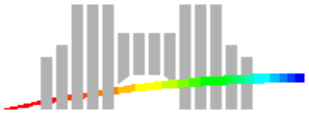


Contribution à l'algorithmique parallèle

Des bibliothèques de calcul au metacomputing

Frédéric Desprez

Projet CNRS - ENS Lyon - INRIA ReMaP

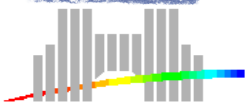
LIP ENS Lyon

INRIA Rhône-Alpes

Plan de l'exposé

- **CV**
- **Introduction**
- **Recouvrements et pipelines**
  - Projets LOCCS/OPIUM
- **Parallélisation de programmes Fortran**
  - Projet TransTool
- **Problem Solving Environments**
  - Projets Scilab//, DIET
- **Conclusions et perspectives**

Curriculum Vitae



[1994] **Postdoctorat**
à l'Université du Tennessee à l'ICL

[1994-1995] **Maître de Conférence** à l'ENSEIRB
Recherche dans l'équipe ALiEnor

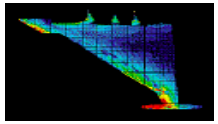
[1995-] **Chargé de Recherche** à l'INRIA, LIP ENS Lyon
projet CNRS-INRIA-ENS Lyon ReMaP

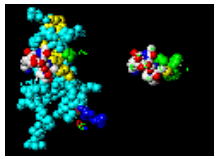
[Sep.2000-] Responsable du projet ReMaP

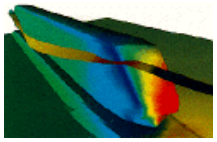
- Encadrement de 6 thèses, 9 DEA, 3 DESS, …
- Projets européens et internationaux EuroTOPS (93-98), TTN ProHPC (97-99) et Paralin (96-99)
- Projets nationaux ARC OURAGAN (98-00) et RNRT VTHD (99-01)
- Enseignement à l'ENS, à l'ENSEIRB, à SUPELEC,
- Comité d'édition de Parallel and Distributed Computing Practice
- Organisation des JIP'94-95, MSA'2000-01 dans ICPP

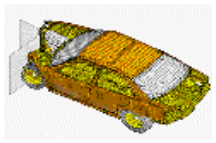
Introduction

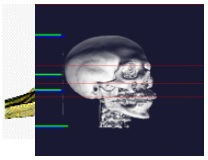
- Calcul numérique utilisé dans de nombreuses applications
- Simulations toujours plus précises ⇒ besoins importants en puissance de calcul et en capacité de stockage
- **Evolution rapide des architectures**
  - Processeurs superscalaires (du commerce) + augmentation des fréquences d'horloge, mémoires de plus en plus hiérarchiques
  - Machines parallèles (vectorielles, SM, DM, SIMD, MIMD, réseaux de stations, SMP, grappes de SMP, plates-formes de metacomputing) ⇒ de + en + d'hétérogénéité
- **Evolution plus lente du logiciel**
  - Pas de langage vraiment adapté au parallélisme
  - Pas de standard
  - La plupart des applications programmées en C/F77 + MPI
  - Problèmes de parallélisation automatique, debug, profiling, etc

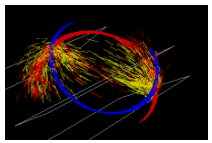


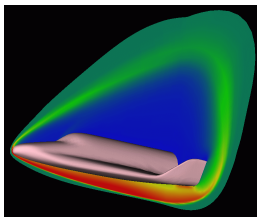






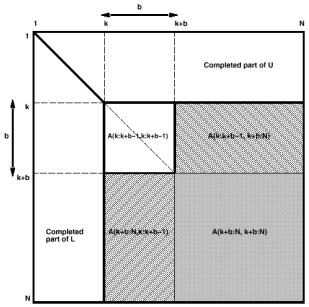


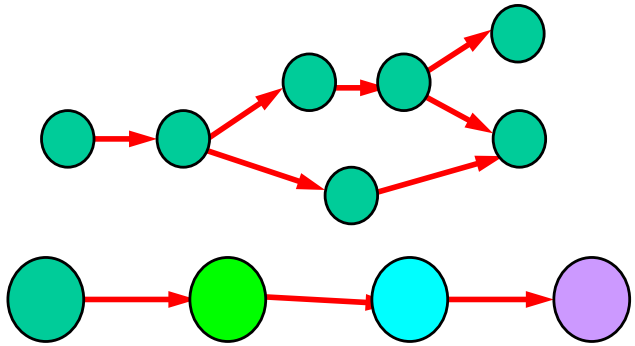




Parallélisation d'applications numériques

- **Plusieurs sources de parallélisme**
  - Parallélisme de données
  - Parallélisme de tâches
  - Parallélisme de flux





- **Plusieurs modèles de programmation en mémoire distribuée**
  - Passage de messages
  - Parallélisation automatique
  - Langages data-parallèles
  - Bibliothèques parallèles
  - Java ? OpenMP + threads?
  - Mélanges de langages, de bibliothèques et de supports d'exécution à hautes performances

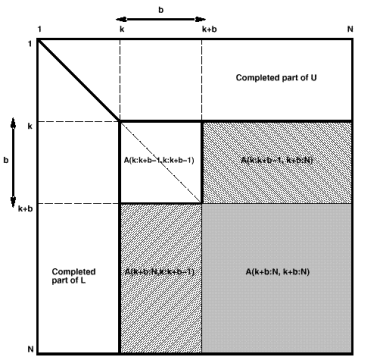
Bibliothèques numériques

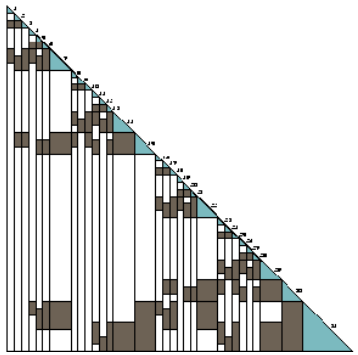
- **Avantages**
  - Le parallélisme est masqué à l'intérieur de la bibliothèque
  - Utilisation de noyaux numériques optimisés sur les processeurs (BLAS de niveau 3, ATLAS, PhiPAC)
  - Eviter le choix de l'algorithme de résolution
  - Maintenance et mise à jour

- **Inconvénients**
  - Choix de la distribution laissé à l'utilisateur
  - Insertion de redistributions entre les étapes de calcul
  - Pas de choix de l'algorithme de résolution
  - Transformation des structures de données des applications
  - Encore limitées (creux)

Problem type Ax = b	SDrv	EDrv	Factor	Solve	Inv	Cond Est	Iter Refin
Triangular				X	X	X	X
SPD	X	X	X	X	X	X	X
SPD Banded	X		X	X			
SPD Tridiagonal	X		X	X			
General	X	X	X	X	X	X	X
General Banded	X		X	X			
General Tridiagonal	X		X	X			
Least squares	X		X	X			
GQR			X				
GRQ			X				

Ax = λx or Ax = λBx	SDrv	Edrv	Reduct	Solution
Symmetric (2 types)	X	X	X	X
General (2 types)			X	X
Generalized BSPD		X	X	X
SVD			X	X





Problem Solving Environments

- **Besoin des utilisateurs**
  - Applications simples à exécuter (parallélisme obligatoirement caché) dans un environnement intégré
  - Exécuter n'importe quand des simulations sans limitation de taille
  - Partager des codes entre les utilisateurs
  - Pouvoir collaborer avec d'autres disciplines
  - Utiliser efficacement et simplement des ressources disponibles à travers le réseau
  - Mettre en commun des puissances de calcul
- Avoir des **environnements intégrés à hautes performances**

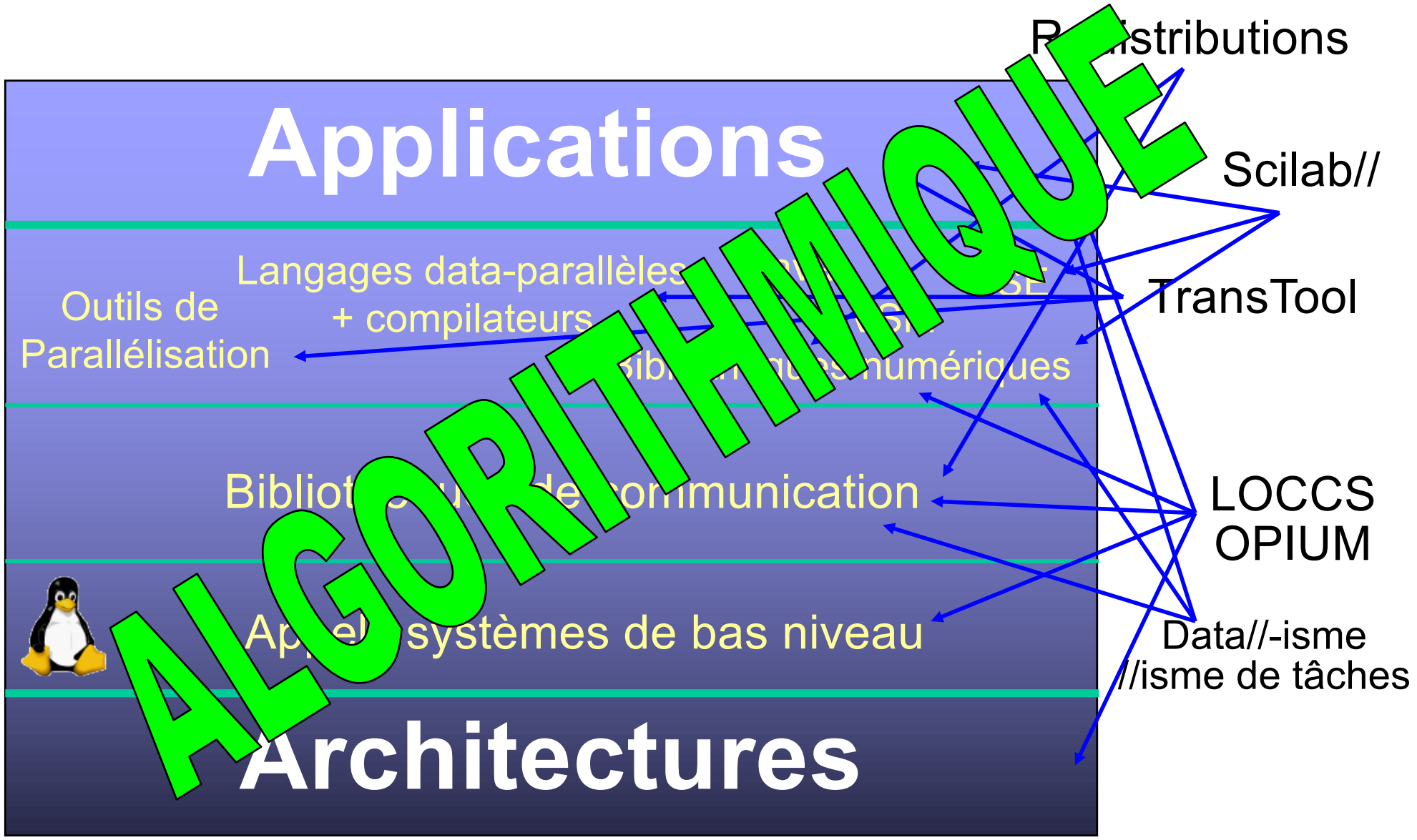
  calcul, visualisation + couplage de codes

ASP *(Application Service Provider)*

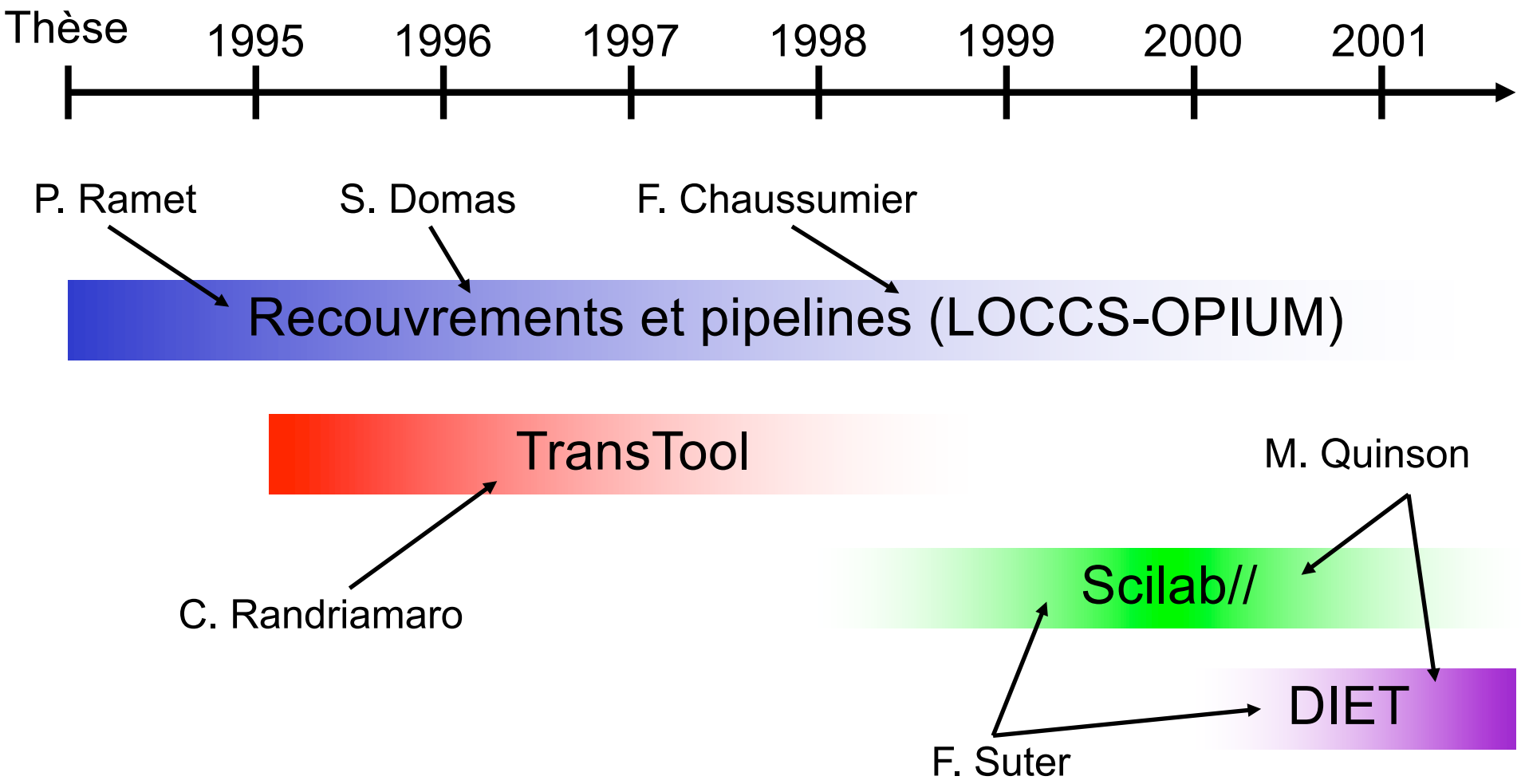
- ✓ **Une idée à long terme :** location de services de calcul via le réseau
  - – Accès des performances en calcul et en capacité mémoire
  - – Logiciels ou données protégés
  - – Utilisation de serveurs de calcul accessibles via le(s) réseau(x)
  - – Mais
  - ☹ Généralement autres domaines d'applications (BD, finances, …)

- ✓ **GridRPC** (Netsolve, NINF, RCS, …)
  - ☹ Toujours difficiles à utiliser pour les non-spécialistes
    - ☹ Pratiquement pas de transparence
    - ☹ Problèmes de sécurité généralement pas traités
  - ☹ Souvent dépendants des applications
  - ☹ Manque de standards (CORBA, JAVA/JINI, sockets, …) pour construire les serveurs de calcul

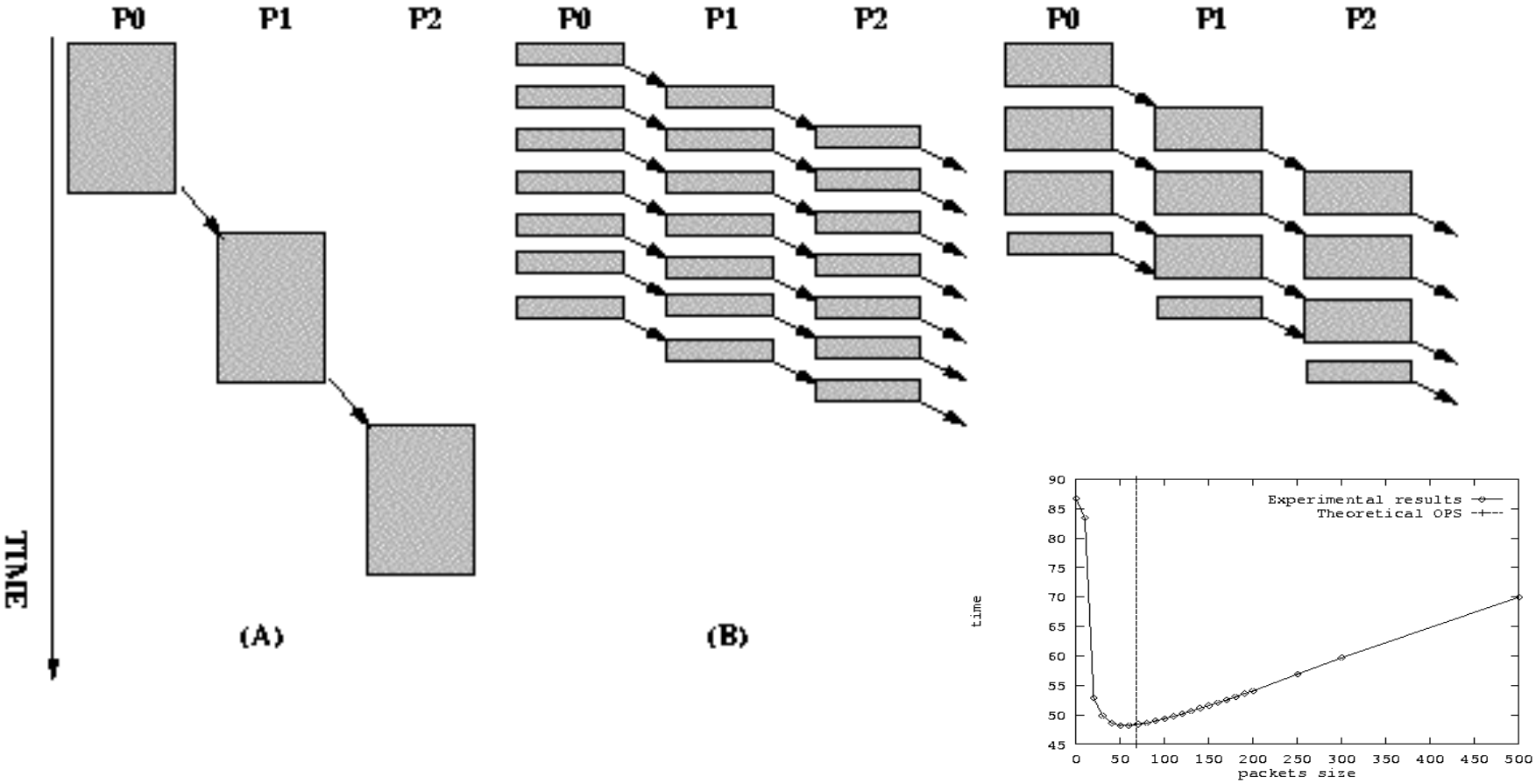
Positionnement de mes travaux



Projets de recherche et thèses

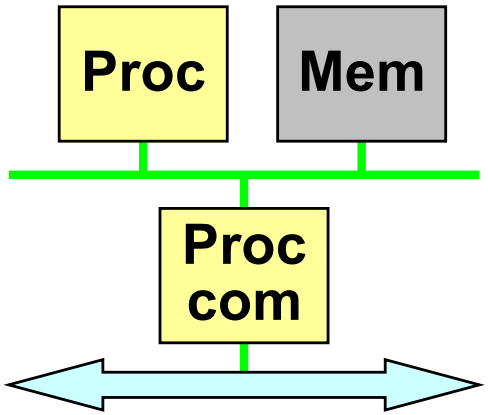


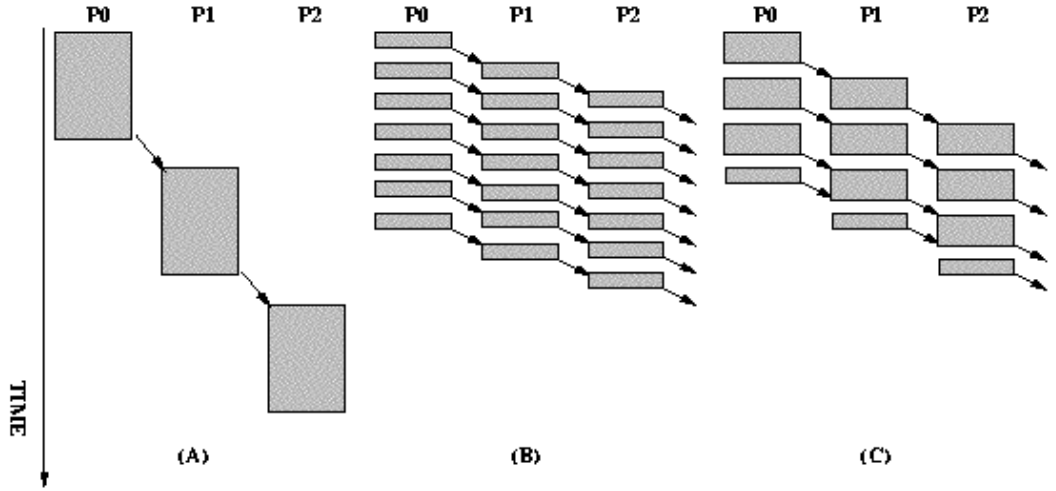
RECOUVREMENTS ET PIPELINES



Recouvrements et pipelines de calculs

- **Buts**
  - Cacher les communications
  - Briser la séquentialité d'un noyau numérique
- Nombreuses applications dans le calcul numérique et l'imagerie
- **Deux approches combinées**
  - Recouvrements des communications (archi + support d'exécution)
  - Pipelines des calculs (algorithmique)

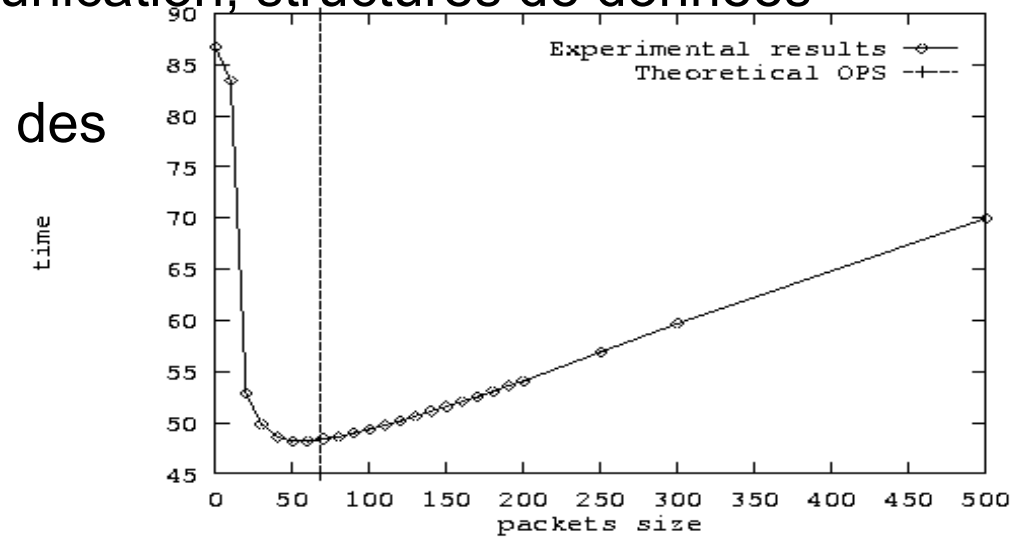




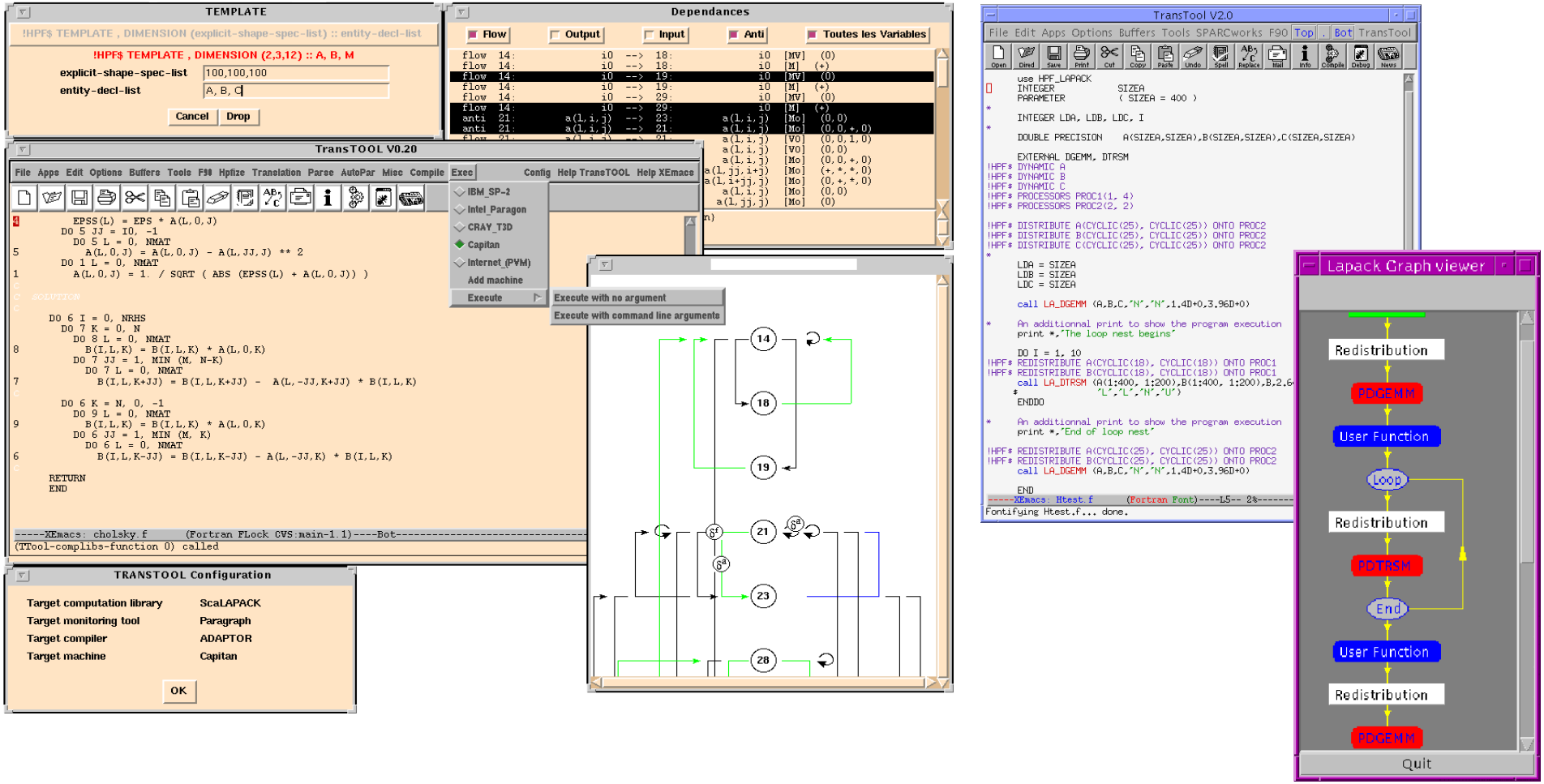
Avec F. Chaussumier, S. Domas, L. Prylli, P. Ramet et J. Roman

Pipelines de calculs

- **Recherche de boucles « pipelinables »**
  - Boucles *cross-processors* par analyse de dépendances
- **Analyse du gain**
  - Modélisation de l'architecture et de l'algorithme
- **Calcul du grain de calcul optimal**
  - Minimisation de la fonction de temps (OPIUM)
- **Génération de code** ou appel à une bibliothèque spécialisée
  - Différents schémas de communication, structures de données régulières ou irrégulières, supports d'exécution, gestion des buffers (LOCCS)
  - Calcul de la taille de paquets pour différentes fonctions
  - Génération de code HPF



PARALLELISATION DE PROGRAMMES FORTRAN

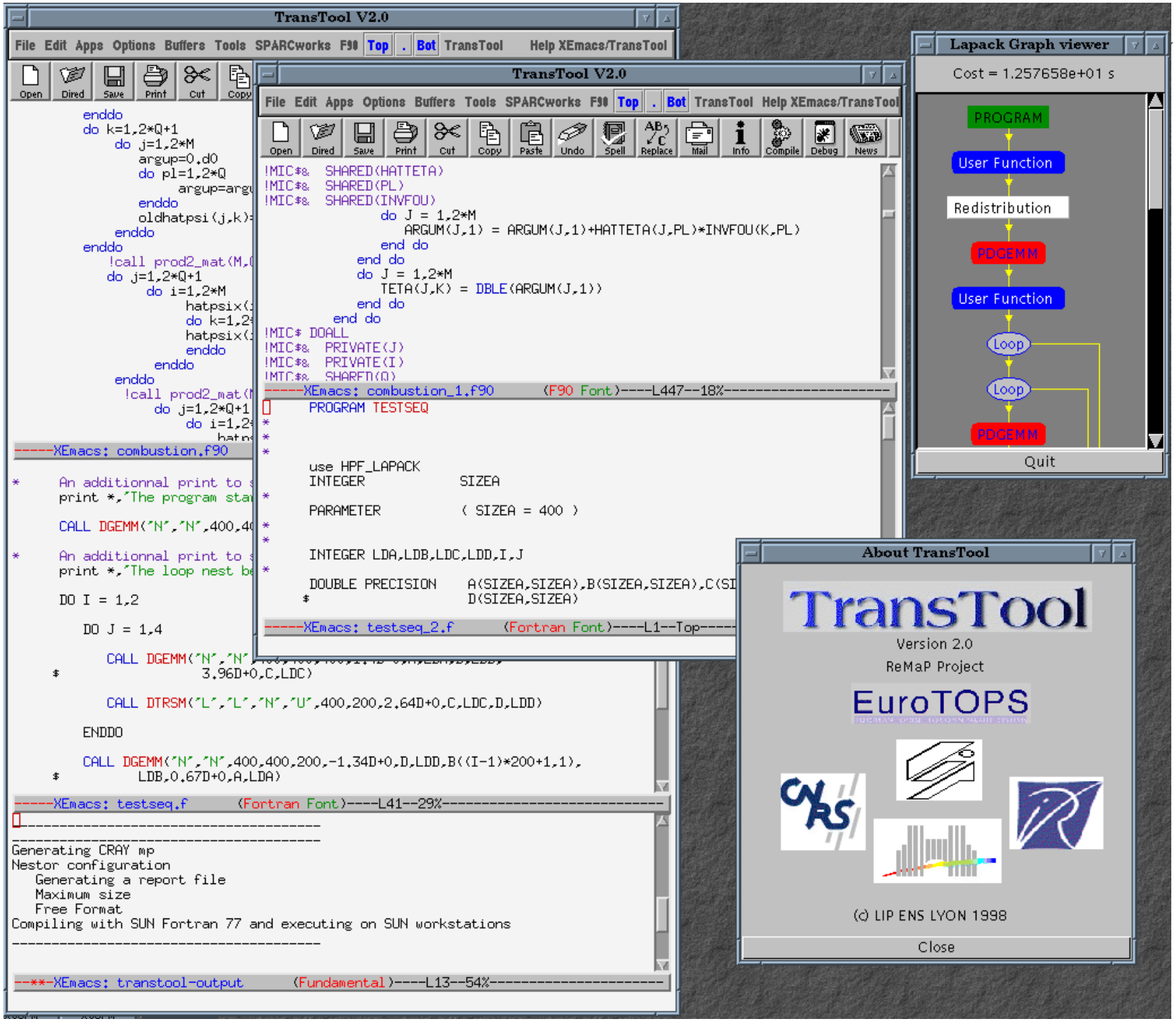


TransTool

- Environnement intégré de parallélisation interactive de programmes Fortran et transformations source à source
- Développé au dessus de XEmacs
- **Plate-forme de présentation de nos développements**
  - Parallélisation et distribution automatique
  - Traduction d'appels de bibliothèques numériques
  - Optimisation de macro-pipelines

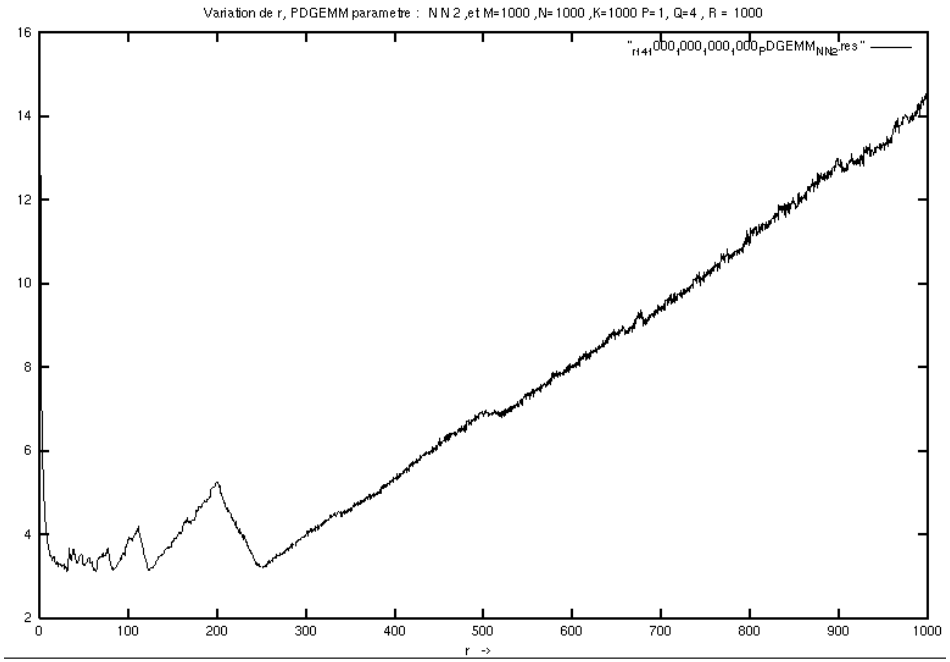
- Génération de code HPF (lié au compilateur cible)
- Plusieurs API pour l'intégration de nouveaux outils (édition, analyse de dépendances, génération d'arbre syntaxique, visualisation de données distribuées)
- Projet EuroTOPS

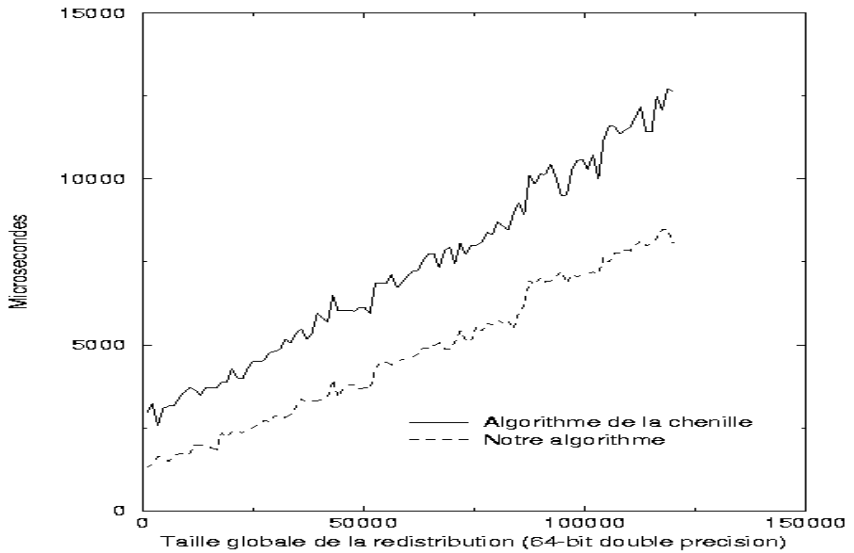
Avec S. Domas, J.-C. Mignot, C. Randriamaro, A. Darte et G.-A. Silber


TransTool V2.0
File Edit Apps Options Buffers Tools SPARCworks F90 Top . Bot TransTool Help XEmacs/TransTool
Open Dired Save Print Cut Copy
TransTool V2.0
File Edit Apps Options Buffers Tools SPARCworks F90 Top . Bot TransTool Help XEmacs/TransTool
Open Dired Save Print Cut Copy Paste Undo Spell Replace Mail Info Compile Debug News
!MIC$& SHARED(HATTETA)
!MIC$& SHARED(PL)
!MIC$& SHARED(INVFOU)
do J = 1,2*M
ARGUM(J,1) = ARGUM(J,1)+HATTETA(J,PL)*INVFOU(K,PL)
end do
end do
do J = 1,2*M
TETA(J,K) = DBLE(ARGUM(J,1))
end do
end do
!MIC$ DOALL
!MIC$& PRIVATE(J)
!MIC$& PRIVATE(I)
-----XEmacs: combustion_1.f90 (F90 Font)----L447--18%----------
PROGRAM TESTSEQ
use HPF_LAPACK
INTEGER SIZEA
PARAMETER (SIZEA = 400)
INTEGER LDA,LDB,LDC,LDD,I,J
DOUBLE PRECISION A(SIZEA,SIZEA),B(SIZEA,SIZEA),C(SI
$ D(SIZEA,SIZEA)
-----XEmacs: testseq_2.f (Fortran Font)----L1--Top-----
-----XEmacs: combustion.f90
An additionnal print to
print *,'The program sta
CALL DGEMM('N','N',400,40
An additionnal print to
print *,'The loop nest be
DO I = 1,2
DO J = 1,4
CALL DGEMM('N','N'
$ 3.96D+0,C,LDC)
CALL DTRSM('L','L','N','U',400,200,2.64D+0,C,LDC,D,LDD)
ENDDO
CALL DGEMM('N','N',400,400,200,-1.34D+0,D,LDD,B((I-1)*200+1,1),
$ LDB,0.67D+0,A,LDA)
-----XEmacs: testseq.f (Fortran Font)----L41--29%----------
Generating CRAY mp
Nestor configuration
Generating a report file
Maximum size
Free Format
Compiling with SUN Fortran 77 and executing on SUN workstations
--**-XEmacs: transtool-output (Fundamental)----L13--54%----------
Lapack Graph viewer
Cost = 1.257658e+01 s
PROGRAM
User Function
Redistribution
PDGEMM
User Function
Loop
Loop
PDGEMM
Quit
About TransTool
TransTool
Version 2.0
ReMaP Project
EuroTOPS
(c) LIP ENS LYON 1998
Close
RIA
RHÔNE-ALPES

ALASca

- **Optimisation des redistributions** pour des distributions cycliques par blocs (de PxQ avec taille de bloc r vers P'xQ' avec r')
  - résultats optimaux en nombre et en volume de messages
- **Traduction d'un code Fortran 77** contenant des appels BLAS et LAPACK vers
  - Un code F77 avec des appels à ScaLAPACK
  - Un code HPF avec des appels à l'interface HPF de ScaLAPACK
  - Trouver les meilleures (re)distributions

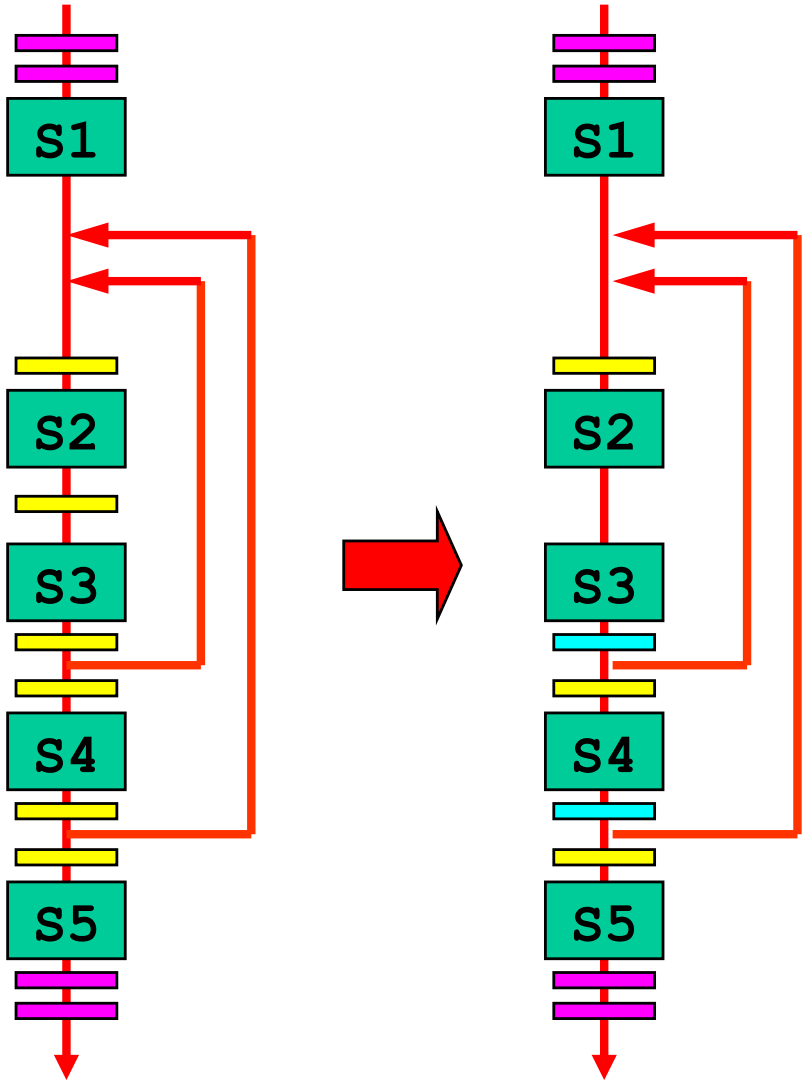




Avec S. Domas, J. Dongarra,J.-C. Mignot, A. Petitet, C. Randriamaro, Y. Robert

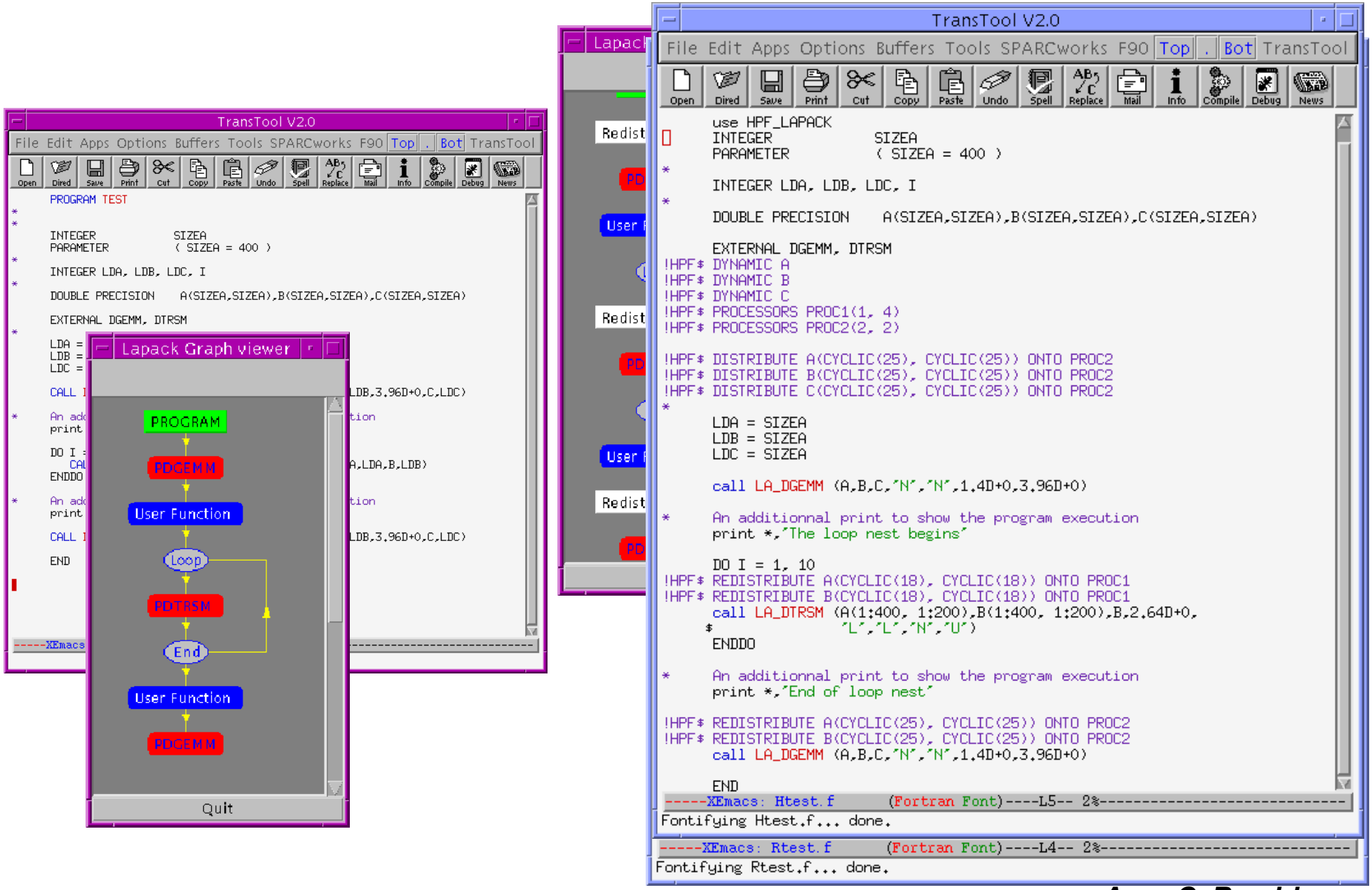
Exemple

```
begin
   Initialize A
   Initialize B
   ...
   DGEMM(A, B, C)          S1
   do i=1, m
     do j=1, n
       DGEMM(C, C, D)      S2
       DGEMM(A, B, C)      S3
     enddo
     (E,F) = LU(D)         S4
     ...
   enddo
   DSYMM(F,A,C)            S5
   …
   Gather matrices C, E, F
end
```



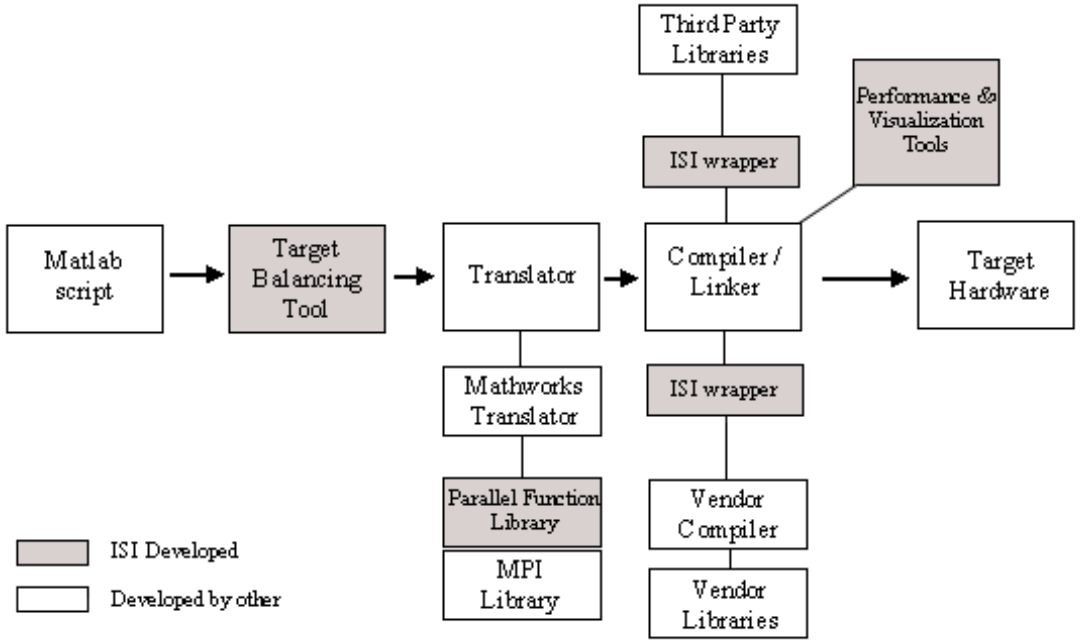
Avec C. Randriamaro

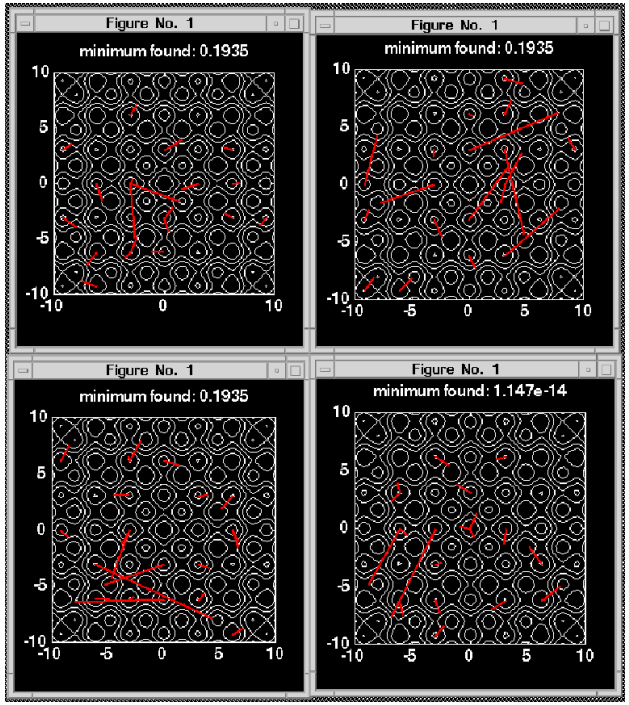
Integration d'ALASca



Avec C. Randriamaro

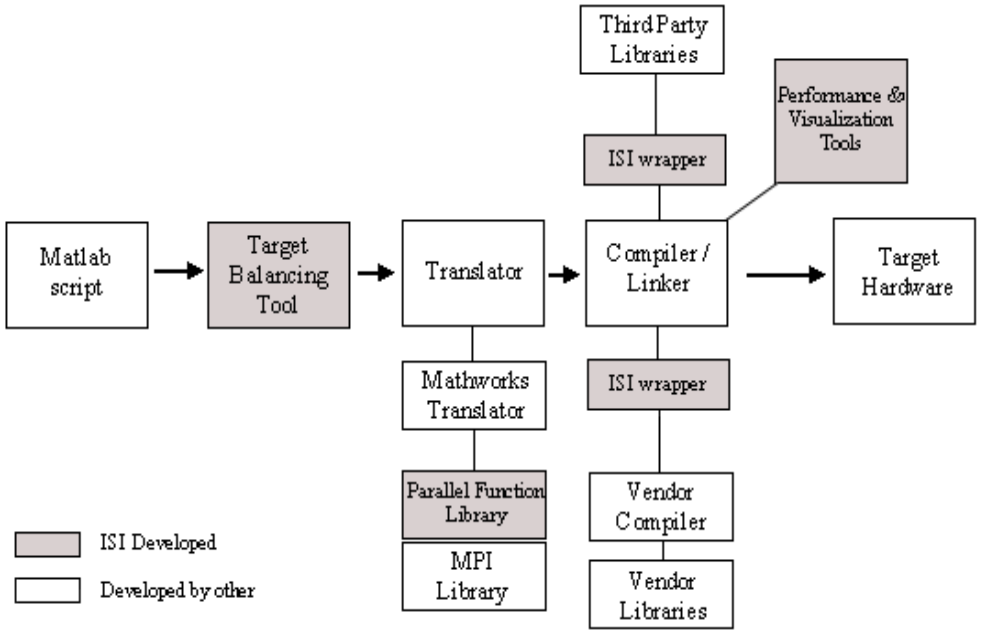
PARALLELISATION D'OUTILS de type MATLAB





Approche compilation

- **Entrée:** un script Matlab
- On ajoute des directives de compilation
- On le compile vers un langage de type Fortran ou C (en s'occupant des problèmes de type et de tailles de variables)
- On ajoute des appels à des bibliothèques séquentielles ou parallèles
- **Buts:**
  - Eviter l'interprétation
  - Plus de problèmes de types
  - Appels à des bibliothèques à hautes performances
  - Utiliser la technologie des compilateurs paralléliseurs

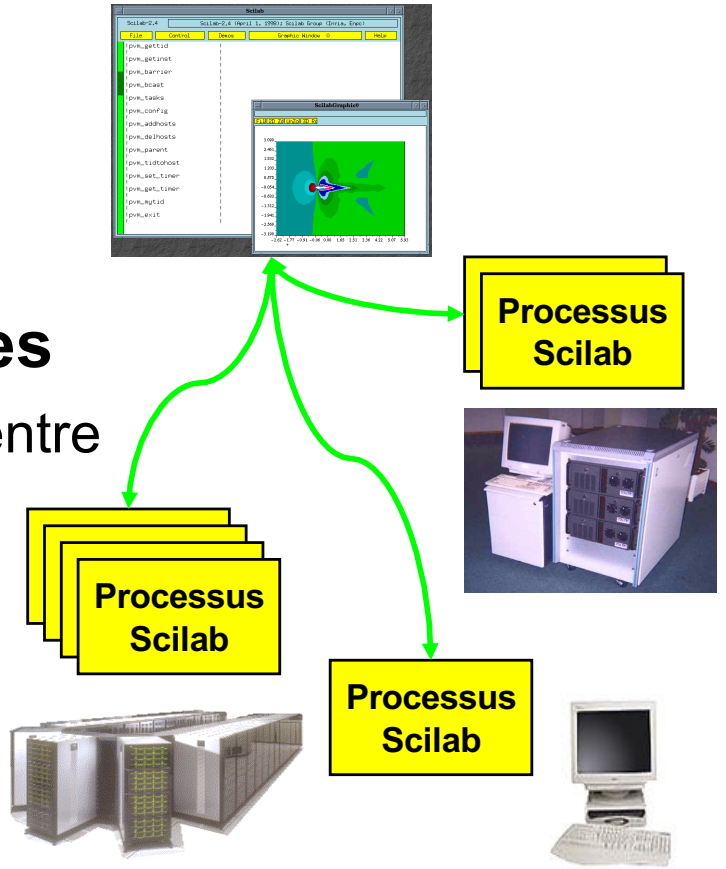


Falcon, Menhir, Match, Paradigm, Conlab, ...

Approche maître-esclave

Conserver l'interactivité de Matlab

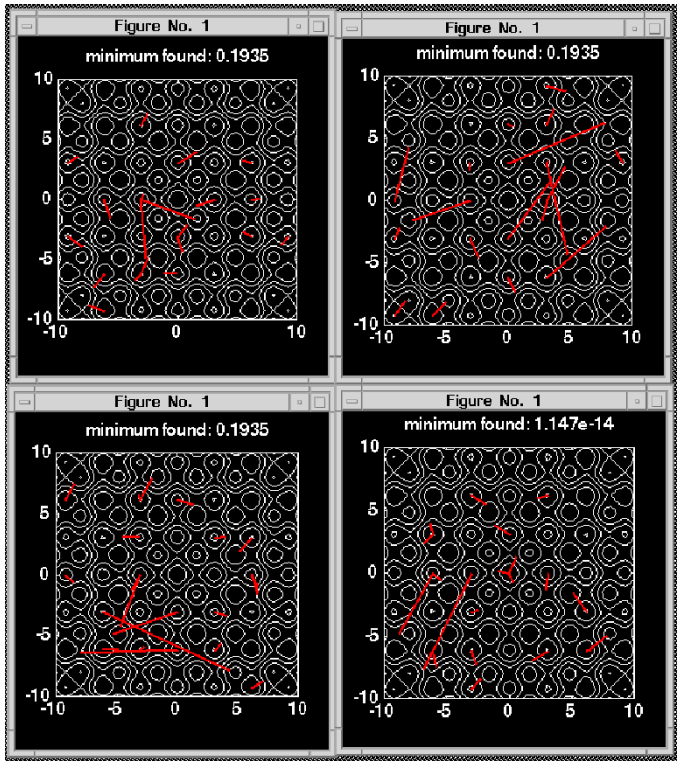
- **Dupliquer des processus** Matlab sur chaque processeur et échanger des messages entre eux (en SPMD ou en maître-esclave)
  - ☺ Facile à développer
  - ☹ Surcoût d'interprétation
- Utiliser des **serveurs de bibliothèques**
  - Besoin d'un protocole de communication entre le client (la fenêtre Matlab) et les serveurs
  - ☺ Plus simple à modifier et à améliorer
  - ☺ Indépendant de Matlab



MultiMatlab, PPServer, MatPar, PSI, Netsolve

Projets qui laissent Matlab interactif

- **MultiMatlab** (Cornell): Duplication + PVM, MPI, BLACS
- **PPServer** (MIT): Serveurs
  - ☺ Interfaces ScaLAPACK, PLAPACK et PETSc
  - 😐 Pas de transparence complète
  - ☹ Environnement homogène
  - ☹ Serveurs non-partagés
- **Matpar** (JPL)
  - Serveur ScaLAPACK
- **PSI** (U. Austin)
  - Interface PLAPACK avec serveur
- **NETSOLVE**



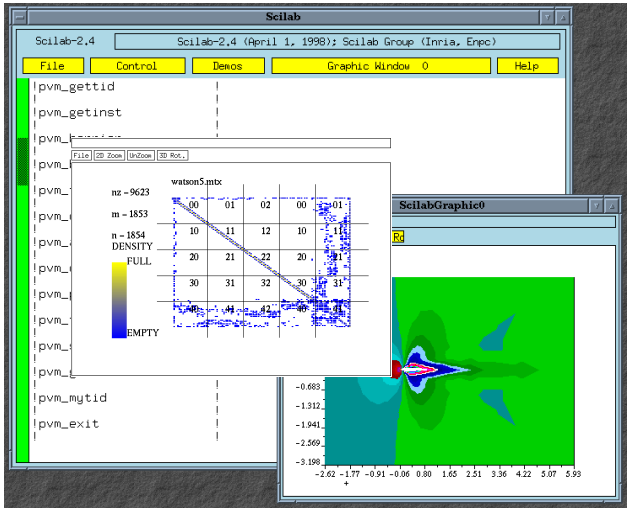
Plusieurs approches

Passage de messages entre tâches Scilab (PVM, MPI),

Interface Netsolve (outil de Metacomputing, UT et ORNL),

Interfaces de bibliothèques parallèles, (ScaLAPACK),

Serveurs de bibliothèques parallèles, (ScaLAPACK, PETSc, PaStiX),



Parallélisme out-of-core (LARIA, Amiens)

Approche CORBA (LIB, Besançon)

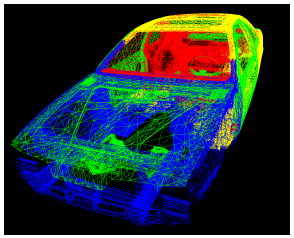
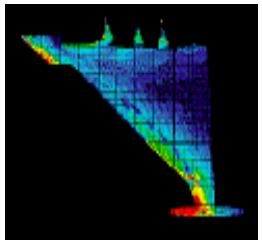
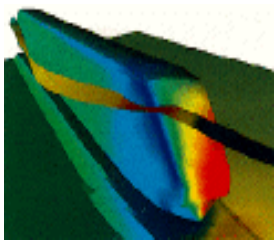
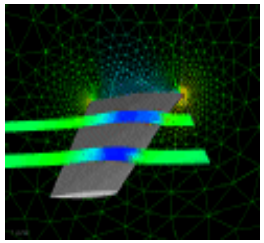
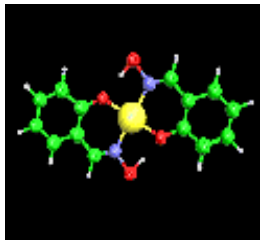
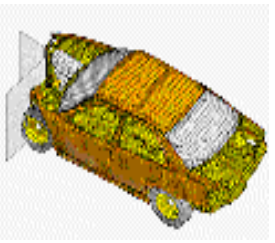
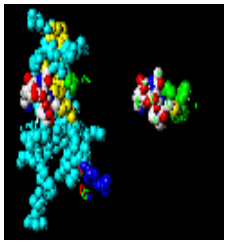
Gestion de parallélisme mixte (tâche/data-parallélisme)

Exécution asynchrone de routines et persistance des données

Approche mixte interactivité/compilation

Approche multi-threads (ordonnancement + migration)

Les différentes couches

SCILAB//

PM-2 High-Perf

Serveurs de calcul / Netsolve

Bibliothèques denses

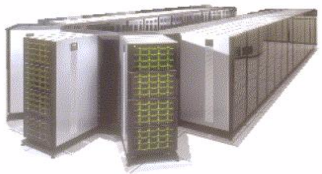
Bibliothèques creuses

BIP-MPI / *MADELEINE* / *PACX-MPI* / PVM / coms Netsolve / CORBA

Outils d 'évaluation de performances NWS, ...








Versions actuelles

Passage de messages

- PVM and MPI
- ☹ Assembleur du parallélisme !

Interface Netsolve

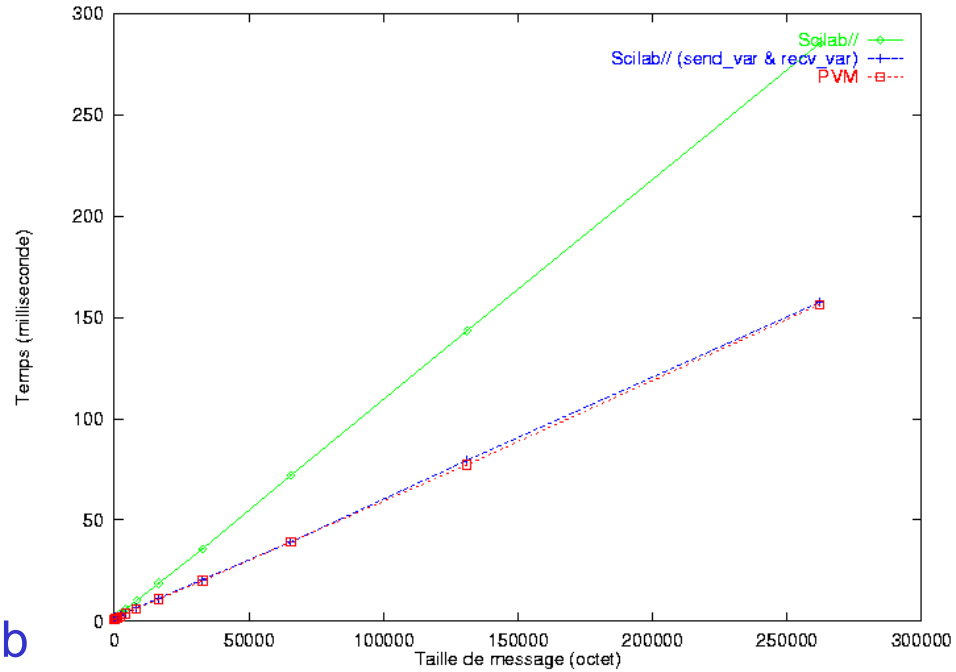
- ☺ Facile à développer
- ☹ Netsolve limité

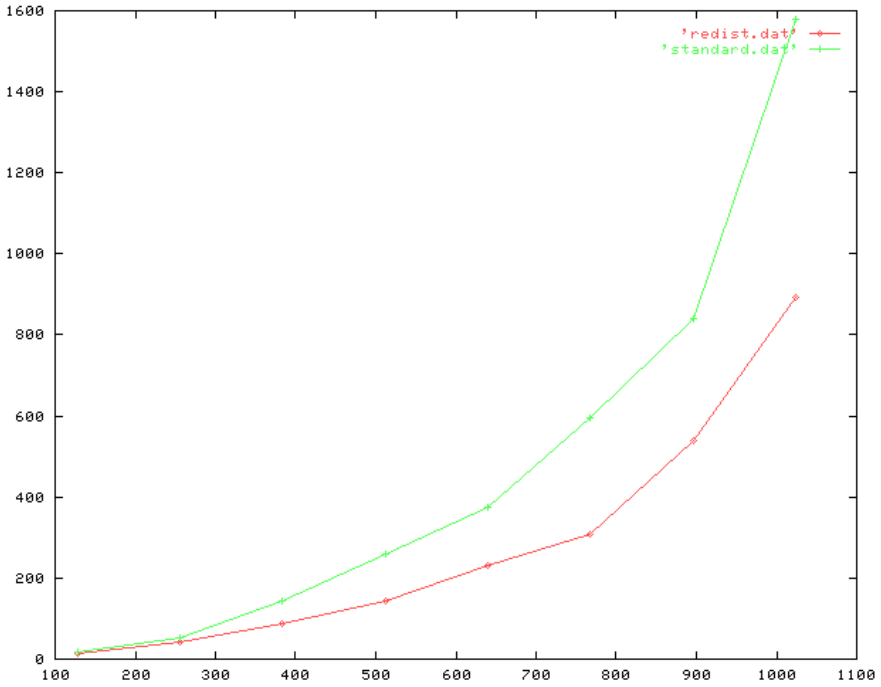
Interface ScaLAPACK + duplication Scilab

- In-core (ScaLAPACK) et out-of-core
- ☺ Relativement facile à développer
- ☹ Duplication de Scilab
- ☹ Types à rajouter (matrice distribuées)

Serveurs de calculs

- ☺ Style client-serveur
- ☺ Plus souple
- ☺ Choix des couches de liaisons
- ☺ Utilisable avec d 'autres interfaces



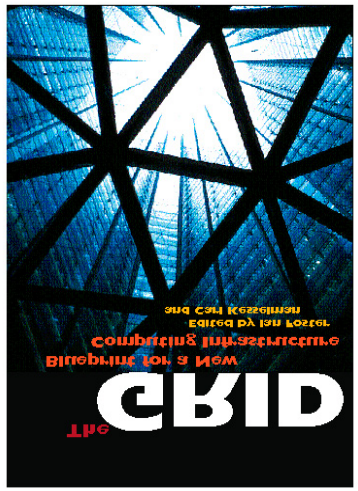


Metacomputing

- De la recherche de la vie extraterrestre aux applications numériques les plus classiques
- De PACX-MPI à Globus
- Des grappes de grappes à l'Internet dans son ensemble

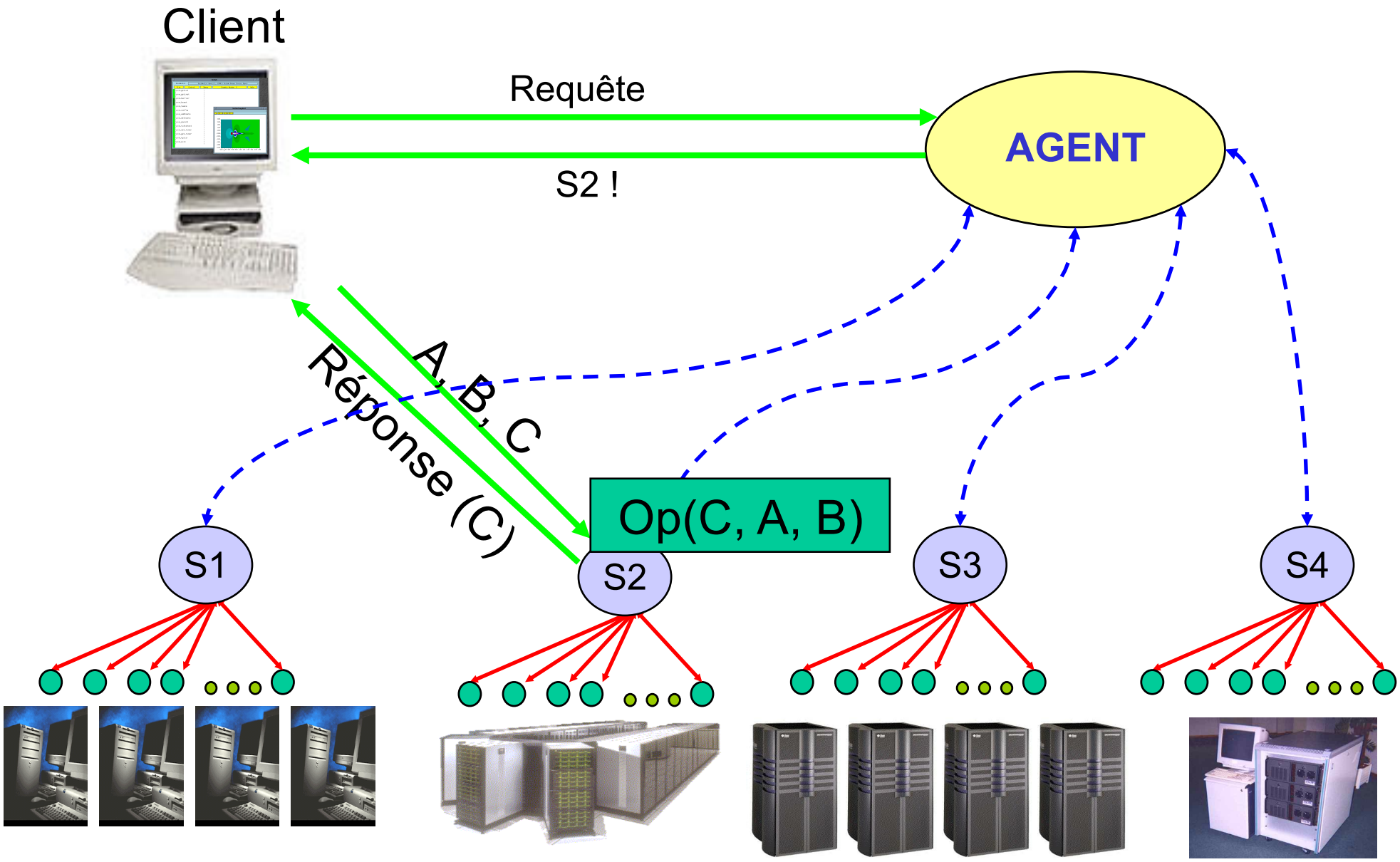
- **Idées:**
  - Utiliser les ressources inutilisées
  - Mettre en commun (et agréger) des ressources de calcul
  - Offrir à l'utilisateur de la puissance de calcul et de la capacité de stockage de manière (relatively) transparente

- **Nombreux problèmes** liés à l'hétérogénéité et à la mise en commun de ressources dispersées
- **Mais** focalisation des travaux sur les couches logicielles et « utilisation » (un peu trop) massive de Globus



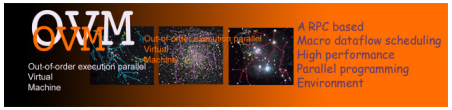
« Mon équipe a développé une solution très innovante, mais nous sommes encore en train de chercher un problème qui va avec. »

RPC Big picture V1



RPC et *grid-computing :* GridRPC

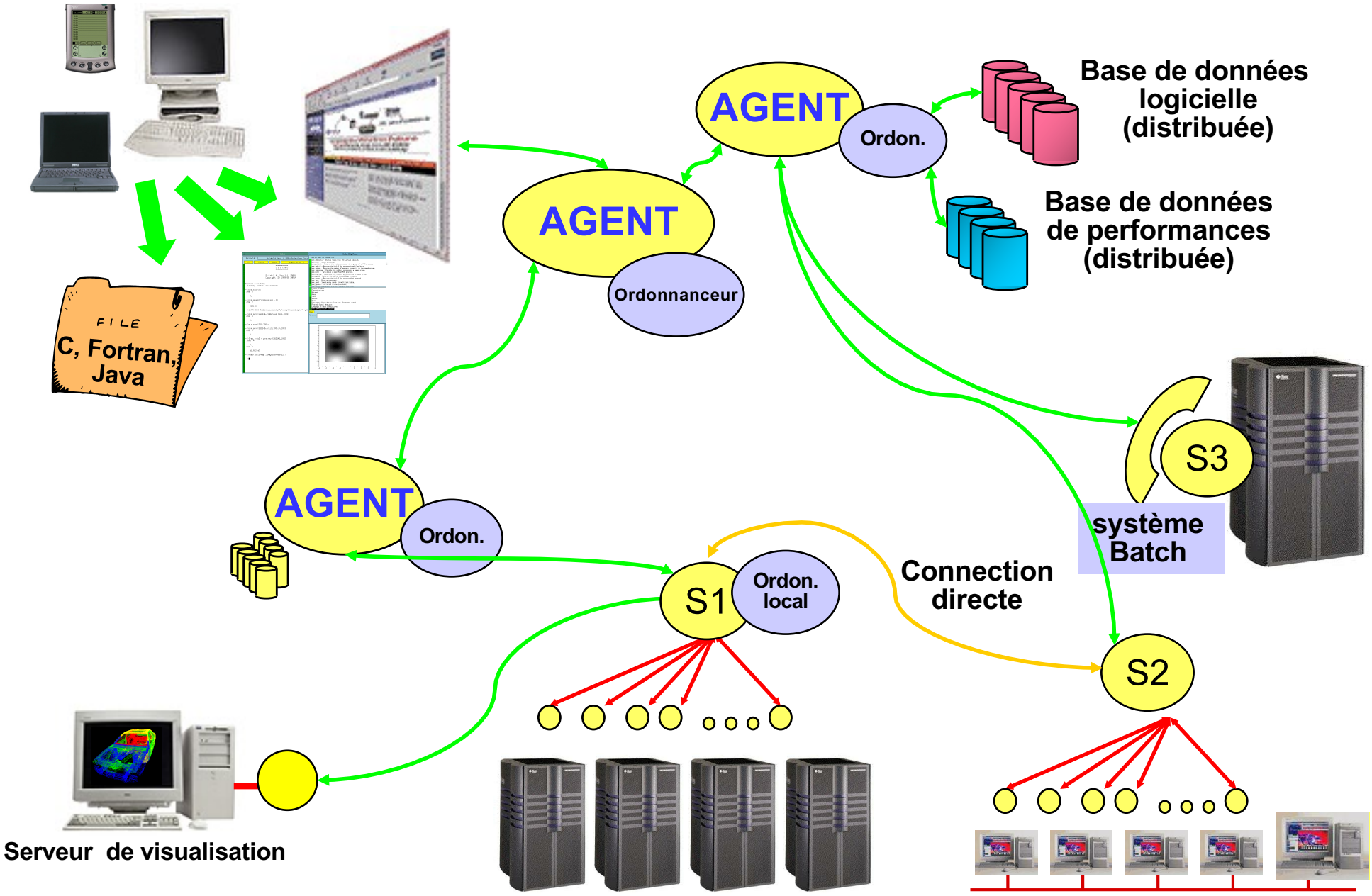
- **Une idée simple**
  - Implémenter le modèle de programmation RPC sur la grille
- **Fonctionnalités nécessaires**
  - Équilibrage des charges (localisation de ressources et évaluation de performances, ordonnancement), sécurité, tolérance aux pannes, IDL, distribution et migration de données, interopérabilité avec d'autres systèmes, …
- **Outils existants**
  - **Ninf** (Electrotechnical Lab, Umezono, Japon)
    - Calcul numérique
  - **Netsolve** (University of Tennessee, USA)
    - Calcul numérique
  - **NEOS** (Argonne National Lab., USA)
    - Problèmes d'optimisation
  - **RCS** (ETH Zürich)
    - Serveurs ScaLAPACK
  - **NIMROD-G** (Monash University in Melbourne, Australie)
    - Au dessus de Globus
  - **OVM** (LRI)
    - Orienté grappes



DIET

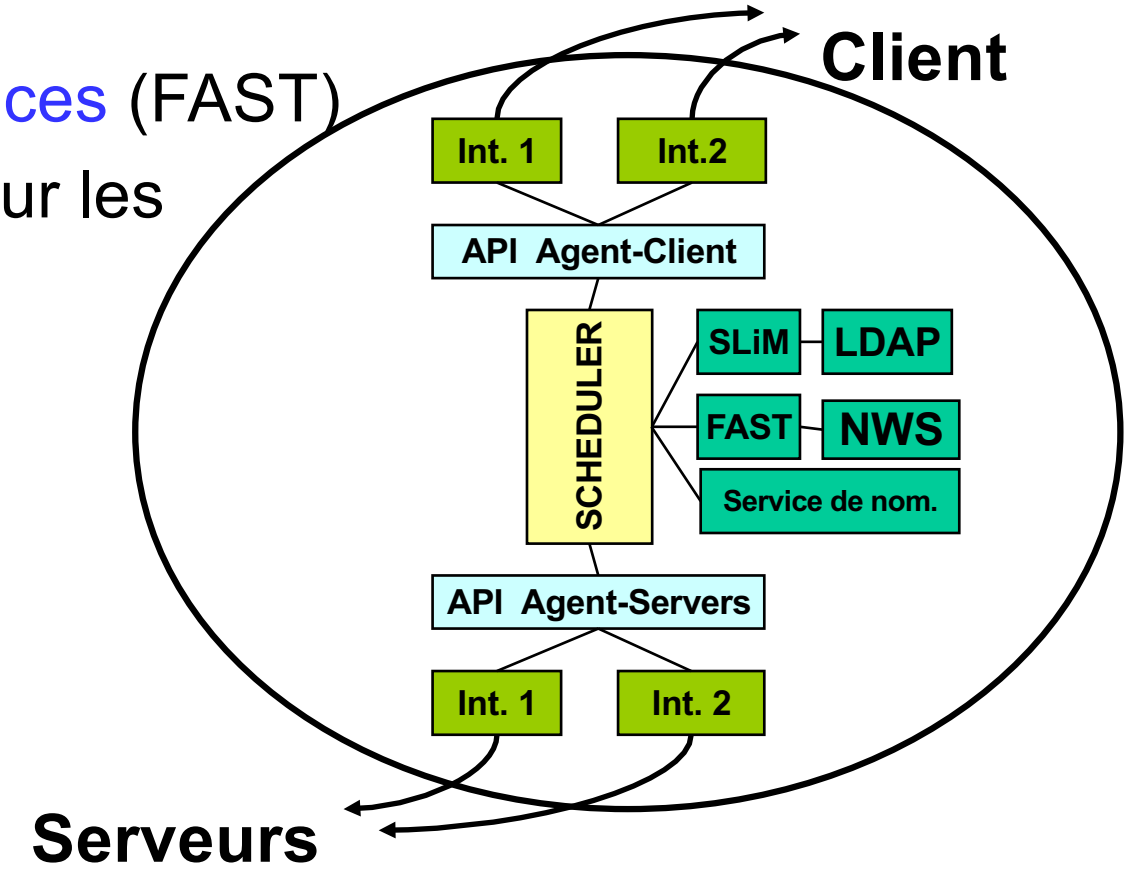
- *Distributed Interactive Engineering Toolbox*
- Bâtir une infrastructure logicielle générique pour un ensemble portails de supercalcul et de *Problem Solving Environments*
- En utilisant des couches logicielles portables (Java, JINI, CORBA, bibliothèques à hautes performances, ...)
- **Plusieurs composants**
  - Ordonnanceur à hautes performances (heuristiques dépendantes des applications)
  - Localisation de ressources
  - Evaluation et prédiction de performances (basé sur NWS)
- VTHD comme plate-forme cible
- RNTL GASP et ACI GRID GRID-ASP

Big picture améliorée



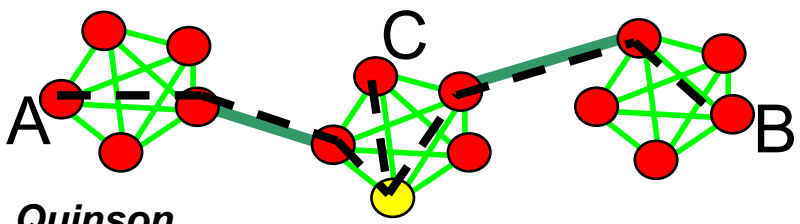
Améliorer l'agent(s)

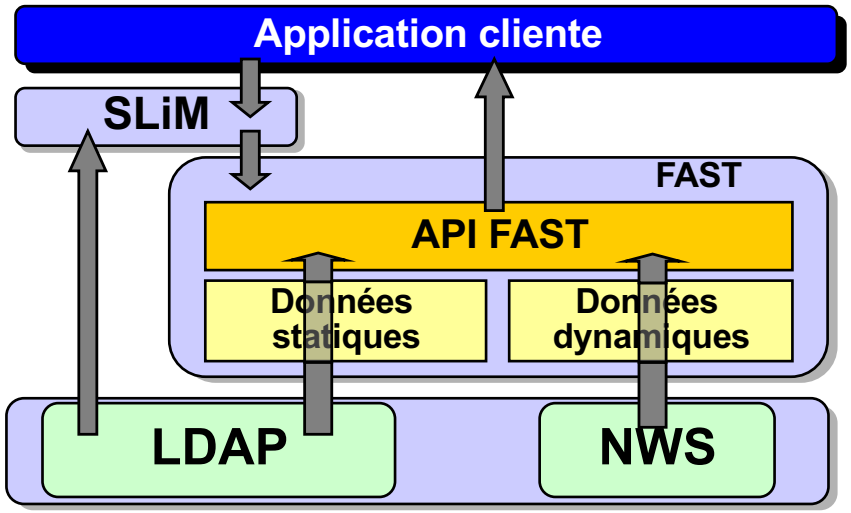
- Ordonnanceur
  - Heuristiques générales ou dépendantes de l'application
- Localisation de ressources logicielles (SLiM)
  - Basé sur LDAP
- Evaluation de performances (FAST)
- Service de nommage pour les données génériques
- Différentes API pour différentes interfaces
- Structure hiérarchique pour la scalabilité



FAST: Fast Agent's System Timer

- Evaluation des performances de la plate-forme pour être capable de trouver un serveur efficace (coûts de redistribution et de calcul) sans tester toutes les configurations ➔ base de données de performances pour l'ordonnanceur
- Basé sur NWS (*Network Weather Service*)
- **Performances de calcul**
  - Charge de la machine, capacité mémoire et performances des queues de batch (données dynamique)
  - Tests des bibliothèques pour différentes données (statique)
- **Performances de communication**
  - Pour être capable de deviner le coût de la redistribution de données entre deux serveurs en fonction de l'architecture réseau et des informations dynamiques
  - Latence et bande-passante (hiérarchique)
- **Ensemble hiérarchique d'agents**
  - Problèmes de scalabilité

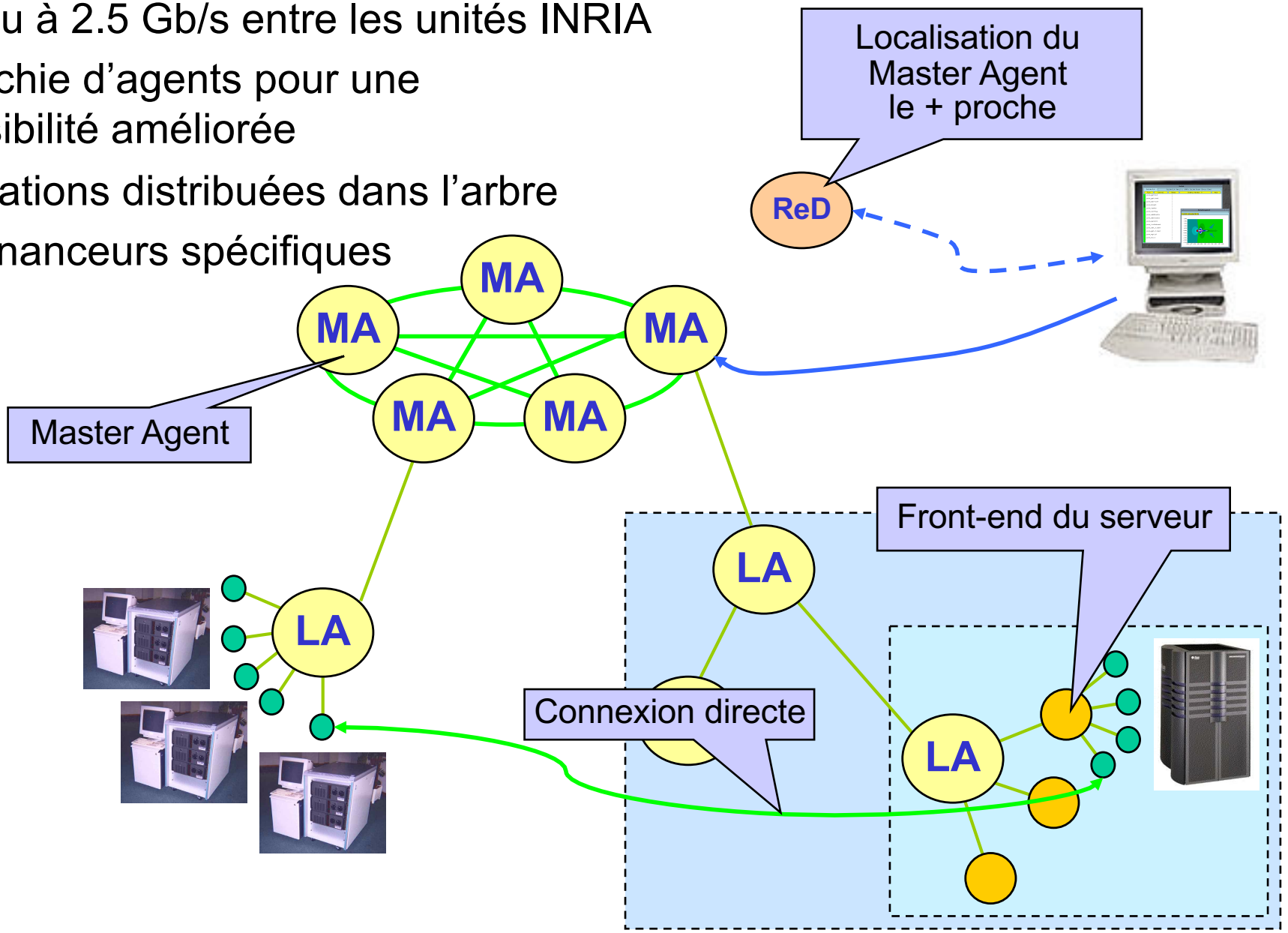




Avec M. Quinson

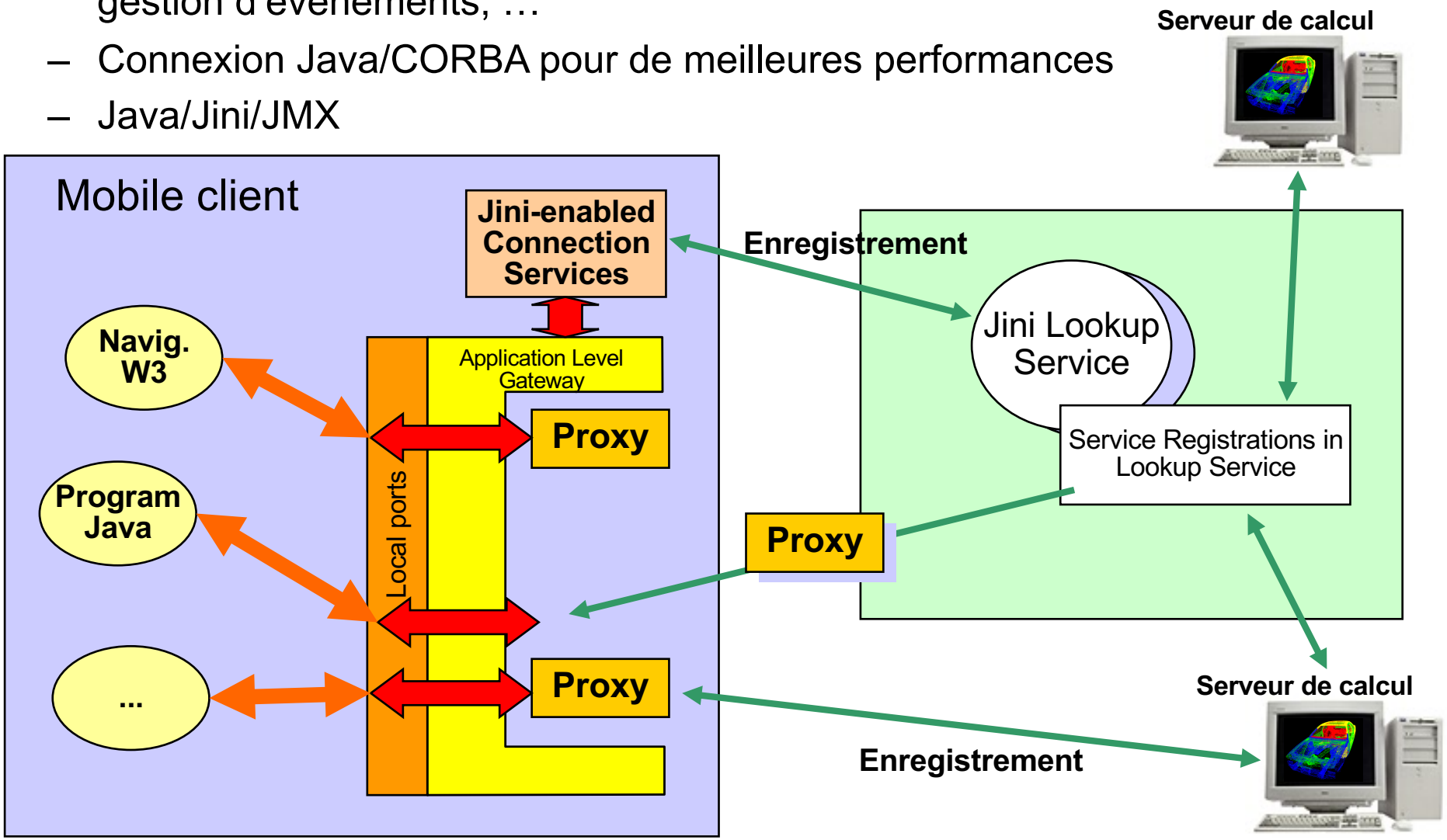
Architecture hiérarchique pour VTHD

- Réseau à 2.5 Gb/s entre les unités INRIA
- Hiérarchie d'agents pour une extensibilité améliorée
- Informations distribuées dans l'arbre
- Ordonnanceurs spécifiques



Et après ?

- **Technologie Java pour le *portal supercomputing***
  - Utilisation de Jini pour l'enregistrement de serveurs, lookup service, gestion d'événements, …
  - Connexion Java/CORBA pour de meilleures performances
  - Java/Jini/JMX

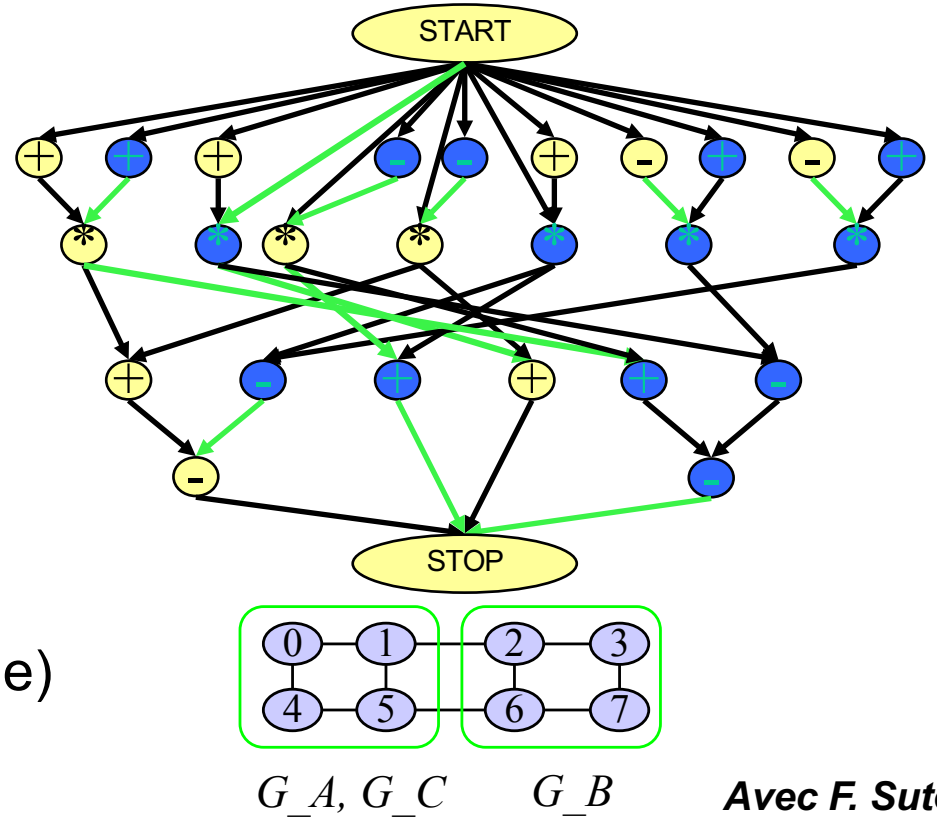


Problèmes liés

- **Sécurité**
  - Comptes utilisateurs et accès
  - Transferts de données
- **Tolérance aux pannes**
  - Serveurs ou agents
- **Intéropérabilité**
  - Description de problèmes
  - **Redistributions de données entre les serveurs**
- **Check-pointing**
  - Entrées/sorties parallèles rapides
- **Scalabilité**
  - **Hiérarchie de serveurs/agents**
- **Recherche de ressources**
  - Matériel et logiciel
- **Ordonnancement**
  - **Ordonnancement on-line d'ordonnancements off-line**
- **Partage de serveurs entre utilisateurs**
  - Problèmes de sécurité
  - Lock/unlock, consistence des données, race conditions
- **Evaluation de performances**
  - **Hétérogénéité**
  - **Systèmes batch**
- **Visualisation de données**
  - Problèmes d'extensibilité
- **…**

Parallélisme mixte

- Exploitation simultanée du parallélisme de tâches et du parallélisme de données
- **Entrée**:
  - Un graphe de tâches
  - Un ensemble fini de processeurs connectés par un réseau
  - Des données (potentiellement) déjà placées sur des sous-grilles
- **Sortie**:
  - Un ordonnancement des calculs
  - Un placement des données
- Première application sur Strassen et Winograd
- **Perspectives:**
  - Automatisation (heuristiques)
  - Intégration dans Scilab//
  - Approche mixte (compil., run-time)



Avec F. Suter

CONCLUSION ET PERSPECTIVES

« Quelques » défis logiciels et autres problèmes difficiles

INRIA RHÔNE-ALPES

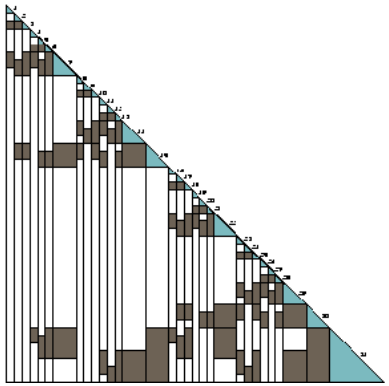
- **Les applications « classiques » du parallélisme supposaient**
  - Une architecture homogène de processeurs (processeurs, réseaux)
  - Des ressources statiques
  - Encore de nombreux problèmes ouverts !
- **Sur la grille**
  - Architecture totalement hétérogène (de collections d'éléments homogènes à l'Internet), ressources connues à l'exécution
  - Comportement dynamique (ajout et disparition de ressources)
  - Encore plus de problèmes ouverts
- **Trop de projets actuels oublient**
  - L'algorithmique
  - La modélisation des algorithmes

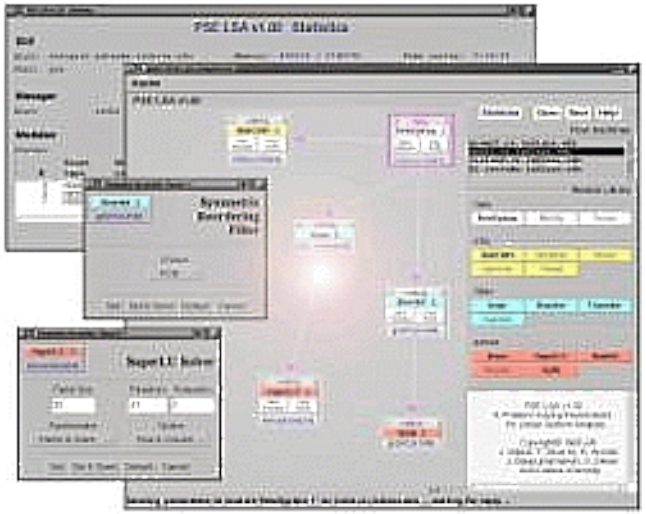
« Quelques » défis logiciels et autres problèmes difficiles, suite

- **Hétérogénéité**
  - Réseaux (couches de communications multi-protocoles), processeurs, hiérarchies mémoire profondes
  - Equilibrage des charges (ordonnancement)
  - Distribution des données (statique et dynamique)
  - Informations sur la charge en temps réel
  - Evaluation des performances
  - Modélisation des architectures (déjà des modèles pour les grappes de SMP, cf P. Fraigniaud (HiHCoHP))
  - Couplage de codes
  - Interopérabilité de logiciels
  - Bases de données de codes (GAMS, Netlib)
- **Administration**
  - Sécurité (transferts, accès, firewalls)
  - Equité, disponibilité, équilibrage des charges

Quelques défis logiciels et autres problèmes difficiles, suite

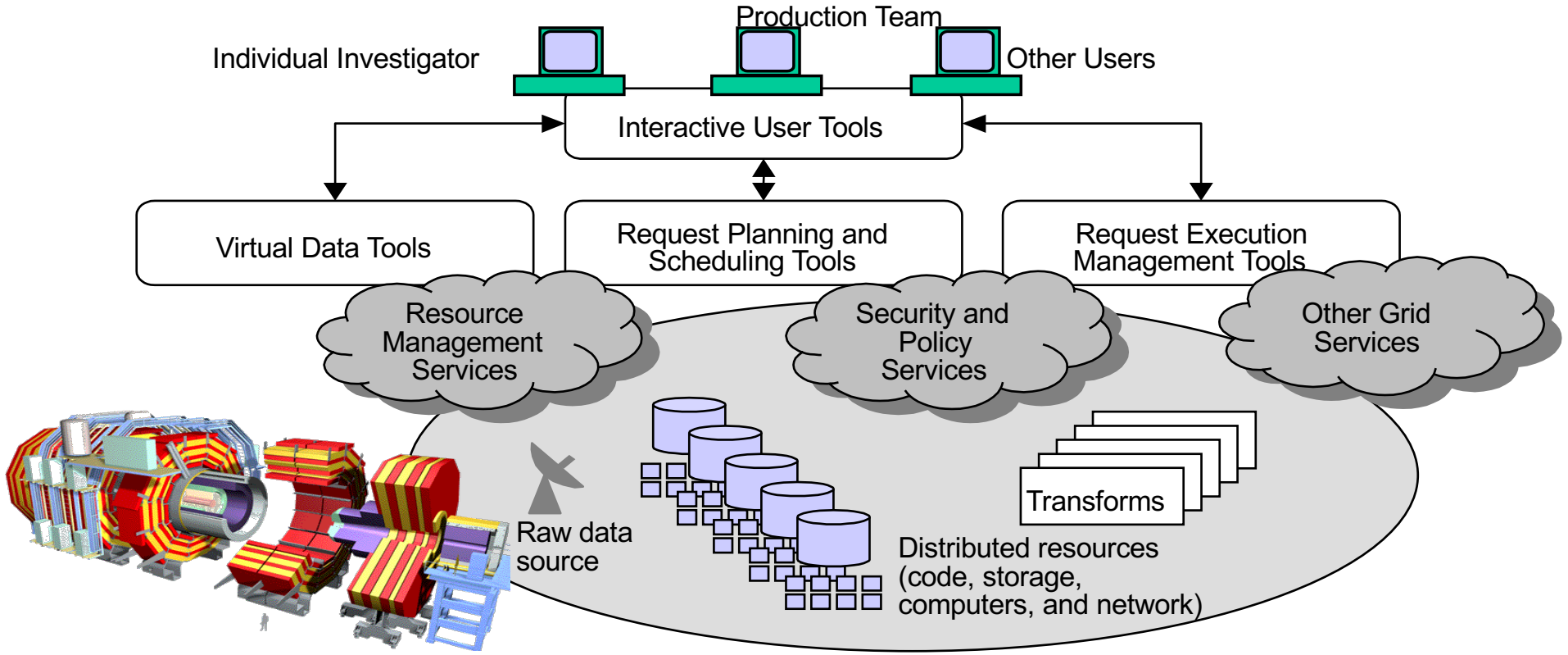
- **Algorithmique numérique et bibliothèques**
  - Algorithmique pour le creux
  - Interopérabilité entre les logiciels (redistributions de données, structures de données)
  - Méthodes hybrides (directes, itératives)
  - Liaisons plus importantes avec les supports d'exécution à hautes performances
  - Auto-adaptabilité des logiciels à tous les niveaux
    (voir ATLAS)
  - Analyse de problèmes et recherche de solveurs
  - Abstraction (programmation par composants) !





Quelques défis logiciels et autres problèmes difficiles, suite

- ***Problem Solving Environments***
  - Plus de transparence, généricité, sécurité
  - Unification des couches logicielles (Grid Forum)
  - Couplage algorithmique et modèles avec les supports d'exécution et les architectures
  - Nombreux projets applicatifs autour de la grille (Datagrid, GridPhyn)



Conclusions et perspectives

- De l'algorithmique parallèle homogène et des aspects bas niveaux des couches de communications ….
- … aux environnements de parallélisation et aux *Problem Solving Environments*

- Orientation vers les aspects middleware pour les PSE …
- … tout en regardant l'algorithmique liée au *Grid Computing* (notamment autour du creux)
- Etudier d'autres applications (MNT) et d'autres domaines (travail collaboratif, télé-enseignement, …)

- RNRT VTHD++, RNTL GASP et ACI GRID-ASP

Remerciements

- **Les étudiants en DEA et en thèse**
  - Pierre Ramet
  - Stéphane Domas
  - Julien Zory
  - Frédérique Chaussumier
  - Fabrice Rastello
  - Cyrille Randriamaro
  - Frédéric Suter
  - Martin Quinson, …
- **Les ingénieurs et postdocs**
  - Jean-Christophe Mignot
  - Eddy Caron
- **Les stagiaires de tous horizons**
  - Nicolas Bert, Lionel Tricon, …