MetaCC – Bilan et perspectives

Georges-André Silber
Centre de recherche en informatique
Ecole des mines de Paris
ACI GRID – Jeune équipe

Action Concertée Incitative
[ACI]
Globalisation des Ressources
Informatiques et des Données
[GRID]

Introduction

- Le but de l'équipe **MetaCC** est d'étudier et de construire une **infrastructure** distribuée sur la grille pour **l'optimisation de code** et la **compilation**.
- **Principes**, **algorithmes** et **outils** nécessaires à une telle infrastructure et à l'optimisation sur la grille.
- **Construction d'applications** optimisées pour la grille à l'aide de cette infrastructure.
- **Développement d'un portail** appelé **compilogrid** (compilogrid.org) et ouvert à la communauté pour utilisation et validation.

Contexte de création

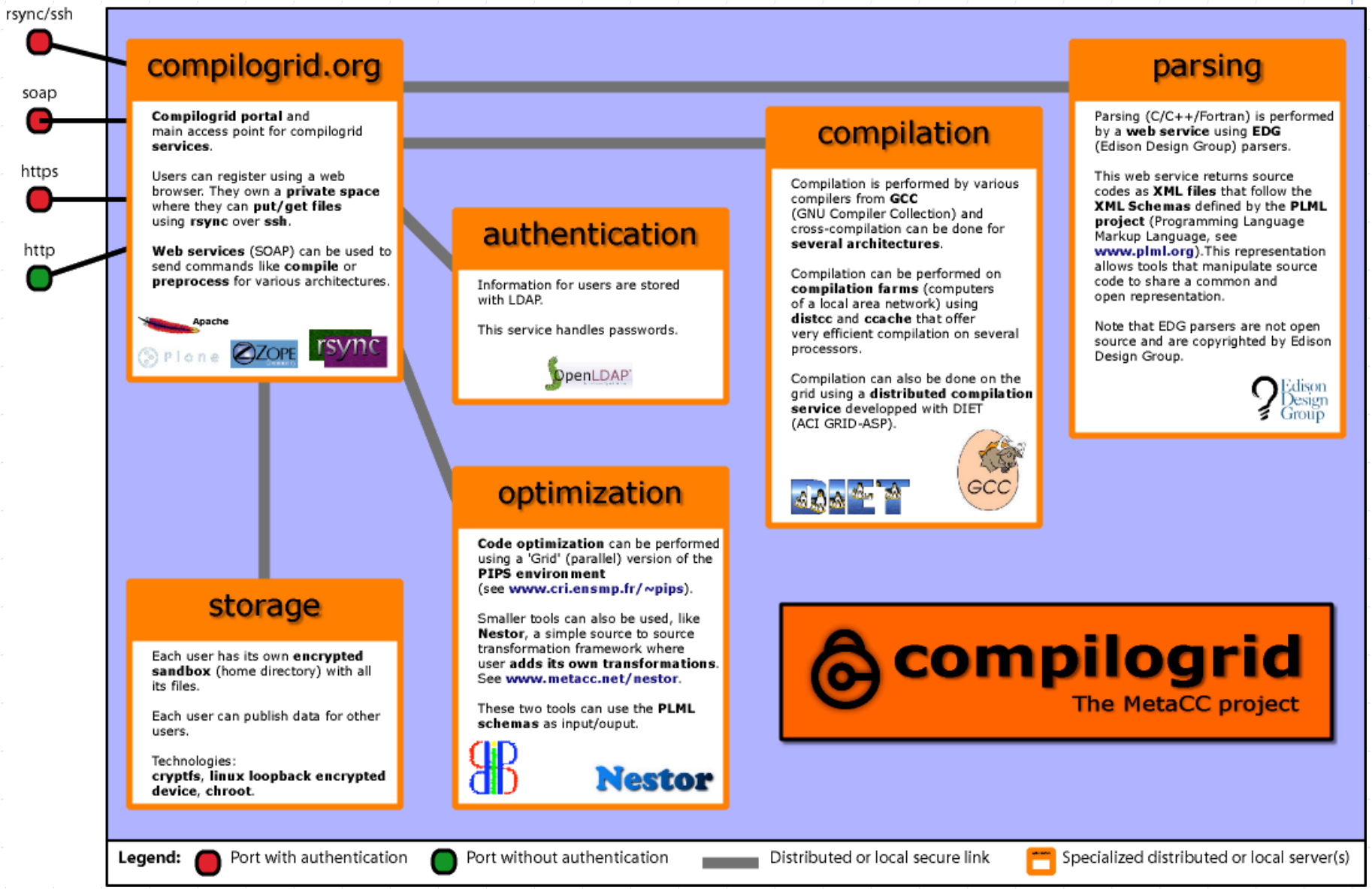
- Expertise en **parallélisation automatique** et optimisation de code (PIPS, Nestor).
- Difficulté d'utilisation des outils.
- Difficulté de déploiement des outils.
- Outils lourds (analyses).
- Analyses et optimisations « jouet » peu utilisables en pratique.
- Dépasser le compilateur généraliste !
- **Service de compilation**

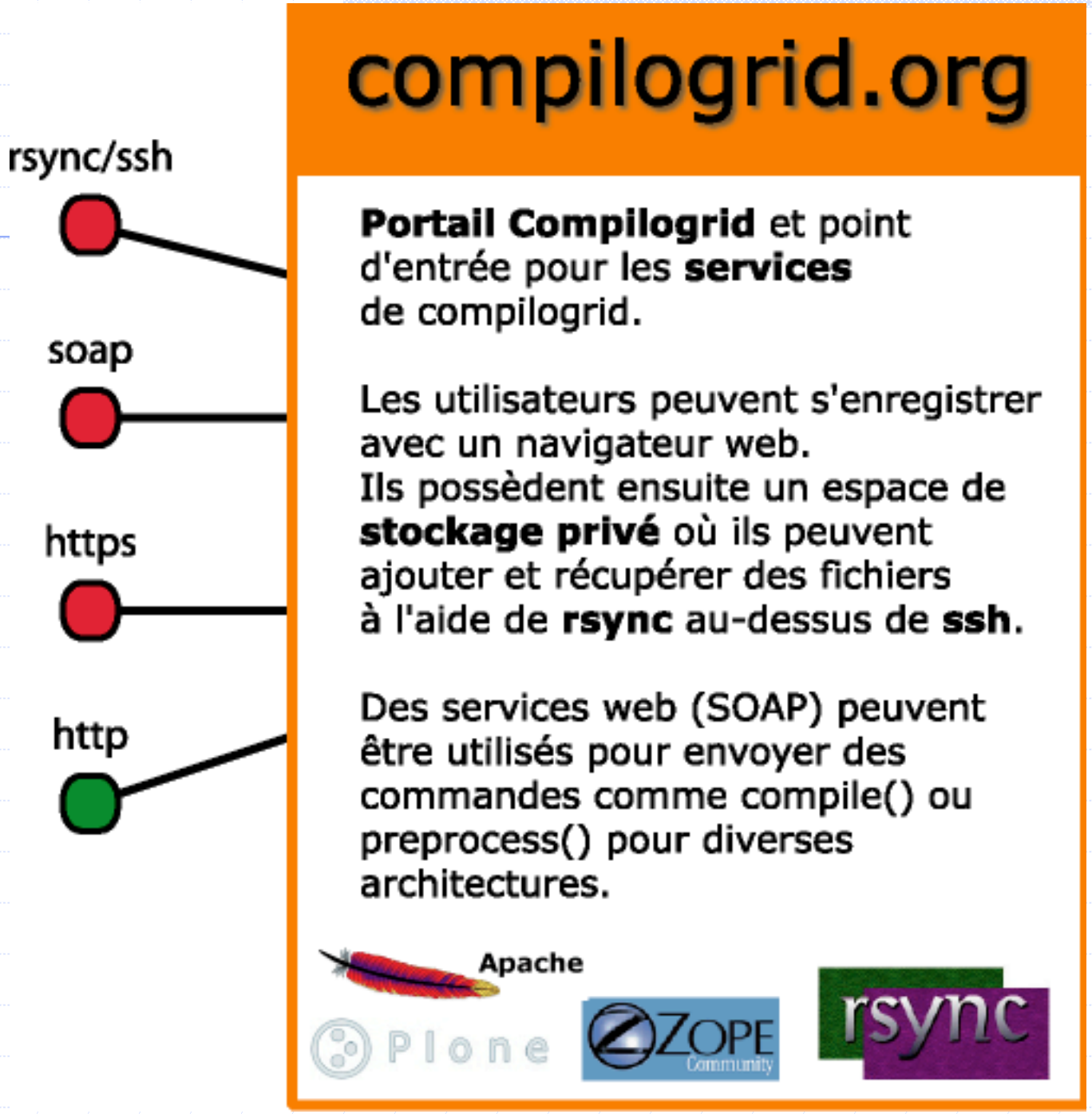
Equipe actuelle et évolution

- Notre équipe a changé depuis 2002:
  - 1 permanent
  - 1 thésard (depuis septembre 2002)
  - 1 stagiaire de maîtrise (fin en avril 2002)
  - 1 stagiaire ingénieur (fin en juillet 2002)
- Recrutement d'un doctorant en septembre 2003

Portail compilogrid

- Le portail **compilogrid** est le terrain de jeu de notre projet.
- Son développement est en cours (fonctionnel fin juillet 2003)
- Le but est de donner aux **utilisateurs distant** la possibilité **d'optimiser**, **analyser** et **compiler** leurs projets logiciels (code source en C, C++ et Fortran) pour diverses architectures (IA32, IA64, SPARC).
- Possibilité d'**enrichir** l'infrastructure.
- Les utilisateurs obtiennent une version **exécutable** pour l'architecture sélectionnée ou un **code source optimisé** et/ou **vérifié**.


rsync/ssh
soap
https
http
compilogrid.org
Compilogrid portal and main access point for compilogrid services.
Users can register using a web browser. They own a private space where they can put/get files using rsync over ssh.
Web services (SOAP) can be used to send commands like compile or preprocess for various architectures.
Apache
Plone
ZOPE
rsync
authentication
Information for users are stored with LDAP.
This service handles passwords.
OpenLDAP
compilation
Compilation is performed by various compilers from GCC (GNU Compiler Collection) and cross-compilation can be done for several architectures.
Compilation can be performed on compilation farms (computers of a local area network) using distcc and ccache that offer very efficient compilation on several processors.
Compilation can also be done on the grid using a distributed compilation service developped with DIET (ACI GRID-ASP).
DIET
GCC
parsing
Parsing (C/C++/Fortran) is performed by a web service using EDG (Edison Design Group) parsers.
This web service returns source codes as XML files that follow the XML Schemas defined by the PLML project (Programming Language Markup Language, see www.plml.org).This representation allows tools that manipulate source code to share a common and open representation.
Note that EDG parsers are not open source and are copyrighted by Edison Design Group.
Edison Design Group
optimization
Code optimization can be performed using a 'Grid' (parallel) version of the PIPS environment (see www.cri.ensmp.fr/~pips).
Smaller tools can also be used, like Nestor, a simple source to source transformation framework where user adds its own transformations. See www.metacc.net/nestor.
These two tools can use the PLML schemas as input/ouput.
Nestor
storage
Each user has its own encrypted sandbox (home directory) with all its files.
Each user can publish data for other users.
Technologies:
cryptfs, linux loopback encrypted device, chroot.
compilogrid
The MetaCC project
Legend:
Port with authentication
Port without authentication
Distributed or local secure link
Specialized distributed or local server(s)


compilogrid.org
rsync/ssh
soap
https
http
Portail Compilogrid et point d'entrée pour les services de compilogrid.
Les utilisateurs peuvent s'enregistrer avec un navigateur web.
Ils possèdent ensuite un espace de stockage privé où ils peuvent ajouter et récupérer des fichiers à l'aide de rsync au-dessus de ssh.
Des services web (SOAP) peuvent être utilisés pour envoyer des commandes comme compile() ou preprocess() pour diverses architectures.
Apache
Plone
ZOPE
Community
rsync

identification

Les informations concernant les utilisateurs sont stockées dans un annuaire LDAP.

Ce service gère les mots de passe et les différents droits d'accès aux autres services.

stockage

Chaque utilisateur a son propre bac à sable (**sandbox**) sous la forme d'un système de fichiers chiffré.

Technologies:
cryptfs, **linux loopback encrypted device**, **chroot**.

A = 2*3

```
<plml:assign>
 <plml:left>
  <plml:var id='A'/>
 </plml:left>
 <plml:right>
  <plml:binexp t='*'>
   <plml:op>2</plml:op>
   <plml:op>3</plml:op>
  </plml:binexp>
 </plml:right>
</plml:assign>
```

parsing

L'analyse syntaxique et grammaticale des programmes est effectuée par un **service web** utilisant le **parseur** d'EDG (Edison Design Group).

Ce service retourne le code source analysé sous la forme d'un fichier XML qui suit un **schéma XML** que nous avons défini: **PLML** (Programming Language Markup Language, www.plml.org). Cette représentation permet aux outils de partager un **représentation commune** du code et de ne pas avoir à se se soucier du parseur.

compilation

La compilation est prise en charge par les compilateurs de **GCC** (GNU Compiler Collection) qui permet la compilation croisée pour de nombreuses architectures.

La compilation s'effectue de manière **parallèle** et **efficace** sur des **fermes de compilation** (ordinateurs proches) grâce à **distcc** et **ccache**.

Cette compilation est également accessible sous la forme d'un **service de compilation distribué** en cours de développement avec DIET (ACI GRID - ASP).

optimisation

L'**optimisation** et l'**analyse** des codes est effectuée avec une version "grille" parallèle de l'environnement PIPS (http://www.cri.ensmp.fr/~pips).

Nestor, un outil plus simple et plus extensible peut être utilisé, où l'utilisateur peut ajouter ses propres modules (metacc.net/nestor).

Ces deux outils utilisent **PLML** en entrée et en sortie.

Nestor

« Gridification » de PIPS

- Outil d'analyse et de transformation « source à source ».
- **Interprocédural.**
- Phases d'analyse et d'optimisation interdépendantes et dynamiques.
- Sa parallélisation/distribution est un problème très difficile.
- Étude en cours (depuis septembre 2002).
- Utilisation de PLML.

Évolution de Nestor

- Nestor est un outil permettant de développer des transformations « source à source ».
- Orienté-objet (élément du code = objet).
- Ouvert (ajout de transformations par bibliothèques dynamiques).
- Évolution
  - possibilité de rajouter des modules sur compilogrid
  - Passage à Java
  - Utilisation de PLML (C/C++/Fortran)

Thèse depuis septembre 2002

- « *Outils et algorithmes pour la compilation et l'optimisation à distance* »
- Pham Quoc Dat, depuis septembre 2002.
- Financement école des mines.
- Portage de PIPS sur la grille.
- Développement **compilogrid**.

Bilan

- Refonte et positionnement de l'équipe
- Affinage du projet
- Tour d'horizon des technologies
  - Utilisation de l'existant / nouveau
- Étude de PIPS //
- Étude PLML
- Démarrage du développement
  - Évolution de PIPS et Nestor
  - Portail compilogrid, middleware
  - Stockage, identification

Bilan (suite)

- Évaluation compilation parallèle
  - Distcc, ccache: très bonne efficacité
- Démarrage d'une thèse
  - Premiers résultats attendus pour septembre 2003
- Papier synthétique
  - « The MetaCC project: towards a compilation grid »

Perspectives

- Développement à poursuivre
- Matériel à mettre en place
  - cluster du pauvre de 12 machines
  - 2 machines pour le portail
- Démarrage d'une nouvelle thèse
- Extension vers l'exécution du code après compilation

Thèse en septembre 2003

- « *Étude et implantation sur une grille de calcul d'un encodeur vidéo MPEG4 parallèle et optimisé à plusieurs niveaux* »
- Financement obtenu de l'Ecole des mines
- Mesurer l'impact d'une optimisation à plusieurs niveaux (grille, cluster, SMP, ILP, EPIC) sur la performance d'une application courante et moyennement gourmande en ressources.
- Proposer de nouvelles approches pour la parallélisation sur la grille.
- Évaluation et enrichissement de **compilogrid**.

Exécution du code

- Web service **d'exécution à distance**
  - **RED** : Remote Execution Daemon
  - « applet à l'envers »
- Permettre l'exécution après compilation
- Compilation en fonction de l'architecture cible (**dynamique** ?)
- Frein: **problèmes de sécurité** !
  - ACI Sécurité: compilation et système d'exploitation sur processeur chiffré.

Conclusion

- Longue mise en place du projet et de l'équipe
- Compilogrid fonctionnel fin juillet 2003
- Ouverture vers la communauté
  - Besoin d'utilisateurs pour valider
  - Besoin de développeurs
- Plate-forme pour la recherche en compilation et optimisation de code ?