

Simulation de réseaux pair-à-pair

Julien Bourgeois

Université de Franche-Comté, France

e-mail: Julien.Bourgeois@univ-fcomte.fr

website: <http://lifc.univ-fcomte.fr/~bourgeoi>

Journée RGE, 3 juin 2010, LORIA, Nancy

Plan

- ▶ Introduction
- ▶ Définition des réseaux P2P
- ▶ Caractéristiques des réseaux P2P
- ▶ Simulateurs de réseaux virtuels
- ▶ Simulateurs niveau paquet
- ▶ Emulateurs
- ▶ Environnement d'exécution
- ▶ Comparatif
- ▶ Conclusion

Introduction

- ▶ 3 possibilités d'expérimentation
 - Simulation, émulation, exécution
- ▶ Exécution
 - Difficile de réunir un nombre suffisant de nœuds
 - Difficile de reproduire des conditions réelles
 - Dsl-lab ?
- ▶ Emulation
 - Emulation réseau mais pas du CPU
 - Accès à une plate-forme est complexe
 - Contraintes d'utilisation

Introduction

- ▶ Simulation
 - Précision des tests
 - Modélisation de l'application
- ▶ Pas de standard pour l'instant !
 - Applications \neq réseaux virtuels
 - Acceptation de modélisation analytique par la communauté
- ▶ Tentative de draft IETF « *Tools for Peer-to-Peer Network Simulation* », 2006
- ▶ Taille critique de la communauté ?
- ▶ Intérêts divergents ?

Caractéristiques

- ▶ Architectures simulées
- ▶ Structure des réseaux simulés
- ▶ Extensibilité
- ▶ Ergonomie
- ▶ Présentation des résultats
- ▶ Utilisation des ressources

Plan

- ▶ Introduction
- ▶ Définition des réseaux P2P
- ▶ Caractéristiques des réseaux P2P
- ▶ Simulateurs de réseaux virtuels
- ▶ Simulateurs niveau paquet
- ▶ Emulateurs
- ▶ Environnement d'exécution
- ▶ Comparatif
- ▶ Conclusion

Définition des réseaux P2P

- ▶ Peer-to-Peer Working Group (P2PWG), William « Bill » Yeager
 - « *Peer-to-Peer is a way of structuring distributed applications such that the individual nodes have **symmetric roles**. Rather than being divided into clients and servers each with quite distinct roles (such as Web clients vs. Web servers), in P2P applications a node may act as both a client and a server.* »
- ▶ Pr Rüdiger Schollmeier, Université technique de Munich (TUM)
 - « *A distributed network architecture may be called a Peer-to-Peer network, if the participants **share** a part of their own hardware resources (processing power, storage capacity, network link capacity, printers,...). These shared resources are necessary to provide the Service and content offered by the network (e.g. file sharing or shared workspaces for collaboration). They are accessible by other peers directly, without passing intermediary entities. The participants of such a network are thus resource providers as well as resource requestors.* »

Définition des réseaux P2P

- ▶ Clay Shirky, écrivain et consultant spécialisé dans les technologies Internet
 - « *P2P is a class of applications that takes advantage of resources (storage, cycles, content, human presence) available at the edges of the Internet. Because accessing these decentralized resources means operating in an environment of unstable connectivity and unpredictable IP addresses, P2P nodes must operate outside the DNS system and have significant or total **autonomy** from central servers.* »
- ▶ Laboratoire de recherche HP
 - « *P2P is about **sharing**: giving to and obtaining from a peer community. A peer gives some resources and obtains other resources in return. [...] A peer-to-peer system is then in which autonomous peers depend on other autonomous peers. Peers are autonomous when they are not wholly controlled by each other or by the same authority e.g., the same user.*

Définition des réseaux P2P

- ▶ Classification
 - Centralisés
 - Napster, Boinc
 - Hybrides
 - FastTrack, JXTA, BitTorrent
 - Purs
 - Gnutella 0.4, FreeNet

Caractéristiques des réseaux P2P

- ▶ Organisation
 - Décentralisation
 - Gestion des entrées
 - Structuration par **réseaux virtuels**
- ▶ Tolérance aux pannes
 - Caractéristique intrinsèque
- ▶ Extensibilité
 - Fait pour gérer des centaines de milliers ou des millions de noeuds
- ▶ Dynamicité
 - Connexion/déconnexion (churn)
 - Mobilité

Caractéristiques des réseaux P2P

- ▶ Hétérogénéité
 - Matérielle, logicielle
- ▶ Anonymat
 - Auteur, éditeur, lecteur, serveur, document, requête
- ▶ Report des investissements
 - Architecture P2P payée par les utilisateurs

Plan

- ▶ Introduction
- ▶ Définition des réseaux P2P
- ▶ Caractéristiques des réseaux P2P
- ▶ Simulateurs de réseaux virtuels
- ▶ Simulateurs niveau paquet
- ▶ Emulateurs
- ▶ Environnement d'exécution
- ▶ Comparatif
- ▶ Conclusion

Simulateur de réseaux virtuels

▶ Généraux

- PeerSim, PIAX, SimPastry, P2PSim, PeerfactSim.KOM, PLP2P, SimP2

▶ Spécifiques

- CANSimulator, FreePastry, MAPLE, NeuroGrid, ONSP, GPS, Query-Cycle Simulator, TOSim

PeerSim

- ▶ Simulateur à événements discrets
- ▶ Ecrit en Java (cross platform)
- ▶ Open source
- ▶ Pas d'interface graphique
- ▶ Configuration fichier texte + Java
- ▶ Simule uniquement le réseau virtuel
 - Performances maximales
 - Pas de détails sous-jacents
- ▶ Objectif :
 - Simulation de millions de noeuds

PeerSim

- ▶ 2 moteurs de simulation
 - Événementiel
 - Horloge globale
 - Activation par envoi de messages
 - Cyclique
 - Simulateur choisi les nœuds activés
- ▶ Intégration du churn
- ▶ Paramétrisation des performances des liens réseau par fichier de trace

PeerSim

- + Rapide
 - + Extensible (10^7 nœuds simulés)
 - + Portable
 - + Maintenu
 - + Communauté d'utilisateurs
-
- Ne simule « que » le réseau virtuel
 - Réseau sous-jacent modélisé statiquement
 - Nécessite une ré-écriture de l'application

Plan

- ▶ Introduction
- ▶ Définition des réseaux P2P
- ▶ Caractéristiques des réseaux P2P
- ▶ Simulateurs de réseaux virtuels
- ▶ Simulateurs niveau paquet
- ▶ Emulateurs
- ▶ Environnement d'exécution
- ▶ Comparatif
- ▶ Conclusion

Simulateurs de réseau niveau paquets

- ▶ Dédié P2P
 - GnutellaSim, Macedon, PlanetSim
- ▶ Simulateurs génériques
 - NS-2, OMNET++, J-Sim, Narses

OverSim



- ▶ Conception de réseaux virtuels avec possibilités de simulation et d'émulation
- ▶ Open Source et présent sur :
 - Linux, Windows, Mac OS X, Nokia Internet Tablets
- ▶ Basé sur OMNeT++ pour la partie simulation
- ▶ Utilise PlanetLab pour l'émulation

OverSim

- ▶ Modèle en 3 couches :
 - Application
 - Réseau virtuel
 - Réseau sous-jacent
- ▶ Application et réseau virtuel programmés en C++ avec API
- ▶ Réseau sous-jacent est composé de:
 - Hôtes, topologie, canal de communication, churn, message + observateur

OverSim

- ▶ Bibliothèque de réseaux virtuels et de réseau sous-jacent
- ▶ Différents modèles statistiques de churn
- ▶ Interface graphique
- ▶ Collecte de données facilitée par observateur
- ▶ Post-processing et visualisation des statistiques

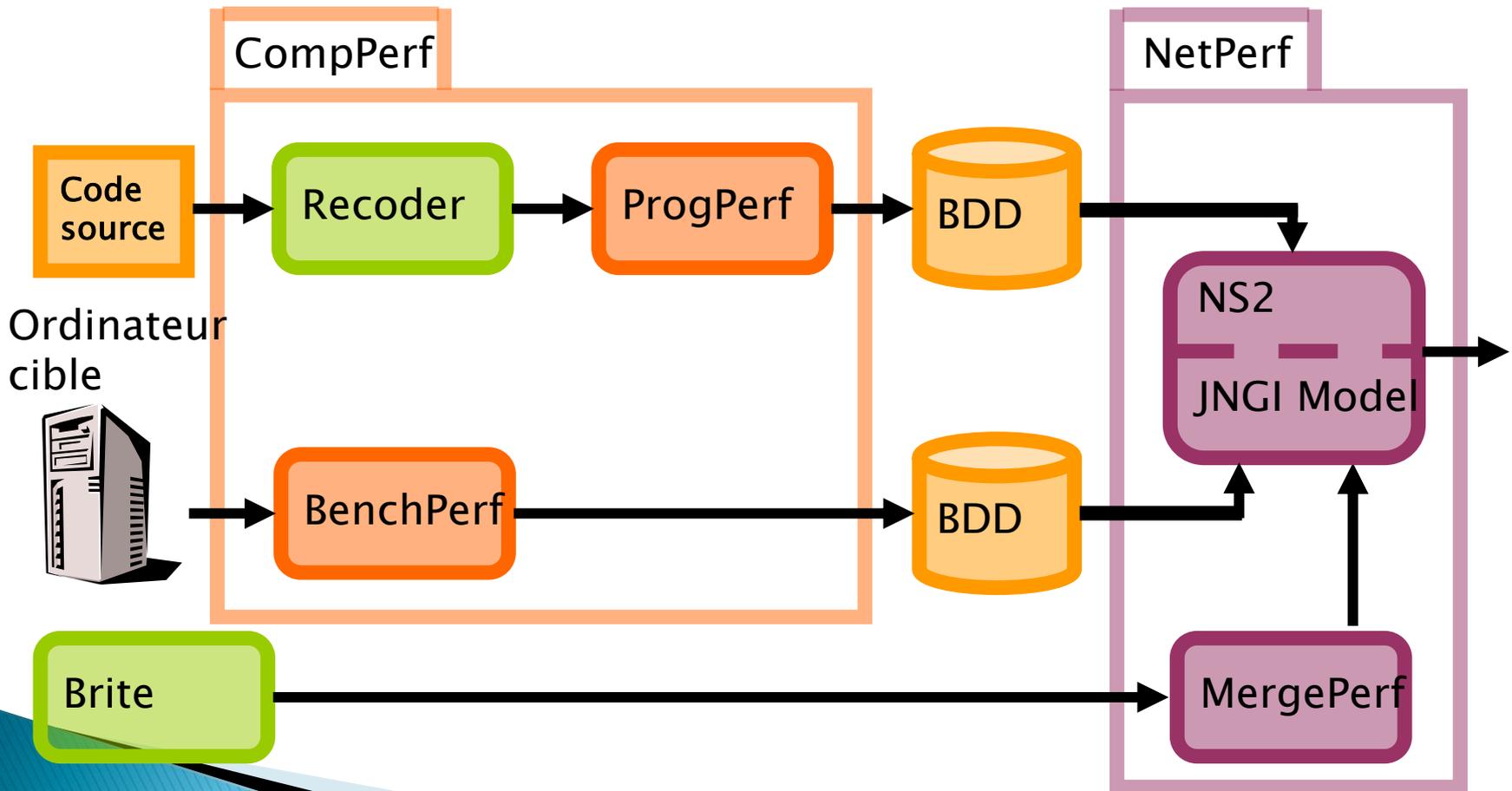
OverSim

- + Multi plates-formes
- + Modélisation précise du réseau
- + Collecte de données efficace
- Extensibilité

P2PPerf/dPerf

- ▶ Simulation d'applications de calcul distribués sur réseaux P2P
 - Doit inclure modèle CPU en plus du réseau
- ▶ Première version P2PPerf
 - Cible Java/JNGI en utilisant NS2
- ▶ Deuxième version
 - Cible application réelle déployée en C/C++/Fortran/MPI/P2Pdc en utilisant SimGrid ou intégrant SimGrid

P2PPerf : Architecture



CompPerf

- ▶ Objectif :
 - Permettre à NetPerf de calculer les temps d'exécution des parties séquentielles
- ▶ Séparer le logiciel du matériel
 - Comptage du nombre d'instructions du programme
→ ProgPerf
 - Définition du coût de chaque instruction pour chaque environnement matériel → BenchPerf

ProgPerf : Introduction

- ▶ Recoder
 - Construction, parcours et modification de l'arbre syntaxique du programme
- ▶ ProgPerf
 - Analyse sans exécution
 - Analyse à partir de traces

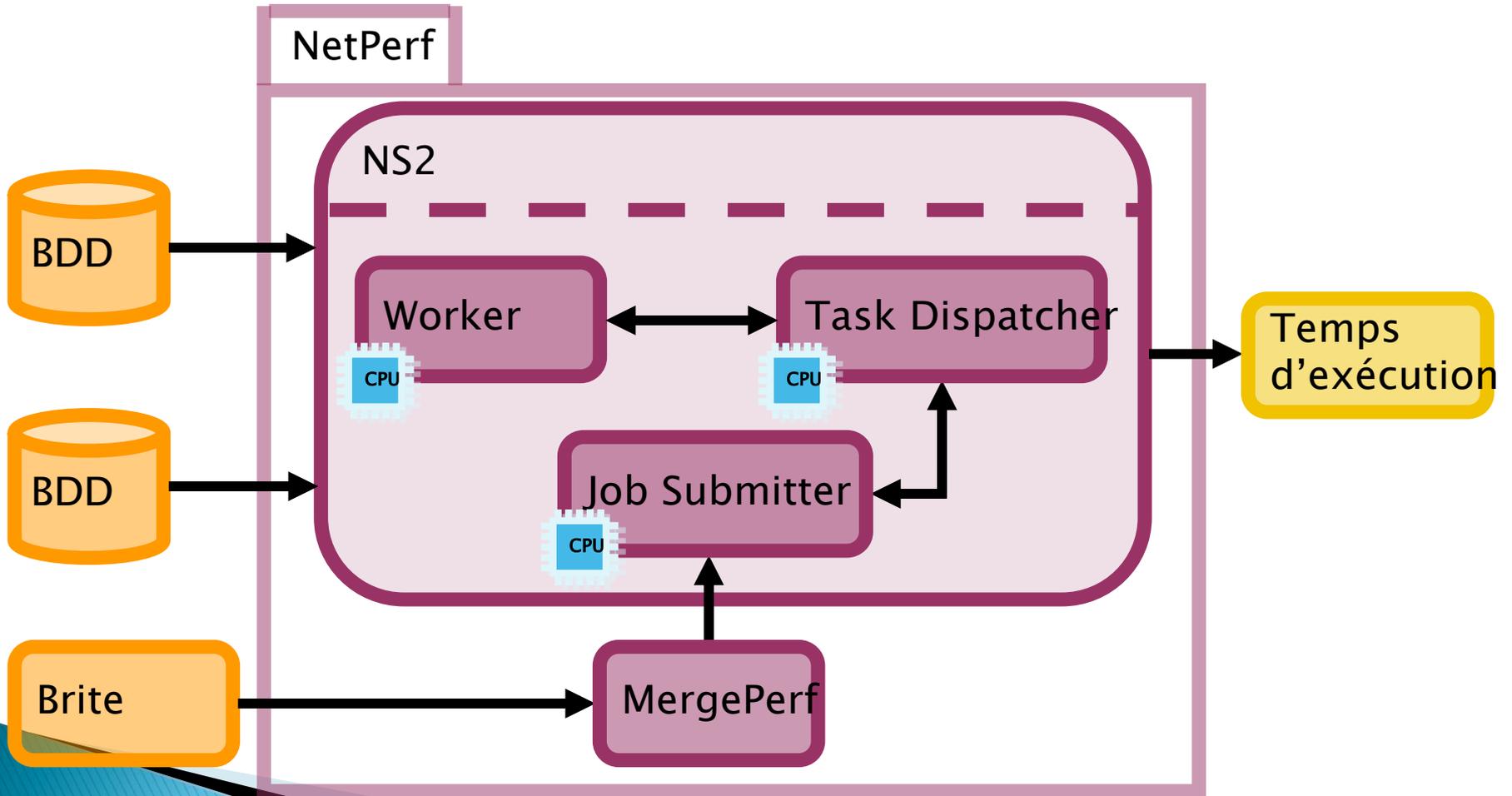
BenchPerf : Description

- ▶ Identifie les paramètres de performance des ordinateurs
- ▶ 354 instructions élémentaires
- ▶ Fonctionne par benchmarking
- ▶ Les résultats sont stockés dans une BDD
- ➔ Réutilisabilité

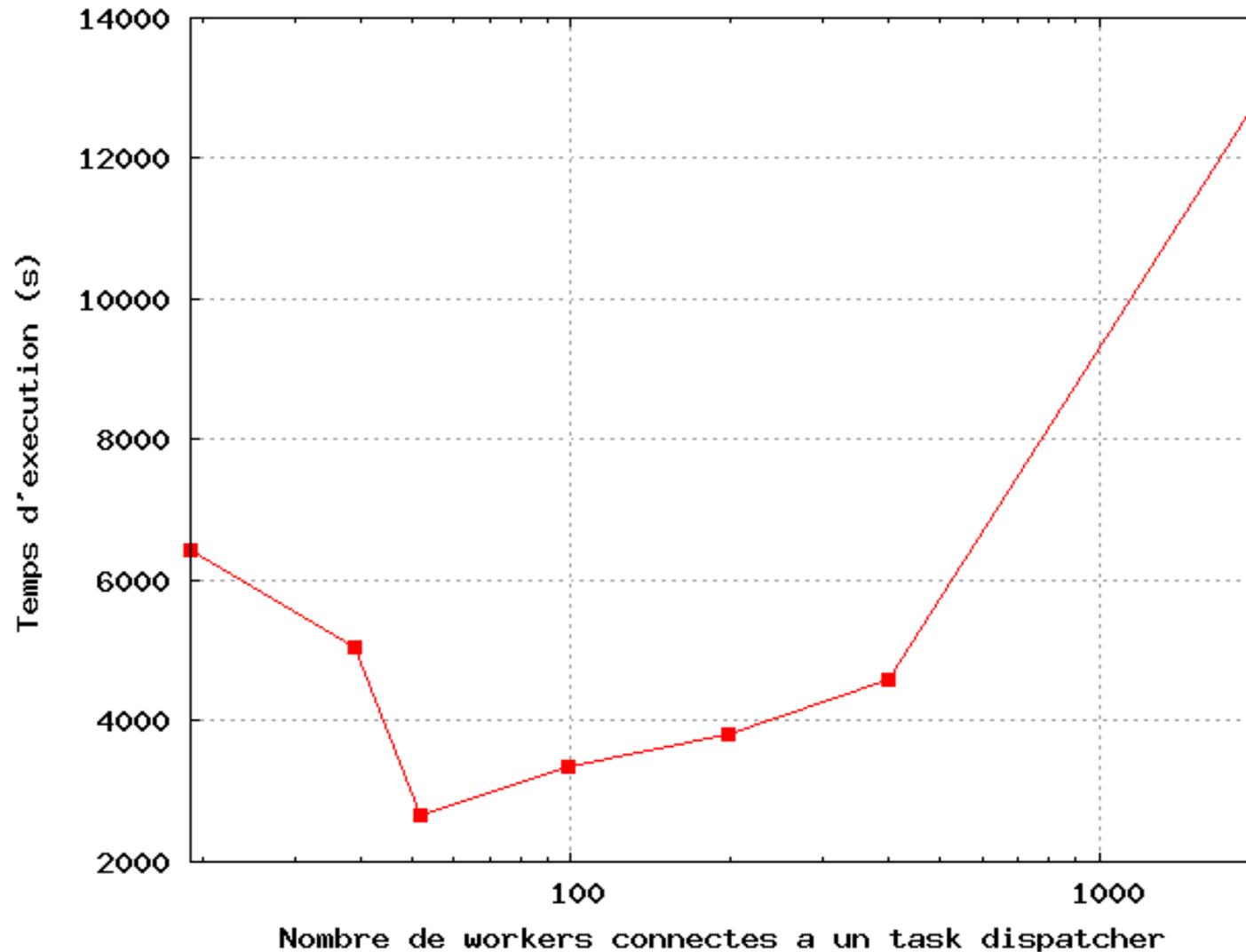
NetPerf : Fonctionnement

- ▶ Utilise NS2 pour les envois de paquets
- ▶ Implémentation d'un module JNGI dans NS2 avec :
 - Job Submitter
 - Task Dispatcher
 - Worker
- ▶ Simule le fonctionnement de JNGI
- ▶ Attribue les temps d'exécution en fonction de ses paramètres d'entrée

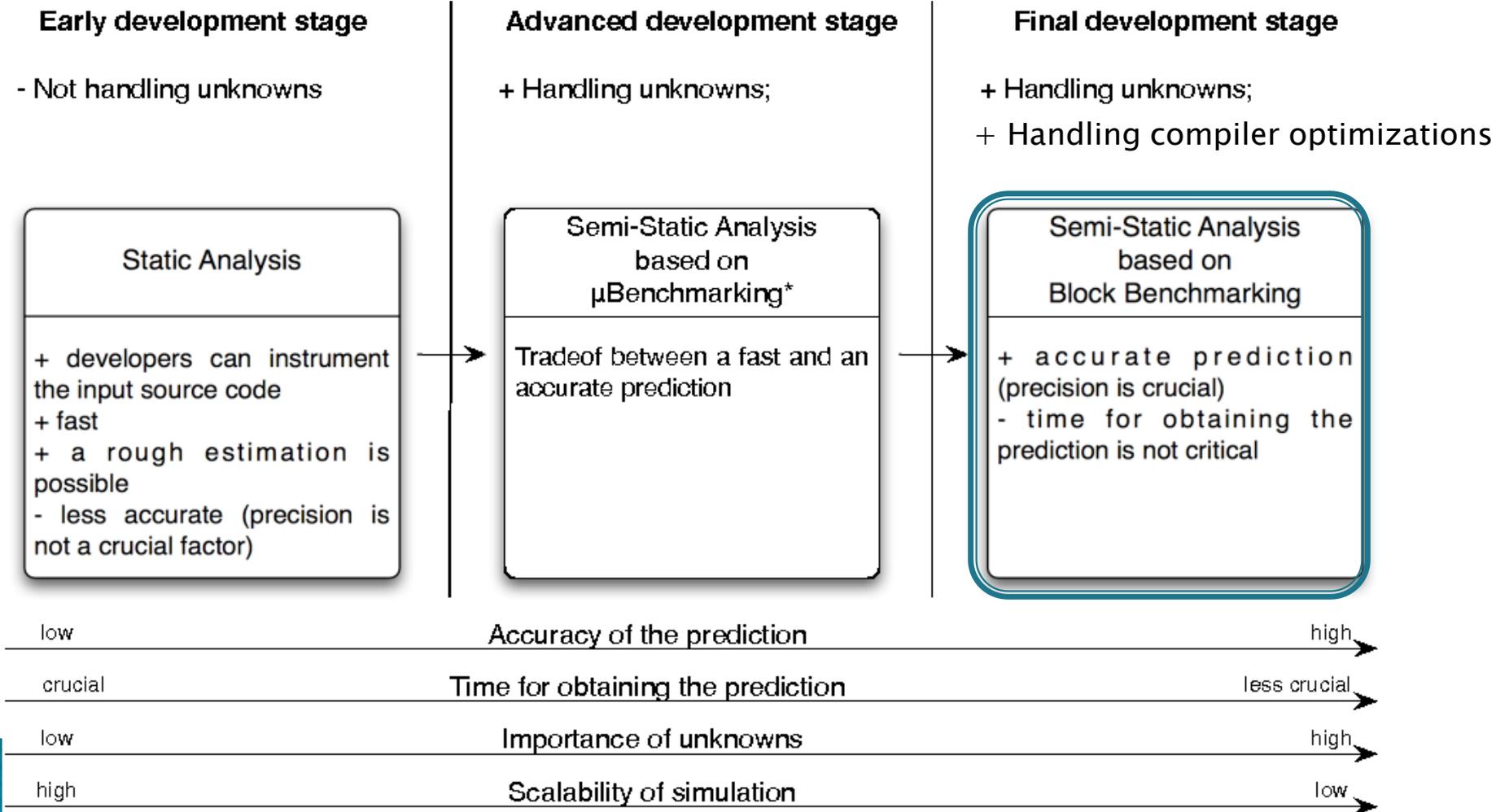
NetPerf : Architecture



Optimisation de JNGI



Life-cycle performance analysis

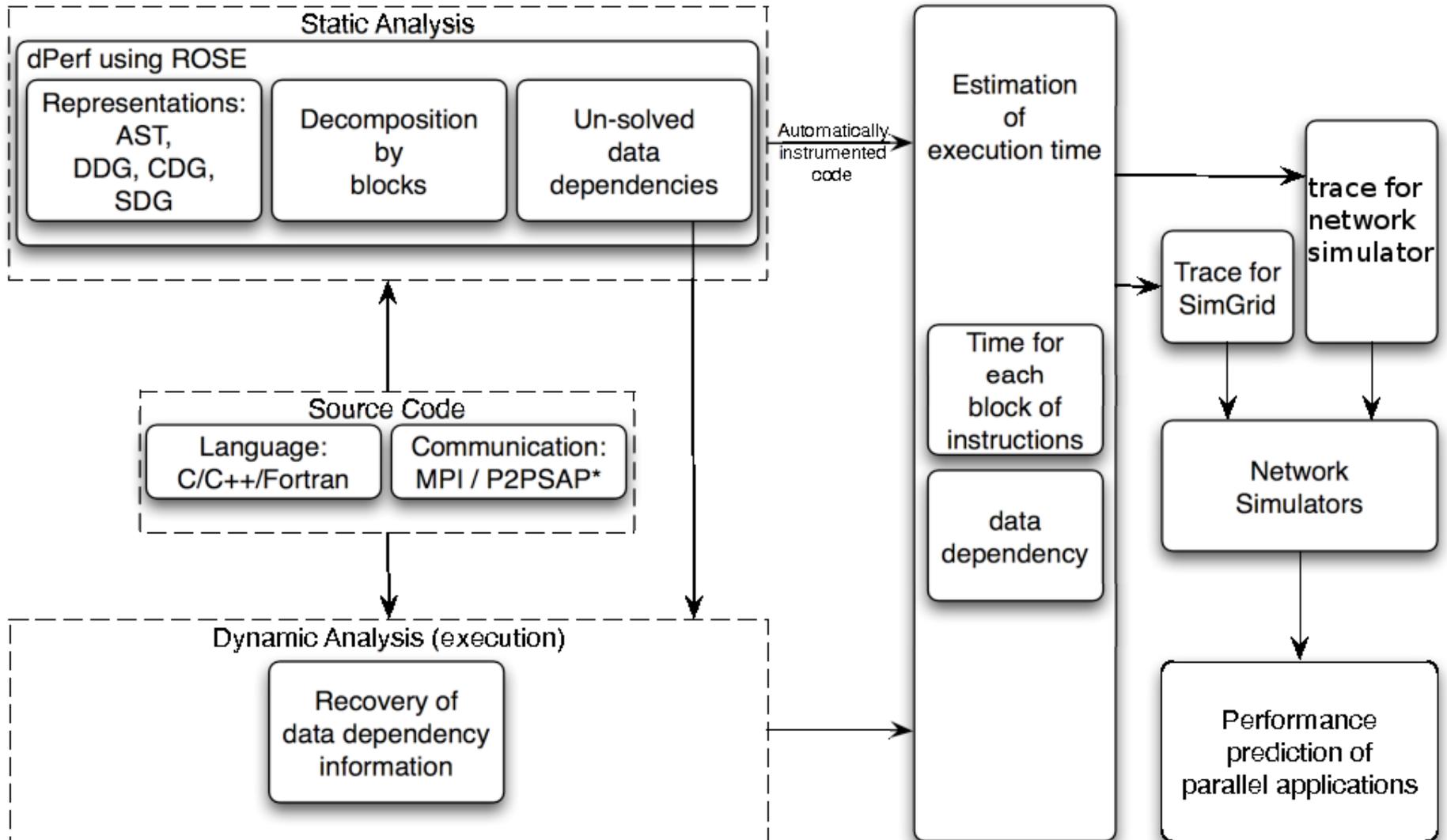


* μ Benchmarking = Benchmarking by micro-instructions

Life-cycle performance analysis

	Static	Semi-static	Block Benchmarking
Accuracy	Low	Average	Good
Slowdown	Faster	Faster	Slower or a bit faster
Unknowns	Can't solve	Can solve	Can solve
Compiler Opt.	Can't use	Can't use	Can use
Program parameters	Can extrapolate	Need execution	Need execution
Memory hierarchy	Can't handle	Can't handle	Handled partially
Number of computers	Can extrapolate	Fixed	Fixed
Network	Configurable	Configurable	Configurable
Program	Simples	Complex	Complex

dPerf



* P2PSAP is a self-adaptive communication protocol developed by the LAAS-CNRS team, for P2P computing systems

Plan

- ▶ Introduction
- ▶ Définition des réseaux P2P
- ▶ Caractéristiques des réseaux P2P
- ▶ Simulateurs de réseaux virtuels
- ▶ Simulateurs niveau paquet
- ▶ Emulateurs
- ▶ Environnement d'exécution
- ▶ Comparatif
- ▶ Conclusion

Emulateurs

- ▶ Environnements d'émulation de réseaux
 - Modelnet
 - EmuLab/PlanetLab
- ▶ Emulateurs dédiés P2P
 - Overlay weaver

Overlay Weaver

- ▶ Environnement de développement de réseaux virtuels
- ▶ Choix de l'émulation
 - Code réutilisable
 - Evaluation plus précise
 - Permet d'aller jusqu'aux tests réel
- ▶ Basé sur Java
- ▶ Configuration fichier texte + Java
- ▶ Repose sur un développeur, Kazuyuki Shudo

Overlay Weaver

- + Complètement modulaire
- ▶ Visualiseur runtime (peu de fonctionnalités)
 - Gestion du churn pas claire
 - Laisée à l'application ?
 - Extensibilité limitée (4000 nœuds)
 - Collecte des données insuffisante

Plan

- ▶ Introduction
- ▶ Définition des réseaux P2P
- ▶ Caractéristiques des réseaux P2P
- ▶ Simulateurs de réseaux virtuels
- ▶ Simulateurs niveau paquet
- ▶ Emulateurs
- ▶ Environnement d'exécution
- ▶ Comparatif
- ▶ Conclusion

Environnement d'exécution

- ▶ Splay (www.splay-project.org)
 - Simplifier le développement
 - Simplifier le déploiement
 - Contrôler l'exécution
- ▶ Structure
 - Langage basé sur Lua
 - Ensemble de démons
 - Contrôleur

Splay

- ▶ Exécution sandboxée
- ▶ Utilisation des ressources contrôlée
- ▶ Permet de spécifier le churn
- ▶ Peut être utilisé au-dessus d'un émulateur (ModelNet)

Splay

- + Exécution sécurisé
- + Facilité de déploiement des applications
- + Possibilité d'émulation

- Ré-écriture des applications
- Nécessite déploiement de l'architecture
- Nécessite l'installation d'un émulateur

Conclusion

- ▶ Prototypes de recherche
 - Documentation incomplète
 - Installation complexe
 - Comportement incohérent
- ▶ Java est le langage le plus utilisé
- ▶ Si prise en compte uniquement réseau, émulation est intéressante

dMEMS 2010

1st Workshop on hardware and software implementation and control of distributed MEMS

Besancon, France
June 28-29th 2010

<http://dmems.univ-fcomte.fr>

- Network of distributed sensors and actuators
- NoC and Soc design,
- Routing and switching in embedded networks, Cross-layer design,
- Data aggregation, data fusion,
- Distributed and peer-to-peer computing, swarm intelligence,
- Distributed applications



	Early Registration	Standard Registration
IEEE Member Full Registration	250 Euros	300 Euros
Non-member Full Registration	300 Euros	350 Euros
IEEE Member Student Registration	150 Euros	200 Euros
Non-member Student Registration	200 Euros	250 Euros
EXB banquet ticket	60 Euros	60 Euros



Questions?