

Environnements de résolution de problèmes sur la grille

F. Desprez

Projet ReMaP
INRIA Rhône-Alpes
LIP ENS Lyon
France



RenPar – Avril 2002 – F. Desprez

Remerciements



- Jack Dongarra (University of Tennessee, ICL lab.) et l'équipe Netsolve en général
- Henri Casanova (UCSD)
- Hidemoto Nakada (Grid Center-AIST, GSIC Center-TITECH) et l'équipe Ninf en général
- L'équipe DIET
 - ReMaP (LIP ENS Lyon) : E. Caron, P. Combes, M. Quinson, F. Suter
 - LIFC (Besançon) : S. Contassot, F. Lombard, J.-M. Nicod, L. Philippe
 - Résédas (LORIA Nancy) : E. Jeannot
 - ARES (INSA Lyon) : E. Fleury

RenPar – Avril 2002 – F. Desprez

Plan de l'exposé



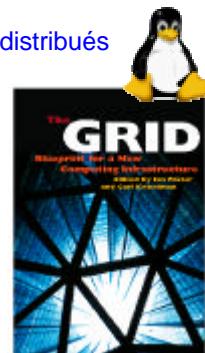
- Introduction sur l'approche GRID-RPC
- Etude de cas de trois environnements et des problématiques reliées
 - NetSolve (University of Tennessee)
 - Ninf (TOKYO)
 - DIET (ENS Lyon, LORIA, INSA, LIFC)
- Conclusion et travaux futurs

RenPar – Avril 2002 – F. Desprez

INTRODUCTION



- Le futur du calcul parallèle : **distribué** et **hétérogène**
 - Metacomputing/Grid Computing = *Utiliser des ensembles distribués de plates-formes hétérogènes*
 - Le *Network Computing* aujourd'hui !
 - Grappes de machines SMP avec des processeurs rapides, des réseaux rapides (et peu coûteux), des logiciels (pratiquement) matures
 - De nombreux projets (Trop?)
 - **Cible:** plusieurs applications dans de nombreux domaines (pas uniquement que des applications numériques ou *embarrassingly parallel*)
 - **Quelques problèmes importants:**
 - algorithmiques (distribution des données, équilibrage des charges, algorithmes tolérants à la latence, ...)
 - systèmes (administration, tolérance aux pannes, sécurité, localisation de ressources, ...)
 - logiciels (interopérabilité, réutilisation de code, ...)
- ➔ Grid forum



RenPar – Avril 2002 – F. Desprez

INTRODUCTION, suite



Une idée à long terme pour le Metacomputing: offrir (ou louer) de la puissance de calcul et de la capacité mémoire à travers l'Internet

☺ **Très grand potentiel**

- Nécessité d'avoir des *Problem Solving Environments* (PSEs) et des ASP (Application Service Provider)
- Les applications ont toujours besoin de plus de puissance de calcul et de capacité mémoire
- Certaines bibliothèques ou codes propriétaires doivent rester sur place
- Des données confidentielles ne doivent pas circuler sur le réseau
- ➔ Utilisation de serveurs de calculs accessibles à travers une interface simple

☹ **Toujours difficiles à utiliser pour les non-spécialistes**

- ☹ Pratiquement pas de transparence
- ☹ Les problèmes de sécurité et d'accounting ne sont généralement pas traités

☹ **Souvent des PSEs dépendants d'une application**

- ☹ **Pas de standards** (CORBA, JAVA/JINI, sockets, ...) pour construire les serveurs de calcul

RenPar – Avril 2002 – F. Desprez

RPC et grid-computing : GridRPC



• **Une idée simple**

- Implémenter le modèle de programmation RPC sur la grille
- Utiliser des ressources de calcul à travers le réseau
- Applications nécessitant de grosses puissances de calcul et/ou de gros stockages de données
- Modèle de programmation à parallélisme de tâches (appels synchrones et asynchrones) + parallélisme de données sur les serveurs, parallélisme mixte

• **Fonctionnalités nécessaires**

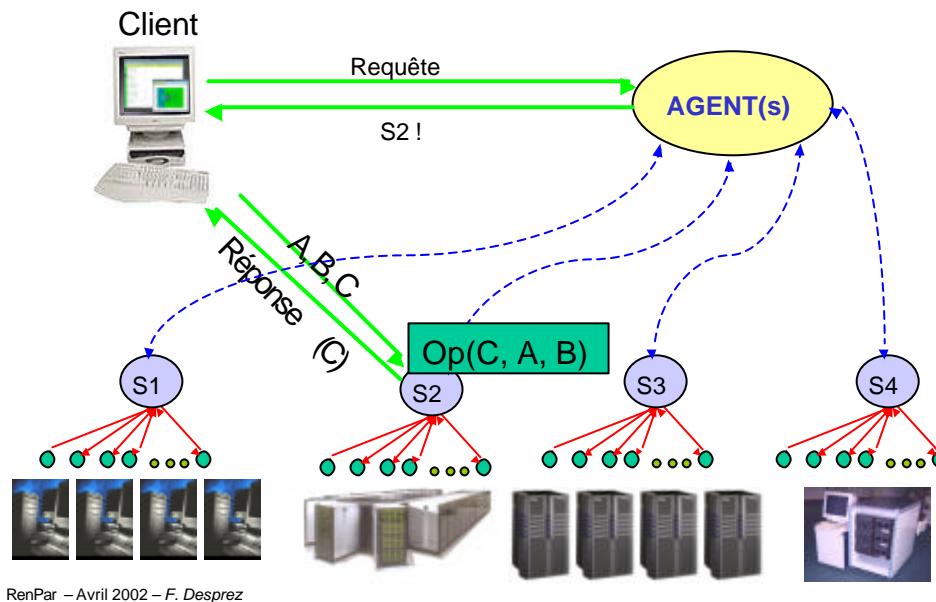
- Équilibrage des charges (localisation de ressources et évaluation de performances, ordonnancement),
- IDL,
- distribution et migration de données,
- sécurité,
- tolérance aux pannes,
- interopérabilité avec d'autres systèmes, ...



RenPar – Avril 2002 – F. Desprez

RPC et grid-computing : GridRPC, suite

INRIA



RenPar – Avril 2002 – F. Desprez

RPC et grid-computing : GridRPC, suite

INRIA

Cinq composants fondamentaux :

- **Client**
Fournit plusieurs interfaces utilisateur et soumet les requêtes aux serveurs
- **Serveur**
Reçoit les requêtes des clients et exécute les modules logiciels pour eux
- **Base-de-données**
Contient des données statiques et dynamiques sur les ressources logicielles et matérielles
- **Ordonnanceur**
Intercepte les requêtes des clients et prend des décisions pour placer les tâches sur les serveurs en fonction des données stockées dans la base-de-données
- **Moniteur**
Observe dynamiquement le statut des ressources de calcul et stocke les informations récupérées dans la base-de-données

RenPar – Avril 2002 – F. Desprez

RPC et grid-computing : GridRPC, suite



- Grain adaptable (<1sec à > une semaine)
- API RPC simple
- Bibliothèques et applications intégrées dans des composants grille
- IDL pour l'interface client, informations minimales

```
double A[n][n],B[n][n],C[n][n];      /* Decl. des données */  
dmmul(n,A,B,C);                      /* Appel fonction locale */  
  
GRPC_call("dmmul",n,A,B,C); /* Appel fonction serveur */
```

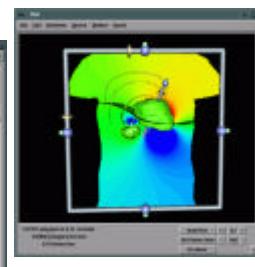
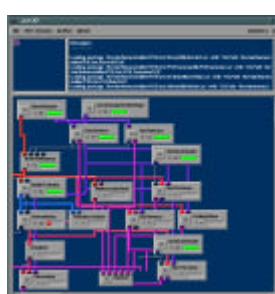
RenPar – Avril 2002 – F. Desprez

RPC et grid-computing : GridRPC, suite



- Middleware entre les portails et les composants GRID
- Outil de base pour le déploiement d'environnements à plus grande échelle (portails Web, *Problem Solving Environments*, Grid Toolkits, ...)
- Succès sur plusieurs applications
- Discussion dans le Working Group APM (*Advanced Programming Models*) du Global Grid Forum

RenPar – Avril 2002 – F. Desprez



SCIRun torso
defibrillator application –
Chris Johnson, U. of Utah

RPC et *grid-computing* : GridRPC, suite



- **Quelques outils existants**

- **NetSolve** (University of Tennessee, USA)
 - Calcul numérique
- **Ninf** (Electrotechnical Lab, Umezono, Japon)
 - Calcul numérique
- **DIET** (ReMaP, ARES, LIFC)
 - Basé sur CORBA, ASP
- **NEOS, Meta-NEOS** (Argonne National Lab., USA)
 - Problèmes d'optimisation combinatoire
- **RCS** (ETH Zürich)
 - Serveurs ScaLAPACK
- **NIMROD, NIMROD-G** (Monash University, Australie)
 - Au dessus de Globus



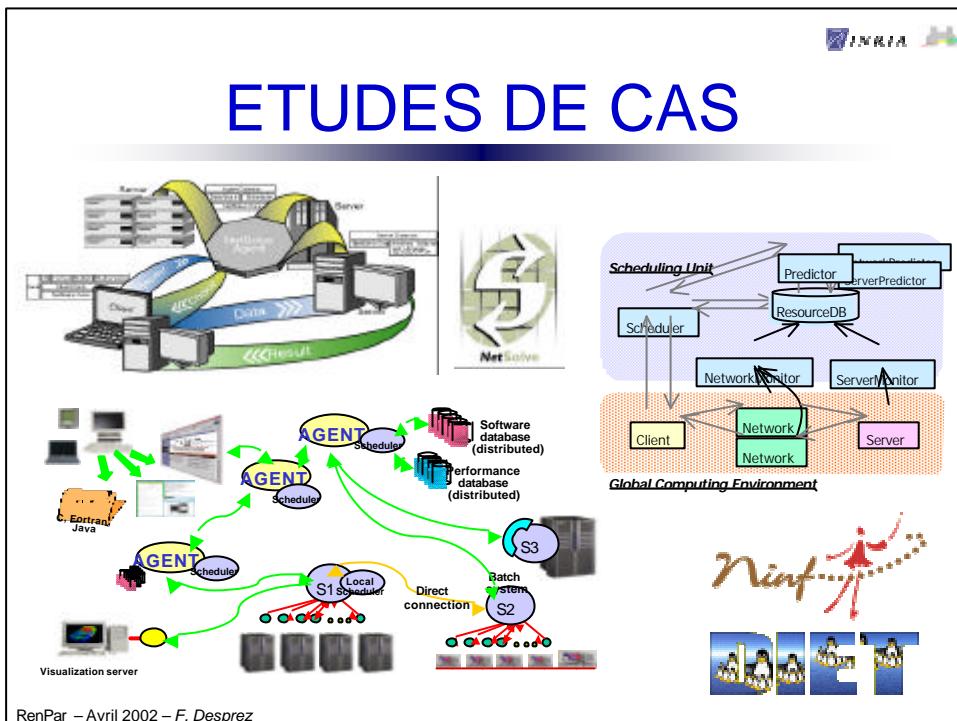
RenPar – Avril 2002 – F. Desprez

Projets proches



- **Problem Solving Environments**
 - Information Power Grid (NASA)
 - Parallel Simulation User Environment (Univ. of Wales, UK)
- **Accès à des serveurs via le Web**
 - Virtual Distributed Computing Environment (Syracuse Univ.)
 - WebFLOW (Syracuse Univ.)
 - XCAT
- **Calcul Pair-à-pair (P2P)**
 - XtremWeb, CGP2P, SETI@home, JXTA, ...
- **De nombreux projets** (visualisation distribuée, travail coopératif, éducation à distance, environnement de simulation dépendant d'applications, ...)
- **Efforts logiciels proches**
 - GrADS, Globus, Legion, APPLEs, APST, Nimrod-G, DISCWorld, CoG toolkit, Web Services

RenPar – Avril 2002 – F. Desprez

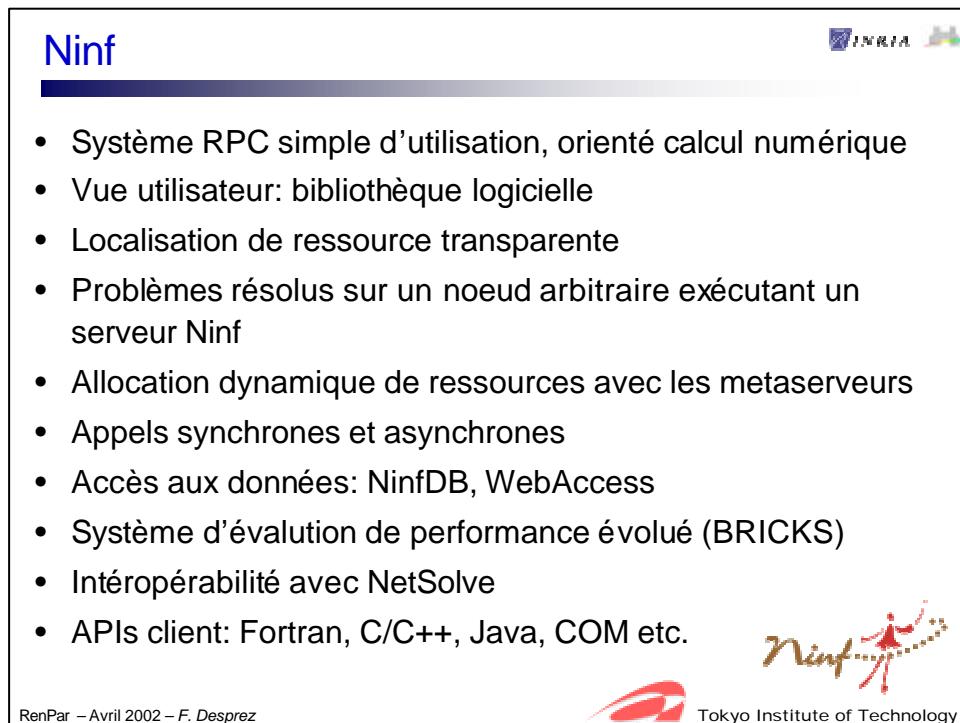
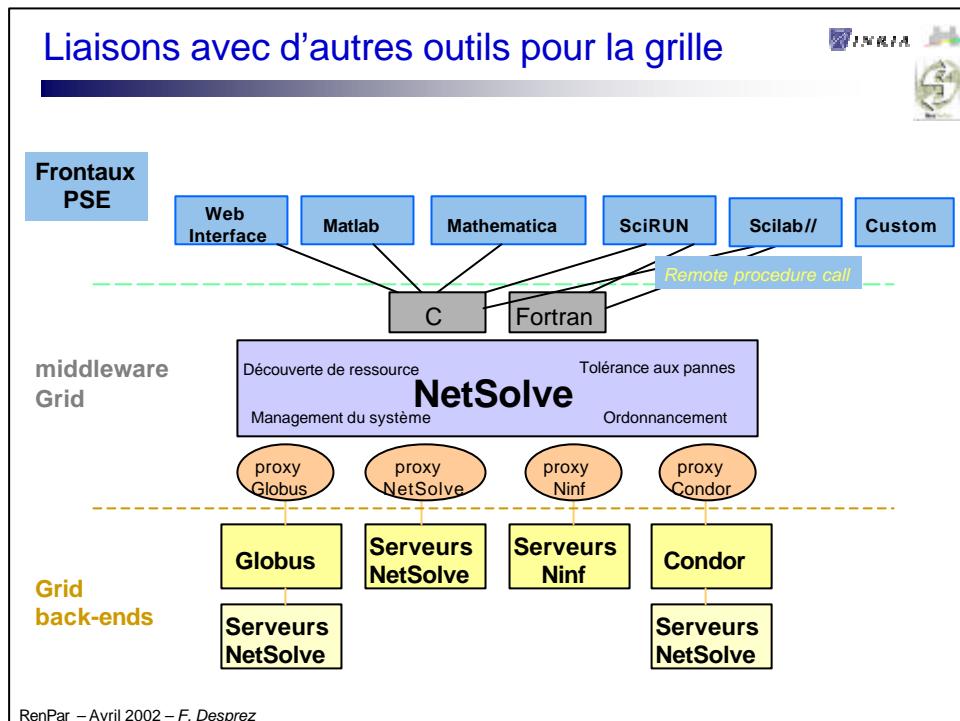


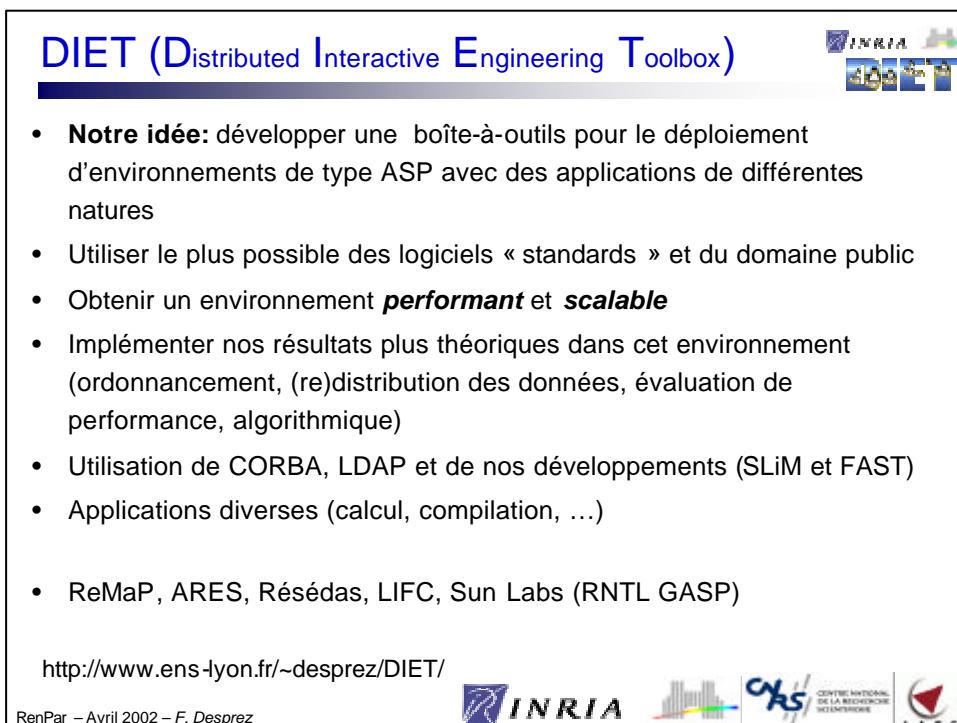
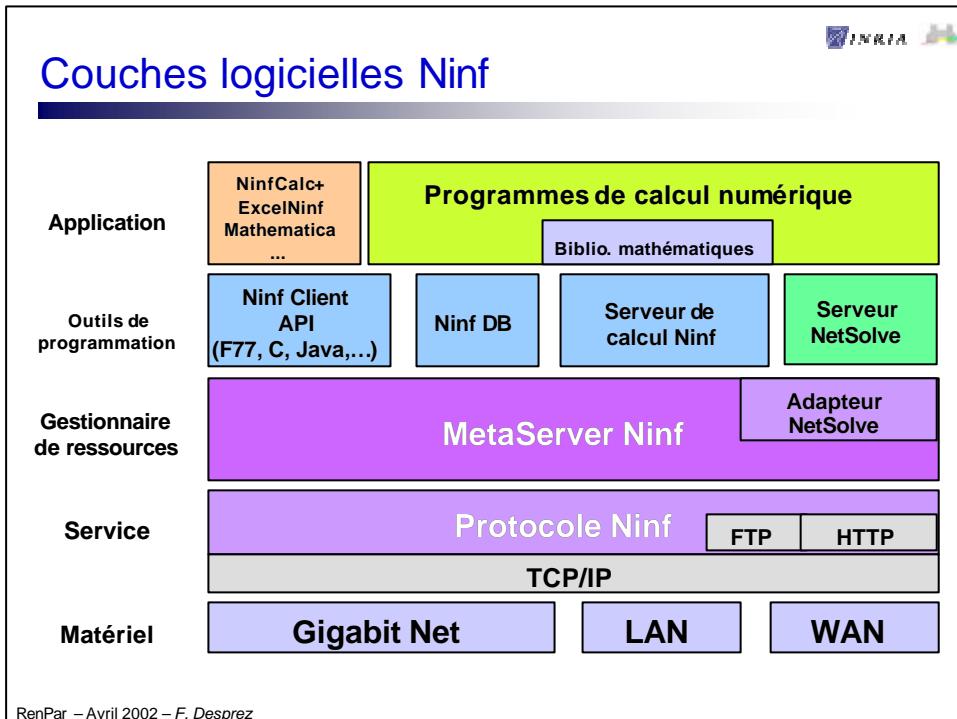
NetSolve

- Système RPC simple d'utilisation, orienté calcul numérique
- Vue utilisateur: bibliothèque logicielle
- Localisation de ressource transparente
- Appels synchrones et asynchrones
- Problèmes résolus sur un noeud arbitraire exécutant un serveur NetSolve
- Accès aux données: IBP
- Intéropérabilité avec Ninf
- Premières bibliothèques de calcul: BLAS, LAPACK, ScaLAPACK, PETSc
- APIs client: Fortran, C/C++, Java, Mathematica, Scilab

RenPar – Avril 2002 – F. Desprez

Innovative Computing Laboratory
COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF TENNESSEE





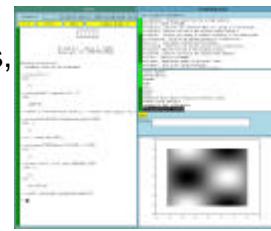
Historique



• 1998-2000: ARC INRIA OURAGAN

Outils pour la résolution de problèmes numériques de grandes tailles

- Parallélisation du logiciel Scilab (PVM, MPI, PBLAS, BLACS, ScaLAPACK, Pastix, NetSolve)
- Utilisation de Scilab face à des serveurs de routines de calcul (séquentielle ou parallèle)
- Optimisation de NetSolve (persistance des données, développement d'un environnement de prédition des temps de calcul et de communication)
- Découverte de problèmes (performances, scalabilité, intéropérabilité des logiciels, manque de standardisation)
- ReMaP, Métalau, Résédas, LIFC, LaBRI



RenPar – Avril 2002 – F. Desprez

Notre première vue des serveurs de calcul



• Idées

- Scilab comme une première application cible
- Pour simplifier l'utilisation de nouvelles bibliothèques (bibliothèques pour les problèmes manipulant des matrices creuses)
- Pour bénéficier des développements actuels autour du metacomputing
- Développer un toolkit pour les serveurs de calcul

• Premier prototype développé à partir de logiciels existants

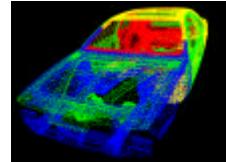
- NetSolve (Université du Tennessee)
- NWS (UCSD et UTK) pour l'évaluation dynamique de performances
- Nos développements sur les bibliothèques (routines de redistribution, solveurs creux, routines pour l'out-of-core)
- LDAP pour la base de donnée logicielle et CORBA pour la gestion des serveurs

RenPar – Avril 2002 – F. Desprez

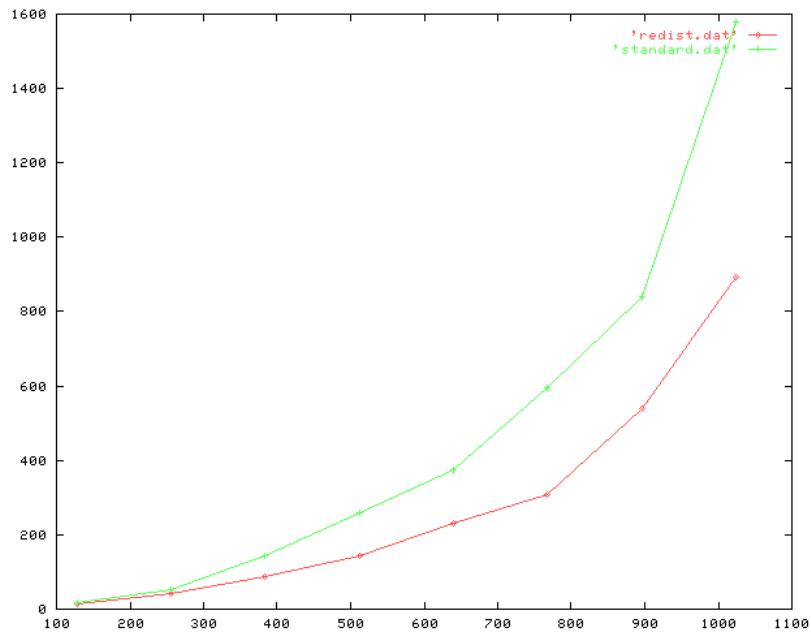
Nos premiers buts



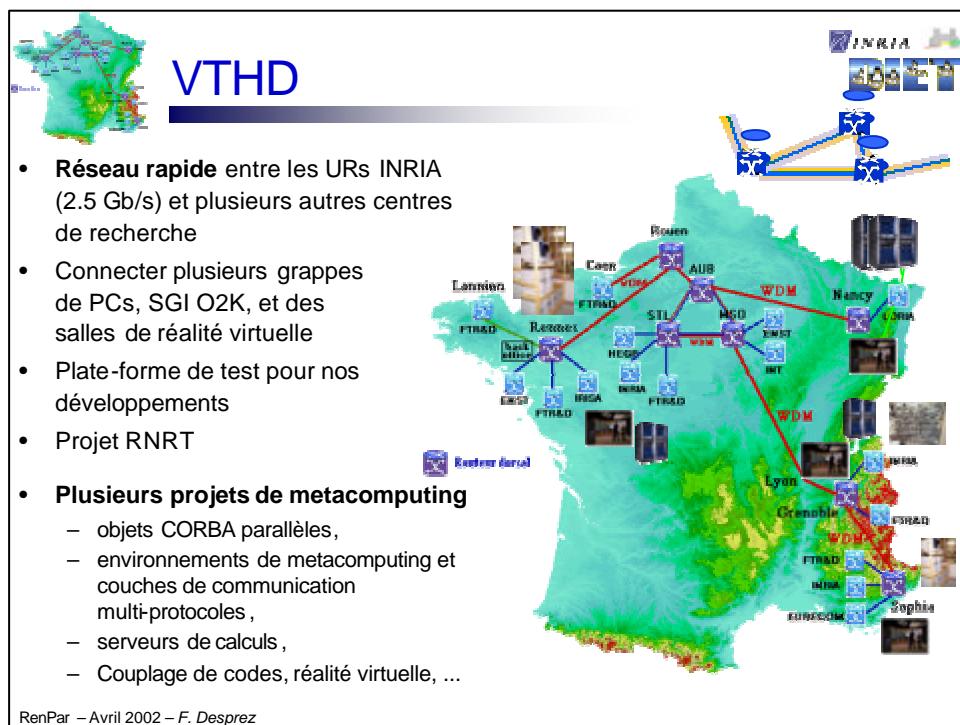
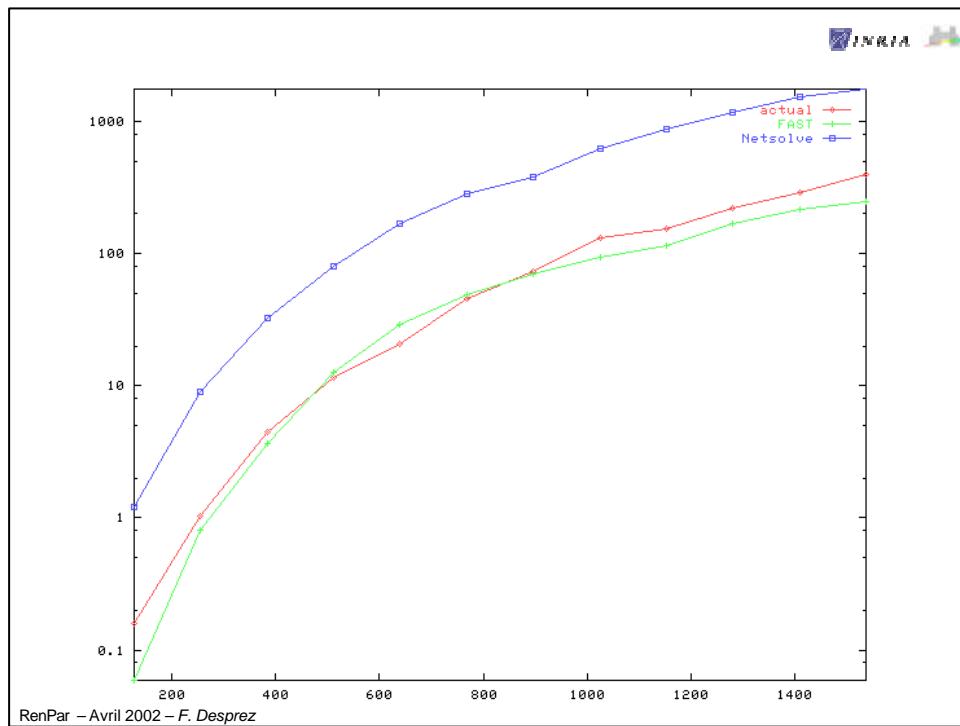
- Ajouter des fonctionnalités à Netsolve pour notre application
 - Persistance des données sur les serveurs
 - Redistribution et parallélisme entre les serveurs
 - Meilleure évaluation des paires [routine, machine] pour les calculs à grain fin
 - Amélioration de l'ordonnanceur (tester des heuristiques d'ordonnancement on-line)
 - Base de donnée portable des bibliothèques disponibles (LDAP)
- Avoir une plate-forme d'expérimentation pour nos développements
 - Approche à parallélisme mixte (parallélisme de données et parallélisme de tâches)
 - Heuristiques d'ordonnancement pour les tâches data-parallèles
 - Algorithmes parallèles sur des plates-formes hétérogènes
 - Evaluation de performances
 - Gestion des serveurs avec CORBA
 - Visualisation interactive de données distribuées
- NSF-INRIA avec l'Université du Tennessee



RenPar – Avril 2002 – F. Desprez



RenPar – Avril 2002 – F. Desprez



Les applications cibles

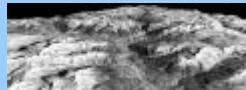


Action Concertée Incitative (ACI)
Globalisation des Ressources Informatiques
et des Données (GRID)

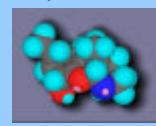


- Chercheurs de quatre disciplines différentes (chimie, physique, électronique, géologie)
- Quatre applications aux besoins et aux modes de fonctionnement différents

Modèles Numériques de Terrain



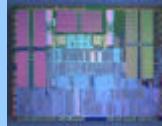
Dynamique Moléculaire



HSEP



Simulation de Circuits



RenPar – Avril 2002 – F. Desprez

<http://www.ens-lyon.fr/~desprez/GRIDASP/>

Modèles numériques de terrain



- Traitement stéréoscopique :
 - Correspondance maximum entre les points de deux images.
 - Calcul des altitudes.

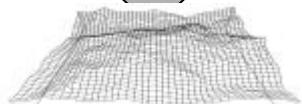


Informations de prises de vues et coordonnées des points homologues initiaux

- Contraintes géométriques
- Différences radio-métriques

Fichiers binaires

MNT



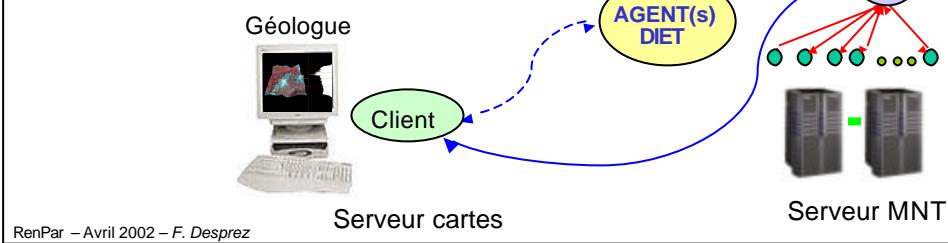
LST

RenPar – Avril 2002 – F. Desprez

Modèles numériques de terrain, suite



- Besoins spécifiques à l'application :
 - Consommation mémoire importante
 - Volume de données important
 - Visualisation.
- Approche ASP :
 - Puissance de calcul :
 - Traitement d'images de très hautes résolutions
Ex : Images Satellite SPOT < 5m
 - Réduction du temps de traitement des images
Ex : Séisme.



RenPar – Avril 2002 – F. Desprez

INTERFACES CLIENTS

```

for (i = 0; i < NUM_HOSTS; i++){
    sprintf(entry[i], "ninf://%s:%d/pi/pi_trial", hosts[i], port);
    if ((ids[i] = Ninf_call_async(entry[i], rand(), times, &count[i])) == NINF_ERROR){
        Ninf_perror("pi_trial"); exit(2);
    }
}
while (1) {
    int id = Ninf_wait_any(); /* WAIT FOR ANY HOST */
    if (id == NINF_OK)
        break;
    for (i = 0; i < NUM_HOSTS; i++) /* FIND HOST */
        if (ids[i] == id) break;
    sum += count[i];
    done += times;
    if (done >= whole_times)
        continue;
    if ((ids[i] =
        Ninf_call_async(entry[i], rand(), times, &count[i])) == NINF_ERROR){
        Ninf_perror("pi_trial"); exit(2);
    }
}

```

RenPar – Avril 2002 – F. Desprez

Interface client



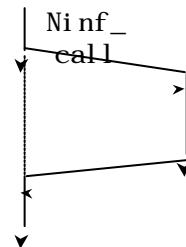
- La plus simple possible
- Multi-interface (C, C++, Fortran, Java, Matlab, Mathematica, Scilab, Web, etc.)
- Proposition d'une interface standard par les équipes de Ninf et Netsolve
- Utilisation d'un Proxy
 - processus séparé qui fait la liaison entre le client et les autres composants

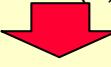
RenPar – Avril 2002 – F. Desprez

API Client Ninf



- Ninf_call(**FUNC_NAME**,);
- **FUNC_NAME** = **NAME / ninf://HOST:PORT/ENTRY_NAME**
- API pour C, C++, Fortran, Java, Lisp, COM, Mathematica, ...



```
double A[n][n],B[n][n],C[n][n]; /* Declaration de données*/  
dmmul(n,A,B,C); /* Appel de fonction locale */  
  
Ninf_call("dmmul",n,A,B,C); /* Appel de fonction Ninf */
```

RenPar – Avril 2002 – F. Desprez

Ninf Interface Description (Ninf IDL)



```
Define dmmul(long mode_in int n,
             mode_in double A[n][n], mode_in double B[n][n],
             mode_out double C[n][n])
" description "
Required "libXXX.o"
CalcOrder n^3
Calls "C" dmmul(n,A,B,C);
```

- IDL information:
 - library function's name, and its alias (**Define**)
 - arguments' access mode, data type (**mode_in**, **out**, **inout**, ...)
 - required library for the routine (**Required**)
 - computation order (**CalcOrder**)
 - source language (**Calls**)

RenPar – Avril 2002 – F. Desprez

Proposition d'une interface client standard



- *Draft Grid RPC API Proposal*, v.0.91, Feb., 2002
- Equipes de Ninf et NetSolve
- Pour permettre de partager les outils et de les faire collaborer
- Subset des appels NetSolve et Ninf
- Pbs avec les codes erreurs dépendants des systèmes

```
• int grpc_function_handle_init(grpc_function_handle_t * handle,
                                 char * host_name, int port, char * func_name);
• int grpc_call(grpc_function_handle_t *, ...);
• int grpc_call_async(grpc_function_handle_t *, ...);
• int grpc_wait(int sessionID);
• int grpc_get_error(int sessionID);
```

RenPar – Avril 2002 – F. Desprez

Exemple de code

```
#include "grpc.h"
#define NUM_HOSTS 8
char * hosts[] =
    {"host00", "host01", "host02", "host03",
     "host04", "host05", "host06", "host07"};
grpc_function_handle_t handles[NUM_HOSTS];
int port = 4000;

main(int argc, char ** argv){
    double pi;
    long times, count[NUM_HOSTS], sum;
    char * config_file;
    int i;
    if (argc < 3){
        fprintf(stderr,
            "USAGE: %s CONFIG_FILE TIMES \n",
            argv[0]);
        exit(2);
    }
    config_file = argv[1];
    times = atol(argv[2]) / NUM_HOSTS;

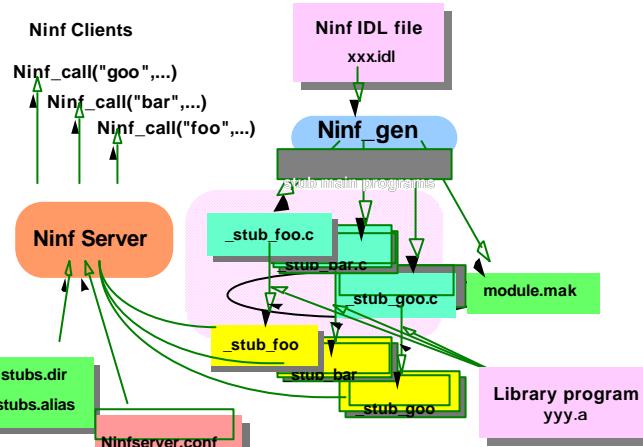
    /* GRPC module initialization */
    if (!grpc_initialize(config_file))
        grpc_perror("grpc_initialize");
    exit(2);
}
```

```
/* initialize function handle */
for (i = 0; i < NUM_HOSTS; i++)
    grpc_function_handle_init(&handles[i],
        hosts[i], port, "pi/pi_trial");

for (i = 0; i < NUM_HOSTS; i++)
    /* parallel invocation */
    if (grpc_call_async(&handles[i], i,
        times, &count[i]) == GRPC_ERROR){
        grpc_perror("pi_trial");
        exit(2);
    }
    /* wait for all the calls */
if (grpc_wait_all() == GRPC_ERROR){
    grpc_perror("wait_all");
    exit(2);
}
/* PI calculation */
for (i = 0, sum = 0; i < NUM_HOSTS; i++)
    sum += count[i];
pi = 4.0 *
    (sum / ((double) times * NUM_HOSTS));
printf("PI = %f\n", pi);
/* GRPC module finalize */
grpc_finalize();
}
```

RenPar – Avril 2002 – F. Desprez

AJOUT DE SERVICES



RenPar – Avril 2002 – F. Desprez

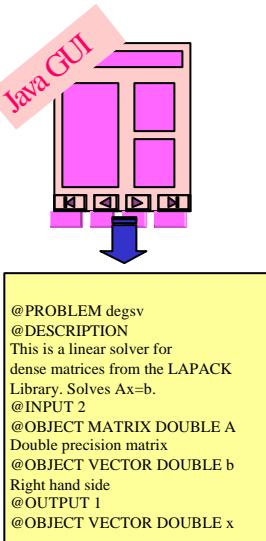
Ajout de services



- Installation de nouveaux serveurs et enregistrement auprès de l'agent
- Description des problèmes et des paramètres en entrée et en sortie
- Génération des interfaces clients
- Permettre le transfert des IDL client depuis les serveurs (pour les laisser les plus *light* possibles)
- Ajout dynamique des exécutables à partir de bibliothèques logicielles (Netlib)
 - Durée limitée, *cache* de services
 - Pbs de versions de systèmes et de machines, de sécurité

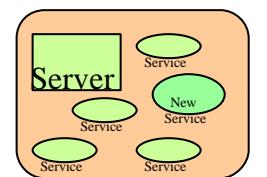
RenPar – Avril 2002 – F. Desprez

Génération de nouveaux services dans NetSolve



- Décrire l'interface (arguments)
- Générer le wrapper
- Installer sur le serveur
- Description de problèmes très bas niveau

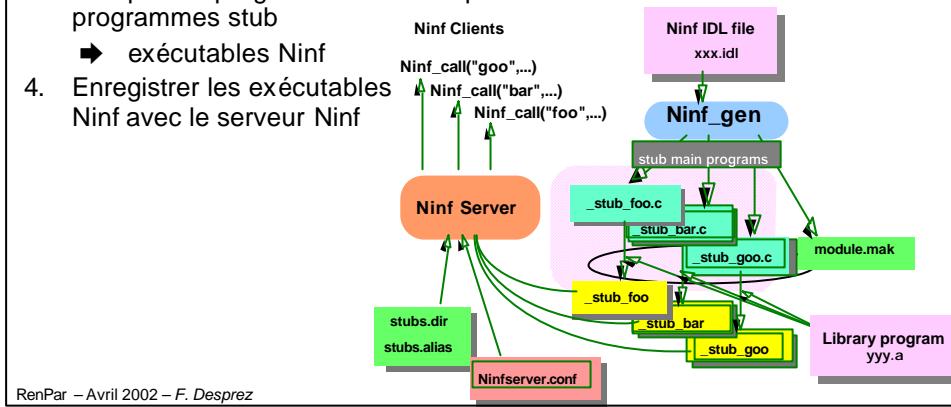
Ajout d'un nouveau service



RenPar – Avril 2002 – F. Desprez

“Gridifier” une bibliothèque/Application

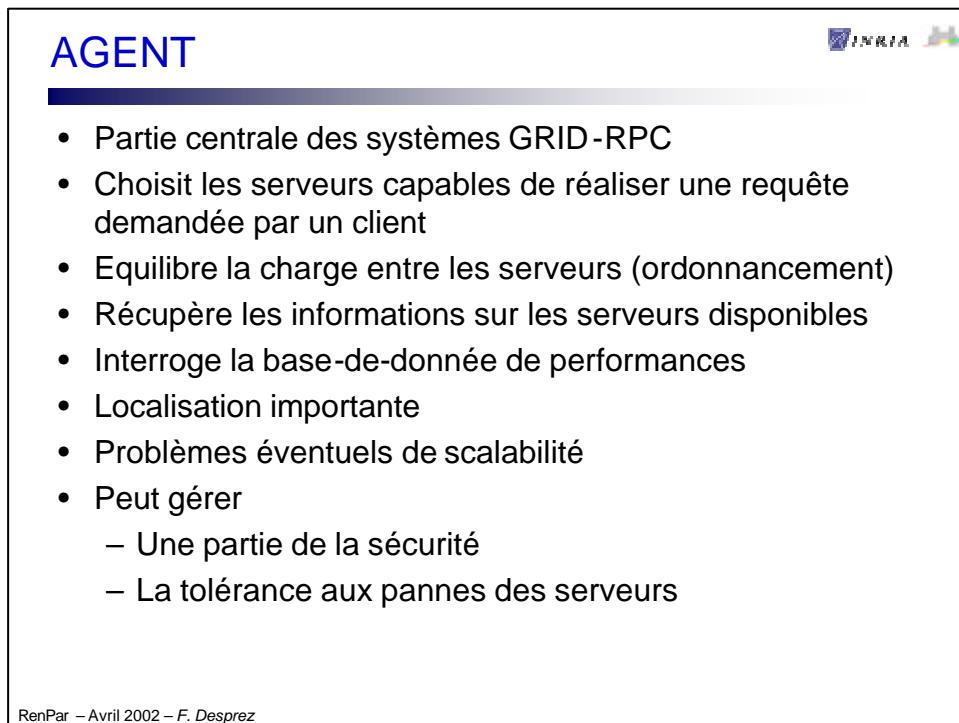
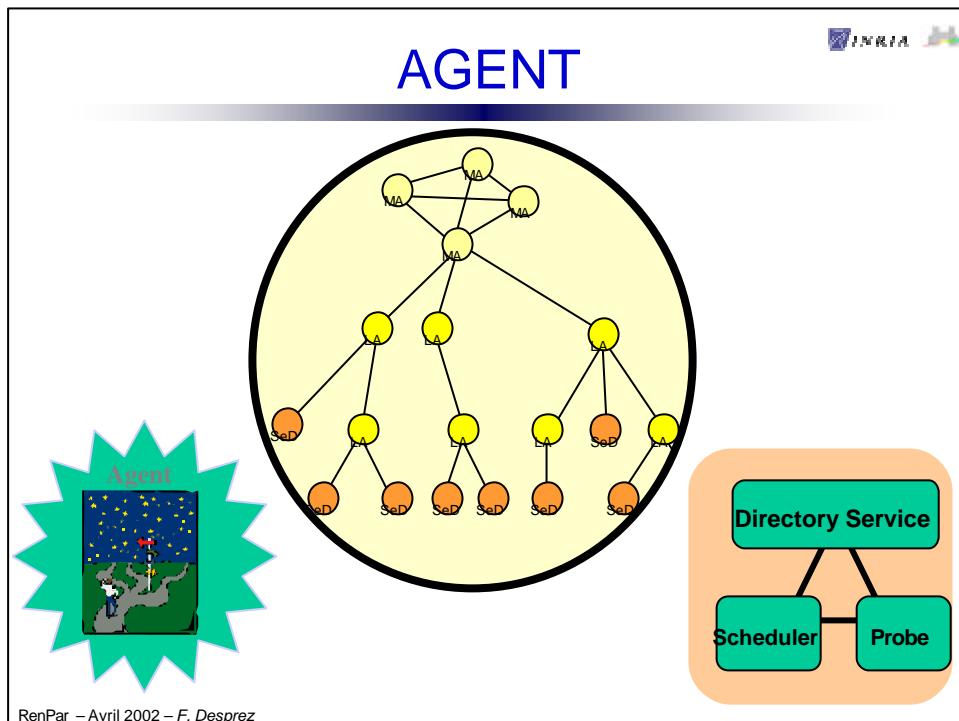
1. Ecrire la description d'interface pour *Gridifier une app/bibliothèque* dans l'IDL Ninf
→ Fichier d'IDL Ninf
2. Exécuter le générateur d'interface Ninf sur le serveur
→ Programmes stub et Makefile
3. Compiler le programme bibliothèque et le lier avec les programmes stub
→ exécutables Ninf
4. Enregister les exécutables Ninf avec le serveur Ninf



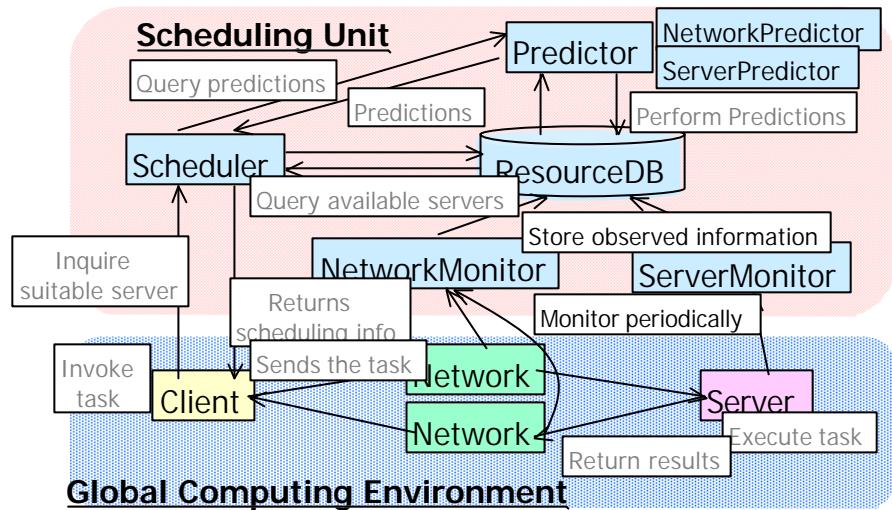
SLiM: Scientific Libraries Metaserver

- Système de nommage de ressources logicielles
- Pour être capable de dire à l'ordonnanceur
 - Quels serveurs sont capables de réaliser quelle opération (en utilisant quelle bibliothèque)
 - Quels sont les utilisateurs de l'environnement (et les opérations qu'ils effectuent)
 - Où sont les données génériques
- Utilisation de LDAP (*Lightweight Directory Access Protocol*)
- Avantages et inconvénients de LDAP:
 - 😊 Distribué
 - 😊 Hiérarchique
 - 😊 Protocole ouvert, standard
 - 😊 Optimisé pour la lecture (pas pour la mise à jour)
- ACI GRID TLSE avec l'ENSEEIHT

RenPar – Avril 2002 – F. Desprez

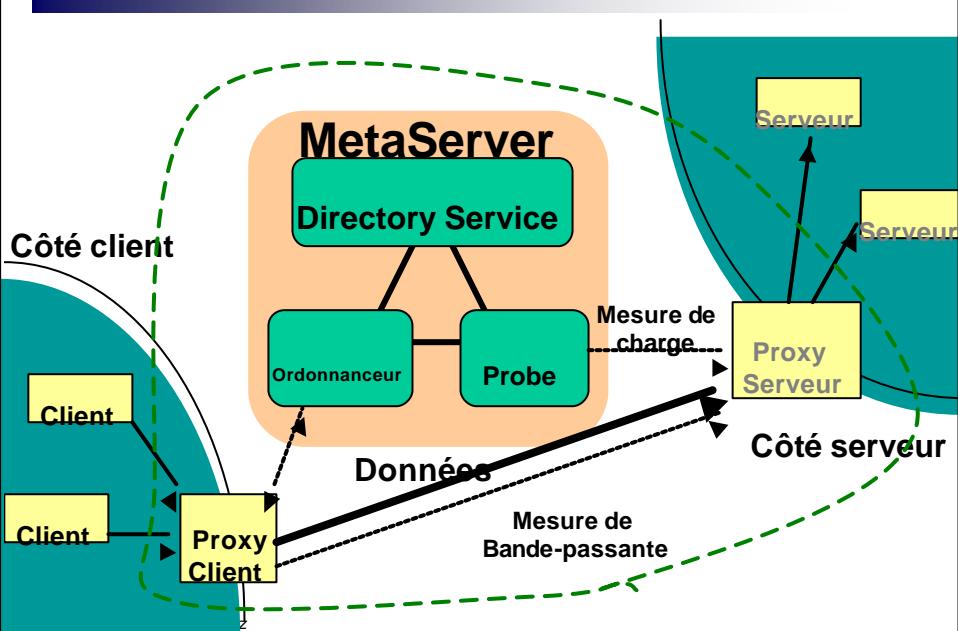
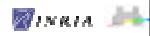


Fonctionnement d'un agent



RenPar – Avril 2002 – F. Desprez

Architecture du MetaServeur Ninf



AGENT NetSolve

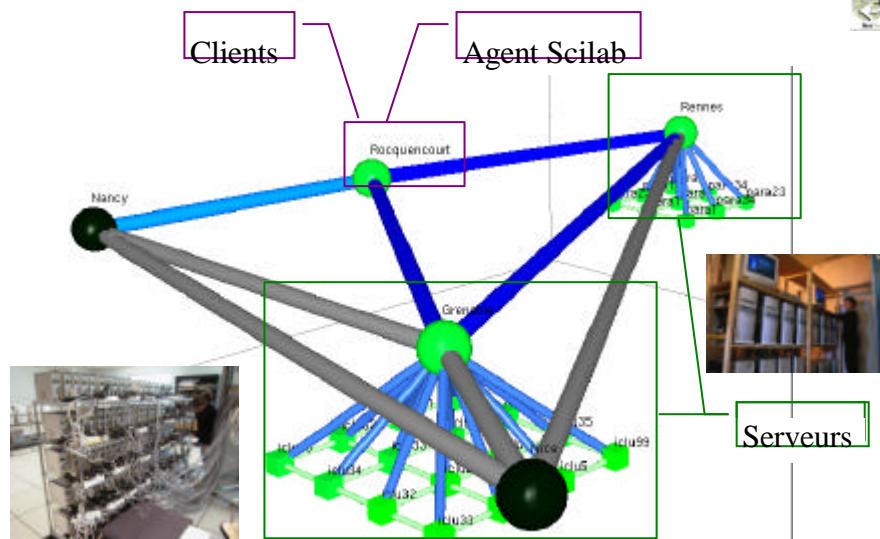


- Serveur de nom pour le système NetSolve
- **Service d'information**
 - Les utilisateurs clients et les administrateurs peuvent demander quels sont les services logiciels et matériels disponibles
- **Ordonnanceur de ressources**
 - Maintient une base de données d'informations statiques et dynamiques sur les serveurs NetSolve pour être capable d'allouer les ressources
 - Performances CPU (benchmark LI NPACK)
 - Bande-passante et latency réseau
 - Charge des serveurs
 - Complexité des algorithmes
 - Calcule le temps d'exécution pour chaque serveur
 - Donne au client une liste des serveurs les plus appropriés



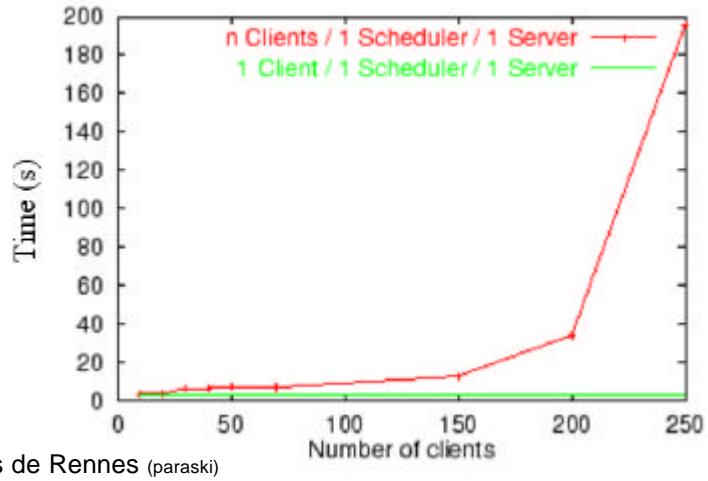
RenPar – Avril 2002 – F. Desprez

Netsolve sur VTHD



RenPar – Avril 2002 – F. Desprez

Comportement de NetSolve



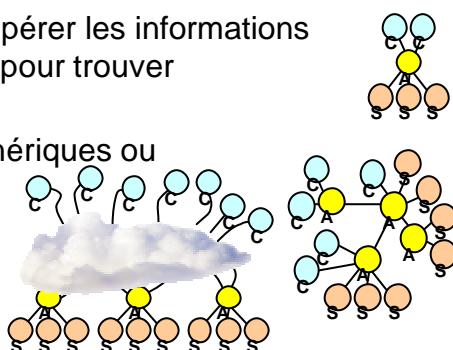
- Réseau : VTHD
- Clients : Grappes de Rennes (paraski)
- Scheduler : Agent NetSolve (Rocquencourt)
- Serveur : paraski26 (paraski)

RenPar – Avril 2002 – F. Desprez

AGENT(s) DIET

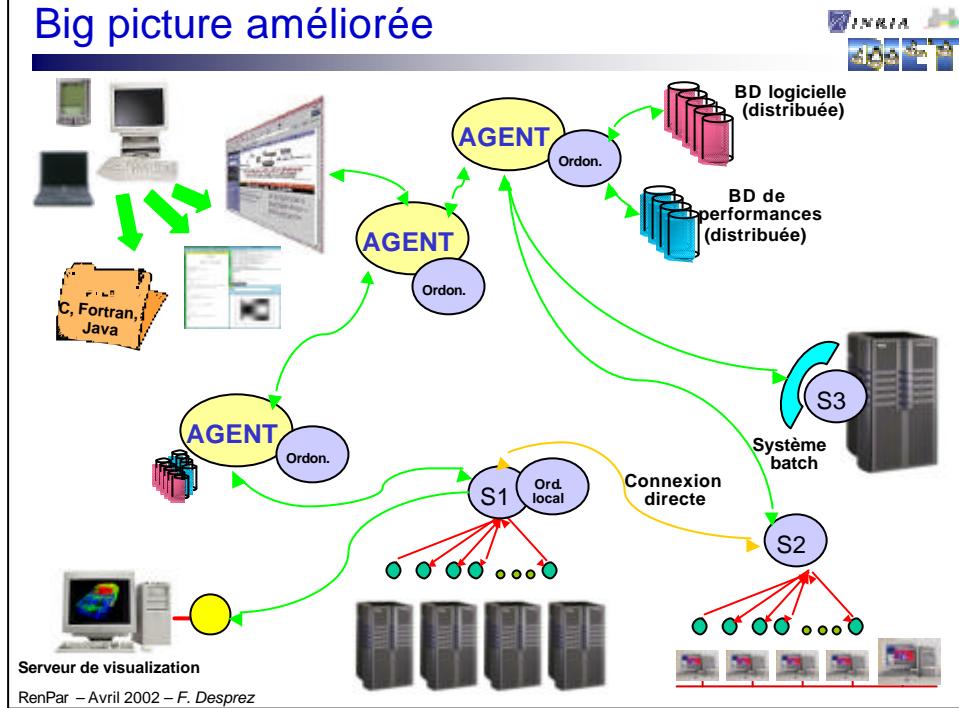


- Ensemble distribué d'agents pour une meilleure scalabilité
- Etude de plusieurs connexions entre les agents (hiérarchique, distribuée, agents dupliqués, ...) et du placement des agents
- Parcours d'arbres avec informations réparties dans les nœuds pour l'approche hiérarchique
- Connexion à FAST pour récupérer les informations sur les ressources et à SLIM pour trouver les applications utiles
- Différents ordonnanceurs génériques ou dépendants des applications
- Corba, JXTA



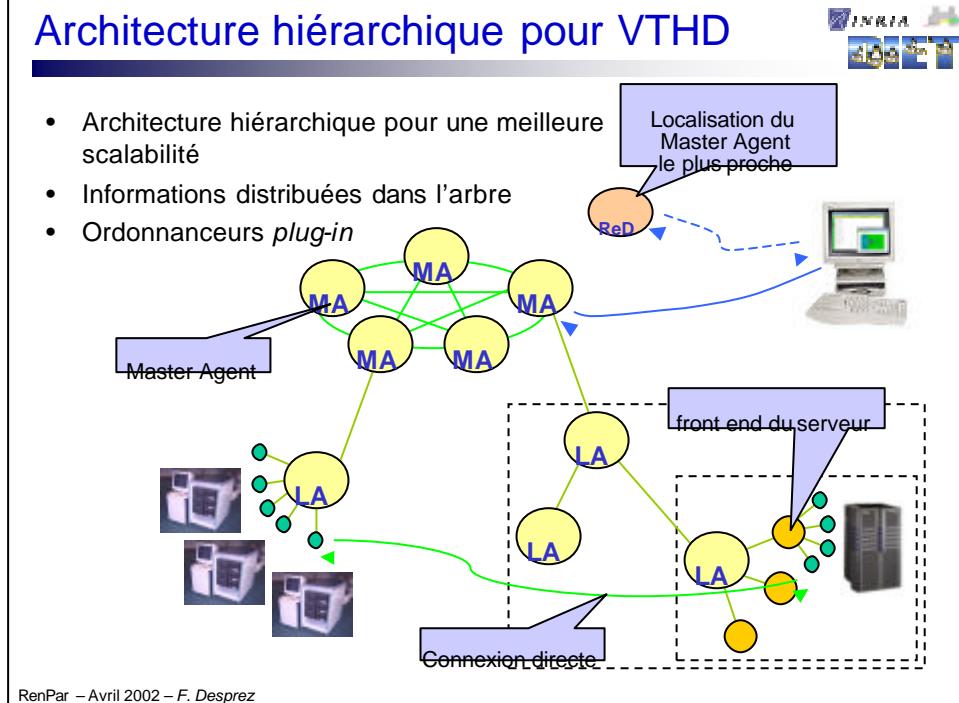
RenPar – Avril 2002 – F. Desprez

Big picture améliorée

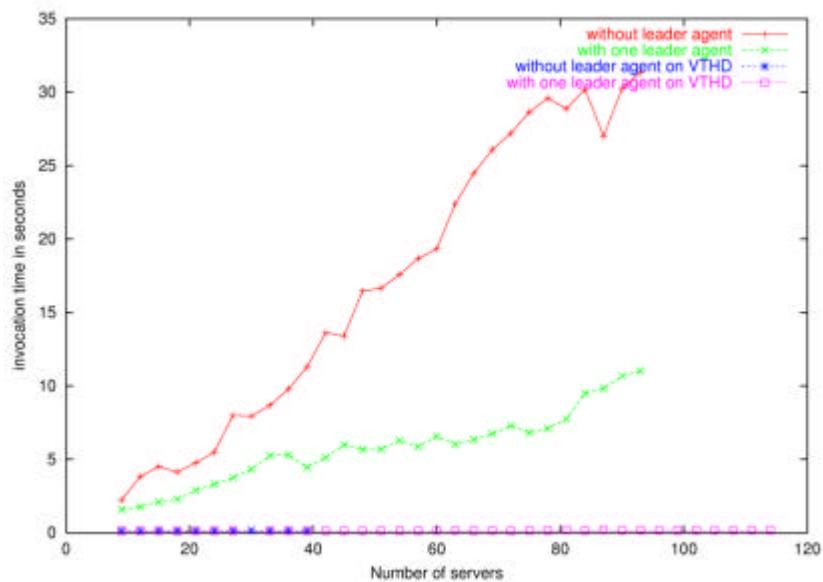


Architecture hiérarchique pour VTHD

- Architecture hiérarchique pour une meilleure scalabilité
 - Informations distribuées dans l'arbre
 - Ordonnanceurs *plug-in*

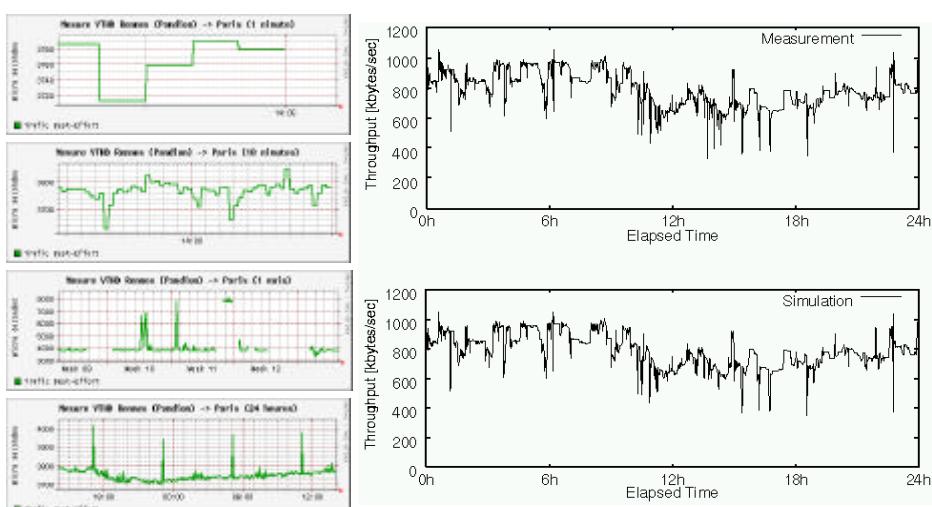


Evaluation de l'invocation du serveur DIET



RenPar – Avril 2002 – F. Desprez

EVALUATION DE PERFORMANCES



RenPar – Avril 2002 – F. Desprez

Evaluation de performance

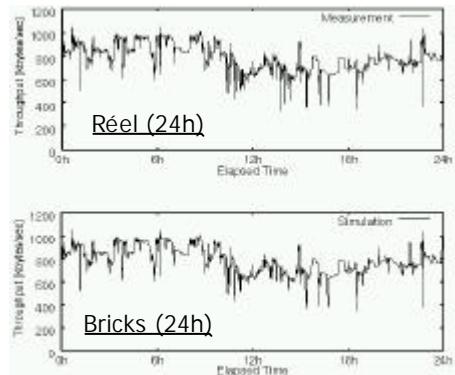
- Evaluation de performances de la plate-forme GRID -RPC
- Trouver un (ou plusieurs) serveur(s) efficace(s) (coûts de calcul de la fonction demandée, charge du serveur, coûts de communication entre le client et le serveur, capacité mémoire, etc.)
→ base de données de performances pour l'ordonnanceur
- Difficile de modéliser/comprendre des réseaux comme Internet ou VTHD
- Temps de réaction court essentiel
- Etre capable de modéliser les applications (pbs des applications qui dépendent des données en entrée)
- Accounting

RenPar – Avril 2002 – F. Desprez



Bricks

- Simulation d'un environnement de calcul global et d'une unité d'ordonnancement
- Permet la simulation du comportement
 - d'algorithmes d'ordonnancement de ressources
 - de modules de programmation pour l'ordonnancement
 - de topologies réseau de clients et de serveurs
 - De schémas de traitement pour les réseaux et les serveurs
- Effectue des benchmarks des composants d'ordonnancement disponibles pour le calcul global
- Intégration de NWS (Univ. Santa Barbara)
- Système à files d'attentes

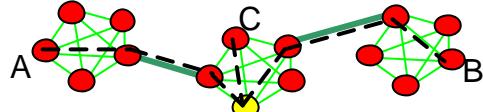


RenPar – Avril 2002 – F. Desprez

FAST: Fast Agent's System Timer

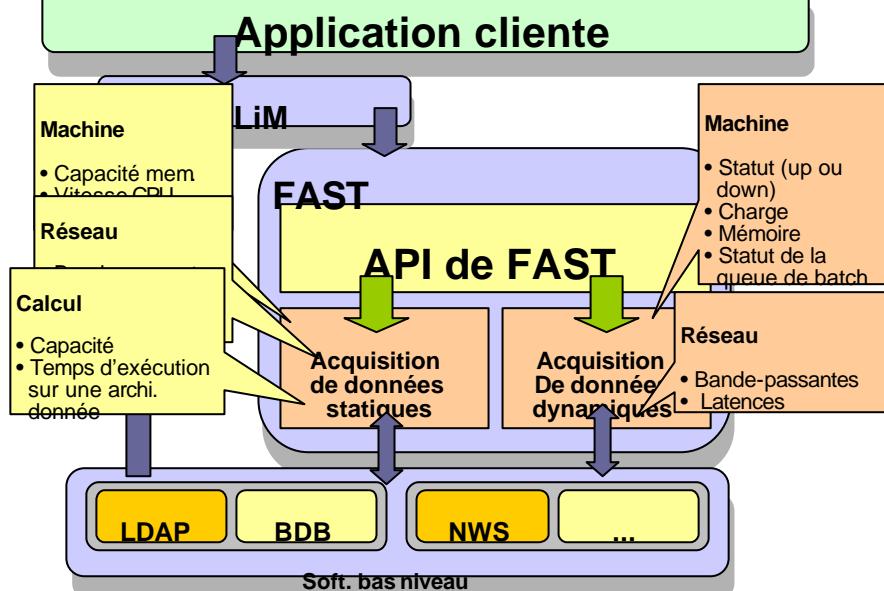


- Basé sur NWS (Network Weather Service)
- **Performances de calcul**
 - charge, capacité mémoire et performances des queues de batch (dynamique)
 - Tests exhaustifs des bibliothèques disponibles (statique)
- **Performances de communication**
 - Etre capable de trouver le coût des redistributions de données entre deux serveurs en fonction de l'architecture réseau et des informations dynamiques
 - Bande-passante et latence (hiérarchique)
- **Ensemble hiérarchique d'agents**
 - Problèmes de scalabilité



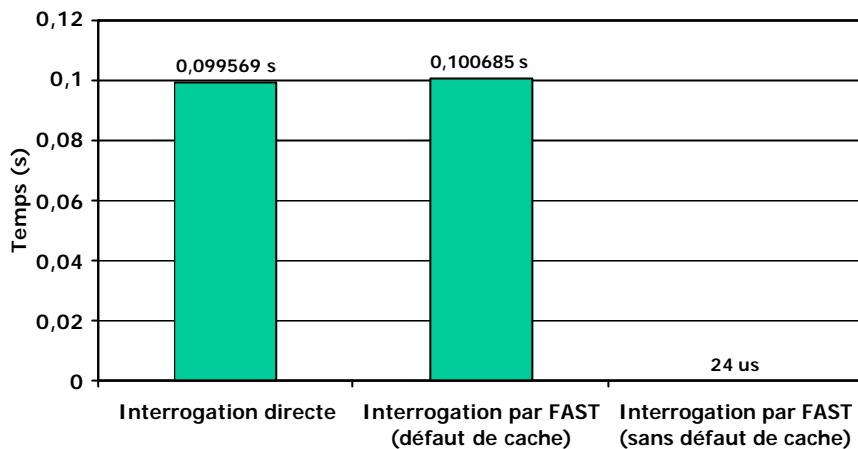
RenPar – Avril 2002 – F. Desprez

SLiM et FAST



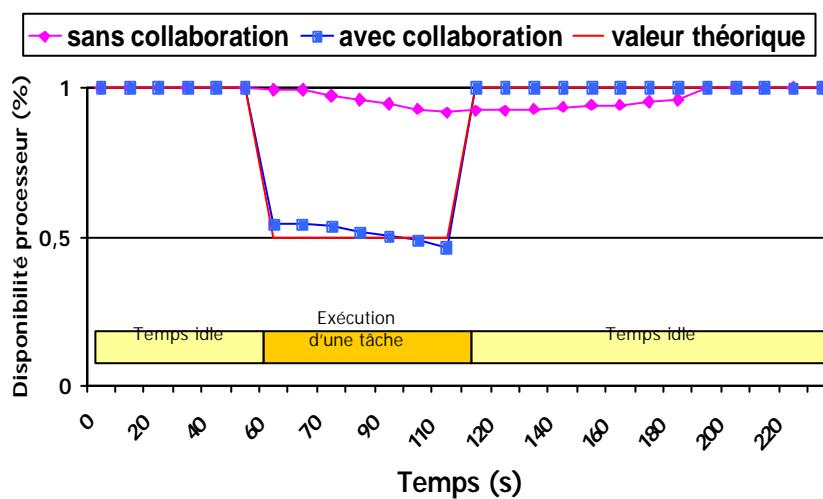
RenPar – Avril 2002 – F. Desprez

Améliorations de NWS - temps de réponse



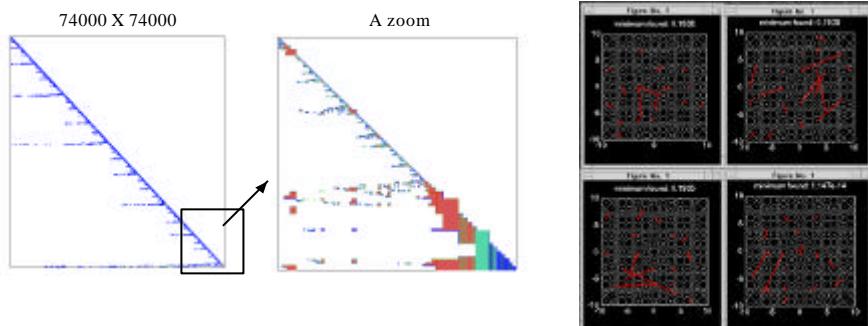
RenPar – Avril 2002 – F. Desprez

Améliorations de NWS - collaboration avec l'ordonnanceur



RenPar – Avril 2002 – F. Desprez

GESTION DES DONNEES



RenPar – Avril 2002 – F. Desprez

Gestion des données

- **Schéma classique :**
envoi données – calcul – récupération du résultat
- Coût important sur la grille (communications)
- Solution : persistance des données (si possible transparente)
- Plusieurs solutions dans d'autres environnements
 - IBP : *Internet Backplane Protocol*
 - GASS (Globus)
- NetSolve : *request sequencing* et IBP
- Ninf : *request sequencing-like* + stockage
- DIET : persistance avec marquage des données à stocker

RenPar – Avril 2002 – F. Desprez

NETSOLVE



- **Request sequencing**

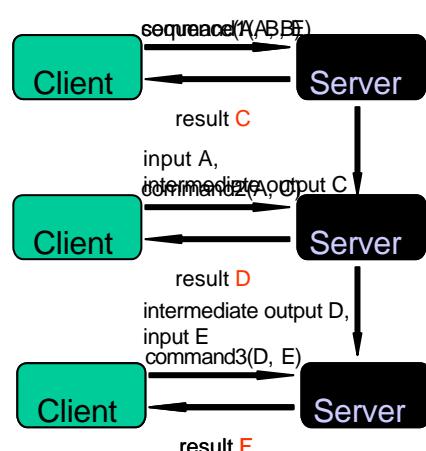
- Enchaînement d'une séquence de requêtes Netsolve
- Analyser les paramètres pour déterminer les dépendances (analyse statique d'un DAG)
- Transmettre l'ensemble des paramètres d'entrée et de sorties et les rendre persistants durant l'exécution
- Ordonnancer les requêtes "classiquement"

- **IBP (Internet Backplane Protocol)**

- Cache de données sur Internet
- Les données à calculer sont déposées dans le cache
- Elles sont utilisées ensuite par les calculs futurs

RenPar – Avril 2002 – F. Desprez

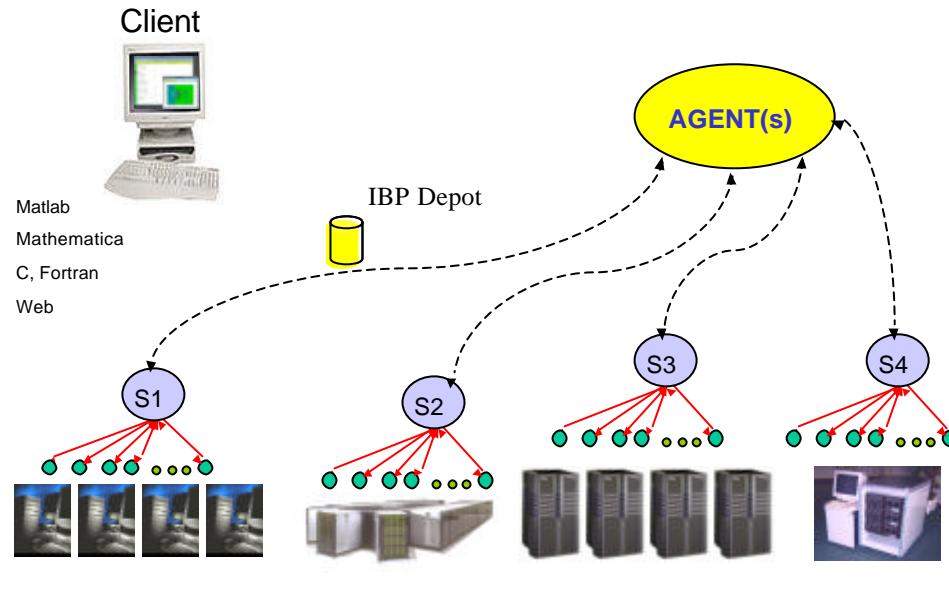
NetSolve (Request sequencing)



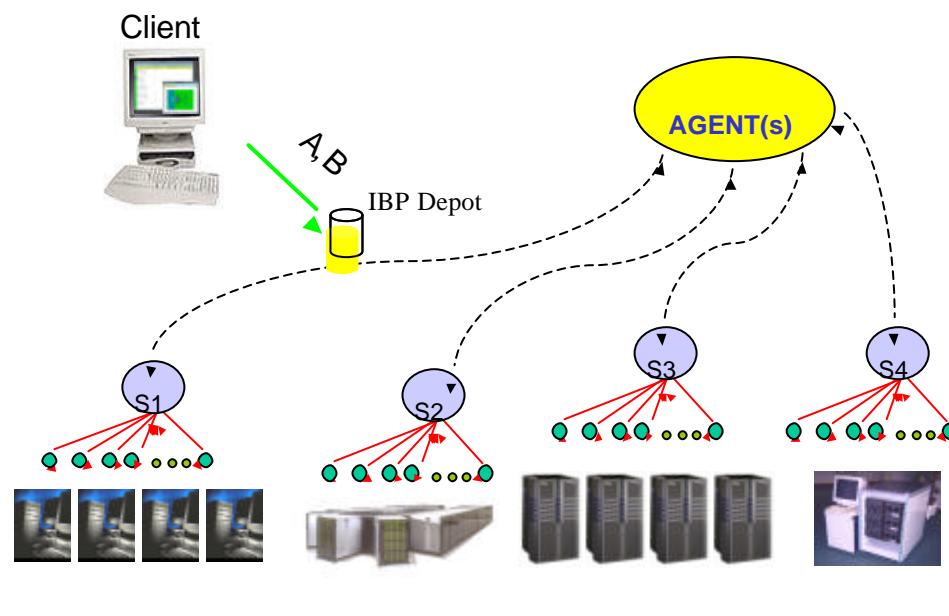
```
netsl_begin_sequence();
netsl("COMMAND1", A, B, C);
netsl("COMMAND2", A, C, D);
netsl("COMMAND3", D, E, F);
netsl_end_sequence(C, D);
```

RenPar – Avril 2002 – F. Desprez

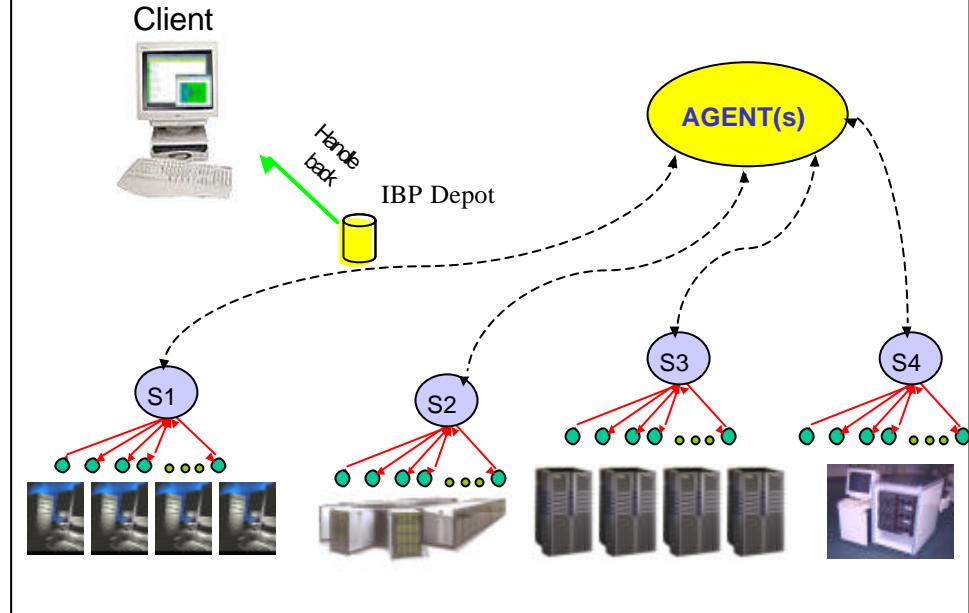
NetSolve: IBP



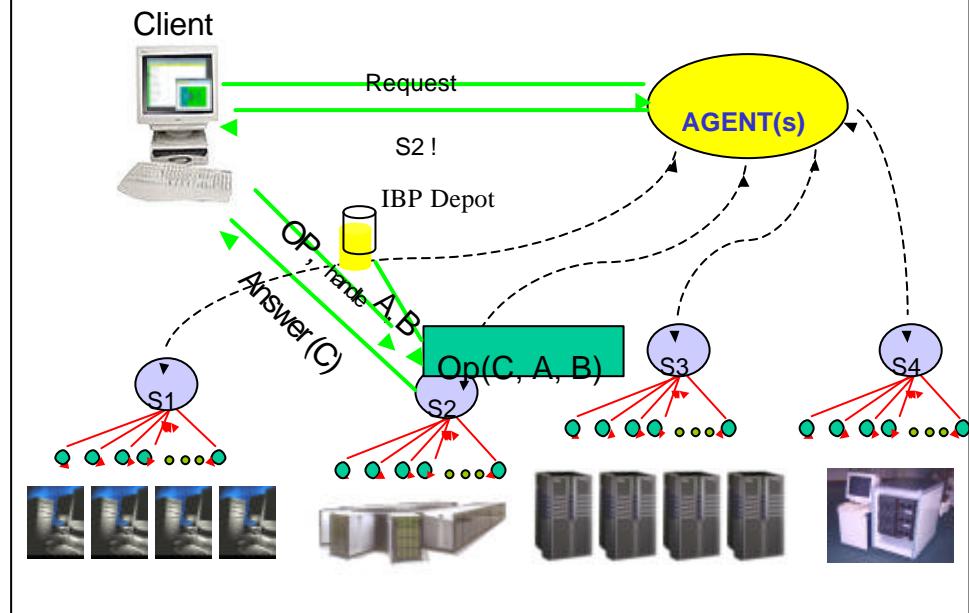
NetSolve: IBP



NetSolve: IBP



NetSolve: IBP

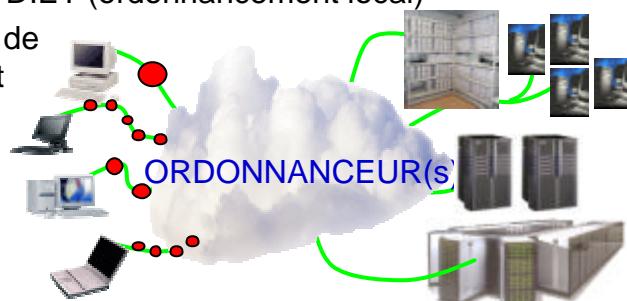


AUTRES CARACTERISTIQUES

RenPar – Avril 2002 – F. Desprez

Ordonnancement

- *Shortest Execution Time First*
- Autres algorithmes possibles (modèle économique, dead-line scheduling , pb classique de l'ordonnancement *on-line*)
- *Request sequencing*
- Mono-agent pour Ninf et Netsolve
- Multi-agents pour DIET (ordonnancement local)
- Modéliser le coût de l'ordonnancement lui-même

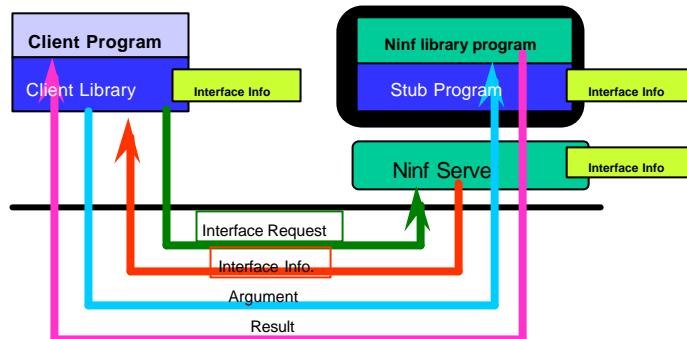


RenPar – Avril 2002 – F. Desprez

Interface de communication



- Généralement sockets avec *connexion par nécessité* (Ninf et Netsolve) ou Corba (DIET)
- Formats binaires pour Ninf et Netsolve
- Utilisation de XML pour d'autres outils GRID
 - nombreux outils et parseurs
 - Protocole structuré mais texte
 - RPC avec XML (SOAP) et description de services (WSDL)



RenPar – Avril 2002 – F. Desprez

Sécurité



- **Indispensable !**
 - Sécurité des communications,
 - authentification des clients pour les serveurs
 - Authentification des serveurs pour les clients
 - Partage de serveurs entre plusieurs clients (couplage d'applications)
 - Délégation d'autorité
- **Netsolve**
 - Utilisation de Kerberos V5
 - Uniquement génération de listes de contrôles d'accès
- **Ninf, DIET**
 - Authentification basée sur SSL (style GSI)
 - **NAA** (NES Authentication Authorization module)

RenPar – Avril 2002 – F. Desprez

Tolérance aux pannes



- **Disparition d'un serveur** avant/après l'acceptation d'une requête
 - Le serveur est retiré de la base-de-données
- **Disparition de l'agent**
 - Généralement plantage du système (disparition des connexions et des enregistrements des serveurs)
 - Problème des systèmes centralisés
- Netsolve
 - Système de notification rudimentaire
- Ninf
 - Fault Manager

RenPar – Avril 2002 – F. Desprez

CONCLUSIONS



RenPar – Avril 2002 – F. Desprez

Conclusions



- Approche valable pour de nombreuses applications
- Bonnes performances et très grande flexibilité
- Possibilité de la coupler avec d'autres systèmes pour une plus grande couverture (cf XtremWeb et ASP dans CASPER)
- Nécessité d'utiliser des standards et de ne pas réinventer la roue (factorisation des développements et des recherches) !
- Nombreux problèmes de recherche (ordonnancement, gestion des données, sécurité, découverte de ressources, déploiement, description de problèmes, tolérance aux pannes, aide à la décision, etc.)
- Liaisons évidentes avec les Web Services (mais attention aux performances)
- Clients légers et wireless ?



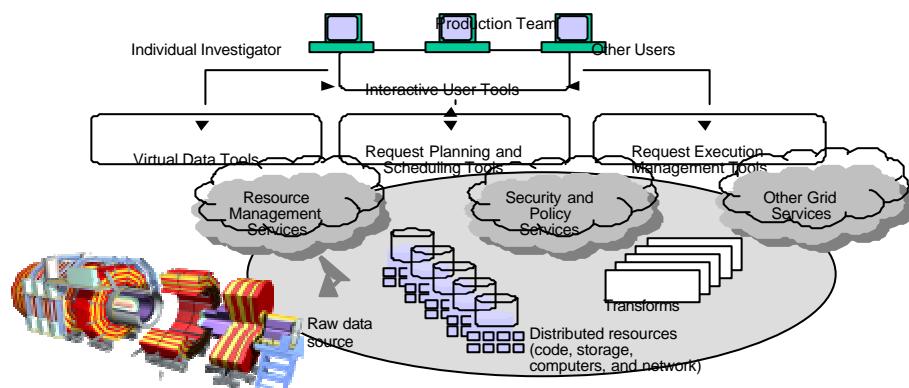
RenPar – Avril 2002 – F. Desprez

INRIA

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)



RenPar – Avril 2002 – F. Desprez

Conclusions et travaux futurs sur DIET



- Développement d'un ensemble d'outils portables pour bâtir des environnements de type ASP
- Multi-applications, multi plates-formes et multi-interfaces
- Réutiliser les développements effectués dans d'autres projets (NetSolve, Ninf, Paris, CGP2P ?)
- Focalisation sur des problèmes tels que la localisation de ressources, l'ordonnancement, le déploiement d'agents, l'algorithmique, l'analyse de performances
- Intégration de CORBA et Java
- Validations sur diverses « vraies » applications
- Portabilité
- Open-source
- RNTL GASP, ACI GRID ASP, RNRT VTHD++

RenPar – Avril 2002 – F. Desprez

<http://www.ens-lyon.fr/~desprez/DIET/>

Problèmes reliés



- Sécurité
 - Comptes et accès utilisateurs
 - Transferts de données
- Tolérance aux pannes
 - Serveurs ou agents
- Intéropérabilité
 - Description de problèmes
 - Redistributions de données entre les serveurs
- Check-pointing
 - E/S parallèles rapides
 - Garbage collection
- Scalabilité
 - Hiérarchie de serveurs/d'agents
- Aide à la décision/PSE
 - Choix automatique des méthodes
- Localisation de ressource
 - Matérielles et logicielles
- Ordonnancement
 - Ordonnancement on-line d'ordonnancements off-line
- Partage de serveurs entre des utilisateurs
 - Problèmes de sécurité
 - Lock/unlock, consistance de données, race conditions
- Evaluation de performances
 - Hétérogénéité
 - Systèmes batch
- Visualisation de données
 - Problèmes de scalabilité
- Dynamicité de la plate-forme
 - Localisation des ressources
 - Placement des agents/serveurs

RenPar – Avril 2002 – F. Desprez

Travaux à court terme



- Sécurité
 - Comptes et accès utilisateurs
 - Transferts de données
- Tolérance aux pannes
 - Serveurs ou agents
- Intéropérabilité
 - Description de problèmes
 - Redistributions de données entre les serveurs
- Check-pointing
 - E/S parallèles rapides
 - Garbage collection
- Scalabilité
 - Hiérarchie de serveurs/d'agents
- Aide à la décision/PSE
 - Choix automatique des méthodes
- Localisation de ressources
 - Matérielles et logicielles
- Ordonnancement
 - Ordonnancement on-line d'ordonnancements off-line
- Partage de serveurs entre des utilisateurs
 - Problèmes de sécurité
 - Lock/unlock, consistance de données, race conditions
- Evaluation de performances
 - Hétérogénéité
 - Systèmes batch
- Visualisation de données
 - Problèmes de scalabilité
- Dynamicité de la plate-forme
 - Localisation des ressources
 - Placement des agents/serveurs

RenPar – Avril 2002 – F. Desprez

On embauche !



- Stages DESS, DEA, Ecoles d'ingénieurs
- Thèses
- Postdocs ou ingénieurs experts sur les projets RNTL GASP, ACI Grid GridASP ou RNRT VTHD++

© 2000 Randy Glasbergen. www.glasbergen.com



"Stop whining — most people do their best work under pressure!"

RenPar – Avril 2002 – F. Desprez

DES QUESTIONS ?

RenPar – Avril 2002 – F. Desprez