

**POLYTECHNIQUE
MONTREAL**



INF8480 - SYSTÈMES RÉPARTIS ET INFONUAGIQUE

TP1 - DÉCOUVERTE DES TECHNOLOGIES DE L'INFONUAGIQUE

Chargés de laboratoire :
Pierre-Frederick DENYS

Automne 2020 - V4.0

1 Introduction

1.1 Prérequis

- **Introduction aux système répartis** : Historique, concepts de base, modèles et caractéristiques des systèmes.
- **Infonuagique** : clients et serveurs de l'infonuagique, services pour les machines virtuelles et les containers.

1.2 But du TP

- Découverte de l'infrastructure Openstack et de son interface web (Horizon)
- Apprendre à définir et configurer des machines et des réseaux virtuels
- Manipuler les paramètres de sécurité des machines virtuelles
- Prendre en compte les enjeux des services exposés sur le web

2 Infrastructure à mettre en place

Le but de ce TP est de s'initier à la solution d'infonuagique Openstack en configurant un cluster Openstack et un environnement de déploiement d'applications. La configuration de l'infrastructure va permettre de déployer deux machines virtuelles dans un environnement contrôlé et un réseau dédié.

Info

Les actions à réaliser sont **en gras** dans le sujet pour les différencier des informations aidant à la compréhension.

3 Mise en place du serveur

3.1 Introduction

Vous devez dans ce TP mettre en place une infrastructure opérant Openstack. Openstack est un ensemble de modules qui permettent de contrôler les ressources (puissance de calcul, stockage, images...) et le déploiement de machines virtuelles. Cette solution permet donc de créer des nuages privés et publics.

Lire la documentation : <https://www.redhat.com/fr/topics/openstack> pour comprendre le rôle des 6 modules principaux.

3.2 Installation de la VM

Dans un cas d'utilisation réel, Openstack est déployé sur plusieurs dizaines de machines physiques. Cependant, à des fins de tests, il est possible d'utiliser une installation "tout-en-un". Vous allez donc utiliser tout les modules d'Openstack dans une seule et même machine virtuelle (vous allez déployer des VM dans une VM).

L'installation d'Openstack et ses modules est automatisée par le projet "packstack".

<https://www.rdo-project.org/install/packstack/>

Dans ce TP, vous devez installer **le logiciel Virtualbox (Linux, Windows, Mac) sur votre PC, et y déployer l'image de la VM à télécharger sur le lien indiqué sur Moodle.**

La VM fournie dispose d'Openstack déjà installé avec packstack.

La VM déployée représente votre cluster Openstack. Vous devez **configurer la redirection de ports de virtualbox pour rediriger le port invité 80 vers le port hôte 8086, le port invité 22 vers le port 2222, le port invité 6080 vers le port 6080.** Cela permet de communiquer avec le cluster déployé dans la VM (invité) directement depuis votre PC (hôte). Et ainsi accéder à l'interface d'Openstack depuis le navigateur de votre PC.

4 Configuration d'Openstack

Si la VM a bien été déployée, et la redirection de port configurée, **vous pouvez accéder à l'URL : <http://localhost:8086> dans le navigateur de votre PC.** Vous devriez avoir l'écran de connexion du dashboard Horizon. Le nom d'utilisateur et le mot de passe de l'administrateur est `admin/6a1a15cf44264eba` . **Connectez-vous.**

Dans Admin>System>System Information, **S'assurer que tous les services des trois onglets Compute, Block, Network services soient à UP.** Sinon voir en annexe pour le dépannage.

4.1 Projet, Rôles et groupes

Se rendre sur l'onglet identity :

- Un **projet** est un groupe isolé de un ou plusieurs utilisateurs qui partagent un accès commun avec des privilèges spécifiques à des ressources (instances, réseau) Par exemple, pour une entreprise d'hébergement, chaque entreprise cliente obtient un « projet » sur l'instance globale. Cela permet d'isoler chaque client.
- Les **groupes** permettent de gérer les privilèges au sein d'un projet (les super admin qui peuvent supprimer et générer les ressources, et les utilisateurs qui peuvent simplement redémarrer des instances par exemple).
- Les **rôles** permettent de définir les privilèges d'un type utilisateur (un utilisateur de type X à le droit de faire une action Y).

Créer un projet `inf8480projet`, créer un rôle `inf8480role`, créer un utilisateur `inf8480user`, avec le mot de passe `inf8480user` Lui associer le rôle `inf8480role` et comme projet primaire `inf8480projet`.

4.2 Gabarits

Dans Admin> Compute> Flavors

- Les **gabarits** permettent de définir un "modèle" d'attribution de ressources pour des machines virtuelles.

Créer un gabarit `inf8480gabarit`, avec 1 vcpu, 2048 de RAM et 10 Go de disque racine. Dans l'onglet `flavor Access`, ajouter `inf8480projet`. Valider.

4.3 Réseau

Dans Admin>Network>Networks Le réseau `external_network` déjà configuré, est un réseau de type "bridge" et permet d'obtenir des IP sur le réseau externe du cluster, c'est à dire sur le réseau de la VM virtualbox (10.0.0.x).

Il faut créer un réseau `internal_network` qui va permettre aux VM du cluster de communiquer entre elles, sans être exposées sur le réseau virtualbox externe au cluster (qui est internet sur un vrai cluster).

Créer le réseau, donner le nom `internal_network`, l'associer au projet `inf8480projet`, réseau de type `local`, créer un subnet `private_subnet` avec comme network Address `192.168.100.0/24`.

Créer enfin un routeur qui va permettre de relier le réseau externe au réseau interne. Lui donner le nom `inf8480router`, l'associer au projet `inf8480projet`, le relier au réseau externe `external_network`. Cliquer ensuite sur le nom du routeur créé, et cliquer sur l'onglet `interfaces`. Ajouter une interface vers le sous-réseau `internal_network`.

Se déconnecter de la console openstack (en haut à droite) et se reconnecter ensuite avec `inf8480user/ inf8480user`.

5 Création des instances



Info

Il est possible que les actions soient lentes (car tout les modules sont déployés dans la même VM). Pendant qu'Openstack travaille, profitez-en pour faire vos lectures par exemple !

Commencez par **créer une image** : dans `Compute>Images` créer une image `inf8480img` à partir de l'image `cirros-0.5.1-openstack.img` fournie dans l'archive où se trouvait l'image de la VM. Le format d'image est QCOW2.

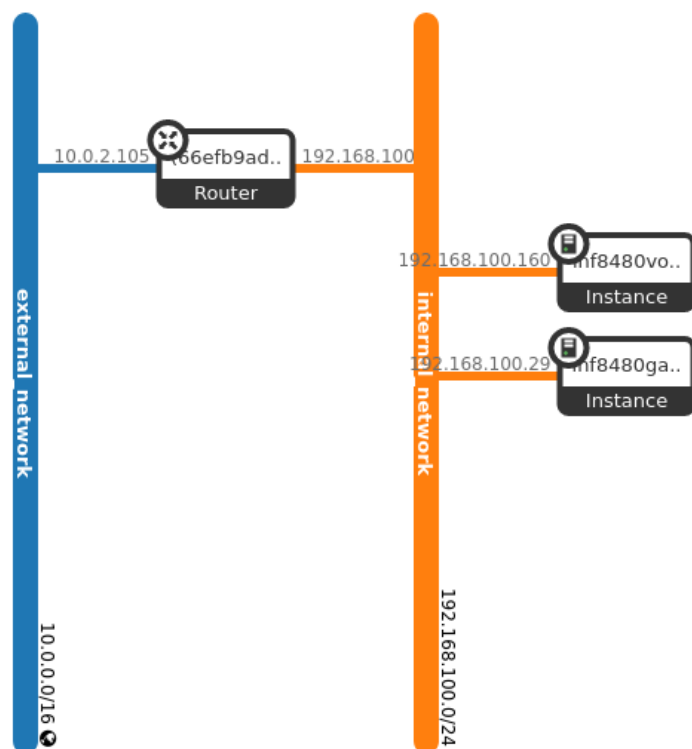
Ensuite, **créer deux volumes** à partir de cet image (on copie le système d'exploitation sur deux disques virtuels qui seront ensuite montés sur les instances) : dans `Volumes> Volumes`, créer un volume `inf8480vol1` à partir de l'image `inf8480img` et de taille 10 GiB. Attendez la fin de la création du volume pour en créer un second `inf8480vol2`.

Dans `Compute>Instances`, cliquer sur `Launch Instance`, **créer une instance** `inf8480vm1`. Dans `Source`, choisir "Volume" dans `Select Boot source` et sélectionner le volume `inf8480vol1`, dans `Flavor`, choisir `inf8480gabarit`, et la connecter au réseau `internal_network`.

Attendez la fin de la création de l'image puis **faire de même pour l'instance** `inf8480vm2`.

Les deux instances devraient avoir le statut `Active` et l'état `Running`. Elles devraient également avoir une IP allouée sur le réseau interne (192.168.100.XXX).

Se connecter au compte admin (en navigation privée par exemple pour garder les deux sessions). Si la configuration réseau des instances est correcte, dans `Network > Network Topology` (compte `admin`) le graphique devrait ressembler à :



6 Connexion aux instances

Cliquer sur le nom de l'instance (via le compte `inf8480user`), cliquer sur l'onglet console et cliquer sur le lien `click here to show only console`. Le lien s'ouvre et ressemble à `10.0.2.15:6080/vnc_ar`. Cependant la connexion échoue car le lien est censé se connecter à l'IP du cluster, or le cluster déployé utilise une redirection de port sur l'hôte. Il faut donc **remplacer dans cet URL l'adresse IP par localhost**.

Vous devriez avoir une console VNC avec un prompt de login. **Connectez vous avec le compte `cirros/gocubsgo`. Enfin lancer la commande `ping 8.8.8.8` pour vérifier que l'instance est correctement connectée au réseau.**

Si le ping réussi sur les deux instances, bravo ! Vous êtes presque prêt pour administrer les 170 000 cores du cluster Openstack de Walmart.

Ouverture : Openstack est utilisé afin de gérer le déploiement et le cycle de vie de machines virtuelles avec possibilités d'isolation très avancées. Openstack est capable de gérer des milliers de noeuds, et des configuration réseau très particulières. Par exemple OVH utilise Openstack pour gérer ses VPS (*virtual private server*). En effet, lorsque vous commandez un VPS chez OVH, en backend, un script lance la création d'une instance Openstack.

7 Vérification et remise

Téléverser **SUR LA VM** le script de correction à l'aide de la commande scp à exécuter **SUR VOTRE PC** dans le dossier où vous avez téléchargé correct.sh.x :

```
[moi@monpc]# scp -P2222 correct.sh.x inf8480@localhost:
```

Connectez-vous en root sur la VM, et exécuter la commande suivante pour permettre au script de se connecter au cluster OPenstack :

```
[root@inf8480]# . keystone_admin
```

Exécuter ensuite ce script avec en paramètre le code header obtenu dans la question 1 du devoir Moodle du TP1.

```
[root@inf8480]# ./correct.sh.x code_header_moodle
```

Si le fonctionnement de votre cluster est correct, le hash obtenu permet de valider la seconde question du devoir Moodle du TP1.

```
[root@inf8480 ~](keystone_admin)# sh correct.sh MTK3NzA3NDMwN2QyNzcxZThiYmIzNQ==
#####
# Correction INF8480 TP1 Automne 2020 V4.0 #
#####
Résutat :
hash ok
Tp vérifié ! Votre hash unique est : YidNVGszTnpBM05ETXd0MlF5TnpjeFpUaGlZbUl6TlE9PSdjWE5rZGp4Mg==
[root@inf8480 ~](keystone_admin)#
```

Vous devez copier-coller le hash obtenu sur la page web dans la question 2 du devoir Moodle du TP1.

Il n'y a pas de rapport à rendre, le but du TP étant de se familiariser avec l'interface d'Openstack et de comprendre les composantes d'un système réparti.



8 Annexes

8.1 Dépannage

8.1.1 Interface lente

Si la VM est trop lente (RAM du PC < 16 GB, pas de SSD), il est possible que la création des instances échoue. Dans ce cas, vous pouvez utiliser à distance les PC du laboratoire L4712, en effectuant le TP avec VNC.

<https://www.polymtl.ca/gigl/laboratoires-et-%C3%A9quipements/d%C3%A9tails-sur-les-labor>

8.1.2 Redémarrer les services Openstack

Si vous constatez que le module nova est à state=down dans l'interface, redémarrer le en exécutant les commandes suivantes dans l'ordre dans la console de la VM virtualbox avec le compte root (root/inf8480) :

- `systemctl restart openstack-nova-conductor`
- `systemctl restart openstack-nova-compute`
- `systemctl restart openstack-nova-scheduler`

Exécuter ensuite la commande : `systemctl status openstack-nova-*` et s'assurer que tout les services sont up and running.

**POLYTECHNIQUE
MONTREAL**



INF8480 - SYSTÈMES RÉPARTIS ET INFONUAGIQUE

TP2 - APPELS DE MÉTHODES À DISTANCE

Chargés de laboratoire :
Pierre-Frederick DENYS

Automne 2020 - V4.1

1 Introduction

1.1 Prérequis

- Intergiciels et objets répartis, Communication inter-processus, Messages de groupes : Sun RPC, gRPC, CORBA, Java RMI et .NET Remoting.

1.2 But du TP

- Introduction à gRPC
- Introduction au calcul réparti et à la répartition de tâches
- Analyse de performance des systèmes répartis à l'aide du traçage

Le TP comporte deux parties indépendantes, il est vivement conseillé que les deux membres du binôme travaillent et comprennent les concepts abordés dans les deux parties du TP.

2 Partie 1 : Implémentation d'une application répartie

2.1 Mise en situation

Vous travaillez dans un laboratoire de recherche, et vous êtes responsable de la baie de serveur disponible pour votre groupe. Vous disposez de plusieurs serveurs, formant une grappe de calcul. Tous les membres du laboratoire ont besoin d'effectuer des tâches de calcul durant plusieurs heures.

Vous souhaitez donc implémenter un gestionnaire qui permet aux clients de soumettre leur tâche dans une file d'attente FIFO, et de décider quel serveur doit effectuer la tâche. Un serveur ne peut effectuer qu'une tâche à la fois. Vous devez utiliser gRPC (gRPC Remote Procedure Call), un système libre d'appel de procédure à distance développé par Google pour implémenter votre gestionnaire.

Dans ce TP, vous vous contenterez d'implémenter un gestionnaire qui envoie un appel gRPC sur un seul serveur. Vous devrez utiliser le traçage système (avec LTTng) afin de visualiser l'exécution des appels gRPC. Pour cela, le code a été instrumenté.

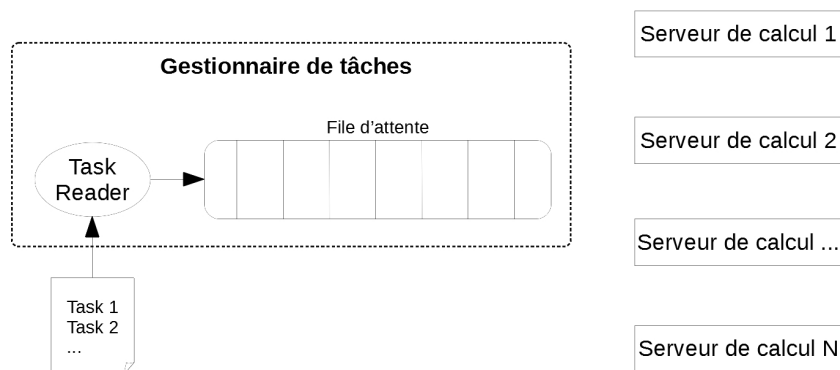


FIGURE 1 – Architecture

2.2 Mise en place

Avant de commencer vous devriez lire attentivement l'ensemble du TP afin de comprendre l'enchaînement des étapes, et vous informer de la section dépannage à la fin du TP.

2.3 Développement d'un programme RPC

Un code incomplet en C++ de l'application vous est fourni dans l'archive `TP2_trace_partie1.tar.gz` :

- `task_scheduler`
- `lttng-traces`
 - `grpc_tracing.h` : déclaration des tracepoints
 - `grpc_tracing.c`
- `manager.cc` : code du manager, **compléter les TODO**
- `server.cc` : code du serveur, rien à modifier.
- `operation.proto` : fichier d'IDL protocol buffers **à compléter**
- `Makefile` : compile le serveur et le manager
- `trace.sh` : lance la session lttng
- `correction.sh` : script pour valider le TP

Vous devez le compléter la déclaration des interfaces IDL, et le code du manager, qui doit prendre en paramètre le nom de la tâche (exemple d'exécution : `./manager tâche45`). Vous pouvez éditer le code sur votre machine de labo ou personnelle, mais la compilation doit se faire dans le container déployé dans la VM (voir paragraphe suivant).

Après modification, refaire une archive `TP2_trace_partie1_modif.tar.gz`.

2.4 Déploiement de la machine virtuelle

La VM fournie (`VM_TP234_INF8480.ova`) est à déployer sur Virtualbox de la même manière que pour le TP1. Les ressources nécessaires sont bien plus faibles, vous pouvez normalement la déployer sur un PC avec 8 go de RAM. **N'oubliez pas de supprimer la VM du TP1.**

Mettez en place la redirection de port invité 22 vers 2222 avec virtualbox.

Cette VM est accessible uniquement en SSH, avec le fichier de clé fourni avec la VM (il n'y a donc aucun mot de passe à rentrer). Vous devez utiliser la commande suivante depuis votre PC pour vous y connecter : (pas de mot de passe root, utilisez sudo)

```
[MONPC]$ ssh -i cle_vm_tp234 -p 2222 inf8480@localhost
```

2.5 Utilisation d'un container

Vous allez tester votre code sur un container docker, avec les bibliothèques C++ de GRPC et Protobuf déjà installées.

Dans la VM, la commande `sudo docker ps` vous permet de tester si le service Docker est bien démarré.

Pour accélérer le déploiement, une image docker `inf8480tp2:v1` est disponible sur la VM, et est basée sur l'image <https://hub.docker.com/r/grpc/cxx/dockerfile>.

La commande suivante vous permet de voir les images disponibles sur la VM :

```
inf8480@inf8480:~$ sudo docker images
```

Lancer maintenant un container nommé `inf8480tp2`, basé sur l'image `inf8480tp2:v1`.

```
inf8480@inf8480:~$ sudo docker run -dit --name inf8480tp2 inf8480tp2:v1
```

La commande `sudo docker ps` permet de vérifier que le container est bien "up and running".

2.6 Tests

Démarche de test :

- Téléverser l'ensemble des fichiers modifiés de l'application sur la machine virtuelle (commande à exécuter sur le PC hôte) :

```
[MONPC]$ scp -i cle_vm_tp234 -P2222 TP2_trace_partiel_modif.tar.gz inf8480@localhost:
```

Puis dans le container (commande à exécuter sur la VM) :

```
inf8480@inf8480:~$ sudo docker cp TP2_trace_partiel_modif.tar.gz inf8480tp2:/root
```

- Connectez vous ensuite à l'intérieur du container (commande à exécuter sur la VM) :

```
inf8480@inf8480:~$ sudo docker exec -it inf8480tp2 /bin/bash
```

Une fois connecté, vous obtenez un prompt du type :

```
root@9aff26152f31:/#
```

Déplacez vous dans le répertoire où les fichiers ont été transférés :

```
root@9aff26152f31:/# cd /root/
```

- Compilez votre trace provider dans le dossier `lttng-traces` :

```
gcc -I. -c grpc_tracing.c
```

- Reculer d'un dossier et compiler le programme avec `make` (le `Makefile` vous permet de compiler les deux programmes `manager` et `server`)



Remarque

Si votre code ne compile pas ou que celui-ci ne fonctionne pas lorsque vous le testez avec les étapes suivantes, vous pouvez vous servir de `scp` puis `docker cp` afin de transférer uniquement le fichier modifié (`operation.proto` par exemple) au lieu de re-transférer toute l'archive .

- Lancer le script `sudo sh trace.sh` (lance le traçage et le programme `server` en tâche de fond). Le script `trace.sh` permet de générer une trace système du fonctionnement de votre programme.
- Lancer votre serveur en arrière plan avec `./server &` et appuyez sur entrée pour avoir à nouveau le prompt.
- Envoyez plusieurs tâches au manager : (`./manager tache23`, `./manager tache45`). Vous devriez recevoir à chaque fois la réponse du serveur.
- Exécuter les commandes suivantes pour terminer le traçage :

```
lttng stop
lttng destroy
```

- Le fichier de trace générés en sortie est situé dans le dossier `trace_files`.
- Téléverser votre dossier de trace (le dossier contenu dans `trace_files`) sur votre ordinateur de labo (faire une archive, utilisez `docker cp` puis `scp` de la même manière que l'on a transféré les fichiers de l'hôte vers le container, mais cette fois du container vers l'hôte).
- Utilisez le logiciel **Trace Compass** (à télécharger ici : <https://www.eclipse.org/tracecompass/>) afin de visualiser la trace (voir annexe pour mode d'emploi) .
- vous devriez obtenir un résultat comparable :


	<srch>	<srch>	<srch> grpc	<srch>
	12:43:29.569 731 511	ustchannel_0 0	lttng_ust_statedump:soinfo	baddr=0x7f0facec5000, sopath=/lib/x86_64-linux-gnu/librt-2.23.so,
	12:43:29.569 835 334	ustchannel_0 0	lttng_ust_statedump:soinfo	baddr=0x7f0faccbd000, sopath=/usr/lib/x86_64-linux-gnu/liburcu-bj
	12:43:29.569 939 305	ustchannel_0 0	lttng_ust_statedump:soinfo	baddr=0x7f0facab5000, sopath=/usr/lib/x86_64-linux-gnu/liburcu-cc
	12:43:29.570 034 303	ustchannel_0 0	lttng_ust_statedump:soinfo	baddr=0x7f0fac7ac000, sopath=/lib/x86_64-linux-gnu/libm-2.23.so,
	12:43:29.570 075 134	ustchannel_0 0	lttng_ust_statedump:end	context._vpid=1487, context._vtid=1492
👉	12:43:29.612 914 570	ustchannel_0 0	grpc_tracing:manager_send	string=tache envoyee, context._vpid=1488, context._vtid=1488
👉	12:43:29.628 382 324	ustchannel_0 0	grpc_tracing:server_start	id=50051, context._vpid=1487, context._vtid=1502
👉	12:43:29.628 419 439	ustchannel_0 0	grpc_tracing:server_end	id=50051, context._vpid=1487, context._vtid=1502
👉	12:43:29.628 778 815	ustchannel_0 0	grpc_tracing:manager_recv	string=tache envoyee, context._vpid=1488, context._vtid=1488

FIGURE 2 – Exemple de trace

- Si vous avez vu la trace de vos messages envoyés dans Trace Compass, vous pouvez terminer en exécutant le script de correction **sur la VM** avec le code header moodle donné dans la première question du quiz sans les guillemets et sans le b

Le script doit être exécuté dans le dossier où se trouve le dossier `trace_files` **Exemple :**

```
ubuntu@tp2:~/task_scheduler_correction$ ./correction.sh.x cDExMjAzOE1UazN0ekEzTkUxVWF6Tk9lZW==
```

```
inf8480@inf8480:~/task_scheduler$ ./correction.sh.x
code_header_moodle
```

Pensez à le rendre exécutable si besoin !

2.7 Vérification et remise

Vous venez de découvrir le traçage système, cette méthode qui permet d'instrumenter le noyau d'un système d'exploitation ou une application afin d'enregistrer une trace des évènements qui

s'y déroulent. Le traçage permet d'analyser les performances d'une application ou d'un système distribué ou temps réel. On peut, en utilisant un logiciel d'analyse de traces comme Trace Compass détecter les latences, le temps d'exécution d'une application... Le traçage est utilisé afin de détecter les problèmes de pertes de messages dans un système distribué par exemple (on détecte un message envoyé, son chemin vers la carte réseau) et en déduire quel est l'élément défectueux.

Un quiz moodle vous permet de déposer vos résultats.

3 Partie 2 : Analyse d'une trace système d'un système distribué

3.1 Mise en situation

Vous êtes analyste au service informatique de Poly, et on vous a signalé que le logiciel des ressources humaines était lent.

Le site est un système distribué, composé de trois parties (application cliente, serveur d'application et serveur d'authentification) qui sont des applications gRPC. Lors de la connexion, le client envoie une requête au serveur d'application, et ce dernier vérifie l'identité du client auprès du serveur d'authentification.

Vous devez donc trouver la cause et la durée des goulots d'étranglement. Les goulots d'étranglement peuvent être dans le client, le serveur, ou le serveur d'authentification.

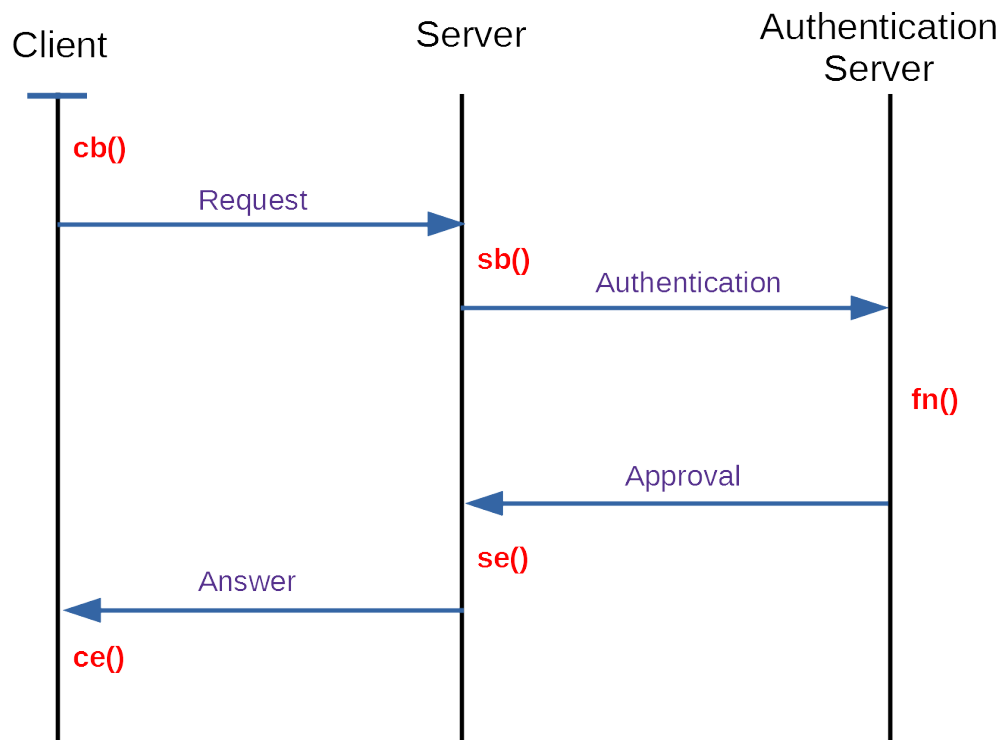


FIGURE 3 – Appel des fonctions

3.2 Analyse



Attention

Vous n'avez pas besoin de la VM dans cette partie! vous devez utiliser Trace compass sur votre ordinateur personnel ou le PC du labo.

Vous devez analyser à l'aide de trace compass la trace fournie 1568729535905. Vous devez calculer le temps passé dans le client, dans le serveur et dans le serveur d'authentification. Attention, le temps d'exécution réel du client n'est pas de `client_start` à `client_end` mais de `client_start` à `server_start` et de `server_end` à `client_end`. Même chose pour le temps d'exécution du serveur.

Vous pouvez voir le delta de temps entre deux événements avec trace compass, en sélectionnant deux événements, avec les touches **Ctrl + Maj** enfoncées.

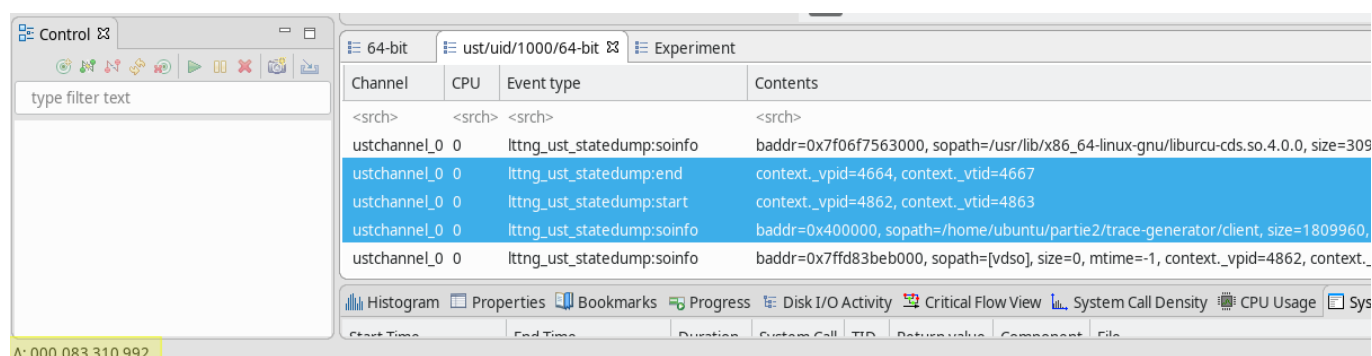


FIGURE 4 – Voir le delta de temps entre deux événements (en bas à gauche)

3.3 Vérification et remise

Vous devez analyser la trace fournie, et répondre aux 3 questions du quiz Moodle proposées sur le site du cours.

4 Annexes

4.1 Traçage système

4.1.1 Définition

Une trace d'un programme est une représentation de l'exécution de ce même programme. Le but est d'instrumenter et d'optimiser la qualité des programmes en termes de performances et de robustesse.

4.1.2 LTTng

LTTng est un outil de traçage et de visualisation des événements produits à la fois par le noyau linux et par les applications.

- **Définition** : <https://lttng.org/docs/v2.10/#doc-what-is-tracing>
- **Tracepoint provider** : <https://lttng.org/docs/v2.10/#doc-tracepoint-provider>

4.1.3 Trace Compass

Trace compass est un outil de visualisation graphique de traces systèmes.

Pour ouvrir une trace, cliquez sur *File* → *Trace Import*, cliquer sur *Browse...* et sélectionner le dossier de la trace.

Puis cocher la case du dossier de la trace à ouvrir, et enfin cliquer sur finish. Le menu à gauche, permet de naviguer entre les traces systèmes et traces applications (ust). Dans ce tp, vous analyserez les traces ust (userspace).

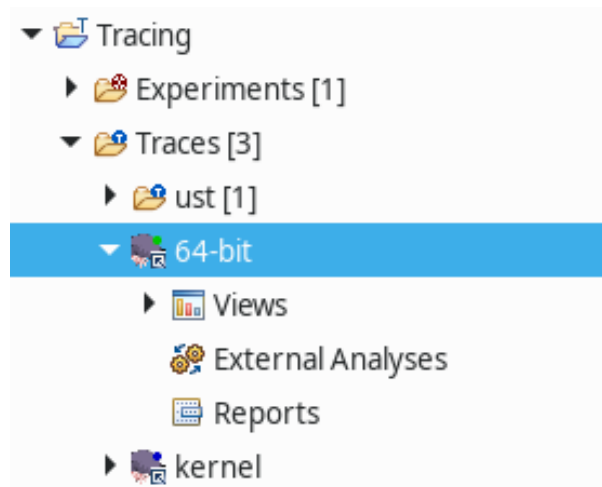


FIGURE 5 – Ouvrir la trace userspace

Vous pouvez le télécharger ici : <https://www.eclipse.org/tracecompass/>



**POLYTECHNIQUE
MONTREAL**



INF8480 - SYSTÈMES RÉPARTIS ET INFONUAGIQUE

TP3 - SYSTÈMES DE FICHIERS DISTRIBUÉS

Chargés de laboratoire :

Pierre-Frederick DENYS pierre-frederick.denys@polymtl.ca

1 Introduction

1.1 Prérequis

- **Services de fichiers répartis et poste à poste** : composantes et interfaces, mécanismes pour l'implémentation. Exemples de Sun NFS, AFS, GFS, Ceph, Napster, Gnutella et BitTorrent

1.2 But du TP

- Découverte de docker et déploiement de containers
- Mise en place d'une architecture de stockage distribuée avec GlusterFS

2 Système de fichiers distribués GlusterFS

2.1 Introduction

Le but du TP est de mettre en place une architecture de stockage haute disponibilité avec glusterfs. L'architecture est composée de deux serveurs de stockage simulés par deux containers, et un poste client simulé par un troisième container. Ensuite, un volume répliqué et un volume distribué seront déployés sur les deux noeuds. Le déploiement va s'effectuer sur une machine virtuelle comme cela se fait dans l'industrie. La machine virtuelle est en local sur vos machines, mais vous ne pouvez y accéder que par SSH.

2.2 Procédure

2.2.1 Installation de GlusterFS

Suivre les instructions du TP2 pour déployer la VM sur virtualbox et vous y connecter en SSH.

2.2.2 Installation de GlusterFS

1. Créer les trois containers avec les commandes suivantes : (les ports ouverts sont documentés ici : <https://www.jamescoyle.net/how-to/457-glusterfs-firewall-rules>)

```
sudo docker run -t -d --privileged=true --name gluster1 --  
    hostname gluster1 -p 24007 -p 49152-49160 ubuntu  
sudo docker run -t -d --privileged=true --name gluster2 --  
    hostname gluster2 -p 24007 -p 49152-49160 ubuntu  
sudo docker run -t -d --privileged=true --name client --hostname  
    client -p 24007 -p 49152-49160 ubuntu
```

2. Connectez vous au premier container avec docker exec.
3. Récupérer l'IP du container avec la commande `ifconfig` (installer le paquet `net-tools`).

4. Dans une autre console, connectez vous au second container. Editer le fichier `/etc/hosts/` afin d'y ajouter le nom et l'IP du premier container. Faire de même sur le premier container. Au final, les fichiers `/etc/hosts/` devront contenir les noms et IP des deux containers.
5. Sur les **deux** containers, installer gluster-fs avec les commandes suivantes :

```
apt-get install software-properties-common
add-apt-repository ppa:gluster/glusterfs-6
apt-get update
apt-get install glusterfs-server
/usr/sbin/glusterd -p /var/run/glusterd.pid
mkdir /home/disk1/
```



Conseil

Le logiciel **terminator** vous permet de lancer des commandes identiques dans plusieurs terminaux avec les groupes.

6. Sur le container **gluster1**, ajouter le second container au *trusted pool*, et créer un volume répliqué.

```
gluster peer probe gluster2
gluster volume create replicated1 replica 2 gluster1:/home/disk1
/ gluster2:/home/disk1/ force
gluster volume start replicated1
```

7. Sur le container **gluster2**, vérifiez le bon fonctionnement du volume avec :

```
gluster volume status
```

8. Vous connecter au container **client** et exécuter les commandes suivantes pour installer glusterfs client, et monter le volume réseau. N'oubliez pas de modifier le fichier `/etc/hosts` afin d'y ajouter les noms et adresses des deux serveurs.

```
apt-get update
apt-get install software-properties-common
add-apt-repository ppa:gluster/glusterfs-6
apt-get update
apt-get install glusterfs-client
mkdir /mnt/replicated1
mount.glusterfs gluster1:/replicated1 /mnt/replicated1
```

9. Sur les **deux serveurs**, lancez la commande qui permet de voir le contenu du disque du volume en temps réel.

```
watch ls /home/disk1/
```

10. Sur le **client**, créer un fichier **bob.txt** dans le dossier `/mnt/replicated1/`. Vous devriez le voir apparaître sur les deux serveurs. Créer un fichier de grande taille dans `/root/` avec la commande suivante :

```
head -c 200M </dev/urandom >bigfile1
```

11. Comparer le temps de copie en local et sur le réseau et noter le rapport entre les deux temps.

```
time cp bigfile1 /mnt/replicated1  
time cp bigfile1 copie
```

12. Maintenant que vous avez un volume de stockage répliqué, mettez en place un volume de stockage distribué `distributed1` selon la même démarche (ne pas oublier de créer le dossier `/home/distributed1` sur les deux serveurs avant de créer le volume. Monter le sur le client dans le dossier `/mnt/distributed1` puis exécuter la commande suivante sur le client dans `/mnt/distributed1` qui crée 20 fichiers. et observer la répartition des fichier avec la commande `watch` du dossier `/home/distributed1` sur les deux serveurs.

```
for (( i=1; i <= 20; i++ )); do touch $i; done
```

le serveur 1 devrait stocker 10 fichiers, et le second les 10 autres fichiers.

3 Partie 2 : Création d'un Dockerfile

3.1 Introduction

Pour "containeriser" une application, une bonne méthode est de la déployer manuellement (comme dans la partie 1) dans un container pour tester, puis d'écrire un *Dockerfile* pour créer une image générique. Ensuite, on utilise `docker-compose` ou Kubernetes ou Docker swarm pour orchestrer les containers.

Dans cette seconde partie, on va se contenter de créer une image d'un serveur de stockage glusterfs, de créer un container avec et de l'ajouter à notre grappe (*trusted pool*) de la partie 1.

3.2 Procédure

- Créer un dockerfile à partir de la documentation https://docs.docker.com/develop/develop-images/dockerfile_best-practices/. Le fichier devra contenir :
 - Image de base `ubuntu`
 - Installation de GlusterFS
 - Exposer les bon ports
 - Pour simplifier, l'édition du fichier `/etc/hosts/` sera effectuée à la main hors du Dockerfile.
- L'image finale `glusterfs_tp3` devra être construite, et lancée avec :

```
sudo docker run -t -d --privileged=true --name gluster3 --  
hostname gluster3 -p 24007 -p 49152-49160 glusterfs_tp3
```

- Se connecter à l'intérieur de ce nouveau container, et demarrer Glusterfs. N'oubliez pas de modifier le fichier `/etc/hosts/` des quatre containers.

```
/usr/sbin/glusterd -p /var/run/glusterd.pid
```

- Sur le container `gluster2`, ajouter le noeud et un réplica au premier volume.

```
gluster peer probe gluster3  
gluster volume add-brick replicated1 replica 3 gluster3:/home/  
disk1/ force
```

- Tester la copie de fichiers comme pour la partie 1.

3.3 Remise

Exécuter **SUR LA VM** le script de correction (que vous pouvez copier à l'aide de la commande `scp`) avec le code header obtenu dans la question 1 du quiz. Si le fonctionnement de votre serveur de fichier est correct, le hash obtenu permet de valider la seconde question du quiz.

```
./correct.sh.x code_header_moodle
```

4 Conclusion

On remarque que l'ajout de noeuds devient fastidieuse avec cette méthode, car il faut déclarer le nom du noeud sur tous les autres noeuds. Cela est dû au fait que l'on utilise une résolution de nom locale (le fichier `/etc/hosts/`). Dans un système réel, on utilise un DNS (domain name server), qui permet de faire la résolution de nom sur le réseau. Ainsi, pour l'ajout d'un noeud, il suffit d'ajouter une entrée sur un seul fichier. Ce type de service sera abordé dans le prochain TP.

Contrairement à certains systèmes de stockage distribué, GlusterFS n'utilise pas de serveur de métadonnées, il n'y a donc pas de serveur central. GlusterFS est une solution très fiable et opensource, utilisée massivement en IA et Big Data et par de grandes entreprises comme Amazon et Red Hat.



5 Dépannage

5.1 Installation des paquets dans les containers

Il se peut que la VM ne dispose pas de la bonne date. Comme c'est celle-ci qui sert de référence de temps aux containers, il se peut qu'un message "is not valid yet" s'affiche suite à l'échec de l'installation des paquets dans le container. Sur la VM, vérifier la date avec la commande `date`. Si celle-ci est en avance ou en retard, lancez la commande suivante :

```
sudo systemctl restart systemd-timesyncd.service
```