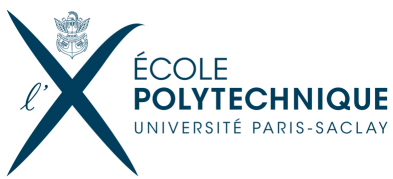
Émulation d'applications distribuées sur des plates-formes virtuelles simulées

Chloé Macur

LORIA – Équipe AlGorille / École Polytechnique



Soutenance de stage de Recherche - 03 Juillet 2014

Motivation générale

- Importance des systèmes distribués (nombre, complexité)
  - Clusters (ensemble de machines homogènes et localisées)
  - Grilles (ensemble de ressources hétérogènes et délocalisées)
  - Systèmes pair-à-pair (ex. : BitTorrent)
  - Cloud
- **SimTerpose : Tester des applications distribuées** (développement, performances, résistance)
  - Générer traces et les rejouer (reproductibilité)
  - Injecter fautes (robustesse)

Plan

État de l'art

- **Applications réelles, environnement réel** (GRID'5000 [2]) :
  - Nécessite la plateforme
  - Complexité de la mise en œuvre
  - Mauvaise reproductibilité
- **Simulation** : modélisation des applications et de l'environnement, interactions calculées via simulateur
  - Nécessite réécriture des applications

État de l'art

- **Applications réelles, environnement réel** (GRID'5000 [2]) :
  - Nécessite la plateforme
  - Complexité de la mise en œuvre
  - Mauvaise reproductibilité
- **Simulation** : modélisation des applications et de l'environnement, interactions calculées via simulateur
  - Nécessite réécriture des applications
- **Émulation** : applications réelles, environnement virtuel
  - Par dégradation (DISTEM [4]) :
    - ajout d'une couche d'émulation à une plate-forme réelle
    - réduction des capacités de l'hôte en ajoutant des délais
    - impossible d'émuler une plate-forme plus puissante
  - Par interception des actions de l'application : SIMTERPOSE
    - actions = calculs et communications
    - exécution possible sur un ordinateur personnel
    - ajout de délais calculés par un simulateur
    - gestion du temps pour émuler un hôte plus puissant

SimGrid

SimGrid [1]

- Simulateur pour l'étude des applications distribuées dans des environnements hétérogènes
- Vise à faciliter la recherche sur les systèmes parallèles et distribués
- Développé par l'équipe AlGorille (entre autres)
- Simulateur : nécessite de réécrire les applications pour les modéliser

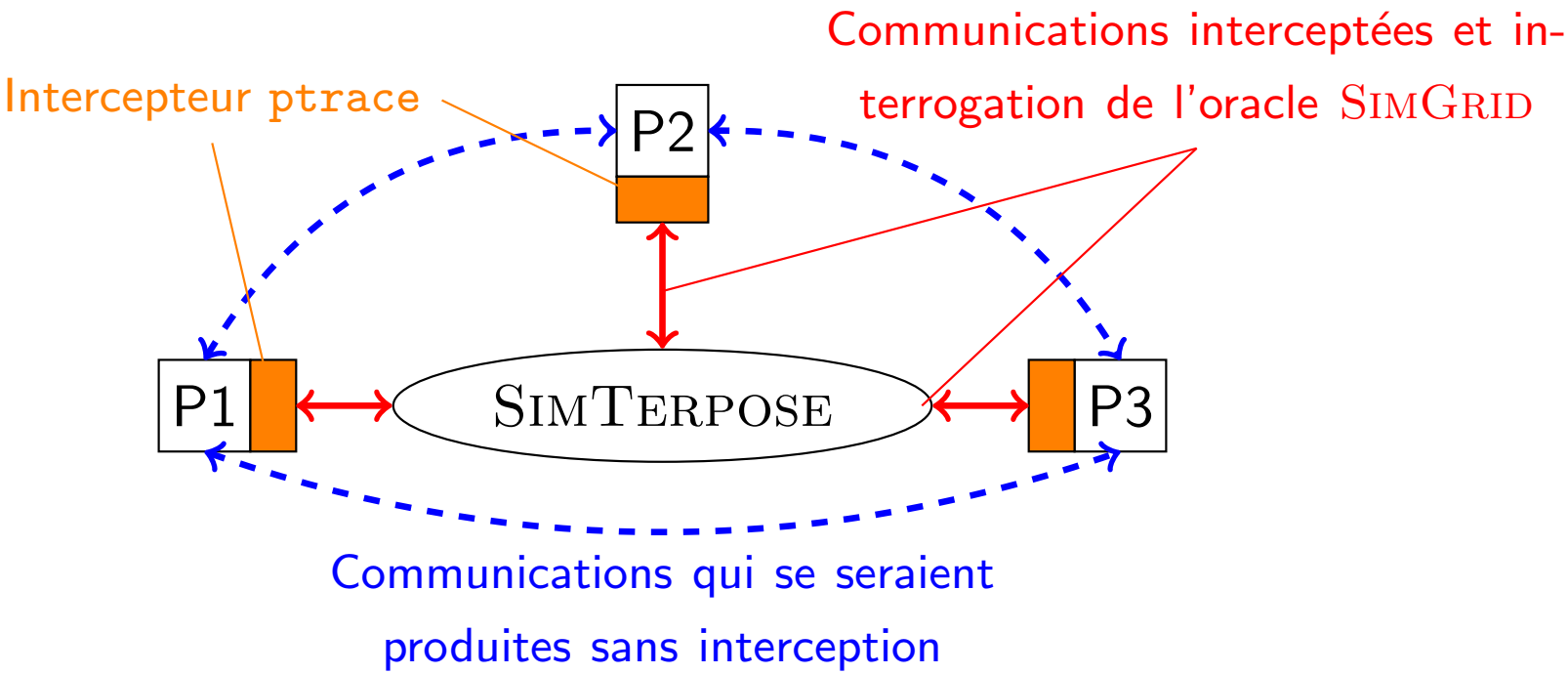
SimTerpose permet d'utiliser SimGrid avec des applications réelles

SimTerpose

Faire croire à des applications qu'elles s'exécutent en environnement distribué

- Simple d'utilisation (ne nécessite pas le code source)
- Intercepte les actions des applications réelles et les modifie
  - **Calculs** : exécutés sur la plate-forme réelle pour réinjecter la durée dans le simulateur
  - **Communications** : modifiées pour imiter un environnement distribué
  - **Délais** (temps de calcul, de communication) : calculés par le simulateur

SimTerpose


Communications interceptées et interrogation de l'oracle SimGrid
Intercepteur ptrace
P2
P1
SimTerpose
P3
Communications qui se seraient produites sans interception

Plan

Réévaluation des méthodes d'interception

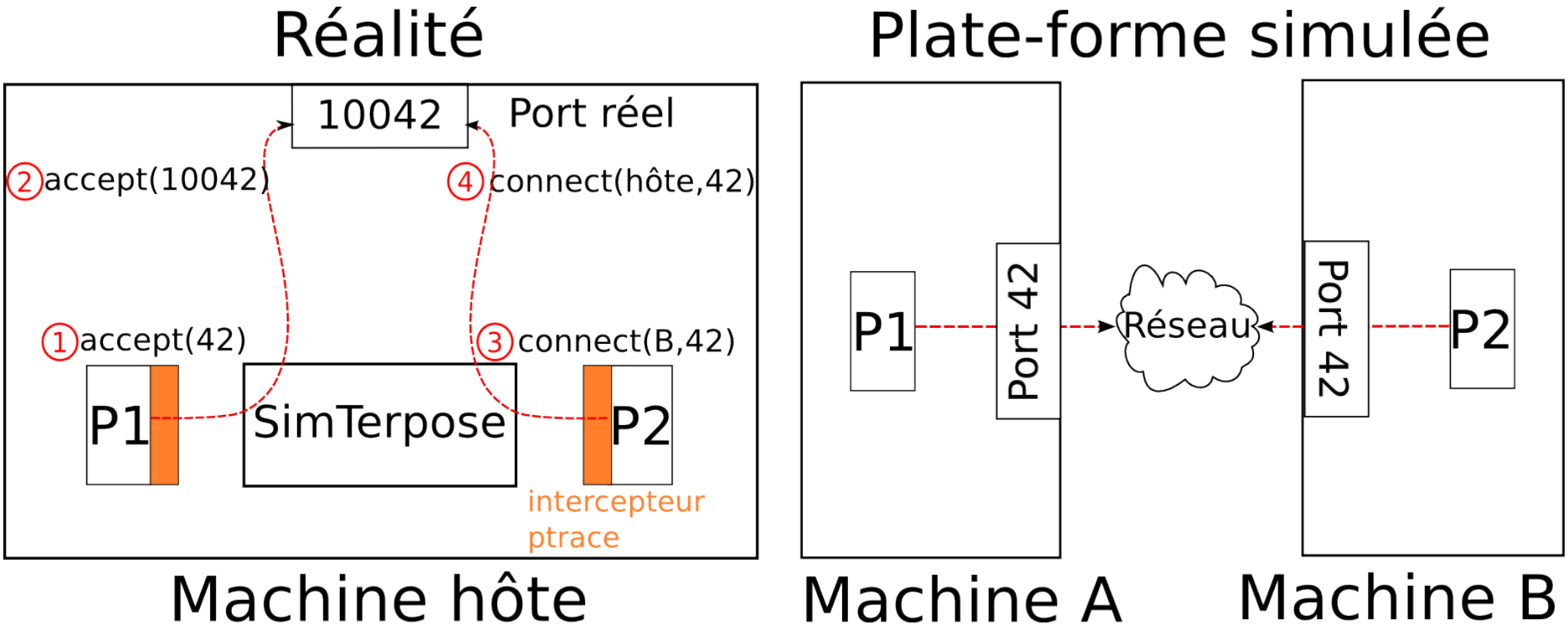
Reprise du prototype :

- Preuve de faisabilité
- Outil non fonctionnel

Réévaluation des méthodes d'interception :

- `ptrace` : appel système qui autorise un processus à contrôler l'exécution d'un autre.
  - Permet d'intercepter les appels système et d'en modifier les registres
  - Exemple :
    ```
    send(int sockfd, const void *buf, size_t len, int flags)
    ```

Réévaluation des méthodes d'interception II



Réévaluation des méthodes d'interception III

- LD_PRELOAD (éditeur de liens dynamiques) : interception au niveau des appels de bibliothèques
  - Préchargement de bibliothèques qui écrasent les fonctions à surcharger
  - Utilisé par `cwrap`[1] pour les *sockets*, le DNS et `setuid`
  - Risque de contournement si oubli de fonction
  - Couplé à `ptrace`, améliore l'interception

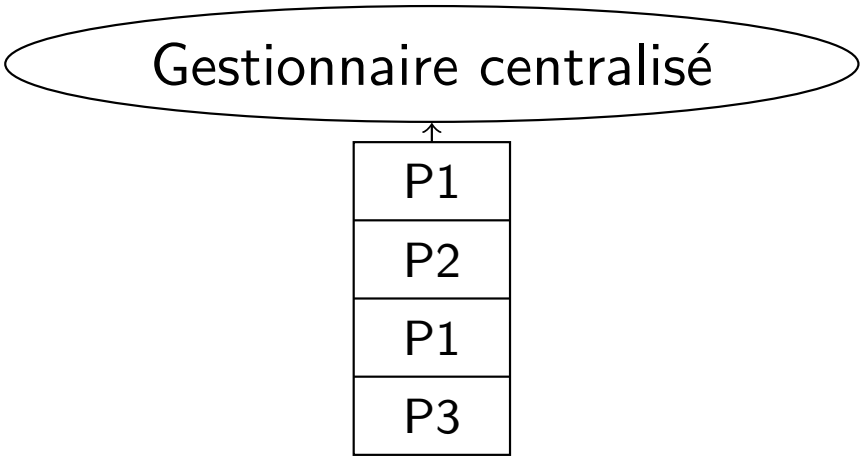
- UPROBES [3] : Insertion de points d'arrêt avec gestionnaires associés
  - ☺ Moins de changements de contexte
  - ☹ Requiert un module noyau pour personnaliser le *handler*

1. `http://cwrap.org/`

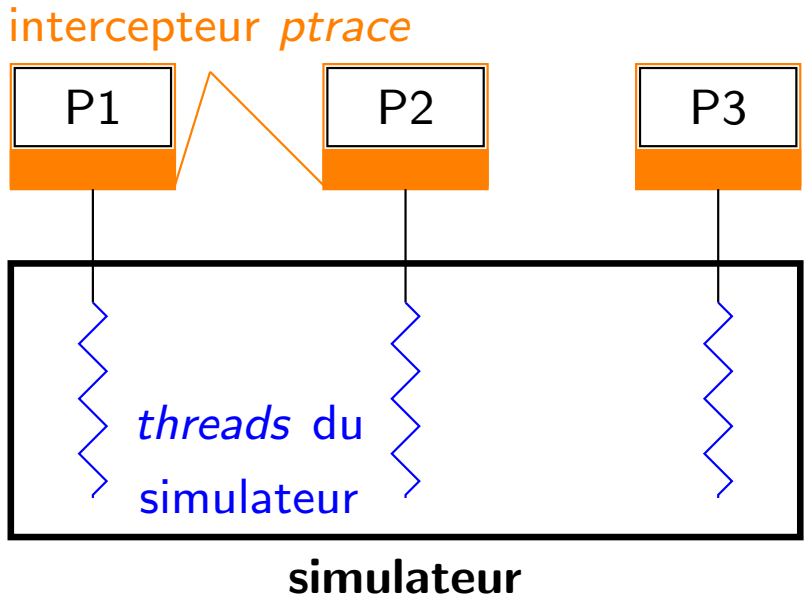
Modification de l'API utilisée

SimGrid offre plusieurs API :

- SimDAG, graphes orientés acycliques
- MSG, applications *Communicating Sequential Processes* (CSP)
- SMPI, applications *Message Passing Interface* (MPI)



SimDAG



MSG

Fonctions liées au temps

Modifier la perception du temps qu'ont les applications

- Intercepter les appels système (via `ptrace`) : `time`, `clock_gettime` et `gettimeofday`
- Ne fonctionne pas :

Virtual Dynamic Shared Object (VDSO)

Améliore les performances

- Réduit les changements de contexte utilisateur/noyau
- Interpole d'après les valeurs précédentes

Fonctions liées au temps

Modifier la perception du temps qu'ont les applications

- Intercepter les appels système (via `ptrace`) : `time`, `clock_gettime` et `gettimeofday`
- Ne fonctionne pas :

Virtual Dynamic Shared Object (VDSO)

Améliore les performances

- Réduit les changements de contexte utilisateur/noyau
- Interpole d'après les valeurs précédentes

Solution envisagée : désactiver le VDSO au démarrage du noyau

- ☺ Parfaitement fonctionnel
- ☹ Diminue les performances et requiert un redémarrage

Solution adoptée : allier `LD_PRELOAD` à `ptrace`

Conclusion et suite du stage

À l'heure actuelle, SimTerpose :

- Modifie les actions des applications et les exécute dans un environnement virtuel
- Simule des applications simples (couple client/server : établissement de connexion, échanges de messages, fermeture de connexion et terminaison des processus)
- Requiert des fonctionnalités supplémentaires : temps, DNS, setuid

Suite du stage :

- Ajouter l'utilisation de `LD_PRELOAD` à celle de `ptrace`
- Tests (taille des expériences, réalisme)

Perspectives

Modifier l'environnement virtuel en injectant diverses fautes dans la simulation. Faciliter l'analyse des applications distribuées en testant leur performance et leur robustesse.

[1] Henri Casanova, Arnaud Legrand, and Martin Quinson.
SimGrid : a generic framework for large-scale distributed experiments.
In *10th IEEE International Conference on Computer Modeling and Simulation*, 2008.

[2] Franck Cappello et al.
Grid'5000 : a large scale, reconfigurable, controlable and monitorable Grid platform.
In *6th IEEE/ACM International Workshop on Grid Computing - GRID 2005*, Seattle, USA, États-Unis, November 2005.

[3] J. Keniston, A. Mavinakayanahalli, P. Panchamukhi, and V. Prasad.
Ptrace, utrace, uprobes : Lightweight, dynamic tracing of user apps.
In *Proceedings of the 2007 Linux Symposium*, pages 215–224, 2007.

[4] Luc Sarzyniec, Tomasz Buchert, Emmanuel Jeanvoine, and Lucas Nussbaum.
Design and Evaluation of a Virtual Experimental Environment for Distributed Systems.
In *PDP2013 - 21st Euromicro International Conference on Parallel, Distributed and Network-Based Processing*, pages 172 – 179, Belfast, Royaume-Uni, February 2013. IEEE.
RR-8046 RR-8046.