

# Défis et perspectives *scientifiques* des grilles informatiques



Thierry PRIOL  
IRISA/INRIA

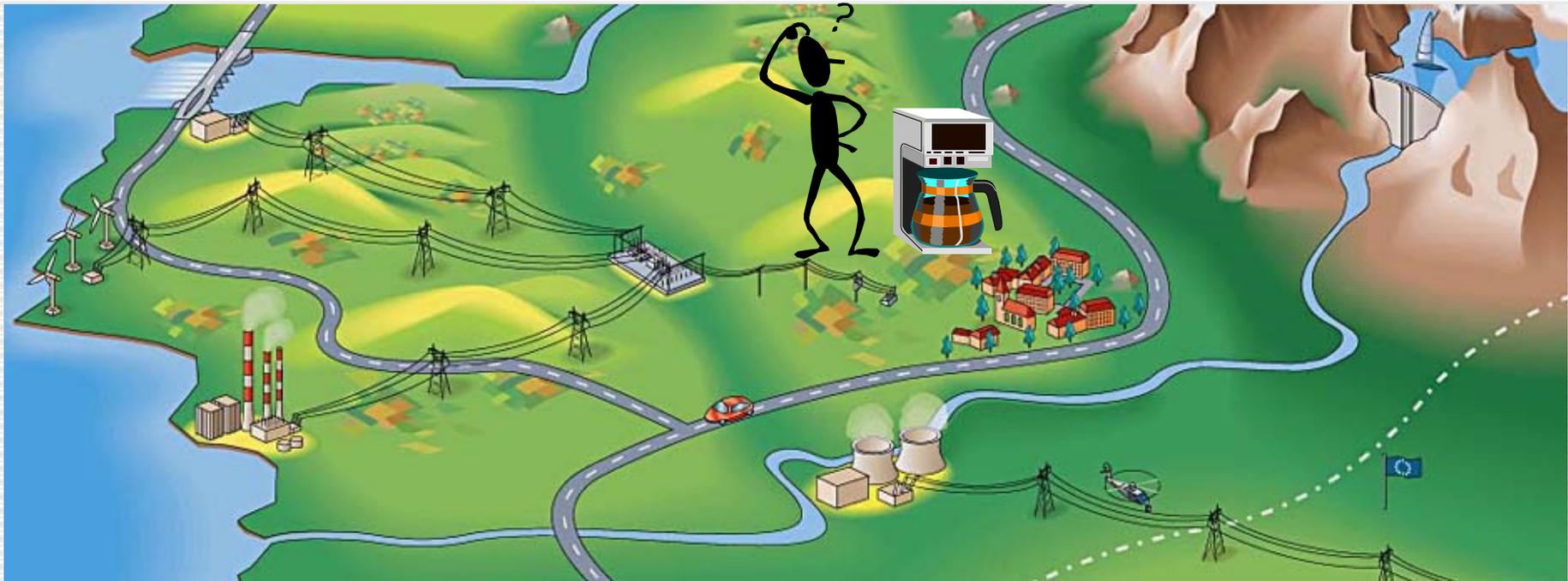
Thierry.Priol@inria.fr

*Contenu de la présentation*

- Concept de grille informatique
- Les types de grilles informatiques et leur déploiement
- Quelques défis
- Conclusions & perspectives



# Le concept de grille



Approche pour la distribution de la puissance électrique =  
le réseaux électrique et la haute-tension

# Le concept de grille informatique (GRID)



Approche pour la distribution de la puissance informatique =  
le réseau Internet et la haute-performance  
(parallélisme et distribution)

# Et ses différentes incarnations...



# Une tentative de classification



- Grille d'information
  - Partager la connaissance

■ Grille de stockage  
■ Stocker à grande échelle

# Systemes distribués ! \*

- Grille de calcul
  - Agréger la puissance de calcul

\* « A distributed system is a collection of independent computers that appear to the users of the system as a single computer ». A. Tanenbaum, Prentice-Hall, 1994

# Modèles de déploiement

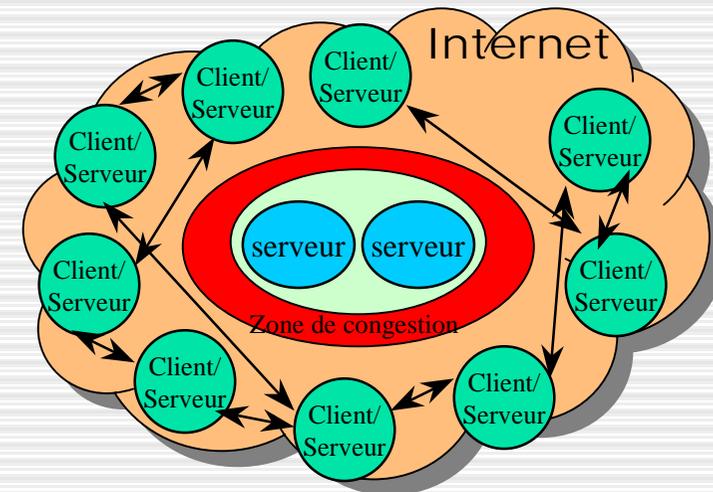
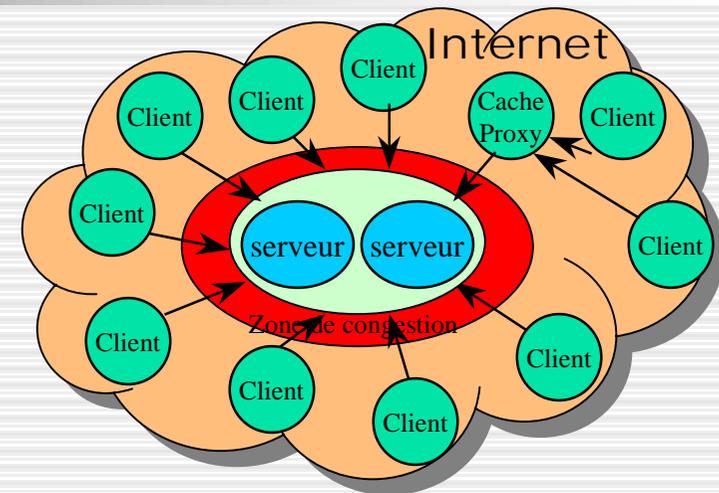


## Client-serveur

- Centralisé ou Distribué
- Utilisation de caches pour éviter la congestion
- Information centralisée

## ■ Pair-à-pair (P2P)

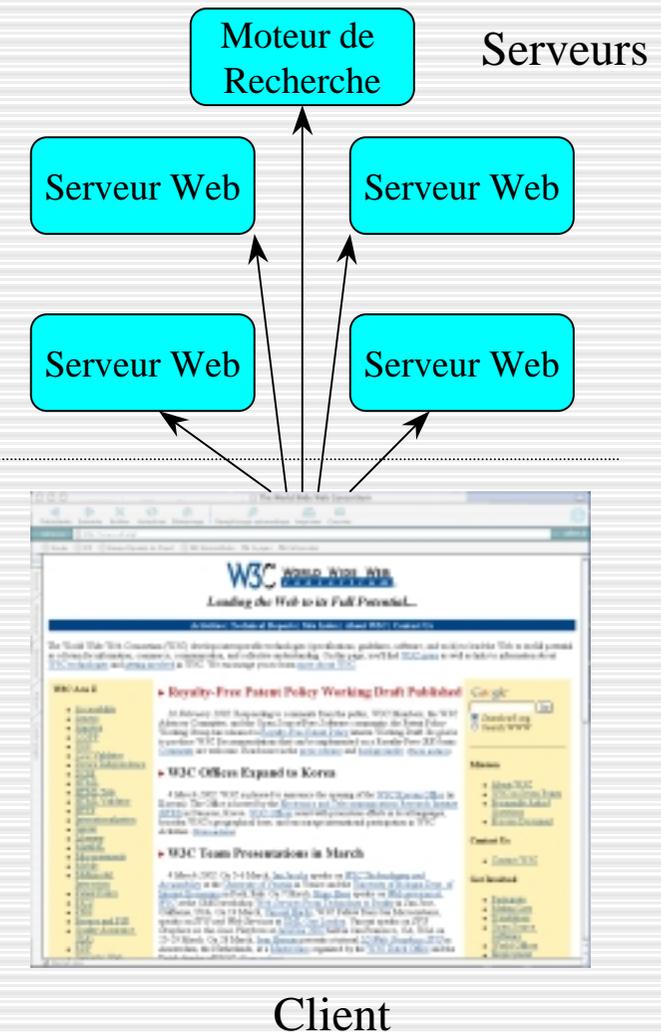
- Chaque pair est à la fois client et serveur
- Distribution de la charge dans le réseau (si parité !)
- Information distribuée



# Modèle client/serveur pour les grilles d'information : le cas des sites Web



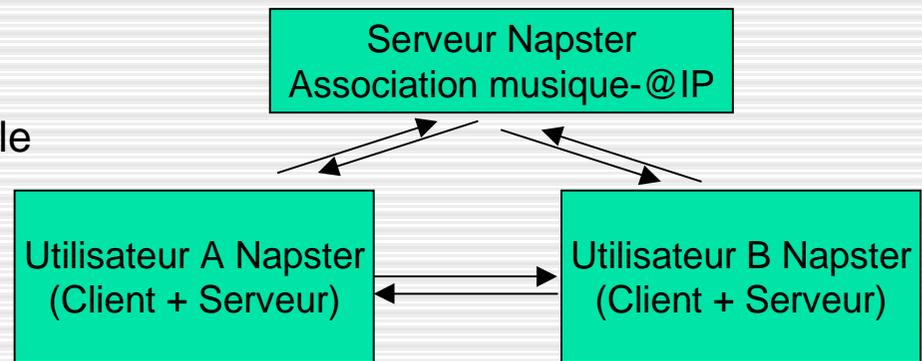
- Sans doute la première incarnation du concept de grille
- Accès à l'information
  - A partir d'une adresse http
  - A partir d'un moteur de recherche
- Transparence lors de l'accès à l'information
  - On ne sait pas toujours d'où vient l'information
- Moteurs de recherche qui ne passent pas à l'échelle
  - Centralisation
  - Ne captent qu'une partie de l'information publiée



# Modèle client/serveur pour les grilles de stockage : le cas Napster



- Entre le client/serveur et le P2P
  - Accès à des données via un site unique contenant un index
- Stockage de données
- Partage des données
  - Données « inaltérables »
  - Copies multiples sans aucun contrôle
- Limites de l'approche
  - Plutôt du client/serveur que réellement du P2P
  - Serveur « vulnérable »
    - Par les tribunaux...
    - Ou par d'autres...



# Modèle client/serveur pour les grilles de calcul : l'Internet Computing



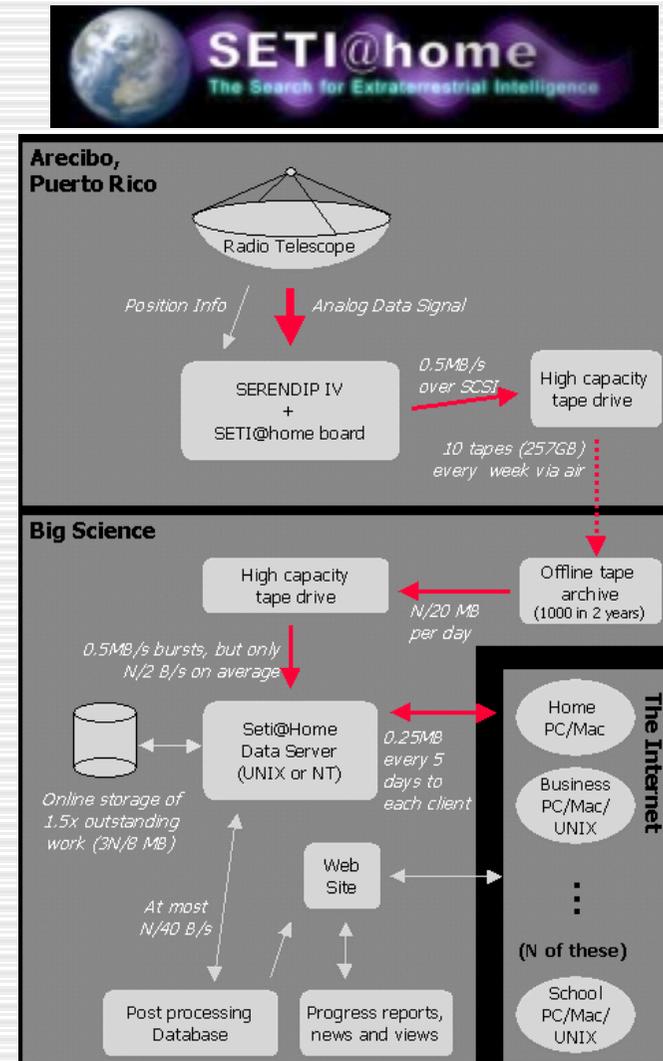
## ■ Principe

- Des millions de PC en attente...
- Récupération des cycles processeurs inutilisés (environ 47% en moyenne dans une entreprise\*) via un économiseur d'écran)

## ■ Exemples

- SETI@home (ce n'est pas du P2P !)
  - Recherche de signaux extra-terrestres
  - 33,79 Teraflop/s (à comparer aux 12,3 Teraflop/s de l'ordinateur le plus puissant au monde au LLNL !)
- Décrypton
  - Etablir la carte des 500 000 protéines du vivant
- RSA-155
  - Casser des codes cryptographiques

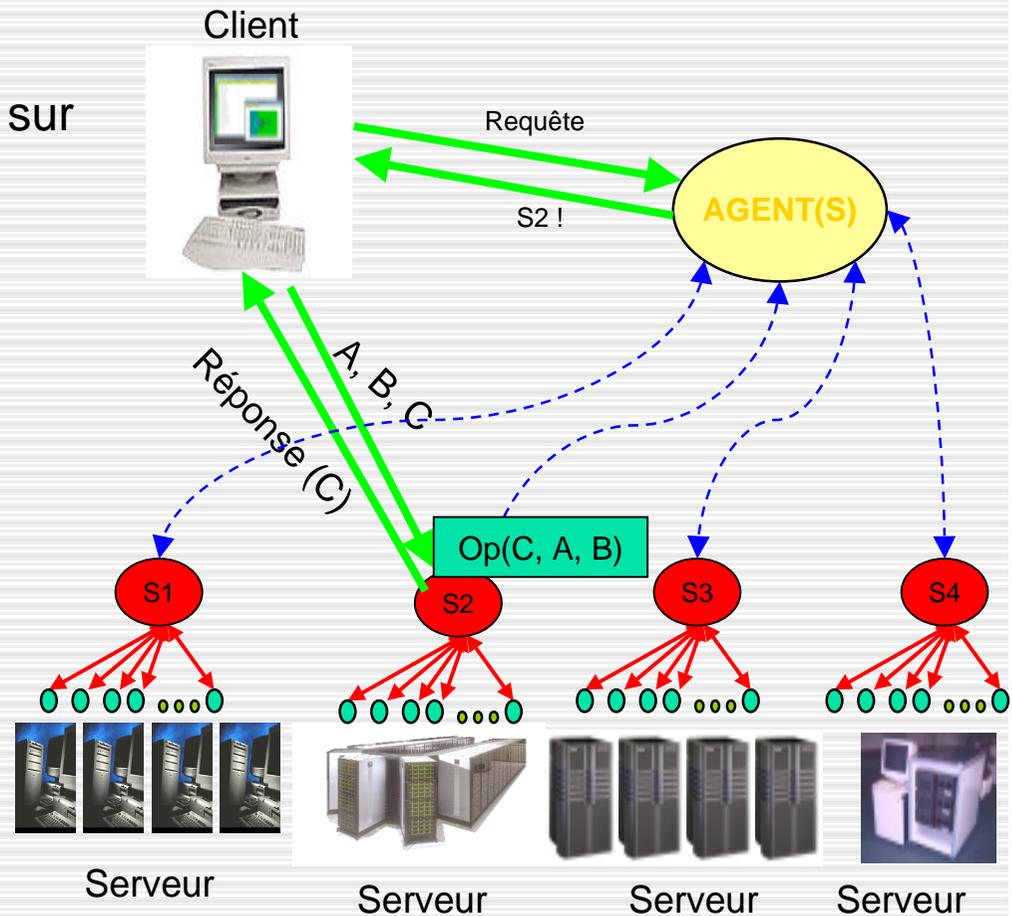
\* d'après une enquête d'Omni Consulting Group



# Modèle client/serveur pour les grilles de calcul : le *metacomputing*



- Principe
  - Acheter du service de calcul sur l'Internet
  - Service = applications pré-installées + calculateurs
- Exemples
  - Netsolve (Univ. Tennessee)
  - NINF (Univ. Tsukuba)
  - DIET (ENS Lyon/INRIA)



# Modèle client/serveur pour les grilles de calcul : le supercalculateur virtuel

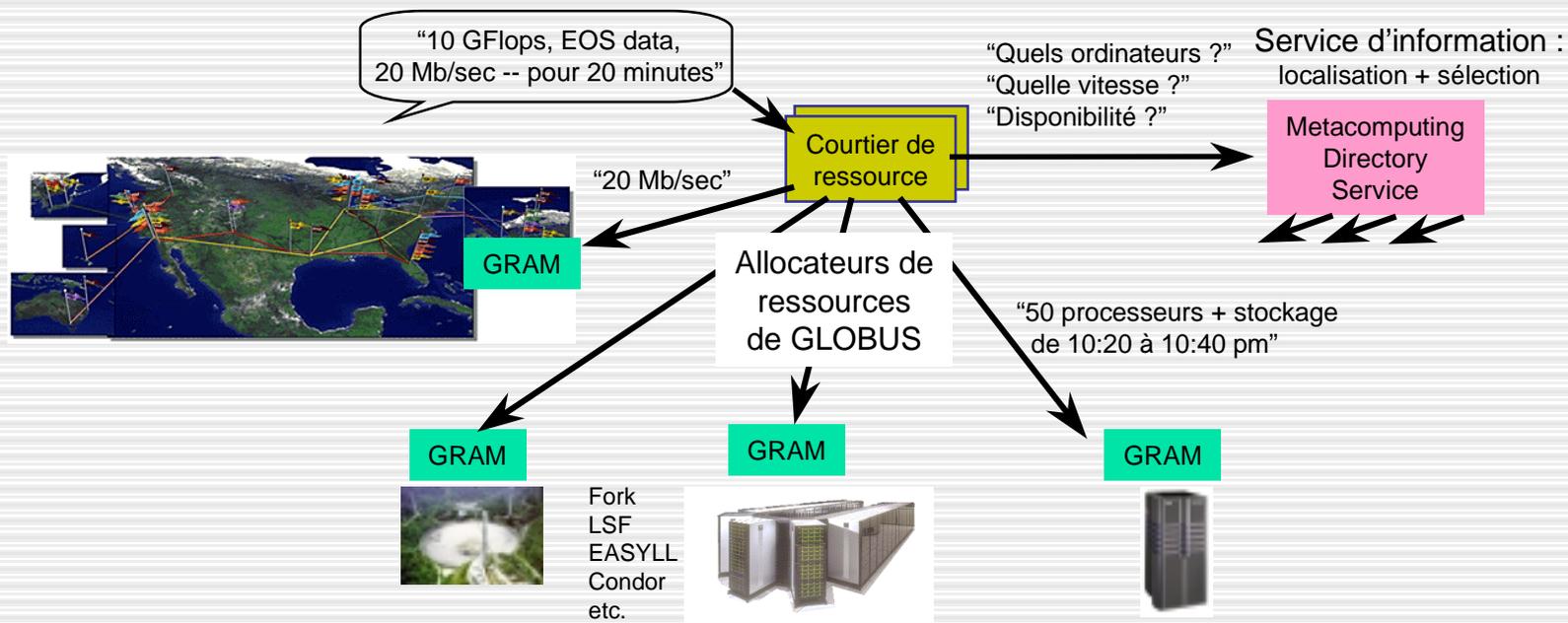


## ■ Principe

- Offrir un supercalculateur parallèle virtuel
- Faire exécuter ses applications sur des ressources distantes

## ■ Exemples

- Globus
- Légion
- Unicore

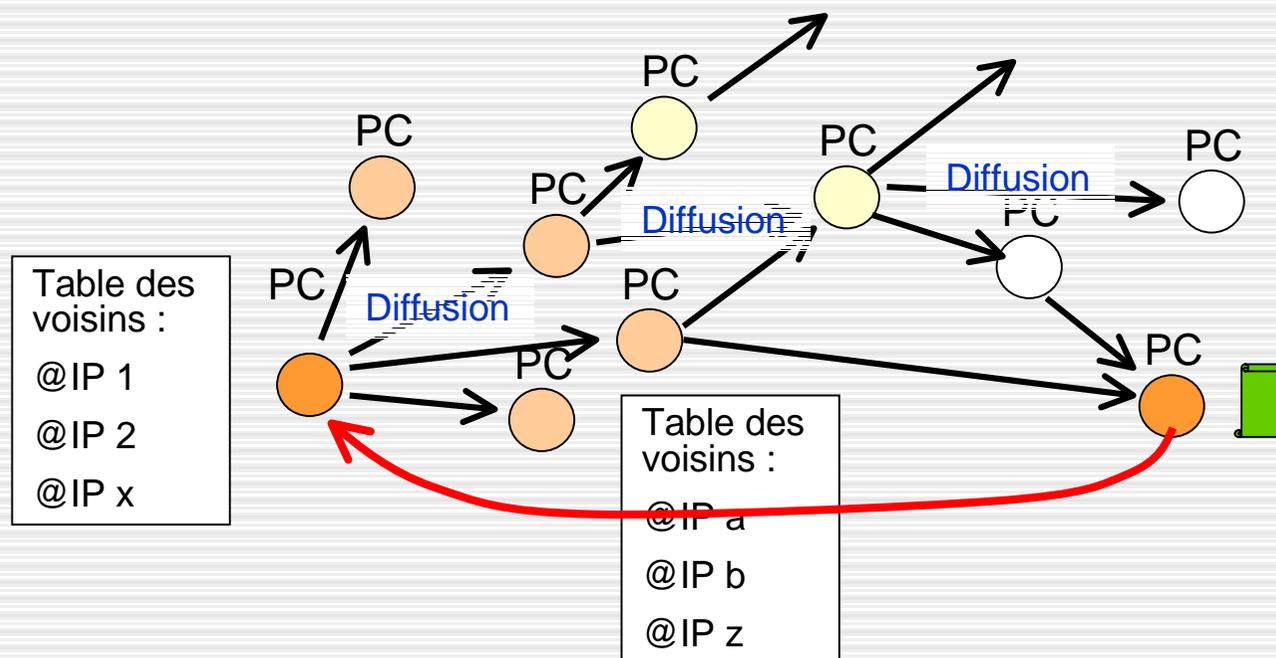


# Modèles de déploiement des grilles : le modèle distribué pair-à-pair



- Essentiellement utilisé dans pour les grilles de stockage
  - Ne pas être vulnérable
  - Pas d'état global
  - Découverte des ressources par diffusion

- Exemples:
  - Gnutella, Freenet
  - KaZaA, JXTA
  - OceanStore
  - FreeHaven



# En résumé



Défi



Déploiement	Client/serveur	P2P
Grilles /		
Grilles d'information	Sites Web Moteurs de recherche (Altavista, Google, ...)	?
Grilles de stockage	Napster Datagrid, ...	Gnutella, KaZaA, Freenet, JXTA, OceanStore, ...
Grilles de calcul	SETI@home, Decryphon Netsolve, Ninf, DIET Globus, Legion, Unicore, XtremWeb1, ...	?

# Quelques grands défis...

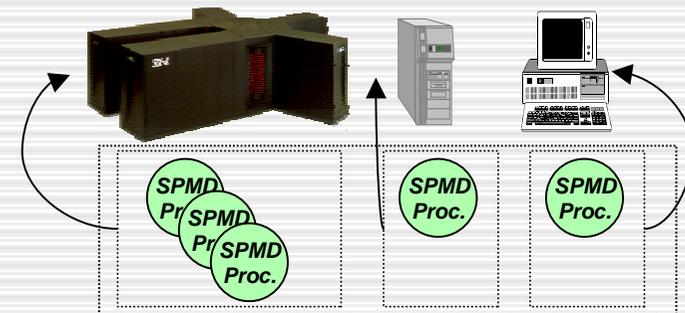


- La programmation des grilles
  - Algorithmique
  - Couplage de codes
  
- Middleware et systèmes
  - Internet computing
  - Metacomputing
  - Vers un Grid-aware OS ?

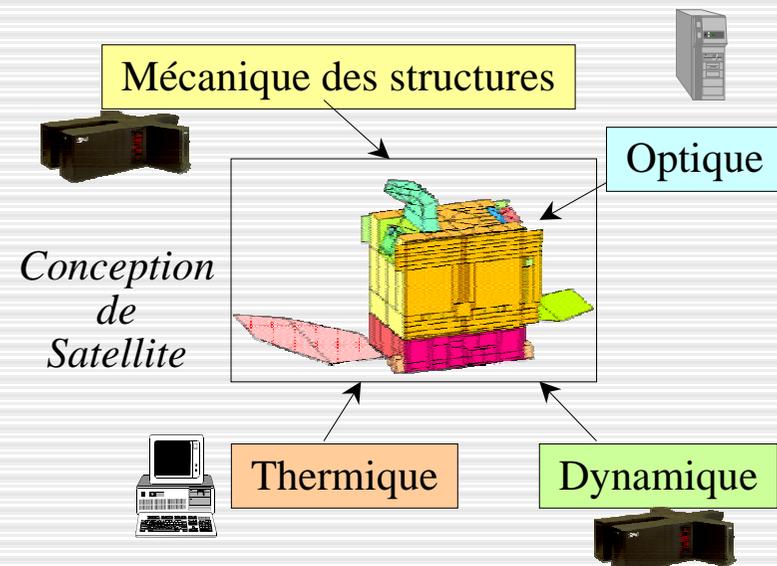
# Programmer les grilles de calcul



- Un champ applicatif vaste avec des besoins variés...
- Codes parallèles
  - Une grille de calcul est vue comme un calculateur parallèle virtuel (la genèse du Grid)
- Couplages de codes
  - Une application est un assemblage de plusieurs codes de calcul



*Application parallèle*



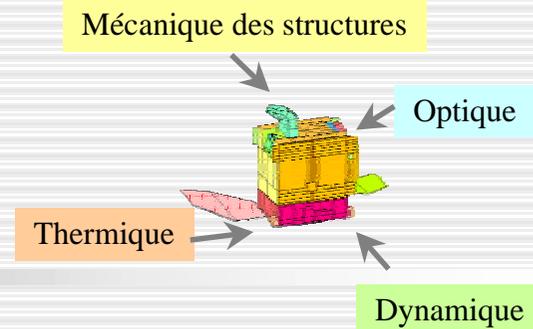
*Application de couplage*

# Vers une algorithmique des grilles de calcul

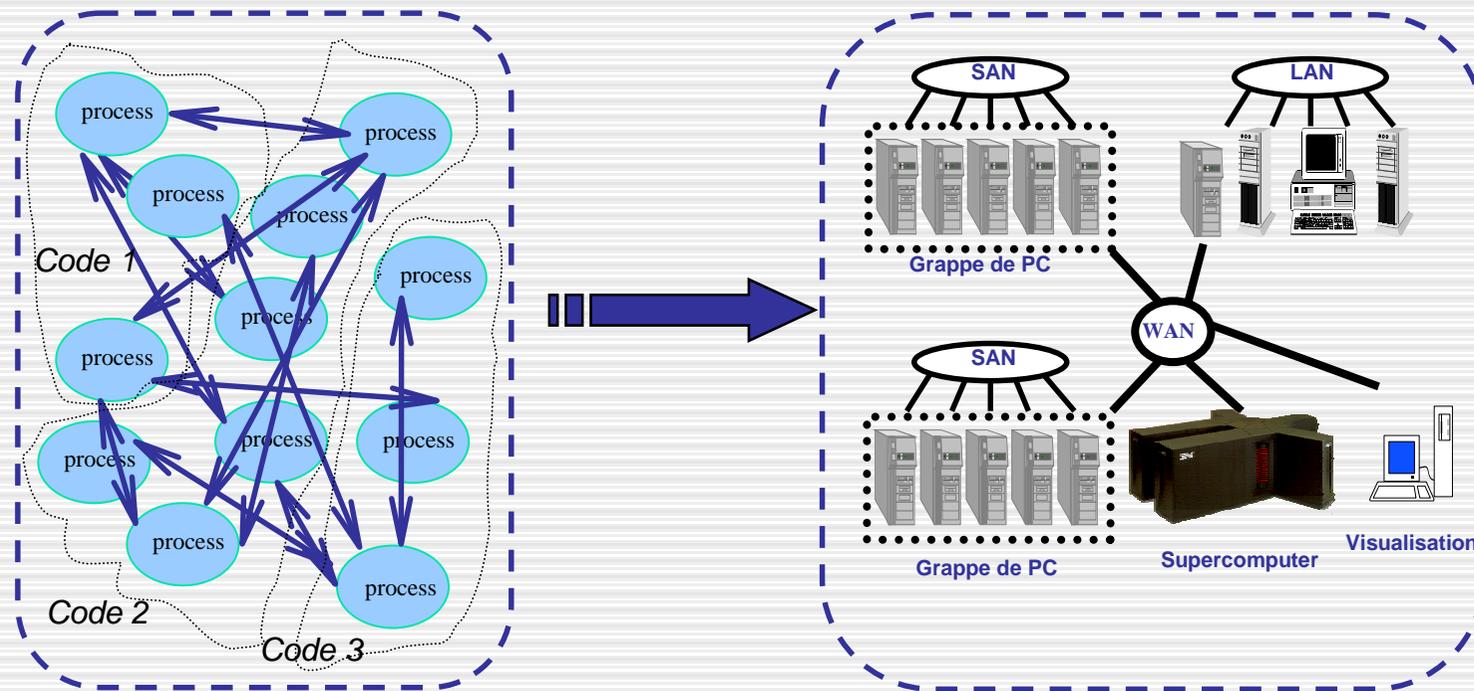


- Programmer une grille de calcul comme un calculateur parallèle
  - Internet computing & supercalculateur virtuel
- Repenser l'algorithmique parallèle
  - Essentiellement conçue pour des architectures parallèles régulières et à configuration statique
- Caractéristiques des grilles de calcul
  - Unités de calcul hétérogènes
  - Non-prédictibilité des performances des réseaux (SAN · WAN)
  - Aspect dynamique des ressources de calcul
- Problèmes et défis
  - Maîtriser la hiérarchie du parallélisme
  - Politique d'ordonnancement des calculs
  - Equilibrage dynamique par redistribution à la volée des données
  - Recouvrement calcul/communication

# Couplage de codes



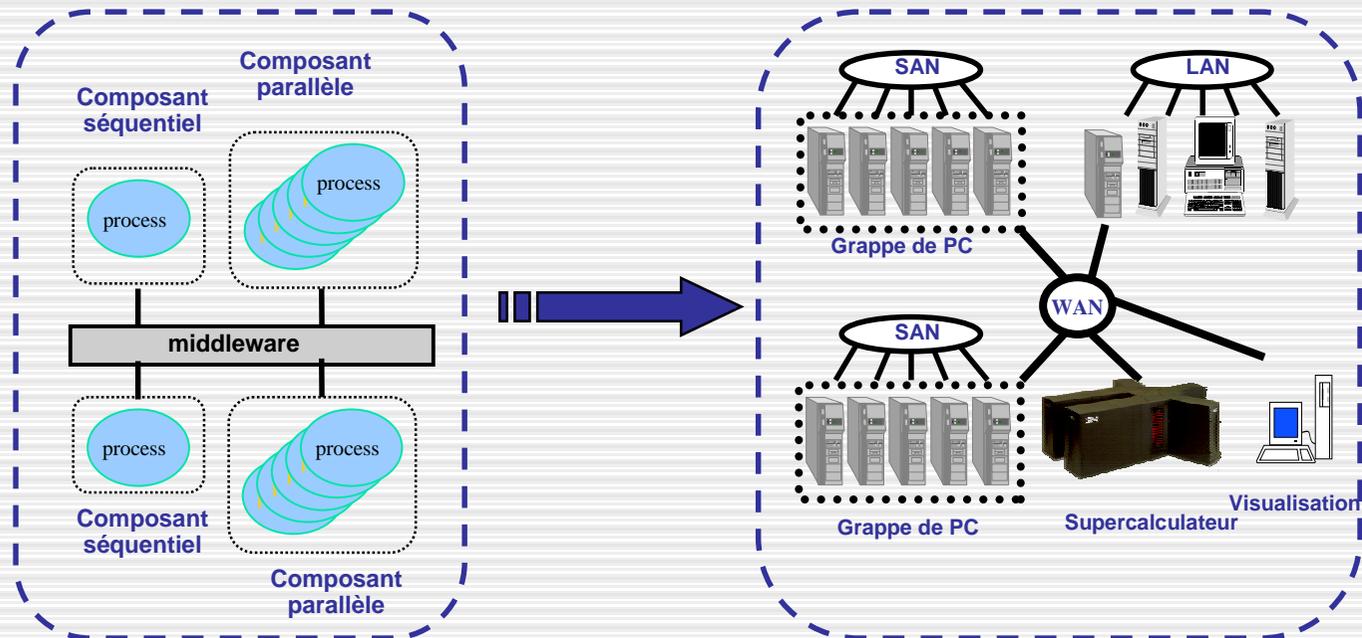
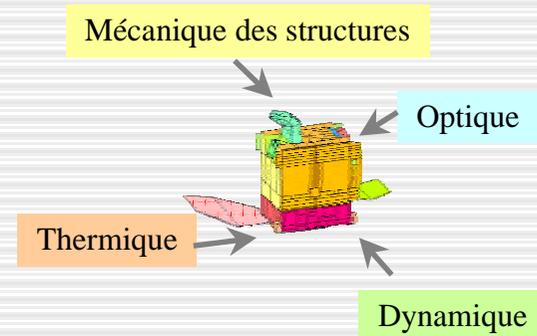
- Utilisation des exécutifs conçus pour la programmation parallèle
  - Une grille de calcul est un ordinateur parallèle virtuel, la programmation par échange de messages s'impose...



# Une approche plus moderne



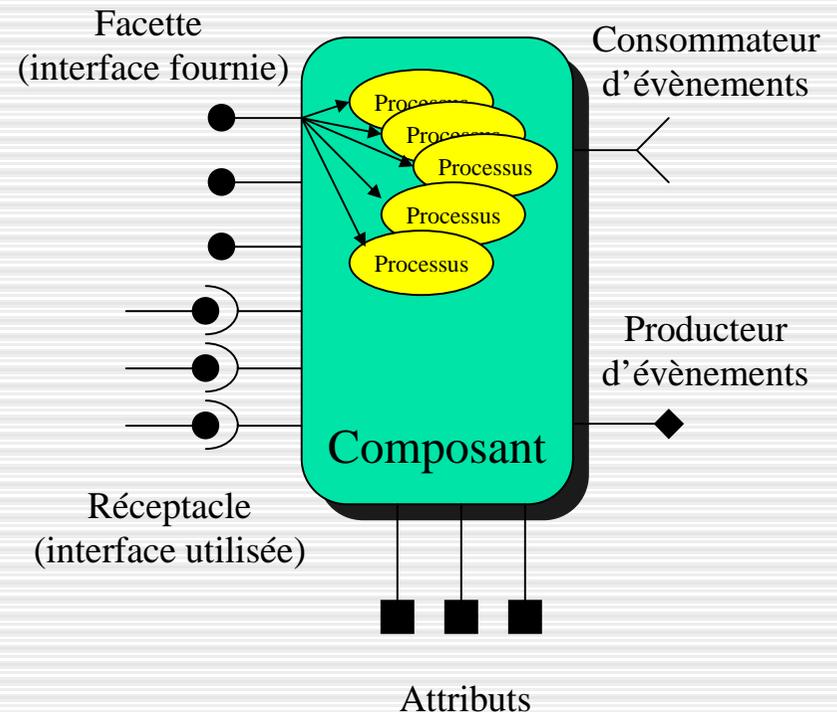
- Objets distribués/composants
  - Structuration de l'application
  - Encapsulation des codes
- Couplage de codes parallèles
  - Interconnexion des objets/composants ➔ un réel défi !



# La programmation par composants pour les grilles de calcul



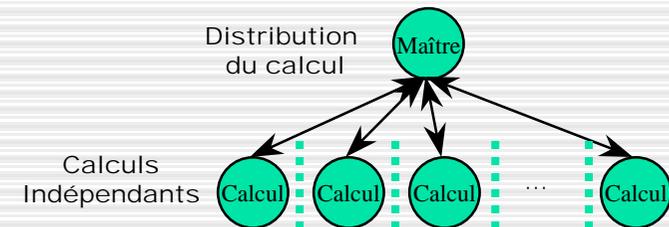
- Construction d'applications à l'aide de composants
- Problème
  - Encapsuler des codes parallèles au sein de composants
- Deux aspects qui s'opposent
  - Cacher la complexité dans un composant
  - Connecter des composants tout en autorisant des communications qui passent à l'échelle



# Les défis du « Internet computing »



- Cela marche bien pour un spectre étroit d'applications
  - *Embarrassingly parallel*



- Usage exclusivement « non-commercial »
  - Casser des codes cryptographiques (défi RSA-155)
  - Recherche de signaux extra-terrestre (SETI@home)
  - Décryptage du génome (Decryphon)
- Cette approche est-elle généralisable ?

# Quels sont les problèmes



- Elargir le spectre d'applications

- De vrais applications parallèles
- Autoriser la communication

- Sécurité

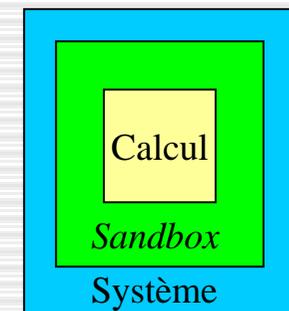
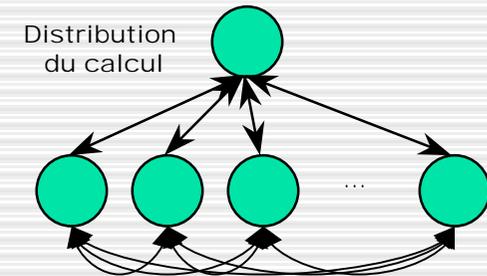
- Etes-vous prêt à laisser exécuter n'importe quoi sur votre PC ?
- Technique du *sandbox* (isolation du code de calcul)
  - Comment communiquer avec le monde lorsqu'on est isolé ?
- Peut-on croire dans les résultats fournis par quelqu'un que l'on ne connaît pas ?

- Parité

- Cela marche si tout le monde joue le même jeu...

- Modèle de déploiement

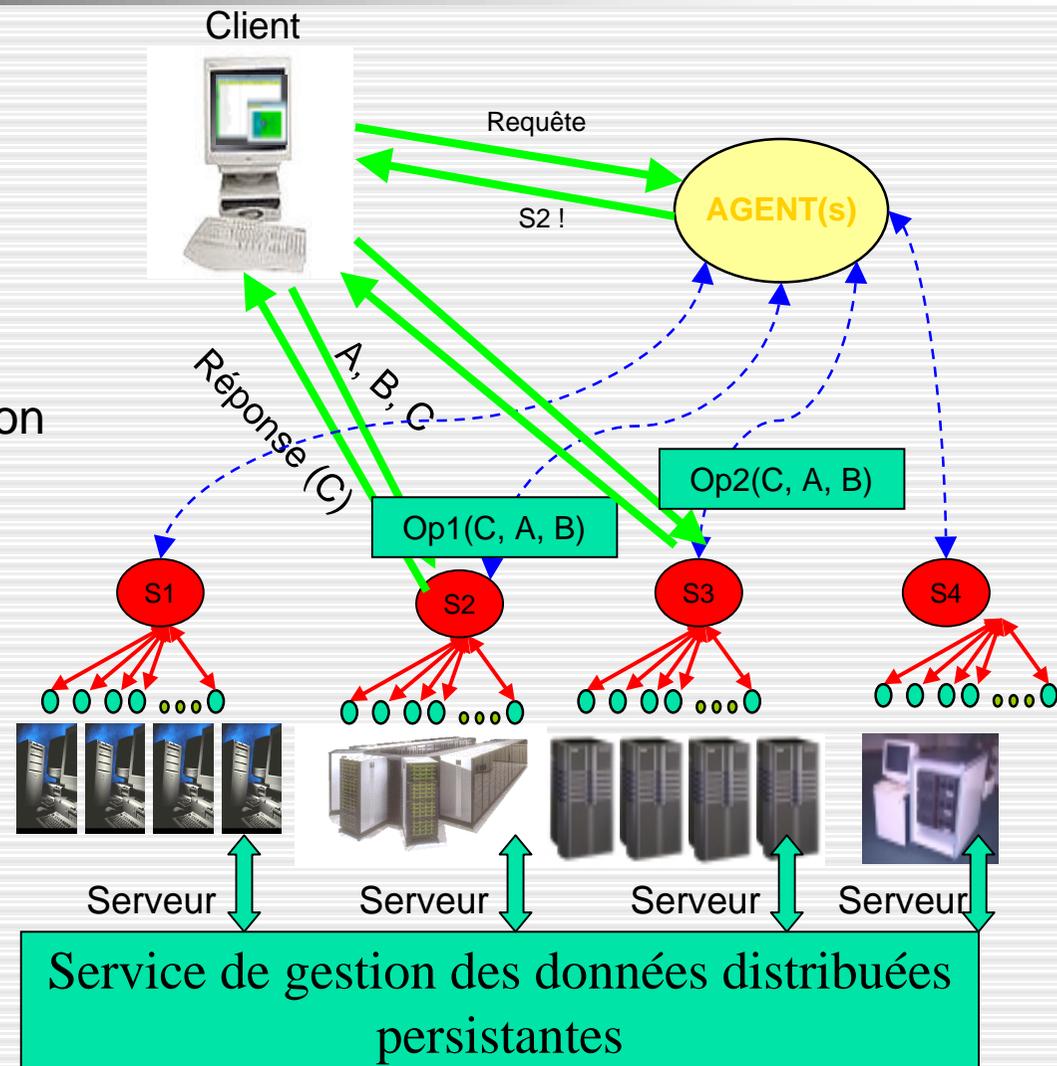
- Essentiellement client/serveur
- A terme, nécessité du P2P



# Les défis du metacomputing



- Stockage des données
  - Pour éviter les transferts multiples entre client et serveurs
  - Gestion des données distribuées et redistribution
- Sécurité
  - Dans les transferts
  - Dans les calculs
- Modèle de déploiement
  - P2P



# Vers de nouveaux systèmes d'exploitation *Grid-aware* ?



- Approches essentiellement « middleware »
  - Une bonne approche dans un monde hétérogène
- Vers un monde hétérogène ou homogène ?
  - Prédominance des PC et de quelques OS
  - Des supercalculateurs vers les grappes de PC
- Système d'exploitation
  - Gestion des ressources
- Vers un *Grid-aware OS* ?
  - Gestion des ressources dans une grille
    - Mémoire, disque, processeur
  - De nombreux obstacles
    - Sécurité, tolérance aux fautes, effet de bord, ...
  - Mais aussi des avantages...
    - Simplicité d'utilisation

# Et les autres défis (horizontaux)



- Interopérabilité
- Sécurité
  - Maîtriser les architectures de sécurité
  - Sécurité et P2P...
- Passage à l'échelle (*Scalability*)
- Tolérance aux fautes
  - Rendre transparent la gestion de la tolérance aux fautes
- Hétérogénéité des réseaux
  - SAN, LAN, WAN, ...
- Humain !
  - Faire collaborer des chercheurs d'horizons différents

# L'ACI GRID et les défis...



- Grilles de calcul
  - Internet Computing
    - CGP2P (F. Cappello, LRI/CNRS)
  - Metacomputing
    - ASP (F. Desprez, ENS Lyon/INRIA)
- Programmation des Grilles
  - Algorithmique:
    - GRID2 - Thème 3 (J-L. Pazat, IRISA/INSA)
    - TAG (S. Genaud, LSIIT)
    - ANCG (N. Emad, PRISM)
  - Composants:
    - RMI (C. Pérez, IRISA/INRIA)
    - CONCERTO (Y. Maheo, VALORIA)

# En conclusion



- Une grande variété dans les grilles informatiques
  - Pas de nouvelle thématique
    - Mais plutôt une adaptation de l'existant
    - Une « combinaison » de technologies du parallélisme et du distribué pour les grilles de calcul
  - Une approche nécessairement pluridisciplinaire
    - Système, réseaux, sécurité, applications, ...
  - Ne pas seulement encourager les aspects purement calcul...
- Bien identifier ce qui est technologique
  - Les effets de mode...

De ce qui est plus fondamental

  - Les nouveaux concepts...
- Encourager toutes recherches visant à utiliser des réseaux à grande échelle pour des applications innovantes...
  - Ne pas se concentrer sur une seule approche