



ACI GRID 2001
Jeune équipe

Concerto

Composants parallèles adaptables



Yves Mahéo

Yves.Maheo@univ-ubs.fr

Membres du projet

- Laboratoire Valoria, Université de Bretagne Sud à Vannes
 - Équipe Orcade
- Coordinateur
 - Yves Mahéo, MCF
- Participants
 - Luc Courtrai, MCF
 - Frédéric Guidec, MCF
 - Pascale Launay, MCF

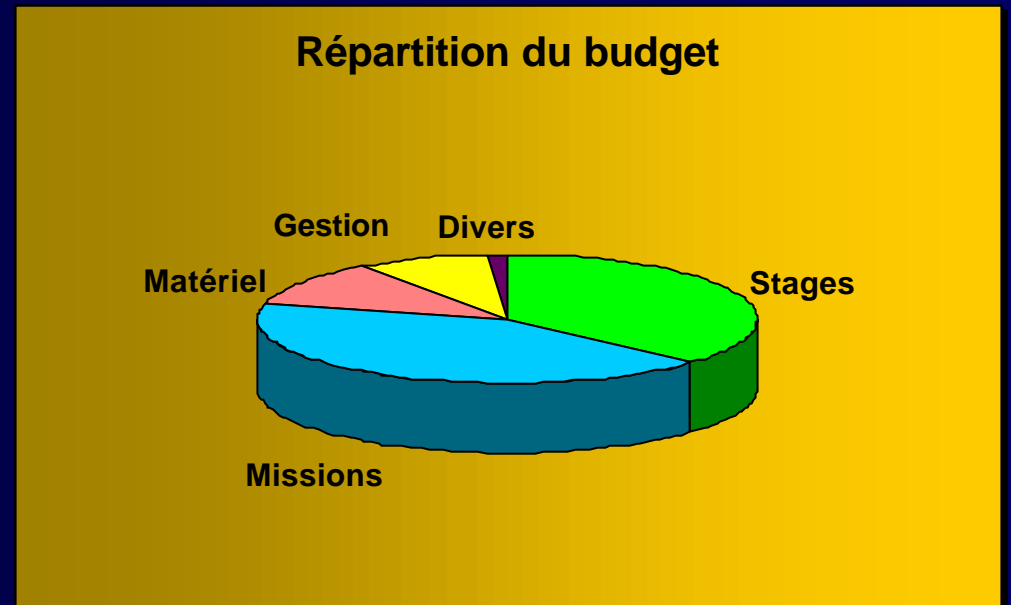
Collaborations

- Irisa
 - J.-L. Pazat (projet PARIS)
 - F. André (projet ARMOR)
- CEA / Université Paris 13
 - Philippe d'Anfray (LAGA)

Financement

- Financement ACI GRID

- Durée : 2 ans (2002-2003)
- 30500 € (200 KF)



- Pas d'autre financement

- Financement de post-doc demandé

Contexte et objectif

- Contexte
 - *Grid Computing*, Grappes
 - Applications parallèles « gourmandes » (simulation numérique,...)
 - Monde Java
- Objectif
 - Définition et mise en œuvre d'un modèle de composants parallèles adaptables
 - Modélisation d'une application par composants
 - Exécution sur grilles de calculateurs
 - Réactivité aux variations de l'environnement

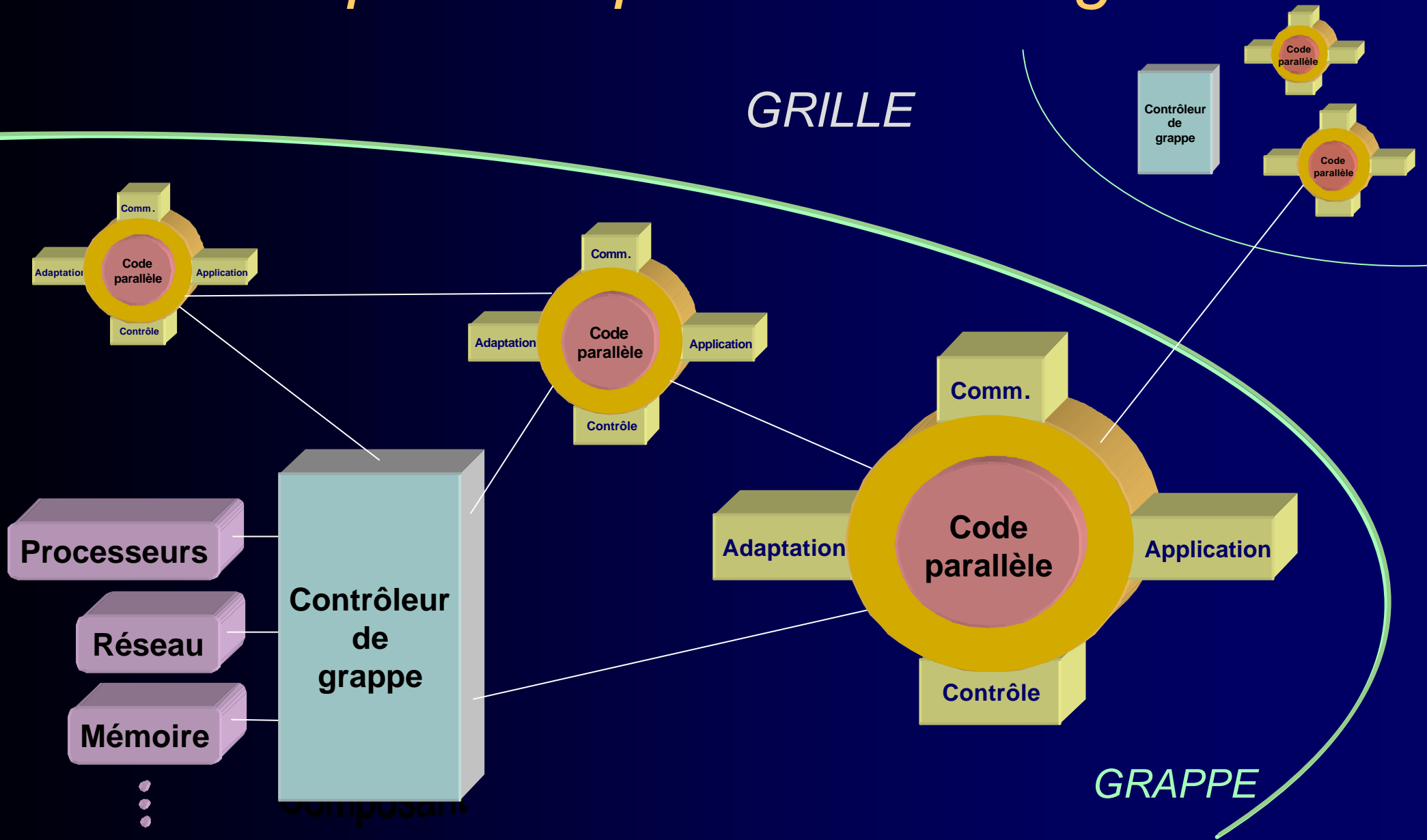
Motivations

- Applications parallèles de plus en plus complexes
 - Développement difficile à maîtriser
 - Intérêt des technologies de composants
 - Grappes de stations de travail / grille
 - Hétérogénéité des plates-formes
 - Partage des ressources
 - Les ressources disponibles pour une application varient
 - d'une exécution à une autre
 - pendant une exécution
- ⇒ Composants *parallèles et adaptables*

Axes du projet Concerto

- Gestion du parallélisme et de la distribution
 - Encapsulation d'un code parallèle dans un composant
- Adaptabilité
 - Prise en compte des ressources
- Support d'exécution
 - Observation de la consommation de ressources
 - Coopération efficace intra-composant

Composants parallèles sur grille



Parallélisme et distribution

Un composant parallèle

- décrit un code parallèle
 - Ensemble de threads Java communiquant par RMI et/ou messages
 - est déployé sur une grappe
 - Placement des threads sur les JVM
 - exhibe sa structure parallèle
 - Interface administrative permettant d'accéder notamment
 - à la liste des threads/JVM
 - au placement des threads sur les nœuds de la grappe
- ☞ La mise en œuvre autorise l'échange de données en parallèle entre composants

Prise en compte des ressources

Objectif : permettre le déclenchement de stratégies d'adaptation en fonction de l'utilisation des ressources

☞ Stratégies mises en œuvre par le composant lui-même

- **Ressources considérées**
 - Caractéristiques physiques
 - topologie, puissance CPU,...
 - bande passante, temps CPU, mémoire, espace disque...
 - Services offerts
 - fichiers, protocoles,...

Prise en compte des ressources (suite)

- Le composant
 - Exprime lors du déploiement ses besoins en termes de suivi de ressources
 - Interroge le contrôleur de grappes sur l'utilisation des ressources
 - Interrogation directe : rapports d'observation
 - Demande de notification lors de l'occurrence d'un ensemble d'événements ayant trait à l'état des ressources
 - Déclenche des procédures d'adaptation
 - codées par le concepteur du composant
 - qui s'appuient sur des services du contrôleur de grappe
- cf. projet RASC (VALORIA)

Support d'exécution

- Support pour la prise en compte des ressources
 - Monitoring de ressources
 - Mécanismes autour de la notification d'événements
- Support pour le composant parallèle
 - Services objets
 - passage de message, appel de méthode à distance, création à distance, chargement de classe,...
 - utilisation du RMI
 - Mise en œuvre adaptée aux grappes
 - cf. Projet Espresso (VALORIA), GRID-RMI

Travaux connexes dans GRID-RMI

- **Projet logiciel ACI 2002-2003**
 - Partenaire : Irisa/PARIS
- **Développement de Do!**
 - Refonte de l'exécutif Do!
 - Utilisation du passage de message
 - Expérimentation sur une application de croissance d'arbres
- **PadicoTM**
 - Support efficace pour la communication entre JVMs

Plan de travail

- Sous-projet 1 : Modèle de composant parallèle
- Sous-projet 2 : Prise en compte des ressources
 - Formalisme de description de l'état des ressources
 - Protocoles de notification des changements
- Sous-projet 3 : Support d'exécution
- Sous-projet 4 : Intégration des mises en œuvre
- Sous-projet 5 : Application test

Agenda et fournitures

- T0 + 6 mois
 - Étude préalable de SP1, SP2 et SP3
 - ☞ Point sur l'utilisation possible des EJB, positionnement / Globus
- T0 + 15 mois
 - Rapports intermédiaires SP1, SP2 et SP3
 - Premières mises en œuvre des prototypes
- T0 + 24 mois
 - Rapport final
 - Démonstration du prototype sur une application

Site Web Concerto

<http://www.univ-ubs.fr/valoria/Orcade/Concerto>