Interactions entre recherche sur les grilles et grilles de production

Ordonnancement, modélisation et simulation

Frédéric Suter

5 décembre 2008

Recherche sur les grilles vs. Grilles de production

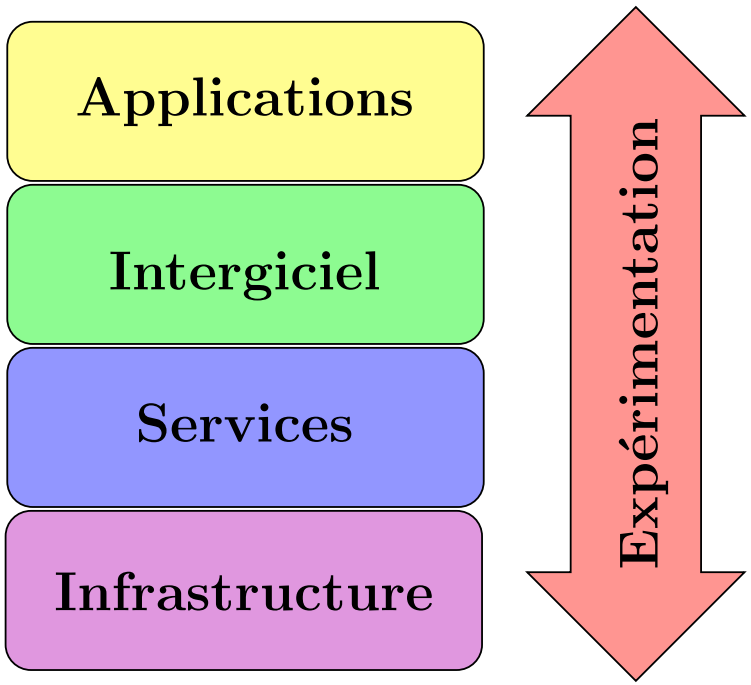
Grilles de production

- Objectif : faisabilité
- Moyens : boîtes à outils
- Exécuter les applications d'aujourd'hui

Recherche sur les grilles

- Objectif : performances
- Moyens : algorithmes
- Préparer les environnements de demain

Une vision de la recherche sur les grilles



- Structuration des applications
  - Algorithmique *Out-of-Core*
- Gestion transparente des ressources
  - Ordonnancement des calculs
  - Prédiction de performances
- Validation expérimentale
  - Simulation « réaliste » d'applications parallèles

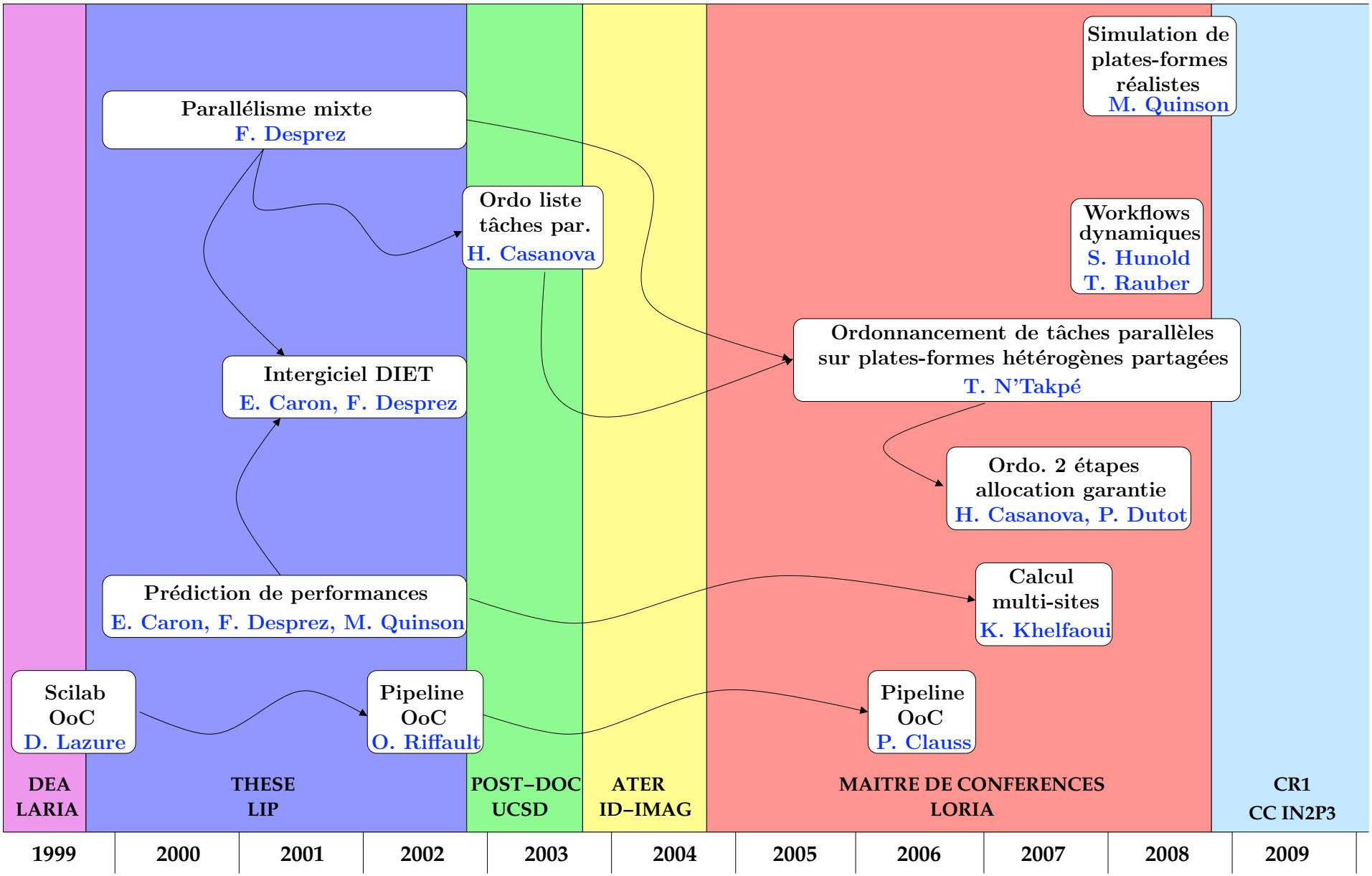
Mots-clés

- Algorithmique et environnements parallèles et distribués

Mon Parcours

1998-1999	DEA d'informatique fondamentale	LaRIA
1999-2002	Doctorant, ÉNS Lyon	LIP
2002-2003	Post-doctorant à l'étranger	UCSD
2003-2004	ATER, Univ. J. Fourier, Grenoble	LIG
2004-2008	Maître de Conférences, Univ. H. Poincaré, Nancy	LORIA
2008-	CR1, CNRS	CC IN2P3

Thématiques de recherches abordées



Plan

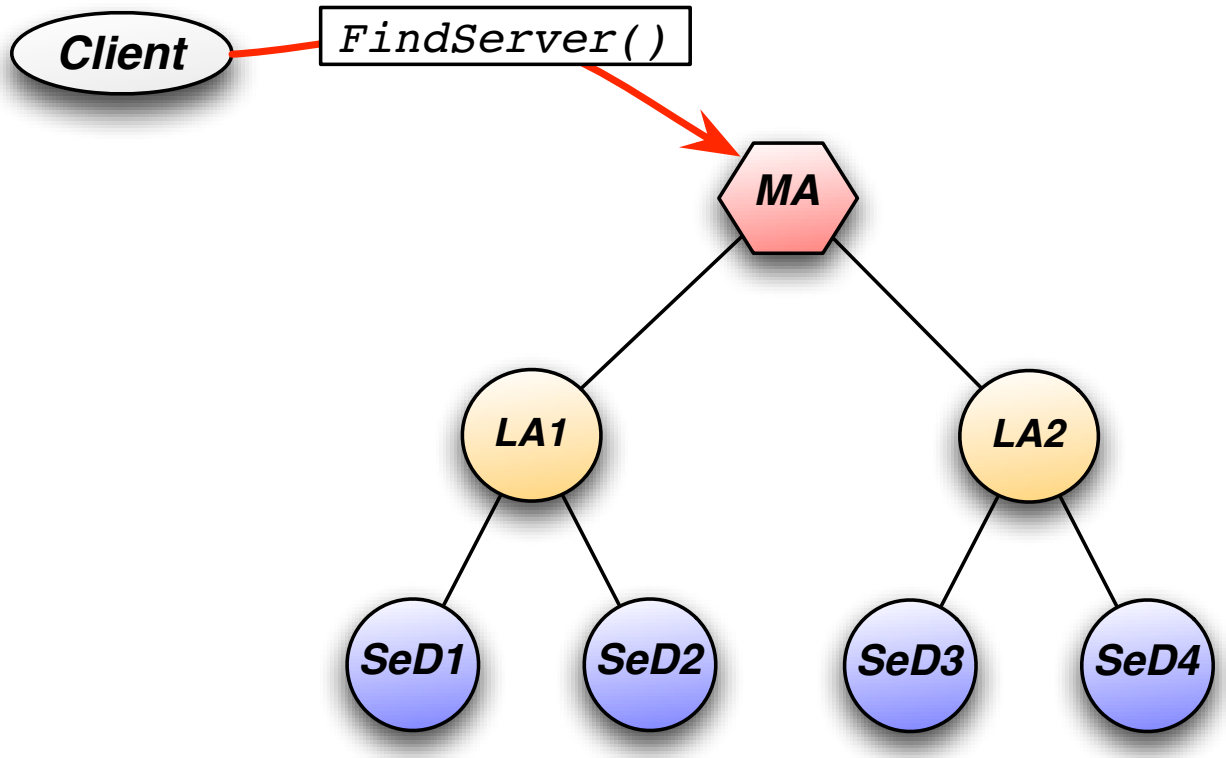
- Introduction et parcours
- Activités de recherche passées
  - DIET et prédiction de performances
  - Algorithmique Out-of-Core
  - Ordonnancement de tâches modelables
  - Simulation « réaliste »
- Activités de recherche futures
  - Ordonnancement
  - Modélisation
  - Simulation
- Projets en cours
- Autres idées à discuter

DIET : un intergiciel hiérarchique

Distributed Interactive Engineering Toolbox

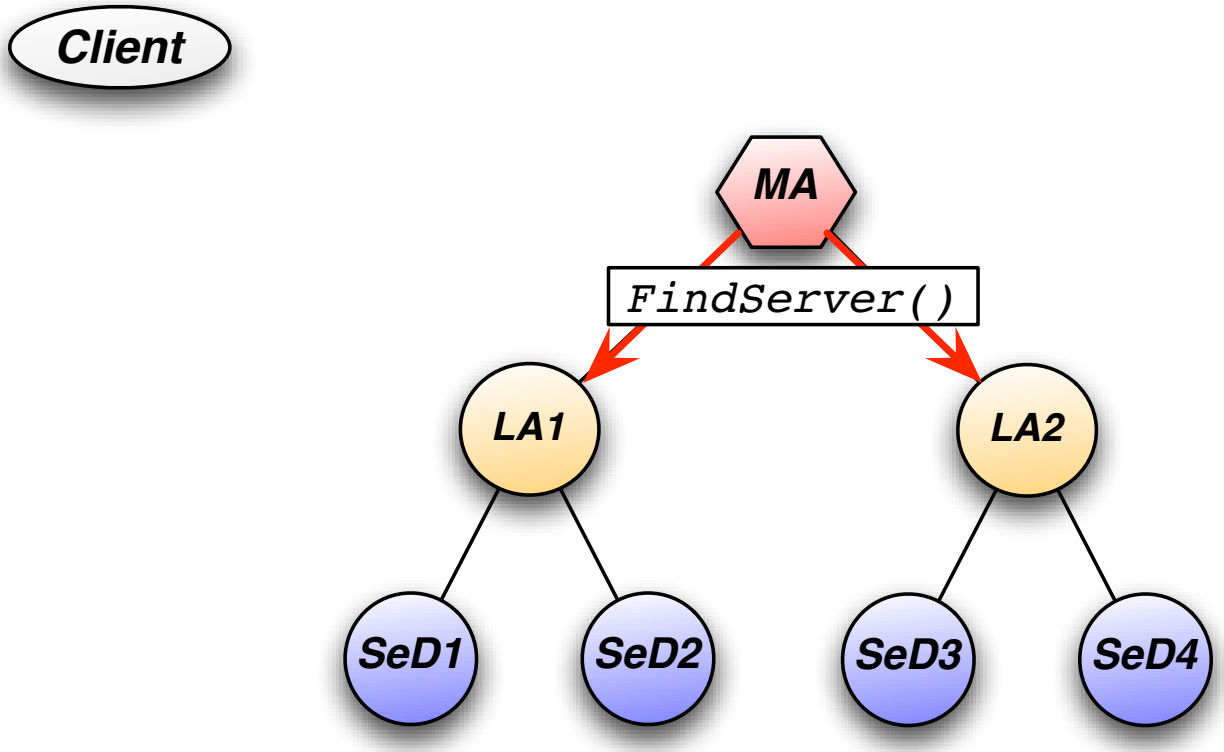
- Projet démarré en 2000 (j'y étais :))
- Développé au LIP ENS Lyon
- Approche GridRPC
- Retenu pour le projet Décrypthon
- Pleins d'outils associés
  - Déploiement (GoDiet)
  - Visualisation (VizDiet)
  - Création / exécution / Monitoring de workflow
  - Réservation de ressources (GRUDU/Dashboard)
  - ...
- Pour plus d'informations
  - `http://graal.ens-lyon.fr/DIET`

Example de fonctionnement



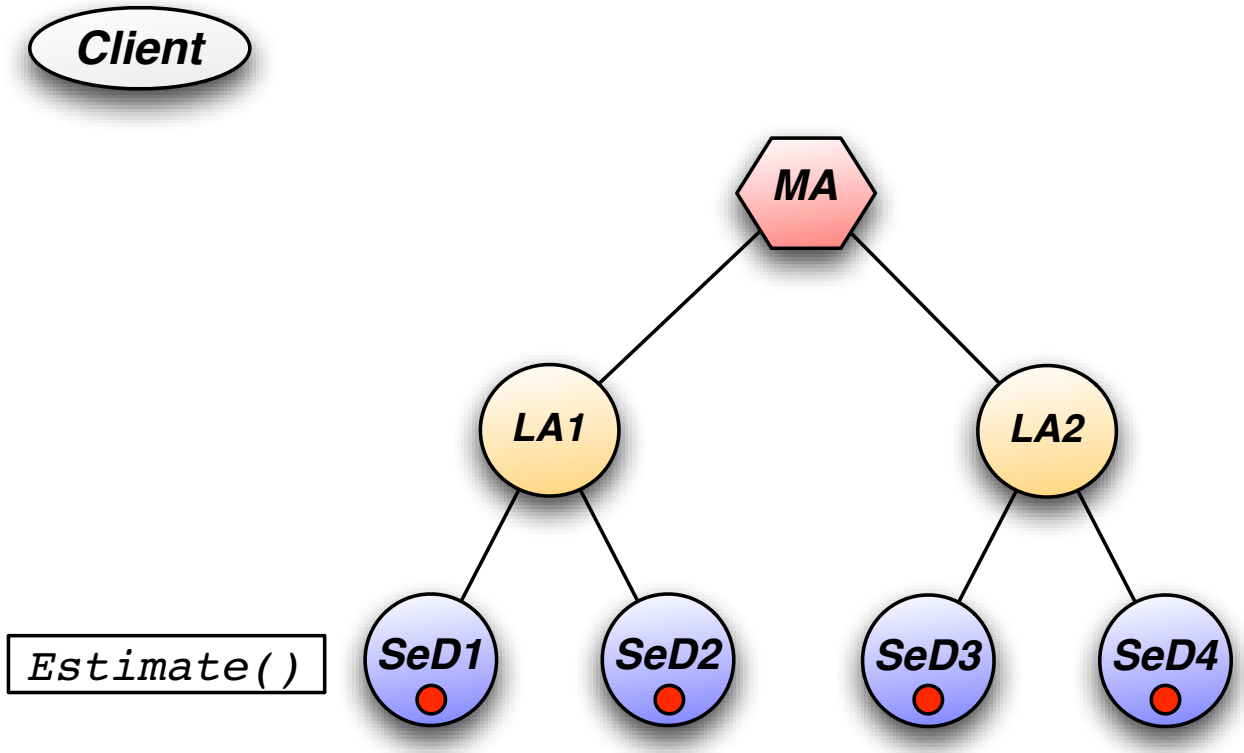
©Benjamin Depardon

Example de fonctionnement



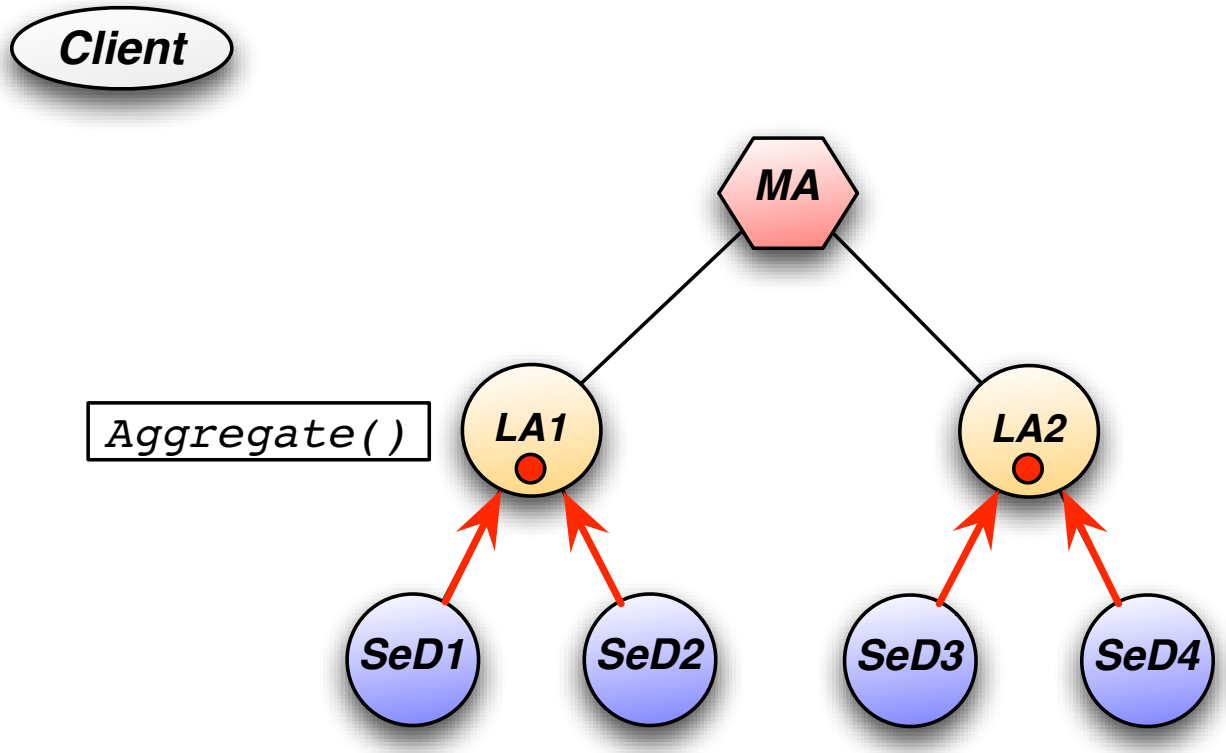
©Benjamin Depardon

Example de fonctionnement



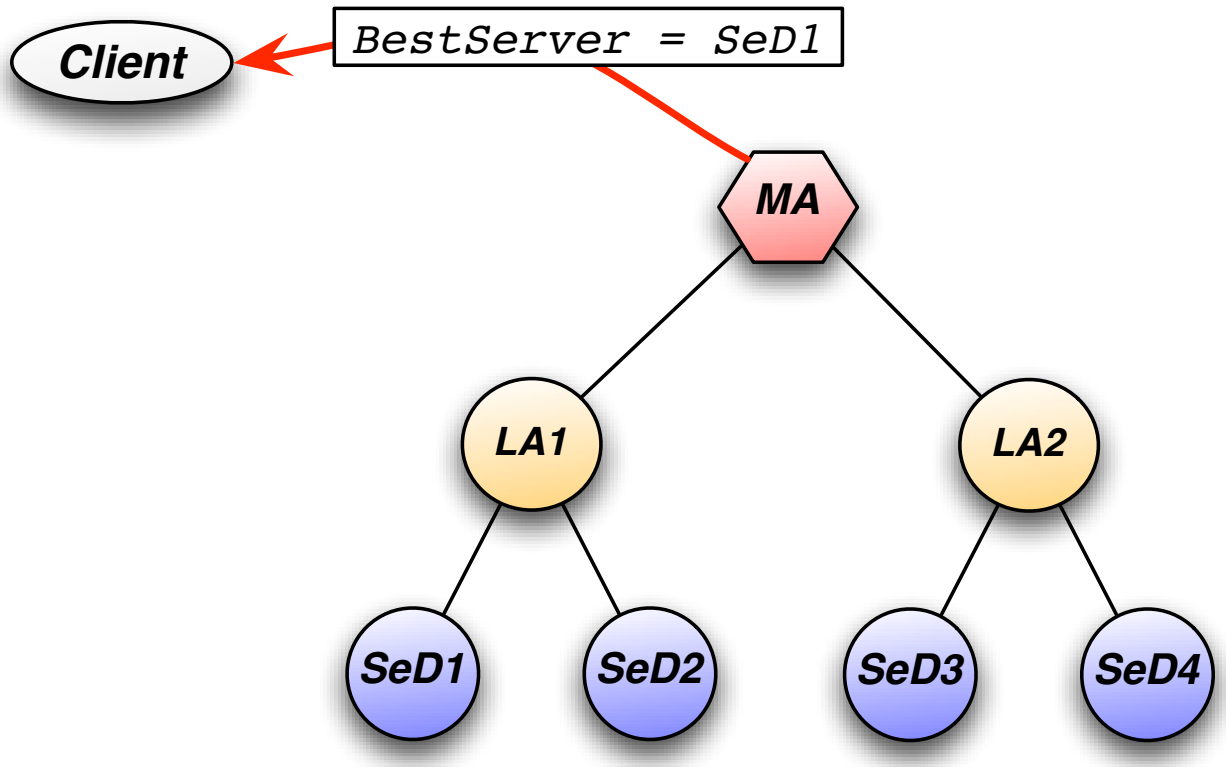
©Benjamin Depardon

Example de fonctionnement



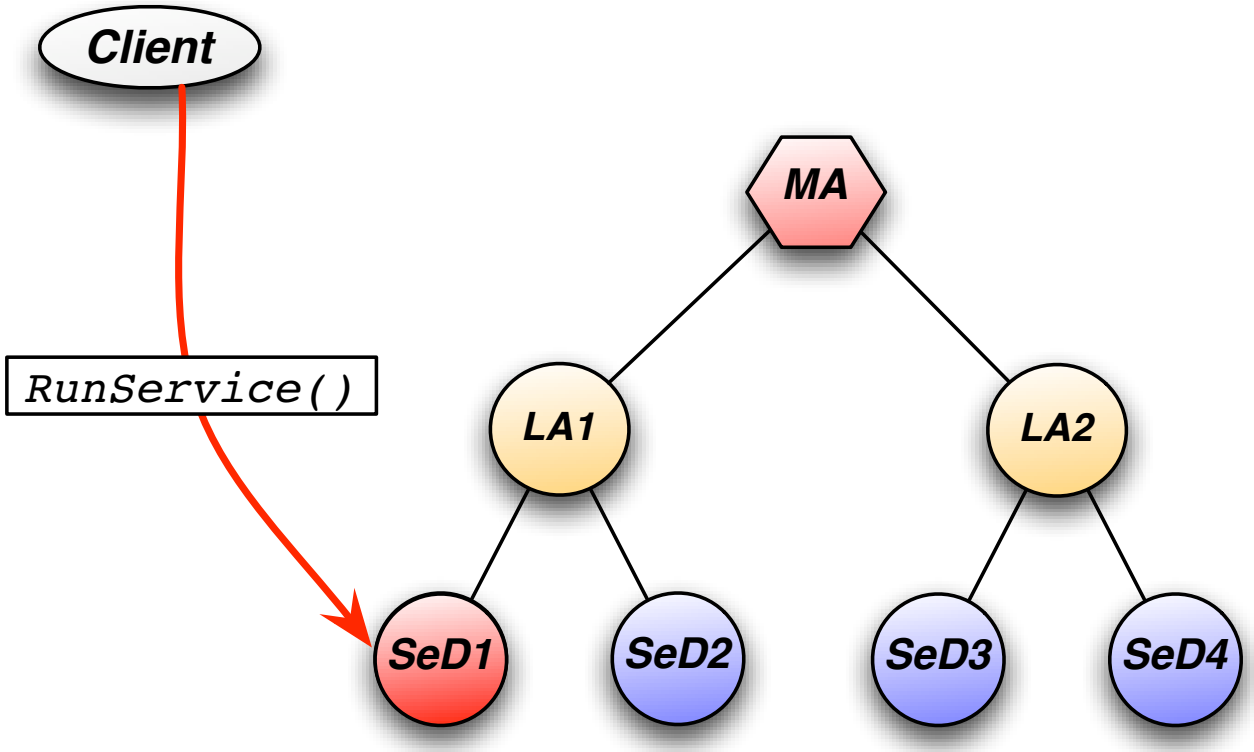
©Benjamin Depardon

Example de fonctionnement



©Benjamin Depardon

Example de fonctionnement



©Benjamin Depardon

Prédiction de performances dans DIET

FAST

- Module de prédiction de performances
- Macro-benchmarking + interface NWS
- Routines séquentielles uniquement

Extension aux routines parallèles

- Algèbre linéaire dense (ScaLAPACK)
- Approche par analyse de code
  - Calcul : nombre et paramètres d'appels aux routines séquentielles
  - Communication : schéma de communication et paramètres réseau (latence et bande passante)
- Deux fonctions modélisées
  - Produit de matrices
  - Résolution triangulaire

  $\longrightarrow$ Briques de base de la factorisation LU

Plan

- Introduction et parcours
- Activités de recherche passées
  - DIET et prédiction de performances
  - Algorithmique Out-of-Core
  - Ordonnancement de tâches modelables
  - Simulation « réaliste »
- Activités de recherche futures
  - Ordonnancement
  - Modélisation
  - Simulation
- Projets en cours
- Autres idées à discuter

Calcul Out-of-Core et pipeline

Calcul Out-of-Core

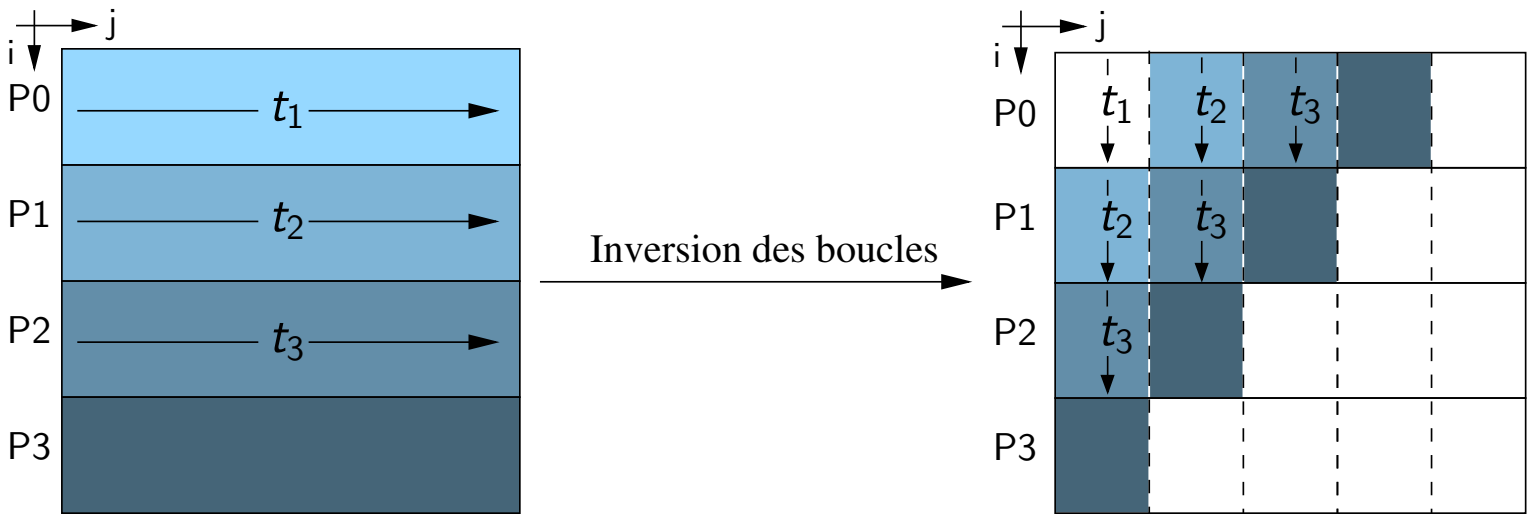
- Nécessaire lorsque la taille des données excède celle de la mémoire
  - Approche Système → améliorer le gestionnaire de swap
  - Approche Algorithmique → améliorer les accès aux données de l'algorithme
  - Point clé : saturation du disque
- 3 étapes
  - Charger un bloc du disque vers la mémoire
  - Calculer dessus
  - Récrire le bloc sur le disque

Pipeline

- Permet le recouvrement calcul/communication
  - Point clé : taille de bloc (dépend du ratio calcul/communication)
- 3 phases
  - Synchronisation (initialisation)
  - Chargement(préparation du régime permanent)
  - Régime permanent (maximum de parallélisme)

Application aux algorithmes par vagues

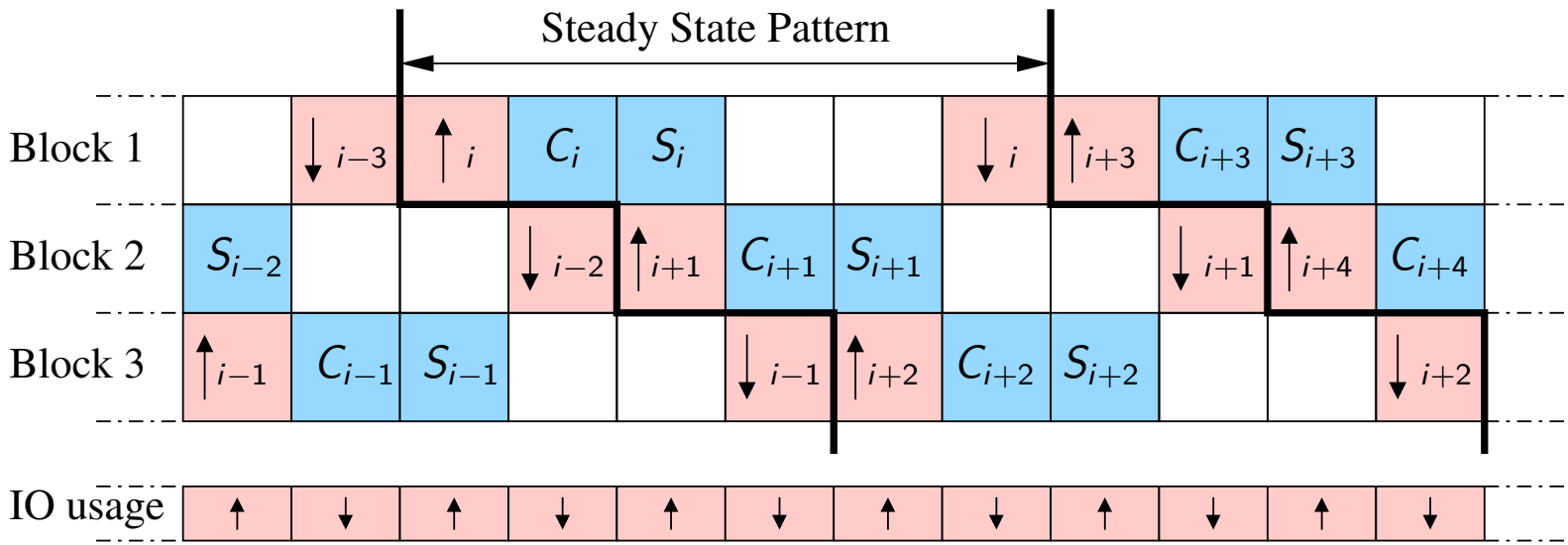
- Appliquer un traitement sur les différents éléments d'une matrice
  - Implique les voisins de l'élément
- Utilisés dans plein d'applications (convolution, Sweep3D, filtres d'images, ...)
- Réorganisation des boucles pour exhiber du parallélisme
  - Macro-pipelining
- Inverser les boucles FOR i et FOR j



Recouvrement des calculs et des comms. par les I/O

Stratégie à 3 blocs mémoire

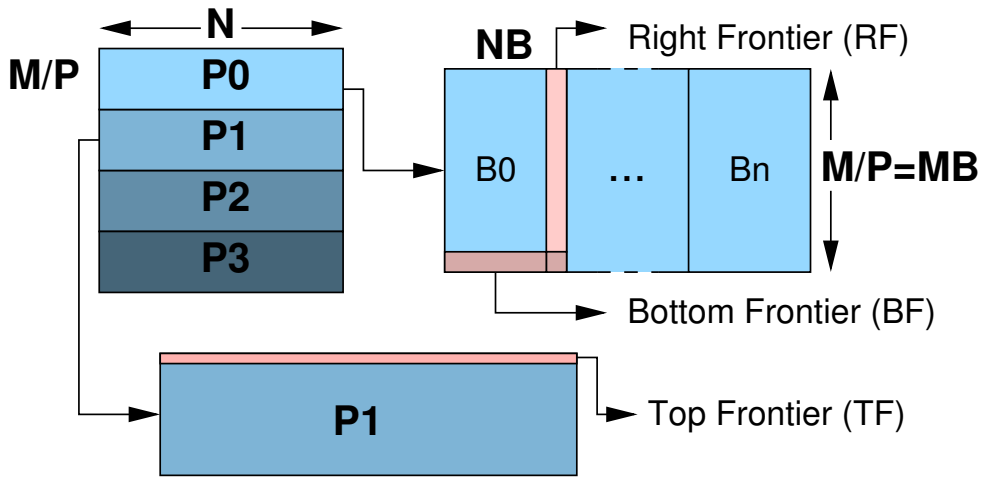
- Usage cyclique
  - Opération I/O sur le premier
  - Calcul sur le second
  - Communication du troisième



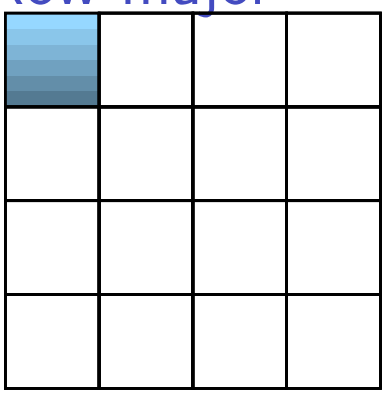
- Sature le disque

Problèmes de *layout* des données

Motifs d'accès



Row-major →

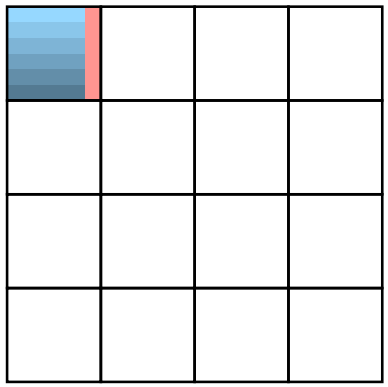


→ Block

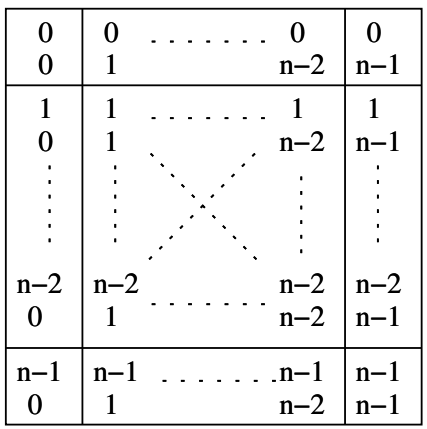
- Évite les petites lectures

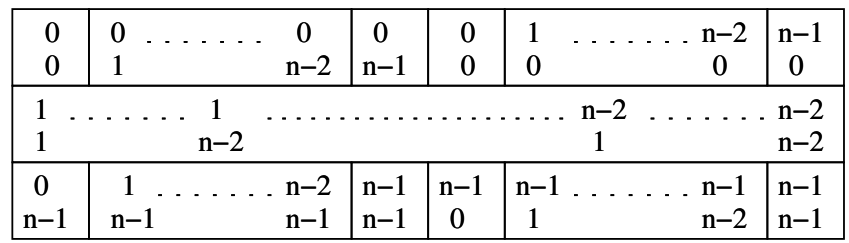
Problèmes de *layout* des données (2)

Reste le problème des lectures non contiguës



Proposition de layout optimisé





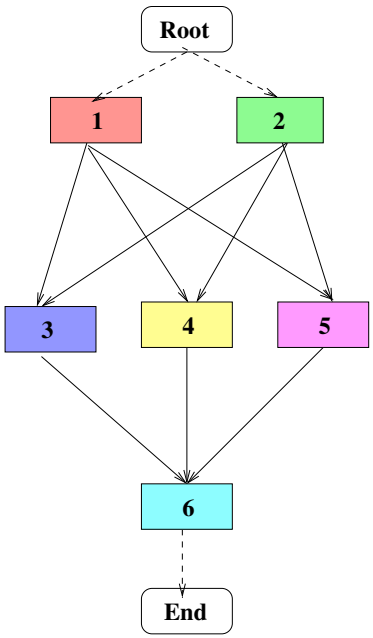
Plan

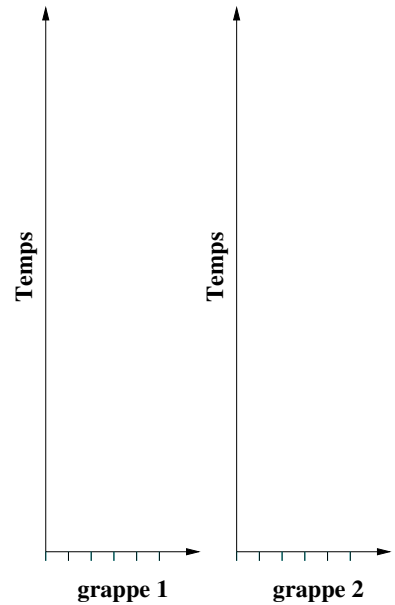
- Introduction et parcours
- Activités de recherche passées
  - DIET et prédiction de performances
  - Algorithmique Out-of-Core
  - Ordonnancement de tâches modelables
  - Simulation « réaliste »
- Activités de recherche futures
  - Ordonnancement
  - Modélisation
  - Simulation
- Projets en cours
- Autres idées à discuter

Ordonnancement de tâches modelables

Problème

Ordonnancer un graphe de tâches modelables sur une plate-forme multi-grappes hétérogène





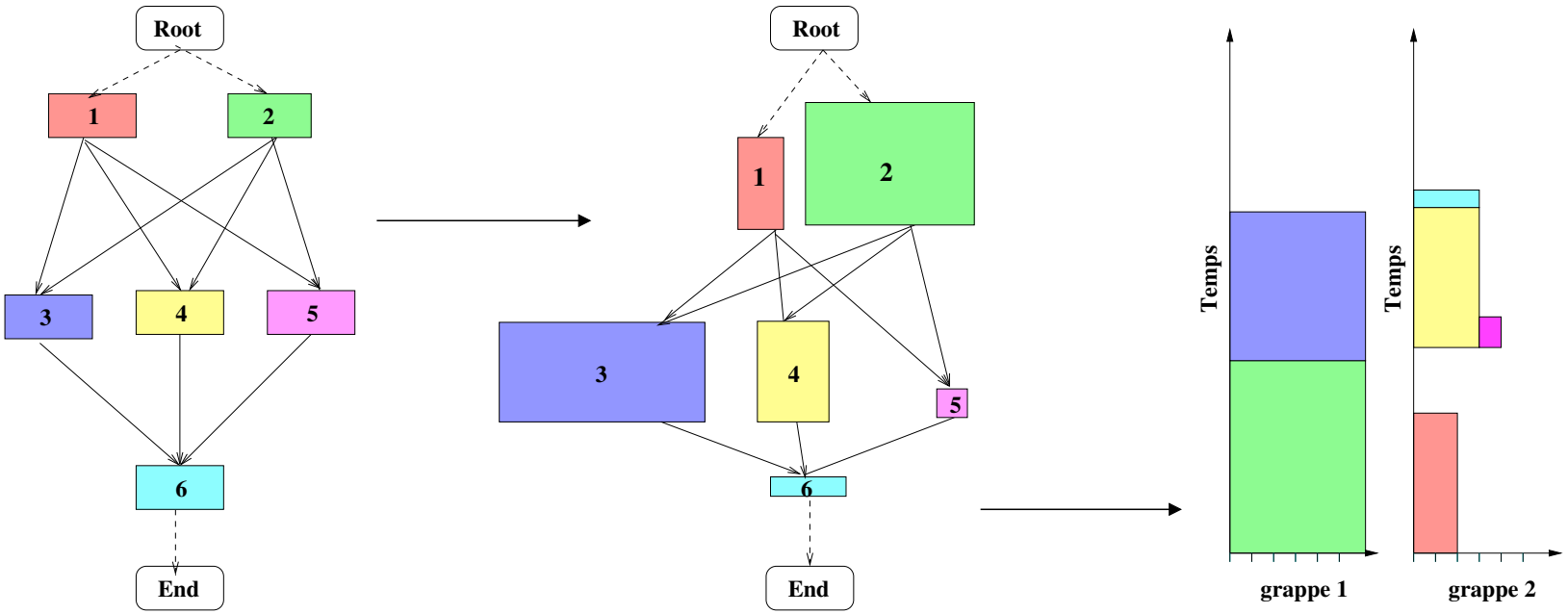
Cas d'utilisation

- Workflows scientifiques (e-science, traitement d'images, etc.)
- Lots de requêtes dans un intergiciel de grille

Ordonnancement de tâches modelables

Problème

Ordonnancer un graphe de tâches modelables sur une plate-forme multi-grappes hétérogène



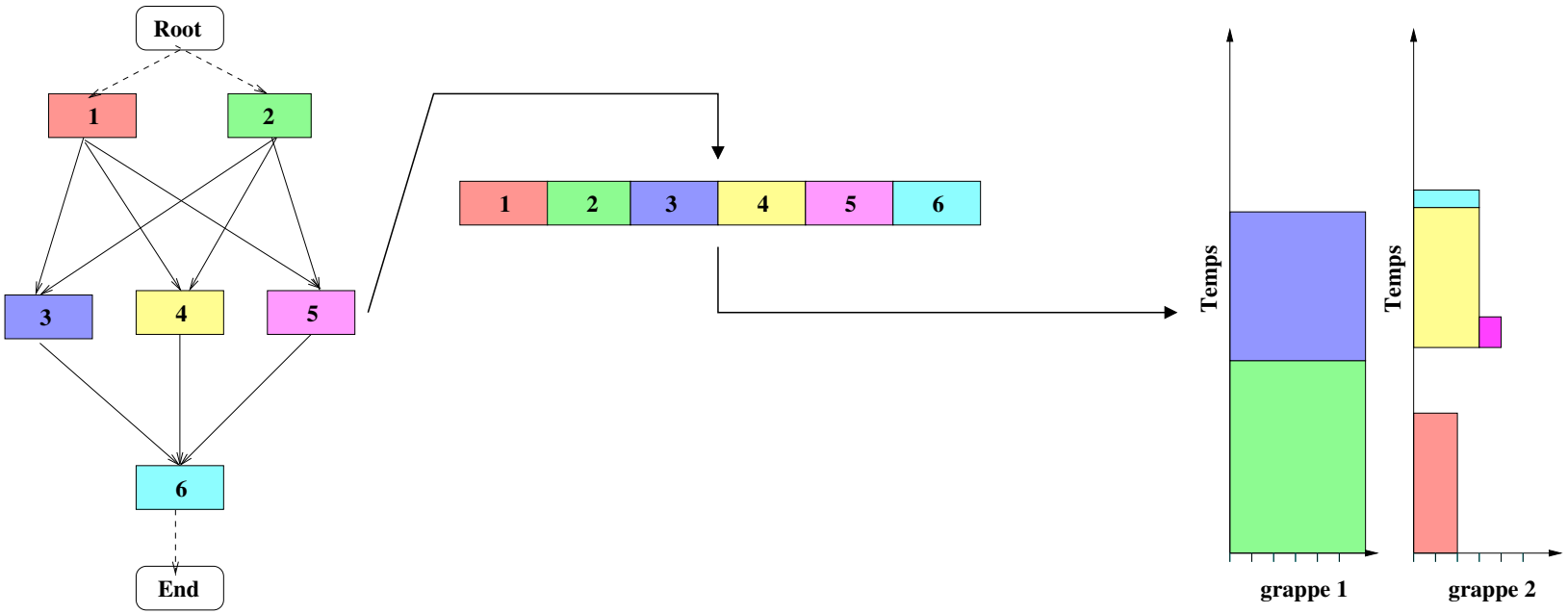
Approches

- En deux étapes
- Algorithme de liste

Ordonnancement de tâches modelables

Problème

Ordonnancer un graphe de tâches modelables sur une plate-forme multi-grappes hétérogène



Approches

- En deux étapes
- Algorithme de liste

Objectifs et contraintes

Objectifs

- Satisfaire les attentes des différents acteurs
  - Client : réduction du temps d'exécution
  - Intergiciel : équilibrage de charge
  - Fournisseur de ressources : maximisation de l'utilisation

Contraintes

- Latence réseau importante entre sites
- Contention réseau
- Respect des dépendances
- Algorithmes utilisables en pratique

Résultats

		Plate-forme		
		Homogènes	*Hétérogènes*	
Type de Calculs	*Séquentiels*	algorithmes de liste	⇒ algorithmes de liste	
	Modelables	⇓ algorithmes en 2 étapes	⇒ ⇓ **Problème ouvert**	⇓ M-HEFT, DMHEFT,Δ-CTS

⇒ HCPA,RATS MCGAS

Comparaisons

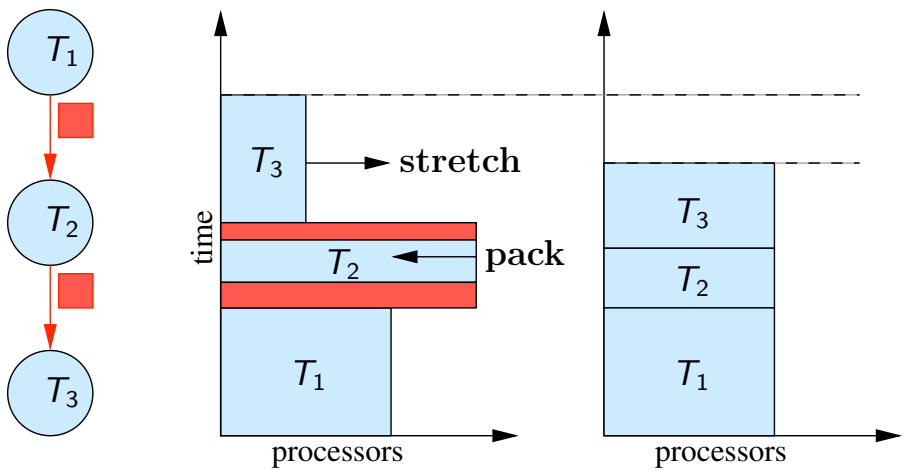
Derniers travaux

- Gestion de la concurrence entre applications
  - Approche par contrainte des allocations

Un exemple : RATS

Redistribution-Aware Two Step Scheduling

- Hypothèse de base
  - Deux calculs consécutifs sur les mêmes processeurs ⇒ Zéro communication
- Le(s) problème(s) de la redistribution
  - Source : processus d'allocation et de placement découplés
  - Pb 1 : des allocations proches ne sont pas identiques
  - Pb 2 : redistributions ⇒ Contentions ⇒ délais
    - Peut compromettre l'ordonnancement
- Solution proposée



L'algorithme de RATS

```
1: déterminer les allocations
2: while ∃ calculs ne sont pas ordonnancés do
3:    for pour chaque calcul prêt do
4:       estimer le temps d'exécution
5:    end for
6:    trier les calculs prêts
7:    while liste des calculs prêts ≠ ∅ do
8:       choisir le premier calcul de la liste
9:       if l'allocation d'un parent satisfait les conditions then
10:         placer le calcul sur l'allocation de ce parent
11:         recalculer les temps d'exécution des calculs prêts utilisant cette alloc.
12:         retrier la liste si nécessaire
13:      else
14:         placer l'allocation initiale
15:      end if
16:   end while
17: end while
```

Plan

- Introduction et parcours
- Activités de recherche passées
  - DIET et prédiction de performances
  - Algorithmique Out-of-Core
  - Ordonnancement de tâches modelables
  - Simulation « réaliste »
- Activités de recherche futures
  - Ordonnancement
  - Modélisation
  - Simulation
- Projets en cours
- Autres idées à discuter

Pourquoi la simulation ?

Avantages par rapport à l'expérimentation

- Pas besoin de construire le système en vrai, ni de développer l'application
- Contrôle et reproductibilité des expériences
- (Presqu')aucune limite aux scénarios testés
- Possibilité pour un tiers de reproduire les résultats

La simulation en bref

- Prédit des aspects du comportement d'un système à l'aide d'un modèle approximé
- Modèle : ensemble d'objets définis par un état + règles d'évolution d'état
- Simulateur : programme calculant l'évolution d'après les règles
- Caractéristiques souhaitées :
  - Précis : correspondance entre simulation et monde réel
  - Extensible : exécution rapide
  - Compréhensible : suffisamment simple
  - Instantiable : pas de paramètre magique
  - Pertinent : permet de voir des choses

Simulation pour le calcul parallèle et distribué

Modèle de plate-forme simpliste

- Débits de calcul et de communication fixes (Flop/s, Mb/s)
- Bus ou clique (pas d'interférences ou simplistes)
- Recouvrement total entre calcul et communications

Modèle d'application simpliste

- Toujours CPU intensif (pas de disque, de mémoire, d'utilisateur)
- Séparation nette entre phases de calcul et de communication
- Temps de calcul souvent ignorés en calcul distribué
- Temps de communication souvent ignorés en HPC

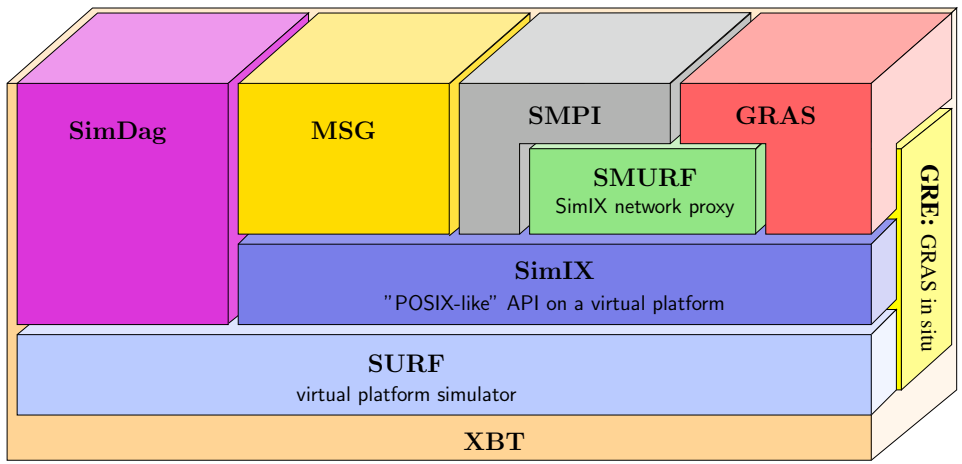
Simulation souvent directe

- Faire remplir un gantt chart ou compter les messages par la machine
- Pas besoin de standard $\rightarrow$ beaucoup de simulateurs *ad-hoc*

Le projet SimGrid (Hawai'i, Grenoble, Nancy)

SimGrid en bref

- Simulation ≡ processus communicants effectuant des calculs
- Point clé : Mélange de simulation mathématiques et à événements discrets à gros grain
- Ressources : Définies par un débit (MFlop/s ou Mb/s) + latence
- Tâches : Utilisent implicitement ou explicitement plusieurs ressources
- Différentes API selon le contexte d'utilisation



`http://simgrid.gforge.inria.fr`

Vers plus de reproductibilité et de réalisme

Plus de reproductibilité

- Indispensable à toute approche scientifique
  - Résultats d'un article doivent être reproduits pour être améliorés
- En informatique : souvent limitée aux auteurs de l'article
  - Simulateurs *ad-hoc*, code non disponible, descriptions sommaires, ...
- Objectif : être plus scientifique ...

Plus de réalisme

- Complexifier les modèles sans perdre l'instantiabilité
- Suivre les évolutions matérielles et logicielles
- Objectif : se rapprocher de l'émulation

Premier effort

- Archive de descriptions de plates-formes réalistes
  - The *Platform Description Archive* : `http://pda.gforge.inria.fr`

The Platform Description Archive

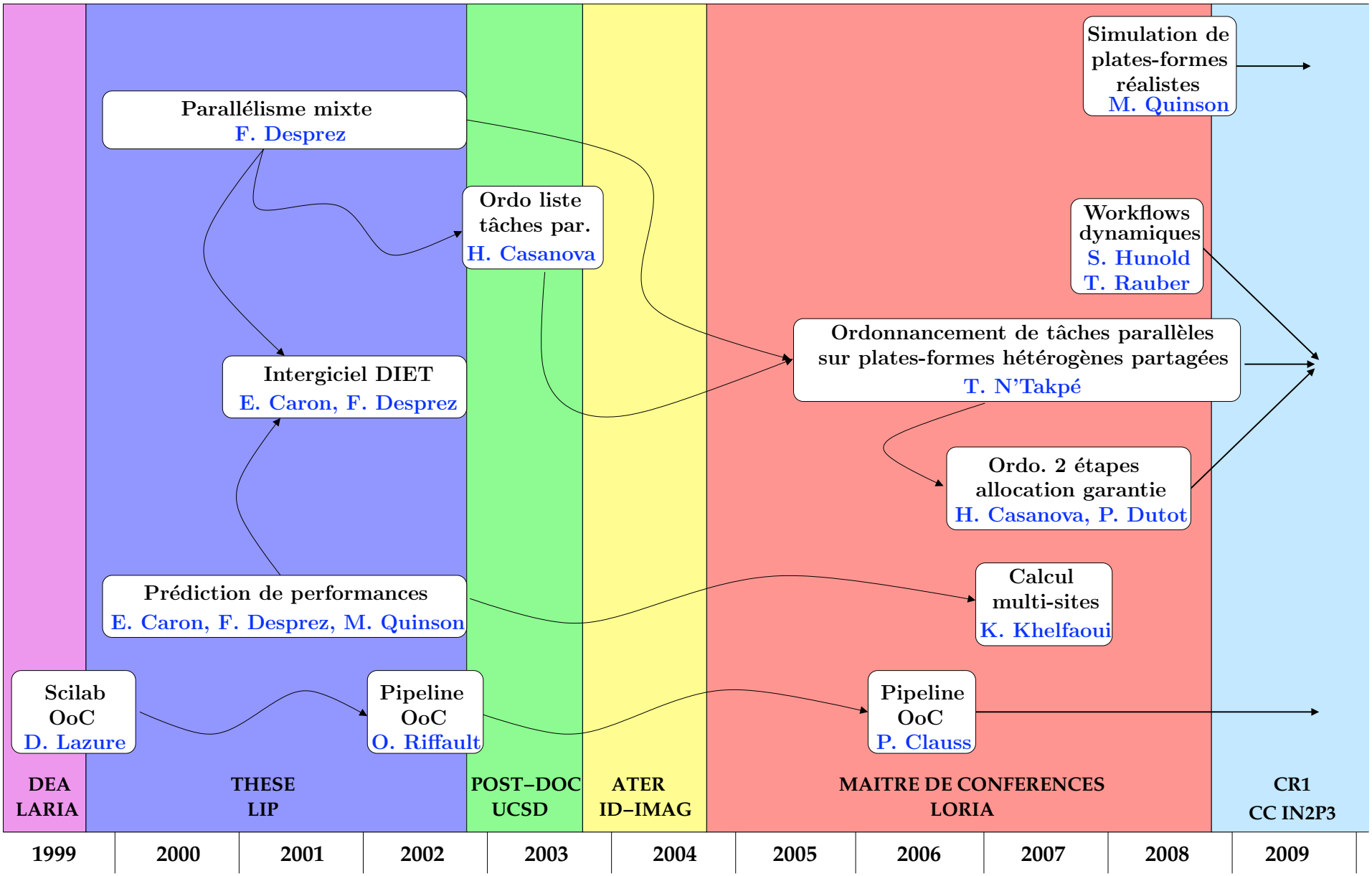
Contenu

- Formalisme de description
  - Lisible, extensible, compact et expressif
- Outils de génération /manipulation de descriptions
  - Simulacrum : SIMUlated pLAtform CReation and User Modification
- Archive de fichiers de descriptions utilisé dans des publications
  - Facilite la reproduction des résultats
- Catalogue de descriptions de « vraies » plates-formes
  - Grid'5000, DAS-3, ...
- Démo sur demande

Plan

- Introduction et parcours
- Activités de recherche passées
  - DIET et prédiction de performances
  - Algorithmique Out-of-Core
  - Ordonnancement de tâches modelables
  - Simulation « réaliste »
- Activités de recherche futures
  - Ordonnancement
  - Modélisation
  - Simulation
- Projets en cours
- Autres idées à discuter

Thématiques de recherches proposées



Ordonnancement

Motivations

- Généralisation des workflows scientifiques
  - Couplage de codes
  - Composition de services
- Banalisation des grilles
  - De plus en plus d'applications
- Évolution majeure des architectures
  - Le futur est au multi-cœurs

Objectif

- Concilier faisabilité et performances sur les grilles
  - Besoin d'interactions avec les fournisseurs d'applications et de ressources

Axes privilégiés

- Multi-applications
- Multi-cœurs
- Workflows dynamiques

Ordonnancement multi-applications

Quoi de neuf ?

- Plate-forme dédiée jusqu'à présent

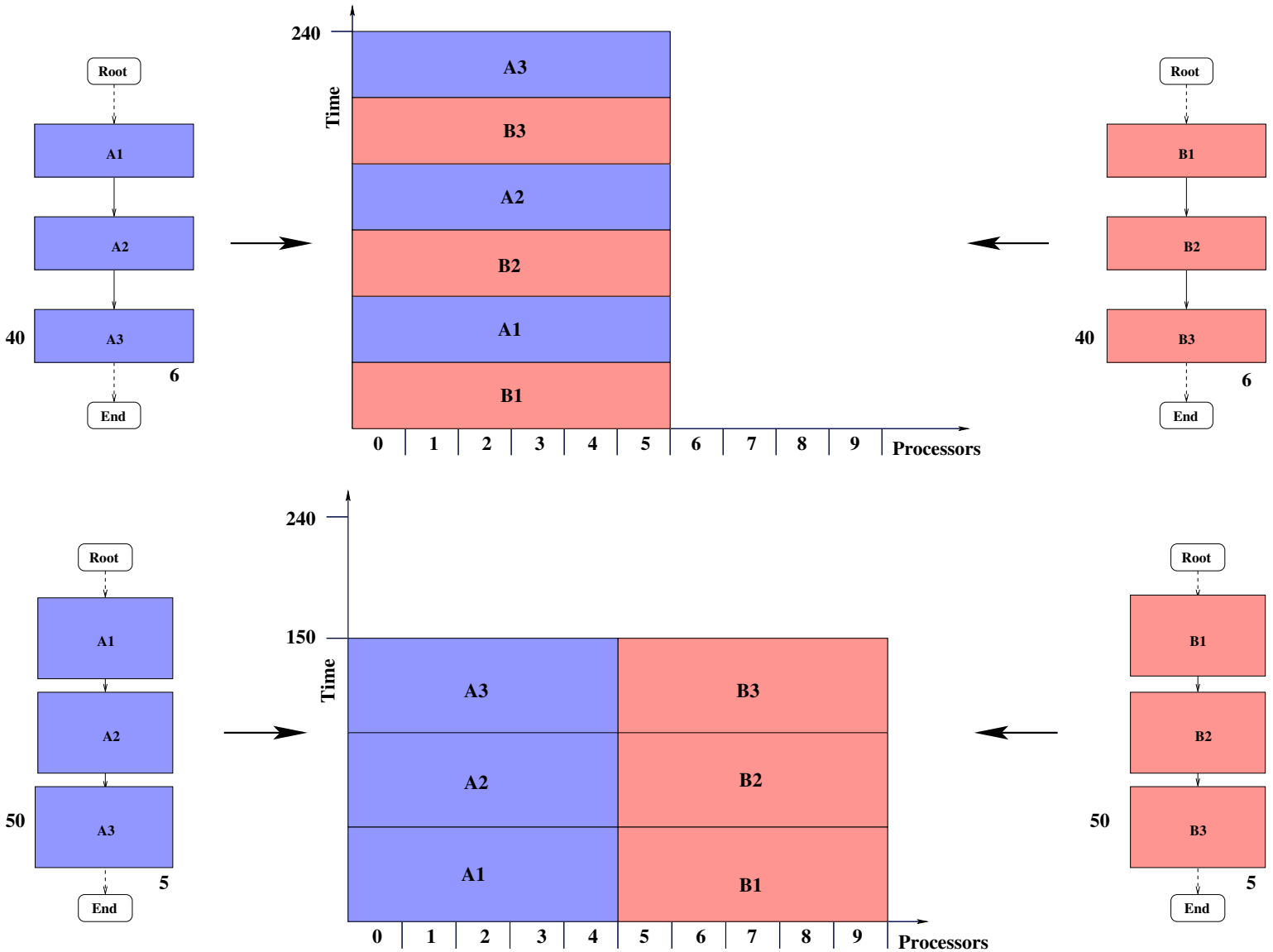
Travaux en cours

- Approche « partage de ressources »
  - Chaque application est contrainte lors de la phase d'allocation
  - Contraintes dépendantes du nombre d'applications et/ou des caractéristiques de chaque application
  - Contraintes statiques ou dynamique

Objectifs

- Limiter l'impact d'allocations égoïstes
- Rester performant
- Être équitable

Illustration



Ordonnancement multi-applications (2)

Travail collaboratif

- Dernier chapitre de mon thésard (soutenance en janvier
- Avec F. Desprez (LIP) et H. Casanova (Univ. Hawai'i)

Un peu plus tard . . .

- Applications soumises dynamiquement

$\longrightarrow$ Recalcul des contraintes et des ordonnancements à la volée

  - Liens avec la thèse de R. Bolze du LIP (31/10/2008)

Interaction Recherche/Production

- Possibilité d'intégration au sein de DIET
  - Projet Décrypthon par exemple

Ordonnancement pour les multi-cœurs

A quel niveau ?

- Plutôt applicatif, mais collaboration avec ordonnanceur système obligatoire

Pourquoi ça m'intéresse ?

- Très proche de ce que j'ai déjà étudié
  - Déterminer s'il faut utiliser x cœurs pour un calcul donné ($x = 1, 2, 4, 8$)

Problèmes

- Modèle de performance
- Plus de time-sharing
- Collaboration avec ordonnanceur système

Interaction Recherche/Production

- Possibilité de collaboration avec CREATIS (à reconfirmer)
  - Parallélisation du simulateur SIMRI
- Application candidate tournant au CC ?

Gestion de la dynamicité

Motivation

- Beaucoup d'applications en production ≡ workflows dynamiques
- Exemple : criblage moléculaire
  - Plusieurs étapes ⇒ workflows
  - Molécule pertinente ou pas ⇒ piste à suivre ou non
    - Impossible à savoir *a priori* ⇒ dynamicité
  - Données à transférer entre les étapes ⇒ besoin d'ordonnancement

Problèmes et Interaction Recherche/Production

- Beaucoup de méthodes classiques supposent la connaissance complète de l'application
- Donc besoin de connaissance précises sur chaque composant ...
  - Qui sait mieux que le concepteur de l'application ?
- et sur la plate-forme
  - Qui sait mieux que les admins/ingés/opérateurs ?

Plan

- Introduction et parcours
- Activités de recherche passées
  - DIET et prédiction de performances
  - Algorithmique Out-of-Core
  - Ordonnancement de tâches modelables
  - Simulation « réaliste »
- Activités de recherche futures
  - Ordonnancement
  - Modélisation
  - Simulation
- Projets en cours
- Autres idées à discuter

Modélisation

Motivations

- De plus en plus d'application *data-driven*
  - Plein de stockage
- Évolution majeure des architectures
  - Le futur est au multi-cœurs

Objectif

- Être capable d'intégrer les ressources disque et multi-cœurs dans la conception d'algorithmes
  - Par exemple dans un outil de simulation
  - Besoin de modèles pertinents

Axes privilégiés

- Modélisation du disque
- Multi-cœurs

Modélisation du disque

A quel niveau ?

- Clairement applicatif
  - Pas intéressé par le bas niveau
  - Mais par ce que « ressent » une application qui manipule des fichiers

Pourquoi ça m'intéresse ?

- A cause du Out-of-Core
  - Besoin d'un modèle pour calculer les bonnes tailles de blocs
- A cause de SimGrid
  - Pas de gestion du disque pour l'instant

Problème et interaction

- Modélisation complexe
  - Dépendante des données
  - Dépendante du support
- Avantage : le CC a plein de spécialistes, non ?

Approche envisagée

Similaire à ce qui a été fait pour le réseau dans SimGrid

- Partir d'un modèle simpliste (latence/débit)
- Ajouter une gestion du partage
- Améliorer jusqu'à être 5 % des performances de TCP
- Modéliser les différentes versions de TCP

Plan de bataille

- Bencher pleins de disques différents (ceux du CC ?)
  - Trouver quelqu'un qui comprend les résultats
- Comparer différents modèles simples
- Définir un modèle d'interaction
  - Quelqu'un sait comment des accès concurrents sont gérés ?
- Faire un article
- Intégrer dans SimGrid

Modélisation des multi-cœurs

En gros c'est pareil

- Benchs
- Modèles simples
- Comprendre et intégrer les mécanismes d'interaction
- Publier
- Augmenter l'outil de simulation

Pas forcément plus de connaissances initiales

- Toute aide ou expérience est la bienvenue !

Plan

- Introduction et parcours
- Activités de recherche passées
  - DIET et prédiction de performances
  - Algorithmique Out-of-Core
  - Ordonnancement de tâches modelables
  - Simulation « réaliste »
- Activités de recherche futures
  - Ordonnancement
  - Modélisation
  - Simulation
- Projets en cours
- Autres idées à discuter

Simulation

Motivations

- Étape dans le processus de validation expérimentale
  - Rapidité d'exécution
  - Exploration d'un grand nombre de scénarios

Objectif

- Proposer des outils d'aide
  - A la validation
  - Au développement (prototypage rapide)
  - Au dimensionnement (extrapoler une configuration matérielle

Axes privilégiés

- Intégration des nouveaux modèles de ressources
- Extension de la Platform Description Archive
- Augmentation du réalisme des applications simulées

Extension de PDA

Version courante

- Outils de génération / manipulation de plates-formes (Simulacrum)
- Catalogue de plate-formes réelles
  - Limité à Grid'5000 et DAS-3
  - Difficile d'avoir les informations voulues pour des grilles de production

Interaction souhaitée

- Obtenir des infos sur LCG
  - Caractéristiques des nœuds de calcul
  - Topologie et caractéristiques du réseau d'interconnexion
- Faire une description de la partie calcul du CC
  - Voire de la partie stockage si modèle de disque

Vers des applications simulées réalistes

Problème

- Simulation d'un modèle de l'application
  - Contrairement à l'émulation ou aux « vraies » expériences
- Focus sur le comportement général d'un algorithme pour différentes tailles de problèmes
  - Ne correspond pas forcément à une application particulière

Approche envisagée

- Définir un ensemble représentatif de noyaux applicatifs (calcul et communication)
- Instrumenter ces noyaux
- Extraire les informations à injecter dans le simulateur (volumes, vitesses d'exécution, . . .)
- Comparer les versions réelle et simulée
  - Ca colle : extrapolation possible
  - Ca colle pas : calibration du simulateur nécessaire

Plan

- Introduction et parcours
- Activités de recherche passées
  - DIET et prédiction de performances
  - Algorithmique Out-of-Core
  - Ordonnancement de tâches modelables
  - Simulation « réaliste »
- Activités de recherche futures
  - Ordonnancement
  - Modélisation
  - Simulation
- Projets en cours
- Autres idées à discuter

ANR ARPEGE USS SimGrid

Pitch et statut

- Ultra Scalable Simulation with SimGrid
  - SimGrid face aux communautés HPC et P2P
- Financé : 805k euros, 76,5k pour le CC (1 an ingé + missions)

Axes

- Modèles : simples pour le P2P, précis pour le HPC (réseau + multi-cœurs) + validation
- Instantiation des modèles : monitoring, découverte de topologies, caractérisation de workloads
- Analyse de simulation : extraction de données, visualisation, inférence d'effets
- Gestion de campagne de tests : planification, stockage, analyse
- Parallélisation du moteur de simulation
- Applications : dimensionnement et stockage pair à pair

ANR ARPEGE SPADES

Pitch et statut

- Servicing Petascale Architectures and DistributEd Systems
  - Du *Desktop computing* aux pools de réservations de ressources
- En attente : 61,4k pour le CC (2 ans post-doc + missions)
- Attention : si financé, passage de 4 à 3 ans et budget réduit d'1/3

Axes

- Découverte de services
- Runtime pour architecture petascale
- Co-scheduling
  - Modèle probabiliste de la disponibilité des ressources
  - Ordonnancement
  - Interaction entre ordonnanceur et batch schedulers
- Gestion des batch schedulers
- Applications : climatologie et cosmologie

ARC INRIA SimGrid'5000

Pitch et statut

- Simuler Grid'5000
- En attente : 78k euros (1 post-doc, 4 stages M2, missions)
- Réponse avant Noël

Axes

- Extension/amélioration de PDA
- Extension d'un simulateur de batch schedulers (SimBatch)
- Formalisation de traces réseaux
- Compatibilité entre API de SimGrid
- Application réelle $\rightarrow$ application simulée
- Ressources supplémentaires (disques et multi-cœurs

Plan

- Introduction et parcours
- Activités de recherche passées
  - DIET et prédiction de performances
  - Algorithmique Out-of-Core
  - Ordonnancement de tâches modelables
  - Simulation « réaliste »
- Activités de recherche futures
  - Ordonnancement
  - Modélisation
  - Simulation
- Projets en cours
- Autres idées à discuter

Vers plus d'interactions

- Plug-in DIET dans JSAGA
  - les gens de Diet sont intéressés
- BQS vs. OAR
  - Même objectif, méthodes différentes. Échanges d'expertise
- Jouer avec le CELL
  - Utilisation de la mémoire du CELL proche des techniques utilisées sur le pipeline Out-of-Core
- Out-of-Core et QCD
  - Peu de processeurs mais beaucoup de mémoire (pas disponible)
- LIMOS Clermont-Ferrand
  - cherchent une grosse appli de bases de données pour faire des caches sémantiques et de la médiation de données
- Institut des Grilles et Aladdin
  - Les deux cherchent à établir des interactions
  - Je suis au milieu ...