Un ensemble d'outils pour faciliter la réalisation d'expériences

Plan

Plan

Fonctionnement

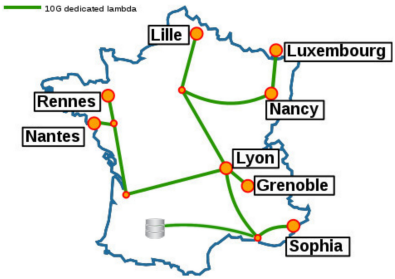
- Groupement d'Intérêt Scientifique
- Contributeurs : Principaux établissements de recherche FR
- Équipes de scientifiques pour orienter les évolutions de la plate-forme
- Équipe technique : 11 ingénieurs ETP

Chiffres :

- 15 ans d'existence
- 600 utilisateurs actifs / an
- 56M cœurs heures distribuées les 12 derniers mois

→ Communauté active : site Web, mailing list, écoles et formations . . .

L'infrastructure Grid'5000



→ 8 sites

→ réseau de cœur 10Gbit/s

Ressources matérielles :

- **Nœuds** groupés en **clusters** présents sur les **sites**
  - ≈ 800 nœuds (15k cœurs), 36 clusters

  → *nova-5.lyon.grid5000.fr*
- Matériel varié :
  - CPUs : Différentes générations Intel (et AMD), ARM64, (et bientôt Power8)
  - GPU Nvidia, Xeon Phi (et bientôt AMD MI50)
  - Réseau : Ethernet 10 Gbit/s, Infiniband, Omni-Path
  - Stockage : SSD, NVMe, PMEM, grands volumes...

Utilisation des Ressources

Accès exclusif au matériel, via un système de réservation

Utilisation des Ressources

Accès exclusif au matériel, via un système de réservation

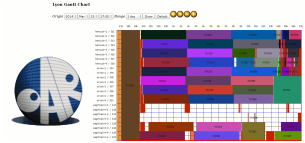
→ Y compris en tant que *root*

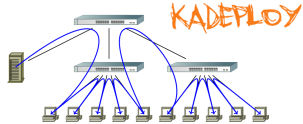
Utilisation des Ressources

Accès exclusif au matériel, via un système de réservation

→ Y compris en tant que *root*

→ Possibilité de déployer son propre OS





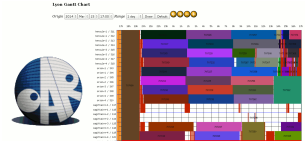
Utilisation des Ressources

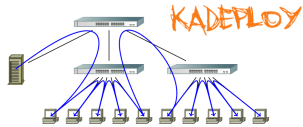
Accès exclusif au matériel, via un système de réservation

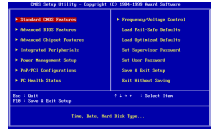
→ Y compris en tant que *root*

→ Possibilité de déployer son propre OS

→ Modification de certains paramètres BIOS (config. CPU pour l'énergie . . .)







Utilisation des Ressources

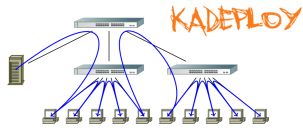
Accès exclusif au matériel, via un système de réservation

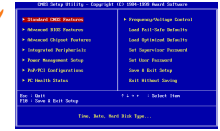
→ Y compris en tant que *root*

→ Possibilité de déployer son propre OS

→ Modification de certains paramètres BIOS (config. CPU pour l'énergie . . .)

→ Isolation réseau pendant l'expérience





Utilisation des Ressources

Accès exclusif au matériel, via un système de réservation

→ Y compris en tant que *root*

→ Possibilité de déployer son propre OS

→ Modification de certains paramètres BIOS (config. CPU pour l'énergie . . .)

→ Isolation réseau pendant l'expérience

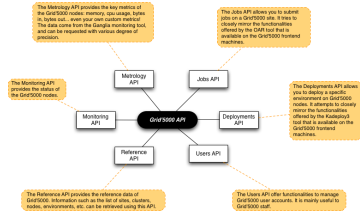
Modèle *Hardware as a Service / Bare-Metal* : on manipule du ≪vrai≫ matériel

Outils pour la réalisation d'expériences

Grid'5000 API :

- Alternative à la ligne de commande
- API REST
- Plusieurs composantes

→ Bibliothèque pour scripter l'utilisation de Grid'5000 dans un langage de haut niveau

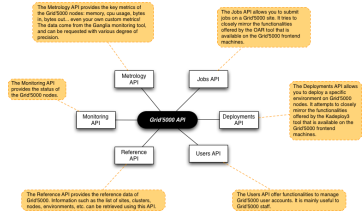


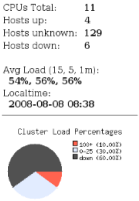
Outils pour la réalisation d'expériences

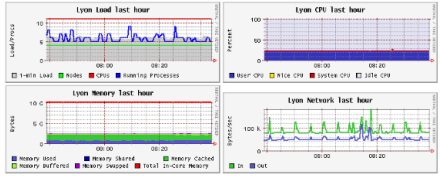
Grid'5000 API :

- Alternative à la ligne de commande
- API REST
- Plusieurs composantes

→ Bibliothèque pour scripter l'utilisation de Grid'5000 dans un langage de haut niveau







Monitoring des performances :

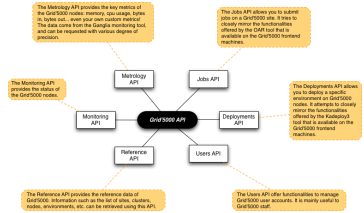
- Métrique ≪classique≫ du nœud : CPU, mémoire . . .
- Mais aussi, électricité consommée à la prise, réseau

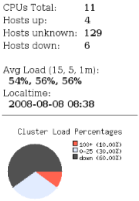
Outils pour la réalisation d'expériences

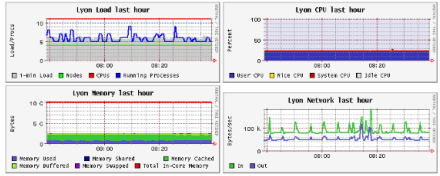
Grid'5000 API :

- Alternative à la ligne de commande
- API REST
- Plusieurs composantes

→ Bibliothèque pour scripter l'utilisation de Grid'5000 dans un langage de haut niveau







Monitoring des performances :

- Métrique ≪classique≫ du nœud : CPU, mémoire . . .
- Mais aussi, électricité consommée à la prise, réseau

Déploiement automatisé d'infrastructures complexes :

- Proposé par l'équipe technique ou par les utilisateurs

→ *OpenStack*, *Cluster Ceph*, *Hadoop*

Plan

Compte et première connexion

Compte Grid'5000 :
→ Vous devriez avoir reçu un email contenant votre **login utilisateur** (ex : *sdelamare*) et vous demandant d'entrer un mot de passe et une **clé SSH**.
Il est ensuite possible de se connecter à la plateforme avec :

```
$ ssh <login>@access.grid5000.fr
```

Pas de clé SSH ?

```
$ ssh-keygen
$ cat ~/.ssh/id_rsa.pub
```

Pas de SSH tout court ?

Si vous êtes dans l'incapacité d'utiliser SSH, il est possible d'accéder à Grid'5000 depuis le navigateur à cette URL (mais ceci est bien moins pratique) :

https://intranet.grid5000.fr/shell/lyon/

Site Web et documentation

→ www.grid5000.fr

Gestion :

- Usage Policy : rêgles d'utilisation (pas plus 2h.coeurs d'un même cluster en journée)
- Support : comment obtenir de l'aide
  s'adresser à la communauté : mailto:users@lists.grid5000.fr
  à l'équipe technique : mailto:support-staff@lists.grid5000.fr
- Manage Account : interface de gestion de son compte (ex : changer la clé SSH)
- Reset Password : si on a oublié son mot de passe

Plateforme :

- Hardware : matériel disponible sur les différents sites
- Gantt : matériel disponible sur les différents sites
- Status : maintenances et incidents en cours sur la plateforme
- API de reference : https://api.grid5000.fr/stable/sites/lyon/clusters/nova/nodes/nova-1

Documentation :

- Users Home : point d'entrée des documentations et tutoriaux
- Getting Started : premiers pas sur Grid'5000

Accès aux sites Grid'5000

On se connecte à la machine d'accès de la plateforme avec :

```
$ ssh <login>@access.grid5000.fr
```

Depuis la machine d'accès on se connecte à un des sites de Grid'5000 (lyon, grenoble, lille, nancy...) :

```
$ ssh lyon
```

Pour copier un fichier depuis sa machine locale vers le site de Lyon, on utilise :

```
$ scp mon_fichier <login>@access.grid5000.fr:lyon/
```

Pour copier un fichier depuis le site de Lyon vers sa machine :

```
$ scp <login>@access.grid5000.fr:lyon/mon_fichier .
```

Les fichiers qui sont placés dans le sous répertoire 'public' de votre répertoire Grid'5000 sont téléchargeable à cette adresse :

https://api.grid5000.fr/stable/sites/lyon/public/<login>/mon_fichier

Accès aux noeuds de Grid'5000, bases

Une fois connecté à la *frontale* d'un site Grid'5000 (*flyon*, par exemple), on peut obtenir un accès aux noeuds disponibles sur ce site :

- Accès interactif à un noeud :

```
$ oarsub -I
```

- . . . spécifiant le cluster :

```
$ oarsub -I -p cluster='nova'
```

- . . . en spécifiant la durée (1h par défaut)

```
$ oarsub -I -l walltime=2:30: -p "cluster='nova'"
```

Attention, la réservation se termine si on quitte le *shell* de connexion ouvert par `oarsub -I`

Accès aux noeuds de Grid'5000, plusieurs noeuds

- Accès interactif à plusieurs noeuds

```
$ oarsub -I -l nodes=3
```

→ la liste des noeuds réservés peut être obtenue avec la commande :

```
$ uniq $OAR_FILE_NODES
```

→ on se connecte d'un noeud à l'autre avec :

```
$ oarsh <node>
```

(pour utiliser directement `ssh`, il faut donner l'option "`-t allow_classic_ssh`" à `oarsub`)

- Un seul coeur

```
$ oarsub -I -l core=1
```

- Un GPU (+ les coeurs associés)

```
$ oarsub -I -l gpu=1
```

Accès aux noeuds de Grid'5000, soumission

Note

OAR est un *Job Scheduler* comme on en trouve dans les centres de calcul, mais il est généralement utilisé différemment : dans le contexte de Grid'5000 on utilise d'avantage de réservation interactives ou de réservations à l'avance alors que dans les centres de calcul, on réalise surtout des soumissions de jobs.

- Soumission d'un job :

```
$ oarsub -p cluster='nova' -l nodes=8 <mon_programme>
```

→ Sera exécuté lorsque les 8 noeuds de nova sont dispos (comme pour l'interactif)

- Soumission d'un job, avec script :

```
$ cat my_script.sh
#!/bin/bash
#OAR -l nodes=2,walltime=3:15:00
#OAR -p "cluster = 'nova'"
#OAR --stdout stdoutfile.log
~/mon_programme
$ oarsub -S my_script.sh
```

Accès aux noeuds de Grid'5000, réservation

- Réservation à l'avance :

```
$ oarsub -r "2020-10-05␣20:00:00" \
    -p "cluster='nova'" -l nodes=8,walltime=4: \
    <mon_programme>
```

Note

En accord avec les règles d'utilisation de Grid'5000, il est fréquent d'utiliser des jobs interactifs en journée, pour la phase de développement d'une expérience et ensuite de faire des soumissions. Pour lancer son expérimentation sur une longue durée et/ou un nombre important de ressource, on utilise généralement des réservations en soirée ou le WE.

→ Pour les soumission, voir aussi : Restricting jobs to daytime or night & week-end

Gestion des jobs

- Depuis la frontale, lister les job en cours :

```
$ oarstat
```

- Obtenir des infos sur le jobs (noeuds, heure de début/fin, etc.)

```
$ oarstat -f -j <numero job>
```

- Pour prolonger une réservation, si la ressource est disponible

```
$ oarwalltime <numero job> +1:30
```

- Pour terminer un job

```
$ oardel <numero job>
```

Que faire depuis le noeuds

- De nombreux logiciels installés par défaut (OS : Debian Stable)
- Logiciels HPC/scientifiques suppémentaires

```
$ module av
```

- Docker & Singularity

```
$ singularity version
$ g5k-setup-docker && docker version
```

- Devenir root

```
$ sudo-g5k
$ sudo apt install ...
```

- Informations sur le job

```
$ cat $OAR_NODE_FILE
$ echo $OAR_JOBID
```

- MPI

```
$ mpirun --mca orte_rsh_agent "oarsh" -machinefile $OAR_NODE_FILE mpi_prog
$ echo $OAR_JOBID
```

- Stockage : Espace disponible dans `/tmp`, sinon voir :
  Group Storage (gros volumes NFS, partagés), Ceph storage service (espace de stockage Ceph), Node disks reservation (réservation des disques locaux)

Déployer son OS

→ Avec l'outil *Kadeploy*

- Réserver ses noeuds avec l'option "-t deploy"

```
$ oarsub -I -t deploy
```

- Choisir un environnement

```
$ kaenv3 -l
```

  - *-min* : rien que le minimum pour fonctionner
  - *-base* : + quelques logiciels, *-nfs* : +compte utilisateurs et accès home
  - *-big* : idem env. par défaut

- Déployer :

```
$ kadeploy3 -k -e <nom environnement> -m <noeud1> -m <noeuds2>
$ kadeploy3 -k -e <nom environnement> -f <fichier avec liste des noeuds>
```

```
$ kadeploy3 -k -e debian10-min -m nova-1
$ kadeploy3 -k -e centos8-x64-min -f $OAR_FILE_NODE
```

- Se connecter :

```
$ ssh root@<node>
```

- Avec l'outil `tgz-g5k`, il est possible de "sauvegarder" l'état d'un noeud pour le redéployer directement ensuite, voir Create a new environment from a customized environment

Aller plus loin

Voir les documentations de User Home, par exemple :

- Scripting : Experiment scripting, REST API guide
- Network reconfiguration : KaVLAN
- Monitoring : Kwollect, Ganglia
- VPN
- HPC related : MPI, GPU, Deep Learning Frameworks, PMEM