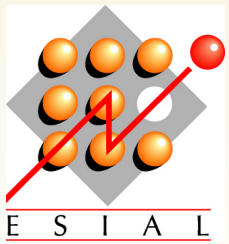
Interception système pour la capture et le rejeu de traces

Marion Guthmuller

LORIA - Équipe AIGorille

Stage de deuxième année ESIAL 2009-2010

Encadrants :
Martin Quinson
Lucas Nussbaum



Plan

Le LORIA (Laboratoire Lorrain de Recherche en Informatique et ses Applications)

- une unité mixte de recherche du CNRS, de l'INRIA, de l'INPL, de l'UHP Nancy 1 et de l'Université Nancy 2 ;
- avec des locaux partagés avec l'INRIA Nancy - Grand Est
- regroupant 150 chercheurs et enseignants-chercheurs, un tiers de doctorants et post-doctorants et des ingénieurs ;
- autour de 5 thématiques :
  - Traitement automatique des langues et des connaissances
  - Fiabilité et sécurité des systèmes informatiques
  - Image et géométrie
  - Perception, action et cognition
  - Informatique et science du vivant

L'équipe AlGorille (Algorithmes pour la Grille)

- Domaine d'application : Réseaux, systèmes et services et calcul distribué
- Thème principal de recherche : calul distribué et applications à très haute performance
- Directeur de Recherche : Jens GUSTEDT
- 4 chercheurs permanents, 2 ingénieurs, 1 Post-Doc et 6 étudiants en thèse

Plan

Contexte : évaluation des applications distribuées

Application distribuée : architecture logicielle permettant l'exécution d'un programme sur plusieurs ordinateurs communiquant entre eux via des réseaux locaux ou Internet, pour mettre en commun des ressources (ex : BitTorrent, SETI@Home, applications HPC)

Évaluation : 3 techniques

- Exécution sur plate-forme réelle (expérience *in situ*) : PlanetLab, Grid'5000
  ☺ exécution directe de l'application étudiée
  ☹ mise en oeuvre lourde et reproduction difficile
- Simulation : mise en oeuvre d'un modèle de l'application et de modèles théoriques pour l'environnement
  ☺ rapide et facile, reproductibilité parfaite
  ☹ application réelle inutilisable (à reprogrammer)
- Émulation : substitution de l'environnement par un logiciel
  ☺ environnement contrôlé, utilisation de l'application réelle

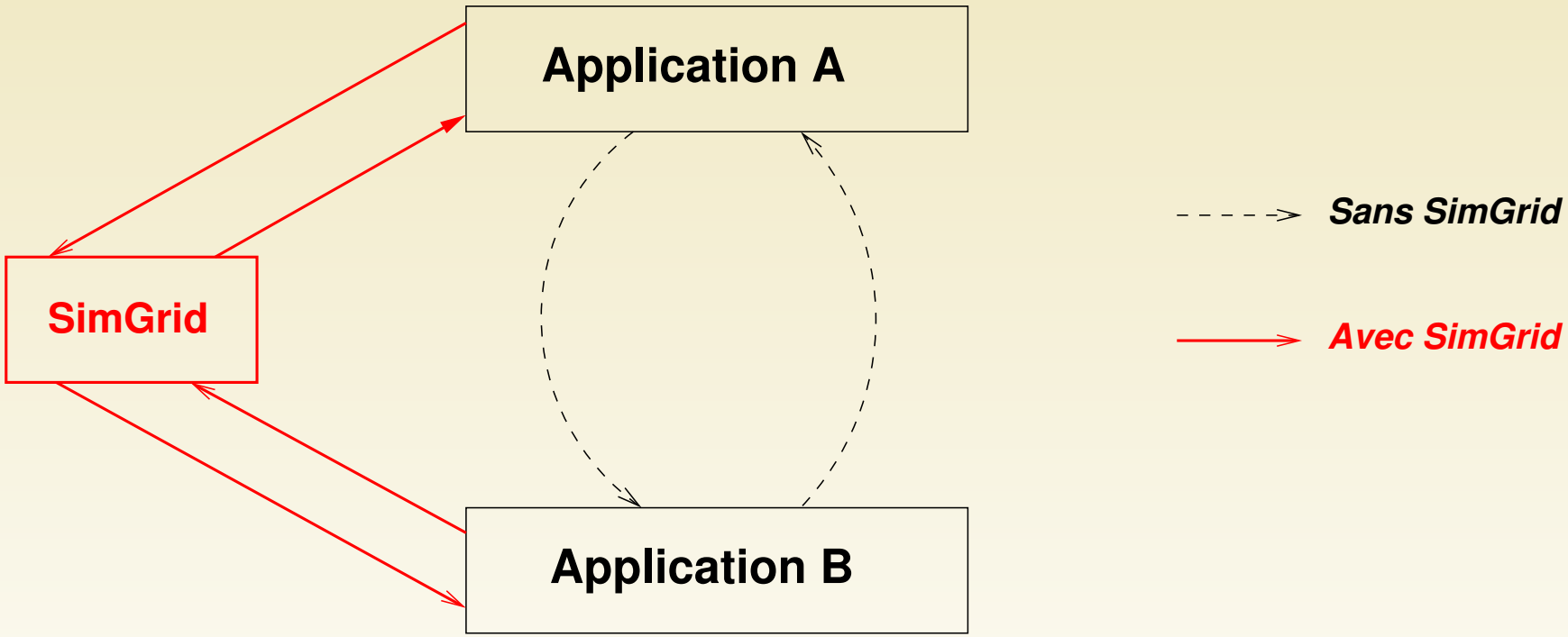
Objectif final à long terme : émuler avec SimGrid

SimGrid : outil pour la simulation d'applications distribuées hétérogènes en environnements distribués $\rightsquigarrow$ faciliter la recherche dans le domaine de la programmation des applications distribuées et parallèles sur des plate-formes de calcul distribué.

Le projet **Simterpose** :

- Fournir un émulateur simple et accessible basé sur SimGrid
- Réalisation :
  - Intercepter les actions de l'application
  - Utiliser SimGrid pour déterminer la réaction de l'environnement aux actions de l'application

Le projet **Simterpose**



Plan

Ma mission : vue d'ensemble

Étudier les moyens d'intercepter les actions de l'application

Actions sur lesquelles l'environnement a un impact

- Communications : `write/read`, `open/close` et sockets
- Calculs

(semaines 1 à 4)

Ma mission : vue d'ensemble

Étudier les moyens d'intercepter les actions de l'application

Actions sur lesquelles l'environnement a un impact

- Communications : `write/read`, `open/close` et sockets
- Calculs

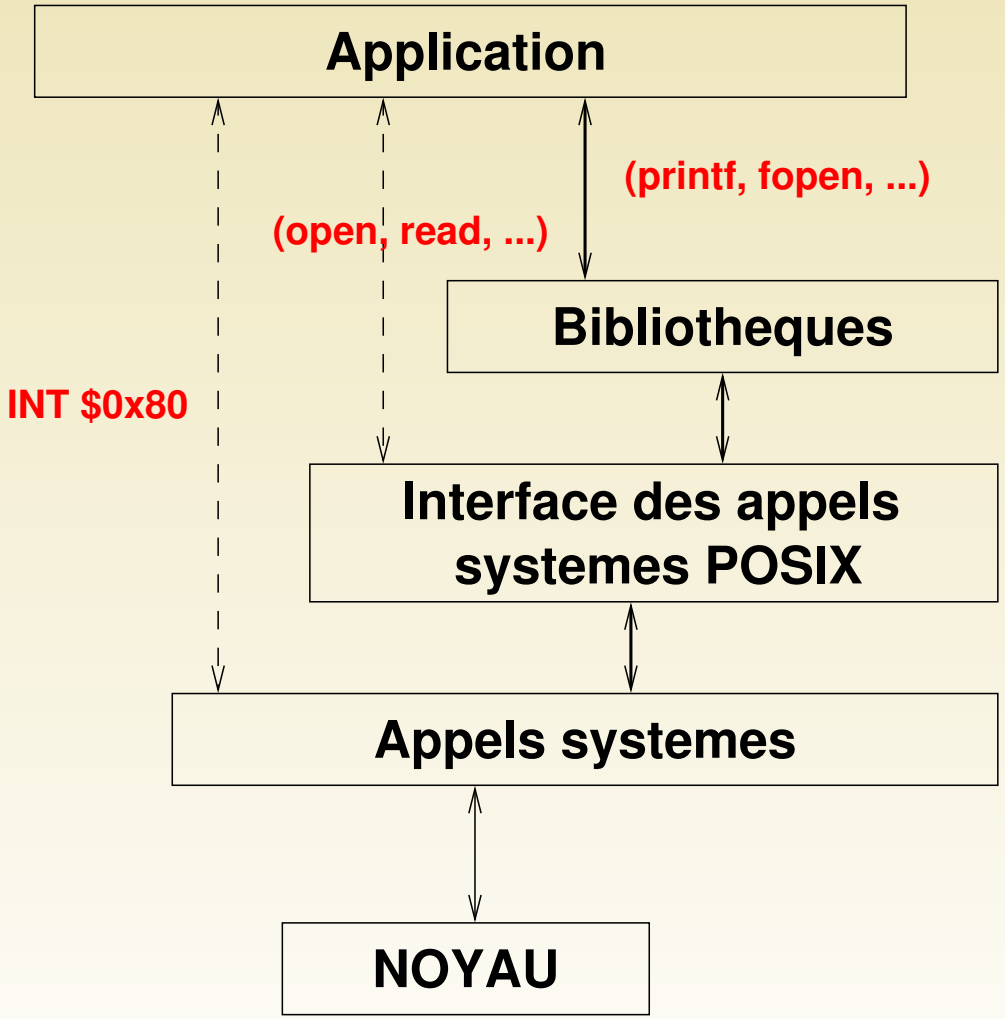
(semaines 1 à 4)

Extraction de la trace d'une application avec `ptrace`

Objectif : rejeu avec SimGrid

(semaines 5 à 10)

Niveaux d'interception



Différentes approches d'interception

Valgrind : outil de programmation pour le profilage de code
- ☹ Surcoût important, complexité dans son utilisation

DynInst : API permettant l'insertion de code pendant l'exécution
- ☺ Coût assez faible
- ☹ API bas niveau, niveau d'abstraction élevé $\rightsquigarrow$ complexe

LD_PRELOAD : édition de liens dynamiques (lld)
- ☺ Faible coût, facilité d'utilisation
- ☹ Surcharge les fonctions des bibliothèques mais pas des appels système

Ptrace : appel système permettant à un processus de tracer les appels système d'autres processus
- ☺ bas niveau, coût moyen
- ☹ portabilité, complexité d'implémentation

Uprobes : interface alternative à `ptrace()`
- ☺ bas niveau, coût faible ?
- ☹ en développement donc peu de documentation

Extraction des actions de l'application

- Sélection des appels systèmes à intercepter : `write/read`, `open/close`, `fork/clone` et tous les appels liés aux sockets
- Interception et récupération de toutes les informations liées à l'appel : valeur de retour, arguments, ...
- Identification des périodes de calcul
- Suivi de la création de processus (`fork()`, `clone()`)

```
..
[24402] getsockopt(4, SOL_SOCKET, SO_REUSEADDR, 1 ) = 0
[24402] bind( 4, {sa_family=AF_INET, sin_port=htons(2226), sin_addr=inet_addr("
     0.0.0.0")}, 16 ) = 0
[24419] connect( 4, {sa_family=AF_INET, sin_port=htons(2226), sin_addr=inet_addr
     ("127.0.0.1")}, 16 ) = 0
[24402] accept( 4, {sa_family=AF_INET, sin_port=htons(56842), sin_addr=inet_addr
     ("127.0.0.1")}, 16 ) = 5
..
[24419] exit_group(0) called
```

Identification des processus communiquants

Lecture des informations sur les sockets : Récupération pour chaque socket du couple (IP,port) local et du couple (IP,port) distant

- à chaque interception
- récupération du numéro de socket associé au file descriptor
- lecture du fichier */proc/net/protocol* où *protocol*=tcp, udp, raw, ...
- enregistrement dans une structure

Identification du processus destinataire : Relier les sockets qui ont des paires de couples (IP,port) inversement identiques.

- à chaque appel lié aux sockets
- parcours de la liste des sockets en cours dans l'application et comparaison des 2 couples (IP,port) locaux et distants.

Résultat de trace dans l'interception système d'un Client/Serveur simple

```
      Timestamp      pidX   wall_time    cpu_time   diff_wall    diff_cpu   type
            syscall          local_addr:port       remote_addr:port      pidY
            return            param
23:15:18:938060      6976       19234       12000           0       12000
      (v)fork                                                              6977
23:15:18:944354      6976       25537       16000        6303        4000
      (v)fork                                                              6978
23:15:21:957648      6976     3038838       16000     3013301           0
      (v)fork                                                              6989
23:15:21:969823      6977     3031988           0           0           0    in
               recv          127.0.0.1: 2226       127.0.0.1:34024      6989
23:15:21:970159      6989       12697           0           0           0    in
               send          127.0.0.1:34024       127.0.0.1: 2226      6977
23:15:21:970356      6989       12895           0         198           0   out
               send          127.0.0.1:34024       127.0.0.1: 2226      6977
      512           (4, "...", 512)
23:15:21:970471      6977     3032640           0         652           0   out
               recv          127.0.0.1: 2226       127.0.0.1:34024      6989
      512           (5, "...", 512)
23:15:21:970594      6989       13133           0         238           0    in
               recv          127.0.0.1:34024       127.0.0.1: 2226      6977
23:15:21:970791      6977     3032963           0         323           0    in
               send          127.0.0.1: 2226       127.0.0.1:34024      6989
23:15:21:970966      6977     3033136           0         173           0   out
               send          127.0.0.1: 2226       127.0.0.1:34024      6989
      512           (5, "...", 512)
23:15:21:971104      6989       13643           0         510           0   out
               recv          127.0.0.1:34024       127.0.0.1: 2226      6977
      512           (4, "...", 512)
```

Plan

Bilan

Professionnel :

- Objectif principal atteint : interception des actions ayant un impact sur l'environnement de l'application
- Approfondissement des connaissances en Réseaux et systèmes
- Découverte du monde de la recherche et d'un projet de plusieurs années
- Confrontation avec une organisation particulière : pas de planning prévisionnel possible

Personnel :

- Envie de continuer dans la recherche

Bilan

Professionnel :

- Objectif principal atteint : interception des actions ayant un impact sur l'environnement de l'application
- Approfondissement des connaissances en Réseaux et systèmes
- Découverte du monde de la recherche et d'un projet de plusieurs années
- Confrontation avec une organisation particulière : pas de planning prévisionnel possible

Personnel :

- Envie de continuer dans la recherche

Perspectives :

- Rejouer la trace avec SimGrid
- Étudier une émulation *online*
- Projet **SimGlite** : étude du middleware de grille Glite

Bilan

Professionnel :

- Objectif principal atteint : interception des actions ayant un impact sur l'environnement de l'application
- Approfondissement des connaissances en Réseaux et systèmes
- Découverte du monde de la recherche et d'un projet de plusieurs années
- Confrontation avec une organisation particulière : pas de planning prévisionnel possible

Personnel :

- Envie de continuer dans la recherche

Perspectives :

- Rejouer la trace avec SimGrid
- Étudier une émulation *online*
- Projet **SimGlite** : étude du middleware de grille Glite

Questions ?