Stage de fin d'année

Cloud HPC : Orchestration d'un cluster de Raspberry Pi

Anzid Zakaria Mr. Gerald Becker

Faculté des sciences
Université de Perpignan

MASTER EN CALCUL HAUTE PERFORMANCE, SIMULATION

Université Perpignan Via Domitia
CRÉATRICE D'AVENIRS DEPUIS 1350

Table of Contents

MASTER EN CALCUL HAUTE PERFORMANCE, SIMULATION

Université Perpignan Via Domitia

INTRODUCTION

Introduction aux Clusters HPC

- Systèmes spécialisés pour résoudre des problèmes complexes nécessitant une puissance de calcul substantielle.
- Interconnexion de nombreux ordinateurs travaillant en parallèle pour des calculs et simulations.

MASTER EN CALCUL HAUTE PERFORMANCE, SIMULATION

Université Perpignan Via Domitia

Importance des Clusters HPC

- Traitement plus rapide et plus efficace de vastes quantités de données.
- Division des gros problèmes en parties plus petites, réparties entre les nœuds du cluster (calcul parallèle).

MASTER EN CALCUL HAUTE PERFORMANCE, SIMULATION

Université Perpignan Via Domitia

Applications des Clusters HPC

- Recherche scientifique : Simulation des systèmes complexes (comportement des matériaux, modèles météorologiques, dynamique des fluides).
- Ingénierie : Simulation des structures et systèmes (composants des avions et des automobiles).
- Analyse financière : Analyse des tendances boursières pour des prévisions.
- Apprentissage automatique : Entraînement des réseaux de neurones profonds nécessitant une puissance de calcul significative.

Exemples de Clusters HPC en France

- GENCI (Grand Équipement National de Calcul Intensif) : Gère les ressources de calcul intensif en France, incluant des supercalculateurs du CEA et CNRS.
- Jean Zay : Supercalculateur à l'IDRIS, utilisé pour la recherche en IA, physique, climatologie, etc.

Exemples de Clusters HPC à l'Échelle Mondiale

- Fugaku : Supercalculateur au Japon, utilisé dans la médecine et la modélisation climatique.
- Summit : Supercalculateur aux États-Unis, utilisé pour des recherches en biologie, astrophysique, et IA.

Pourquoi et Quand Choisir un Cluster de Raspberry Pi ?

- Alternative économique aux clusters HPC traditionnels : Bas coût, faible consommation d'énergie, et facilité d'interconnexion.
- Moins puissants que les clusters HPC traditionnels, mais efficaces pour des tâches de calcul distribué à échelle modeste.
- Fournir une plateforme abordable et éducative pour apprendre les principes de la mise en place, gestion et optimisation des clusters de calcul.

MASTER EN CALCUL HAUTE PERFORMANCE, SIMULATION

Université Perpignan Via Domitia

Technologies et Outils Complémentaires

- Conteneurisation avec Docker : Encapsulation des applications pour un déploiement et une gestion simplifiés.
- Orchestration avec Kubernetes : Automatisation de la gestion des conteneurs, incluant l'équilibrage de charge, gestion des ressources et résilience aux pannes.
- Gestion des Tâches avec Slurm : Planification et gestion des tâches de calcul, optimisant l'utilisation du cluster.

MASTER EN CALCUL HAUTE PERFORMANCE, SIMULATION

Université Perpignan Via Domitia

Objectifs

- Configurer et interconnecter plusieurs Raspberry Pis pour former un cluster de calcul haute performance.
- Utiliser Docker pour encapsuler les applications.
- Utiliser Docker Swarm pour définir un nœud maître et les nœuds travailleurs.
- Mettre en œuvre Kubernetes pour orchestrer les conteneurs.
- Intégrer Slurm en tant que gestionnaire de tâches.

MASTER EN CALCUL HAUTE PERFORMANCE, SIMULATION

Université Perpignan Via Domitia

Introduction aux Raspberry Pi

- Utilisation de quatre Raspberry Pi 4 Model B Rev 1.4
- Processeur ARM Cortex-A72 quad-core
- Fonctionnalités: support pour fp, asimd

MASTER EN CALCUL HAUTE PERFORMANCE, SIMULATION

Université Perpignan Via Domitia

SSH (Secure Shell)

- Gestion à distance et administration
- Connexions sécurisées aux systèmes Linux embarqués
- Maintenance et contrôle via une seule interface

Concept de la Conteneurisation

- Conteneurisation: virtualisation avec des espaces utilisateurs isolés
- Isolation: accès limité aux ressources
- Portabilité: exécution sur divers types d'infrastructure

Avantages

- Efficacité des ressources: moins de ressources utilisées
- Démarrage rapide: en quelques secondes
- Gestion simplifiée: avec Docker
- Mise à jour et déploiement rapides
- Sécurité: isolation réduisant les risques

Processus de Création d'Images Docker

1. Création d'un Dockerfile
2. Spécification de l'image de base avec `FROM`
3. Exécution des commandes d'installation avec `RUN`
4. Copie des fichiers de l'hôte vers le conteneur avec `COPY`
5. Définition des variables d'environnement avec `ENV`
6. Définition de la commande à exécuter avec `CMD`
7. Construction de l'image avec `docker build`

MASTER EN CALCUL HAUTE PERFORMANCE, SIMULATION

Université Perpignan Via Domitia

Example Dockerfile

```
FROM ubuntu:22.04

RUN apt-get update && apt-get install -y gcc libomp-dev

COPY fibo /usr/local/bin/

ENV OMP_NUM_THREADS=4

#ENTRYPOINT ["/usr/local/bin/fibo"]

CMD ["/usr/local/bin/fibo", "50"]
```

MASTER EN CALCUL HAUTE PERFORMANCE, SIMULATION

Université Perpignan Via Domitia

Docker Swarm pour l'Orchestration des Nœuds

- Gestion efficace du cluster
- Orchestration centralisée des nœuds
- Configuration simple et déploiement rapide
- Convient pour des projets nécessitant une gestion de conteneurs efficace

MASTER EN CALCUL HAUTE PERFORMANCE, SIMULATION

Université Perpignan Via Domitia

Kubernetes

- Plateforme open source d'orchestration de conteneurs
- Portabilité et flexibilité: déploiement cohérent
- Mise à l'échelle automatique des ressources
- Gestion des échecs et résilience
- Déploiements facile
- Isolation et sécurité
- Efficacité des ressources

Slurm, Gestionnaire de Tâches

- Gestion des ressources à grande échelle
- Coordination et allocation des ressources
- Optimisation des ressources disponibles
- Soumission des travaux et allocation équitable

MASTER EN CALCUL HAUTE PERFORMANCE, SIMULATION

Université Perpignan Via Domitia

Configuration des Raspberry Pis

- Préparation des Raspberry Pi avec Raspberry Pi OS 64 bits.
- Installation via Raspberry Pi Imager.
- Association au réseau Wi-Fi.
- Activation du service SSH(Création d'un fichier ssh vide dans notre repertoire overlays/boot avant de lancer la Raspberry)
- Attribution d'une adresse IP statique (En modifiant le fichier dhcpcd.conf)

MASTER EN CALCUL HAUTE PERFORMANCE, SIMULATION

Université Perpignan Via Domitia

Association au réseau Wi-Fi

On a créé un fichier: /wpa_supplicat/wpa_supplicant.conf :

```
country=FR
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
    ssid="zakpi"
    psk="123456789"
    key_mgmt=WPA-PSK
}
```

MASTER EN CALCUL HAUTE PERFORMANCE, SIMULATION

Université Perpignan Via Domitia

Configuration de Docker

- Installation de Docker sur tout les Noeuds.
- Démarrage automatique de Docker. du status de Docker.

```
$ curl -fsSL https://get.docker.com | sh
$ sudo systemctl enable docker
$ sudo systemctl start docker
$ sudo systemctl status docker
```

MASTER EN CALCUL HAUTE PERFORMANCE, SIMULATION

Université Perpignan Via Domitia

Configuration de Docker Swarm

- Initialisation du nœud manager.
- Rejoindre des nœuds workers au Swarm.
- Vérification du cluster Swarm.

```
$ docker swarm init --advertise-addr <IP_du_manager>
$ docker swarm join --token <token> <IP_du_manager>:2377
$ docker node ls
```

MASTER EN CALCUL HAUTE PERFORMANCE, SIMULATION

Université Perpignan Via Domitia

Verification du status de notre Cluster.

```
pi@rpi1:~ $ sudo docker node ls
ID                            HOSTNAME   STATUS    AVAILABILITY   MANAGER STATUS   ENGINE VERSION
2aumacqi6cmwe0ruqyrmgnm6d *   rpi1       Ready     Active         Leader           26.1.4
lmjo4l8wwitch7n94alvl39td     rpi2       Ready     Active                          26.1.4
n5cjs2vid1hniid1lnconkkyv     rpi3       Ready     Active                          26.1.4
j3rez3d273svt8j3kbrngy5ai     rpi4       Ready     Active                          26.1.4
pi@rpi1:~ $
```

MASTER EN CALCUL HAUTE PERFORMANCE, SIMULATION

Université Perpignan Via Domitia

Configuration de SLURM

- Installation et configuration de Munge.
- Installation de Slurm.
- Configuration de slurm.conf
- Partage via la commande scp.
- Démarrage des services Slurm.

Configuration de slurm.conf

```
# SLURM Configuration File

ClusterName=pi_cluster
SlurmctldHost=rpi1
SlurmdSpoolDir=/var/spool/slurmd
SlurmUser=slurm
SlurmdUser=root
StateSaveLocation=/var/spool/slurmctld

# Nodes definition
NodeName=rpi[1-4] CPUs=4 State=UNKNOWN

# Partition definition
PartitionName=debug Nodes=rpi[1-4] Default=YES MaxTime=INFINITE State=UP
```

MASTER EN CALCUL HAUTE PERFORMANCE, SIMULATION

Université Perpignan Via Domitia

Verification du status de slurm

```
pi@rpi1:~ $ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
debug*       up   infinite      4   idle rpi[1-4]
```

MASTER EN CALCUL HAUTE PERFORMANCE, SIMULATION

Université Perpignan Via Domitia

Configuration de Kubernetes

- Choix de k3s.
- Modification du fichier /boot/cmdline.
- Configuration du nœud maître.
- Configuration des nœuds esclaves.
- Vérification de l'état du cluster.

```
$ curl -sfL https://get.k3s.io | sh -
$ sudo systemctl status k3s
$ sudo cat /var/lib/rancher/k3s/server/node-token
```

MASTER EN CALCUL HAUTE PERFORMANCE, SIMULATION

Université Perpignan Via Domitia

Joindre les Noeuds travailleurs

```
pi@rpi2:~ $ curl -sfL https://get.k3s.io | K3S_URL=https://192.168.165.117:6443 K3S_TOKEN=K104010c1d3fb9eb17fddbdea98e
7fe4f197963d0bc35fdbec39b266c4b4d6c21::server:ad1f84701fe696660dbdecc8c6d201a5 sh -
[INFO]  Finding release for channel stable
[INFO]  Using v1.29.6+k3s1 as release
[INFO]  Downloading hash https://github.com/k3s-io/k3s/releases/download/v1.29.6+k3s1/sha256sum-arm64.txt
[INFO]  Downloading binary https://github.com/k3s-io/k3s/releases/download/v1.29.6+k3s1/k3s-arm64
[INFO]  Verifying binary download
[INFO]  Installing k3s to /usr/local/bin/k3s
[INFO]  Skipping installation of SELinux RPM
[INFO]  Creating /usr/local/bin/kubectl symlink to k3s
[INFO]  Creating /usr/local/bin/crictl symlink to k3s
[INFO]  Skipping /usr/local/bin/ctr symlink to k3s, command exists in PATH at /usr/bin/ctr
[INFO]  Creating killall script /usr/local/bin/k3s-killall.sh
[INFO]  Creating uninstall script /usr/local/bin/k3s-agent-uninstall.sh
[INFO]  env: Creating environment file /etc/systemd/system/k3s-agent.service.env
[INFO]  systemd: Creating service file /etc/systemd/system/k3s-agent.service
[INFO]  systemd: Enabling k3s-agent unit
Created symlink /etc/systemd/system/multi-user.target.wants/k3s-agent.service → /etc/systemd/system/k3s-agent.service.
[INFO]  systemd: Starting k3s-agent
pi@rpi2:~ $
```

MASTER EN CALCUL HAUTE PERFORMANCE, SIMULATION

Université Perpignan Via Domitia

Status du cluster

```
pi@rpi1:~ $ sudo k3s kubectl get nodes
NAME    STATUS    ROLES                  AGE      VERSION
rpi1    Ready     control-plane,master   37m      v1.29.6+k3s1
rpi2    Ready     <none>                 12m      v1.29.6+k3s1
rpi3    Ready     <none>                 8m42s    v1.29.6+k3s1
rpi4    Ready     <none>                 7m52s    v1.29.6+k3s1
```

MASTER EN CALCUL HAUTE PERFORMANCE, SIMULATION

Université Perpignan Via Domitia

Déploiement avec Docker et kubernetes

- Objectif: Déployer une application avec Docker et Kubernetes.
- Outils: Docker pour la conteneurisation, Kubernetes pour l'orchestration.

MASTER EN CALCUL HAUTE PERFORMANCE, SIMULATION

Université Perpignan Via Domitia

Compilation de l'application

- Commande: `gcc -fopenmp -o fibo mainfibo.c`
- Compilation de l'application avant la conteneurisation.

Création du Dockerfile

```
FROM ubuntu:22.04

RUN apt-get update && apt-get install -y gcc libomp-dev

COPY fibo /usr/local/bin/

ENV OMP_NUM_THREADS=4

#ENTRYPOINT ["/usr/local/bin/fibo"]

CMD ["/usr/local/bin/fibo", "50"]
```

MASTER EN CALCUL HAUTE PERFORMANCE, SIMULATION

Université Perpignan Via Domitia

Création de l'image Docker

- Commande: `sudo docker build -t /repertoire/ou/on/a/Dockerfile`
- Construction de l'image Docker contenant l'application compilée.

Upload au Docker Hub

- Étapes:
  - Tag: `sudo docker tag fibo zakpi/fibo`
  - Push: `sudo docker push zakpi/fibo`
- Description: Publication de l'image sur Docker Hub pour la rendre accessible.

Création du fichier .yaml

- Définition de la configuration de déploiement pour Kubernetes.

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: fibo3
spec:
  replicas: 3
  selector:
    matchLabels:
      app: fibo3
  template:
    metadata:
      labels:
        app: fibo3
    spec:
      containers:
      - name: fibo3
        image: zakpi/fibo3
        resources:
          limits:
            cpu: "4"
            memory: "512Mi"
        ports:
        - containerPort: 80
```

MASTER EN CALCUL RFORMANCE, SIMULATION

Université Perpignan Via Domitia

Déploiement dans Kubernetes

- Déploiment: `sudo k3s kubectl apply -f dep.yaml`
- Vérification: `sudo k3s kubectl get pods`
- Journal/log: `sudo k3s kubectl logs POD_ID`

MASTER EN CALCUL HAUTE PERFORMANCE, SIMULATION

Université Perpignan Via Domitia

Vérification du deploiment

```
pi@rpi1:~/fibonacci $ sudo k3s kubectl get pods
NAME                      READY   STATUS             R
fibo3-5697c87d7b-dxf4q    0/1     CrashLoopBackOff   1
fibo3-5697c87d7b-sq422    0/1     CrashLoopBackOff   1
fibo3-5697c87d7b-wjdwh    0/1     CrashLoopBackOff   1
```

MASTER EN CALCUL HAUTE PERFORMANCE, SIMULATION

Université Perpignan Via Domitia

Logs

```
pi@rpi1:~/fibonacci $ sudo k3s kubectl logs fibo
Entrez n : fib(-361334760) = -361334760
temps pris : 0.000012 sec
```

MASTER EN CALCUL HAUTE PERFORMANCE, SIMULATION

Université Perpignan Via Domitia

Déploiement avec Docker Swarm

- Commande: `sudo docker service create --name fibo-service --replicas 3 zakpi/fibo-img1:latest`
- Description: Orchestration des conteneurs avec Docker Swarm.

MASTER EN CALCUL HAUTE PERFORMANCE, SIMULATION

Université Perpignan Via Domitia

Utilisation de docker-compose.yml

- Fichier: `docker-compose.yml`
- Objectif: Configuration des services, réseaux et volumes pour une application multi-conteneurs.

Déploiement avec docker-compose.yml

- Commande: `sudo docker stack deploy -c docker-compose.yml fibo3_stack`
- Description: Déploiement et gestion simplifiée d'une application via Docker Compose.

MASTER EN CALCUL HAUTE PERFORMANCE, SIMULATION

Université Perpignan Via Domitia

docker-compose.yml

```yaml
version: '3.8'

services:
  fibo3:
    image: zakpi/fibo3
    deploy:
      replicas: 3
      resources:
        limits:
          cpus: '4'
          memory: 512M
    ports:
      - "80:80"
```

MASTER EN CALCUL HAUTE PERFORMANCE, SIMULATION

Université Perpignan Via Domitia

Défis rencontrés

Défis rencontrés

- **Complexité de la configuration:** Installation et configuration de multiples technologies (Docker, Kubernetes, Docker Swarm, SLURM).
- **Gestion des erreurs:** Suivi et résolution des problèmes via les journaux et les commandes de vérification.

Perspectives

Améliorations potentielles

- **Intégration de SLURM:** Planifier et allouer les ressources de manière plus efficace pour optimiser la gestion des tâches de calcul.
- **Scaling:** Extension du cluster avec plus de nœuds Raspberry Pi ou intégration de nœuds plus puissants pour des applications plus complexes.

Nouveaux objectifs

- **Expérimentation avec d'autres orchestrateurs:** Explorer des alternatives à Kubernetes et Docker Swarm pour voir leurs avantages comparatifs.
- **Amélioration de la tolérance aux pannes:** Renforcement de la résilience et de la haute disponibilité du cluster.
- **Optimisation de la performance:** Tests et ajustements pour maximiser l'efficacité et la vitesse du cluster HPC.