MetaCC

Compilation et exécution nomades

Georges-André Silber

Centre de recherche en informatique

École des mines de Paris

`http://www.metacc.net`

CENTRE DE RECHERCHE EN INFORMATIQUE

ACI GRID

MINISTERE DE LA RECHERCHE

ECOLE DES MINES DE PARIS

Compilation et exécution nomades

Permettre à un utilisateur d'accéder à ses données, de compiler et d'exécuter du code n'importe où, depuis n'importe où.

Compile and execute everywhere!

- Offrir des services disséminés sur un réseau.
- Pour tout ce qui est coûteux, critique.
  - Stockage (sources, données, exécutables, analyses).
  - Analyse et transformation (source à source).
  - Compilation (source vers exécutable).
  - Exécution (machines diverses).

L'univers Internet

- Le monde GRID (`http://www.globus.org`).
  - Transferts et accès sécurisés, acquisition de ressources, équilibrage de charges, ...
  - *Lingua franca* de représentation et d'exécution du calcul ?
- Le monde WWW (`http://www.w3.org`).
  - Transferts et accès sécurisés, équilibrage de charges, acquisition de ressources, ...
  - *Lingua franca* de représentation et mouvement de données ?

Approche

- Les autres approches GRID (ex: GrASP) supposent un environnement de programmation dédié:
  - réécriture des codes,
  - utilisation de blocs de base (composants).
- Frein à l'utilisation de la grille ? Industrie:
  - Fortran 77 et C optimisés/bidouillés.
  - Étude de cas avec PIPS.

Down–top

- Majoritairement: vision de haut en bas.
- Nous avons l'approche de bas en haut.
- Adaptation des outils existants au Metacomputing.
  - `emacs`
  - `autoconf`, `automake`, `libtool`, `make`, `gcc`
  - `ld`, `ar`, `as`, `ld.so`, `libc.a`

Plan

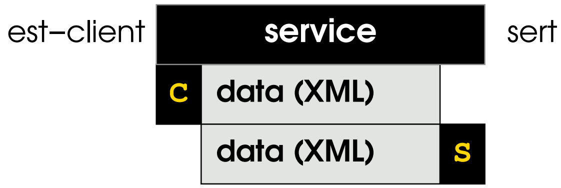
- Problématique
- Travaux et réalisations
- Présentation de l'équipe
- Communication interne et externe
- Calendrier

Problématique

Services

- Pour réaliser tous ces services, il y a une foule de problèmes à résoudre.
  - Transparence (facilité d'utilisation).
  - Sécurité (sources, données, exécutables).
  - Protocoles d'échanges, formats des données.
  - Performance à tous les niveaux.
  - Middleware pour faire communiquer les services.
  - Compilation (optimisation de code).
- Tentative d'isolement des problèmes.

Service virtuel

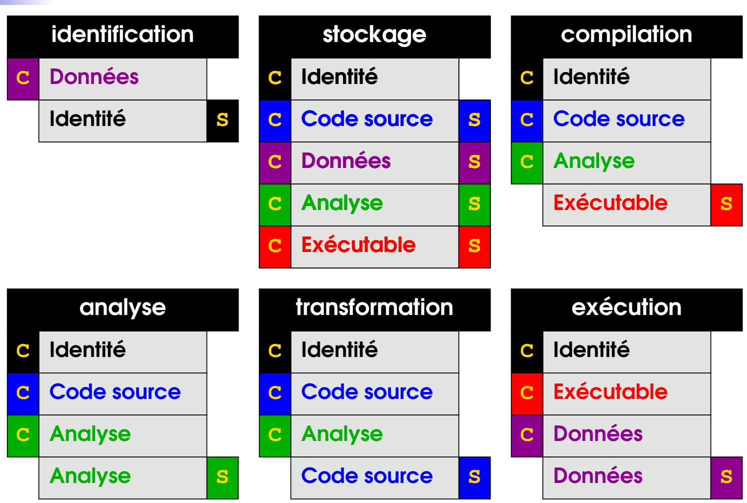


- Machine de Turing.
- data est subdivisé en code/données.
- Un code `info` de contrôle du service.

Spécialisations

- Dérivation en six services spécialisés :
  identification, stockage, analyse,
  transformation, compilation, exécution.
- Spécialisation des canaux de données :
  identité, code source, exécutable,
  analyse, données, paramètres.
- Un code `info` particulier par service.

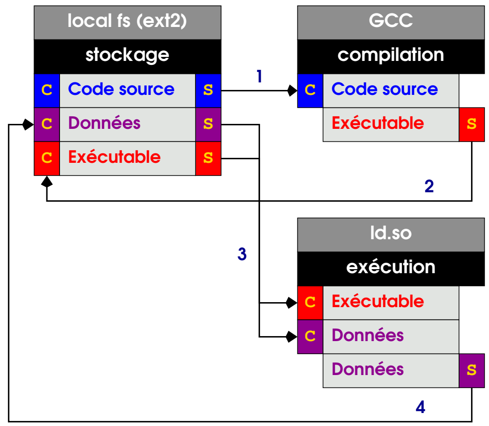
Services spécialisés



Réalisation

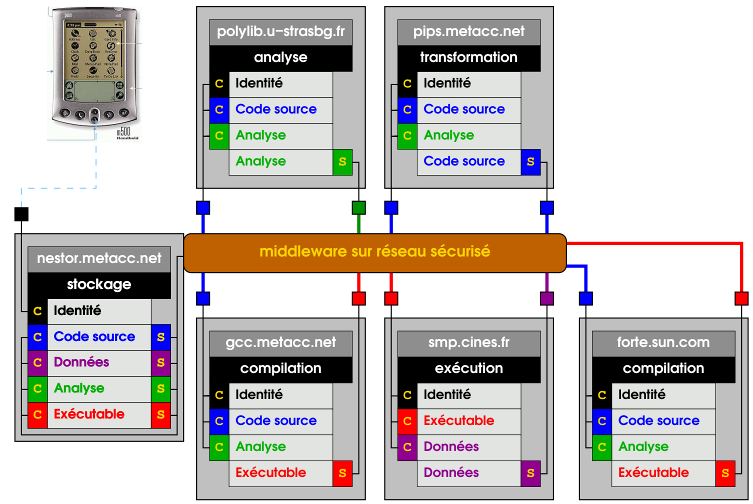
- Réalisation des services spécialisés (implémentation).
- Par exemple:
  - `ftp` réalise le service stockage,
  - `petit` réalise le service analyse,
  - `nestor` réalise le service transformation,
  - `gcc` réalise le service compilation,
  - `ld.so` réalise le service exécution.

Contexte local




```
# gcc matmul.c      1, 2
# ls
a.out data matmul.c
# a.out data        3, 4
# ls
a.out data matmul.c
result
#
```


Contexte nomade



Avantages

- Dépasser la frontière du compilateur généraliste.
- Communauté de la compilation.
  - Test en vrai grandeur !
- Architecture. Simulateur d’exécution.
- Utilisateur:
  - Machine virtuelle puissante.
  - Write Once Everywhere, Run Efficiently Everywhere.
  - Sans réécriture de code.

Études et réalisations

Problèmes transversaux

- Service identification.
  - SSL, SSH, Kerberos, autres ?
- Service stockage.
  - `rsync`, `sftp`, caches, Avaki ?
- Implantation du code `info`.
  - Service, architecture, charge, etc.
  - Sur le modèle d'Apache (`http://localhost/server-status`)

Protocoles et formats

- Un protocole par service.
  Comment faire communiquer les services ?
- Un format (langage) par canal de données.
  Comment faire se comprendre les services ?

Formats

- XML est utilisé comme moyen de structuration des données unifié.
- Quelques avantages de XML:
  - fichiers texte (portables), parseurs gratuits disponibles,
  - transformation de XML $\rightarrow$ n'importe quoi (XSLT, XPath),
  - récupération d'information (XFragments, XQuery),
  - des grammaires explicites (DTD, XSchema),
  - sécurité (XML Encryption, signature),
  - un standard (W3C) !

Formats — PLML

- Programming Language Markup Language.
  Une famille de dialectes XML.
  - C/C++
  - Java
  - Fortran (77, 95, 2000)
  - SciLab
  - Assembleur
- `http://www.metacc.net/plml.html`
- Travail en cours avec Paul Feautrier (A3).

MetaPIPS, polylib

- Réalisation du service transformation et analyse.
  - Parallélisation du code de PIPS.
  - `http://www.cri.ensmp.fr/~pips`.
- Polylib, pip, omega library, paf, Nestor.

Compilation

- GCC – Gnu Compiler Collection (différentes architectures).
  C, C++, Fortran, Java, ...
- ORC – Open Research Compiler (IA-64).
- Prise en charge des différents paramètres.
- Services très spécialisés (`gcc -O3`).
- Compilateurs commerciaux ?

RED

- Remote Execution Daemon.
- Réalisation du service exécution.
- `rsync+ssh`, `chroot`, `rexec`.
- Traces d'exécution.
- Contrôle de l'exécution (code `info`).
- *Vraie machine* ou simulateur.
- JVM.

Middleware

- Rôle de *glue* entre les services.
- Ordonnancement des tâches à accomplir.
- Choix du “meilleur” chemin de données.
  - Collaboration avec ACI ASP (Diet).
- Performances des transferts.
  - Collaboration avec ACI RESAM.
- Choix des paramètres pour les services.

Client léger

- Contrôle de la glue.
- Simple analyseur syntaxique.
  - erreurs de syntaxe traitées en local !
- Source $\rightarrow$ XML.
- Applet, PalmOS, `emacs`, PHP, ...

Présentation de l'équipe

CRI — Équipe ATIP

- Analyses statiques et transformations de codes.
- *Optimisation, maintenance, sécurité, rétro-ingéniérie.*
- Projets : ABC, Conso.
- Logiciels : `PIPS`, `Nestor`.
- Membres : Irigoin, Ancourt, Jouvelot, Coelho, Silber.
- Relations industrielles : EDF, Sagem.
- Relations scientifiques : A3 (INRIA), ReMaP, CEA, ...
- Veille technologique : compilateurs.

CRI — Équipe ADM

- Internet, multimédia et architectures documentaires.
- *Moteurs de recherche, e-learning, sites web.*
- Projets : BIAM, droit.org, VIDAL, OTAN, Préfecture.
- Membres : Mahl, Jouvelot, Daverio, Olivier.
- Veille technologique : technologies web.

MetaCC

Jeune équipe financée par l'ACI Grid 2001.

- **Georges-André Silber** (coordinateur)
  Parallélisation automatique de nids de boucles, transformation de code source à source (Nestor).
  Thèse décembre 1999, Alain Darte, LIP.
- **Fabien Coelho**
  Compilateurs pour langages data-parallèles (HPF), analyses de programmes.
  Thèse octobre 1996, François Irigoin, CRI.
- **Laurent Daverio**
  Technologies WEB, base de données, systèmes d'exploitation.

Collaborations

- Projet A3 INRIA (Rocquencourt)
- Projet CompSys (Lyon)
- Projet ReMaP (Lyon)
- ACI ASP (Lyon)
- ACI RESAM (Lyon) ?
- CEA (Saclay) ?

Communication

Communication externe

- `http://www.metacc.net`, domaine `metacc.net`.
- Conférences, workshops, journaux.
  - Europar 2002
  - MSA 2002
  - SC 2002
- Notes de travail et rapports (CRI).

Calendrier

Ordonnancement

T0+6 Protocoles et formats v0.1.
Implémentation jouet des services v0.1.
Dissémination (Bleau, Evry, Paris, Sophia, InternetFR).
Dissémination amis (Lyon, Paris).
Glue “a la main”.

T0+12 v0.1 $\rightarrow$ v1.0.
Document de référence (communautaire).
Implémentation services v1.0 + instances.
Glue v0.1 + clients légers.

T0+18 Glue v1.0.
Tests.

T0+24 MetaCC v1.0.

Informations

- Rapport interne CRI E-249.
- `http://www.metacc.net`