

,

École d'hiver GRID 2002 :
*Calcul Distribué, Méta-Computing, Globalisation des
Ressources*

Comité d'organisation :

- **Responsable :** Emmanuel JEANNOT, LORIA, Nancy
- Jens GUSTEDT, LORIA, Nancy
- Jean-Louis PAZAT, IRISA-INSA, Rennes
- Stéphane VIALLE, SUPELEC, Metz

Avant propos

La démocratisation des ressources de calcul et des réseaux à haut débit permet d'envisager de nouvelles applications permettant, en utilisant la répartition de ces ressources, de résoudre des problèmes demandant toujours plus de temps de calcul ou de place mémoire. En France, un certain nombre d'initiatives scientifiques (ARC INRIA OURAGAN, VTHD, objets CORBA parallèles, XtremWeb, ACI GRID, etc.) donnent lieu à des premiers résultats dans ce domaine. La thématique "GRID" fait intervenir des domaines de recherche historiquement disjoints : systèmes répartis, calcul distribué, réseaux, parallélisme, bases de données, sécurité. De plus, de nombreux projets pluridisciplinaires visant à utiliser la grille voient le jour en ce moment. L'École GRID'2002 est une formation issue de la recherche et s'appuyant sur des compétences et des résultats du domaine. Le but est de partager et transmettre cela aux doctorants, chercheurs et ingénieurs désireux d'acquérir une culture scientifique leur permettant de comprendre les enjeux de la globalisation des ressources et de lancer des recherches dans ce domaine. Cette école a pour objectif de réunir la communauté sur ce thème émergent. Nous espérons que cette école facilitera une réflexion pluridisciplinaire et favorisera l'émergence de nouveaux travaux de recherche.

Cette école est organisée en cinq journées décomposées en cinq thèmes : outils/environnement/supports, architecture logicielle, applications hautes performances, observation et gestion dynamique, algorithmes et modèles.

Les organisateurs remercient :

- les conférenciers pour la qualité de leur chapitres et de leurs exposés, le succès de l'école GRID'2002 leur revient.
- le CNRS pour la formation permanente (en particulier Catherine Ferveur), l'INRIA pour la formation doctorale, l'ACI GRID GRID2, et le GDR ARP pour leur soutien financier.
- Josianne Reffort qui a gérée avec beaucoup de compétence les invitations
- Armelle Demange et Anne-Lise Charbonnier des relations extérieures du LORIA qui ont assurées le secrétariat de l'école avec beaucoup de compétence.
- le comité scientifique de l'école (Jean-Pierre Briot, Franck Cappello, Frederic Desprez, Bertil Folliot, Jean-Louis Pazat, Christian Perez, Yves Robert, Jean Roman, el-Ghazali Talbi, Denis Trystram et Jean-Marc Vincent)

Jens Gustedt, Emmanuel Jeannot, Jean-Louis Pazat, Stéphane Vialle

Table des matières

| | |
|---|-----------|
| Table des matières | V |
| Table des figures | XI |
| Table des tableaux | XV |
| I Outils / Environnement / Supports | 1 |
| 1 Environnements pour grilles de calcul: l'approche Globus | 3 |
| 1.1 Introduction | 3 |
| 1.2 Architecture logicielles pour les grilles | 4 |
| 1.3 Le projet DataGrid | 6 |
| 2 Composants logiciels et grilles de calcul | 15 |
| 2.1 Introduction | 15 |
| 2.2 Vers des composants logiciels | 16 |
| 2.3 Le modèle de composants logiciels de Corba | 18 |
| 2.4 Composant logiciel et haute performance | 21 |
| 2.5 Conclusion | 24 |
| 3 Calcul à grande échelle | 29 |
| 3.1 Introduction | 29 |
| 3.2 Des utilisateurs et des grilles | 30 |
| 3.3 Langage pour le calcul à grande échelle | 31 |
| 3.4 Algorithmique et programmation | 34 |
| 3.5 Globalisation et calcul scientifique intensif, en guise de conclusion . . | 37 |
| 4 Concerto : gestion de ressources pour composants parallèles adap- | 41 |
| ables | |
| 4.1 Introduction | 41 |

| | | |
|-----------|---|-----------|
| 4.2 | Composants parallèles | 43 |
| 4.3 | Modélisation et contrôle des ressources | 45 |
| 4.4 | Conclusion | 51 |
| 5 | Compression adaptative et dynamique de données | 55 |
| 5.1 | Introduction | 55 |
| 5.2 | Compression et communication | 56 |
| 5.3 | Algorithme AdOC | 58 |
| 5.4 | Bibliothèque AdOC | 61 |
| 5.5 | Expériences | 61 |
| 5.6 | Conclusion | 62 |
| 6 | Partage de mémoire sur une infrastructure pair-à-pair | 65 |
| 6.1 | Introduction | 65 |
| 6.2 | Pourquoi les réseaux <i>pair-à-pair</i> ? | 66 |
| 6.3 | Partage de mémoire en contexte pair-à-pair | 67 |
| 6.4 | JXTA: une infrastructure pour des services pair-à-pair | 69 |
| 6.5 | Définition d'un service de partage de mémoire dans JXTA | 70 |
| 6.6 | Conclusion | 72 |
| 7 | Communication de groupes typés dans <i>ProActive</i> | 75 |
| 7.1 | Introduction | 75 |
| 7.2 | Cadre et précédents travaux | 76 |
| 7.3 | Communication de groupes typés | 78 |
| 7.4 | Conclusion et perspectives | 83 |
| II | Architecture Logicielle | 85 |
| 8 | GridRPC: Approche RPC pour la simulation sur la grille | 87 |
| 8.1 | Introduction | 87 |
| 8.2 | Études de cas | 88 |
| 8.3 | Fonctionnalités et problématiques | 90 |
| 8.4 | Conclusions et travaux futurs | 95 |
| 9 | Calcul dans les systèmes distribués à grande échelle | 99 |
| 9.1 | Introduction | 99 |
| 9.2 | Une caractérisation des systèmes Pair à Pair | 100 |
| 9.3 | Le projet XtremWeb | 102 |
| 9.4 | Recherche dans les systèmes de Calcul Global P2P | 105 |
| 9.5 | Conclusion | 108 |

| | | |
|------------|--|------------|
| 10 | Introduction à la notion de Grille de calcul : le projet DATAGRID | 111 |
| 10.1 | La notion de grille de calcul | 111 |
| 10.2 | Pourquoi maintenant ? | 113 |
| 10.3 | Les ingrédients d'une grille | 114 |
| 10.4 | Les applications scientifiques | 117 |
| 10.5 | Le projet DATAGRID | 118 |
| 10.6 | Perspectives | 120 |
| 10.7 | Conclusion | 121 |
| 11 | Pérennité dans les systèmes de stockage Pair à Pair | 123 |
| 11.1 | Introduction | 123 |
| 11.2 | État de l'art | 124 |
| 11.3 | Les mécanismes de redondances | 125 |
| 11.4 | Mécanismes de réparation dynamique | 127 |
| 11.5 | Conclusion | 129 |
| 12 | Expériences sur les systèmes distribués de grande taille | 133 |
| 12.1 | Introduction | 133 |
| 12.2 | Déploiement d'application à grande échelle | 134 |
| 12.3 | Évaluation d'un système pair-à-pair | 137 |
| 12.4 | Discussion et conclusion | 140 |
| 13 | Supervision et réseaux P2P | 145 |
| 13.1 | Introduction | 145 |
| 13.2 | Supervision des services dans une infrastructure P2P | 146 |
| 13.3 | P2P pour la supervision | 150 |
| 13.4 | Synthèse | 152 |
| 14 | Simulation pour l'ordonnancement distribué | 155 |
| 14.1 | Introduction | 155 |
| 14.2 | État de l'art | 156 |
| 14.3 | SimGrid | 158 |
| 14.4 | MetaSimGrid | 159 |
| 14.5 | Une caméra pour la simulation | 160 |
| 14.6 | Conclusion | 162 |
| III | Applications hautes performances | 165 |
| 15 | Distributed Data Mining Algorithms: A Brief Review | 167 |
| 15.1 | Introduction | 167 |
| 15.2 | Distributed Classifier Learning | 168 |

| | | |
|-----------|--|------------|
| 15.3 | Collective Data Mining | 171 |
| 15.4 | Distributed Association Rule Mining | 172 |
| 15.5 | Distributed Clustering | 173 |
| 15.6 | Privacy Preserving Distributed Data Mining | 175 |
| 15.7 | Future Directions | 176 |
| 16 | Calcul sur la Grille : L'expérience GrADS avec ScaLAPACK | 183 |
| 16.1 | Introduction | 183 |
| 16.2 | Adaptation des Bibliothèques Logicielles | 184 |
| 16.3 | Description de l'Expérience | 185 |
| 16.4 | Résultats | 188 |
| 16.5 | Conclusions | 191 |
| 17 | Coopération de métaheuristiques distribuées sur grilles de calcul | 195 |
| 17.1 | Introduction | 195 |
| 17.2 | L'environnement PARADISEO | 196 |
| 17.3 | Applications | 198 |
| 17.4 | Vers le déploiement de modèles distribués sur grilles | 200 |
| 17.5 | Conclusions et perspectives | 201 |
| 18 | Résolution des Problèmes d'OC Difficiles sur Grille de Machines | 205 |
| 18.1 | Introduction | 205 |
| 18.2 | Méthodes Branch-and-X parallèles | 206 |
| 18.3 | Coopération de métaheuristiques distribuées | 214 |
| 19 | GRID-TLSE: un site d'expertise en algèbre linéaire creuse CERFACS, IRIT, LaBRI et LIP | 225 |
| 19.1 | Introduction | 225 |
| 19.2 | Principaux composants logiciels du projet GRID-TLSE | 226 |
| 19.3 | Site d'expertise pour les matrices creuses | 227 |
| 19.4 | Infrastructure pour le grid computing | 229 |
| 19.5 | Conclusion | 231 |
| 20 | Couplage par composants logiciels de codes d'hydrogéologie | 237 |
| 20.1 | Introduction | 237 |
| 20.2 | Couplage algébrique | 239 |
| 20.3 | Couplage géométrique | 241 |
| 20.4 | Composants logiciels et grille de calcul | 242 |
| IV | Observation et gestion dynamique | 247 |

| | |
|---|------------|
| 21 Agents Résistants aux pannes | 249 |
| 21.1 Introduction | 249 |
| 21.2 Tolérance aux fautes dans les plates-formes multi-agents | 250 |
| 21.3 Adaptation et tolérance aux fautes | 252 |
| 21.4 Présentation générale de DarX | 253 |
| 21.5 Réalisation de DarX | 254 |
| 21.6 Performances | 259 |
| 21.7 Conclusions et Perspectives | 260 |
| 22 Traçage événementiel pour l'analyse d'exécutions réparties | 265 |
| 22.1 Introduction | 265 |
| 22.2 Environnement de traçage | 268 |
| 22.3 Analyse de trace | 273 |
| 22.4 Conclusion | 278 |
| 23 Grappe virtuelle I-Cluster : gestion distribuée des ressources d'un Intranet | 283 |
| 23.1 Introduction | 283 |
| 23.2 Solutions actuelles | 284 |
| 23.3 La gestion distribuée des ressources dans I-Cluster | 287 |
| 23.4 Conclusion | 289 |
| 24 Mesure des performances réseau dans une grille | 293 |
| 24.1 Introduction | 293 |
| 24.2 Problèmes de base | 294 |
| 24.3 Mesure des performances | 297 |
| 24.4 Exemples et applications | 299 |
| 24.5 Conclusion and perspectives | 303 |
| V Algorithmes et modèles | 307 |
| 25 Algorithms and Software to Schedule and Deploy Independent Tasks in Grid Environments | 309 |
| 25.1 Introduction | 309 |
| 25.2 The APST project | 310 |
| 25.3 Divisible Workload Scheduling | 312 |
| 25.4 Conclusion | 319 |
| 26 Ordonnancement en régime permanent pour plateformes hétérogènes | 325 |
| 26.1 Introduction | 325 |
| 26.2 Routage de paquets | 326 |

| | | |
|-----------|--|------------|
| 26.3 | Limitations des stratégies statiques | 331 |
| 26.4 | Conclusion | 333 |
| 27 | Ordonnancement de tâches malléables | 335 |
| 27.1 | Introduction | 335 |
| 27.2 | Graphe de tâches malléables | 337 |
| 27.3 | Définition Formelle et analyse théorique | 338 |
| 27.4 | Ordonnancement de tâches malléables | 343 |
| 27.5 | Conclusion | 350 |
| 28 | Ordonnancement pour le modèle temps partagé | 355 |
| 28.1 | Introduction | 355 |
| 28.2 | NetSolve | 356 |
| 28.3 | Modèles utilisés pour les expériences de simulation | 358 |
| 28.4 | Métriques observées | 359 |
| 28.5 | Gestion de l'historique des tâches | 359 |
| 28.6 | Heuristiques | 360 |
| 28.7 | Conclusion et travaux futurs | 362 |
| 29 | Asynchronisme et équilibrage de charge dans la grille de calcul | 365 |
| 29.1 | Introduction | 365 |
| 29.2 | Classes d'algorithmes itératifs parallèles | 366 |
| 29.3 | Exemple d'un algorithme IACA | 368 |
| 29.4 | Implantation et expérimentations | 369 |
| 29.5 | Équilibrage de charge dans les IACAs | 370 |
| 29.6 | Conclusion | 372 |
| 30 | Ordonnancement de tâches identiques sur réseau hétérogène | 375 |
| 30.1 | Introduction | 375 |
| 30.2 | Définitions | 376 |
| 30.3 | Algorithme | 378 |
| 30.4 | Propriétés | 378 |
| 30.5 | Optimalité | 381 |
| 30.6 | Conclusion | 382 |
| 31 | Modélisation de pipelines hétérogènes | 385 |
| 31.1 | Introduction | 385 |
| 31.2 | Résultats dans le cas général | 386 |
| 31.3 | Pipelines sur cluster de SMP | 389 |
| 31.4 | Conclusion | 393 |

Table des figures

| | | |
|------|--|-----|
| 2.1 | Cycle de vie d'un composant Corba. | 19 |
| 2.2 | Ports d'un composant Corba. | 20 |
| 2.3 | IDL3 d'un composant. | 20 |
| 2.4 | Les composants Corba parallèles permettent d'avoir de multiples flots de communication entre deux codes parallèles, supprimant tout goulot d'étranglement. | 21 |
| 2.5 | Padico TM permet l'accès concurrent aux ressources | 23 |
| 2.6 | Débits de Corba, MPI et des sockets en Java sur Padico TM | 23 |
| 3.1 | Organisation globale | 33 |
| 3.2 | Modèle de composants pour YML | 34 |
| 4.1 | Modélisation de quelques types de ressources | 47 |
| 4.2 | Modélisation des rapports d'observation | 47 |
| 4.3 | Modélisation des « motifs » servant à la sélection des ressources (partie gauche) et à la description des stratégies de recherche (partie droite). | 49 |
| 5.1 | Algorithme de base | 58 |
| 5.2 | Algorithme du thread de compression | 60 |
| 5.3 | Temps de transferts oilpann.hb (gauche), temps de calcul avec dgemm et avec NetSolve (droite) | 62 |
| 6.1 | Architecture générale. | 70 |
| 8.1 | Architecture de Diet. | 92 |
| 8.2 | Architecture de FAST. | 92 |
| 11.1 | Tolérance aux pannes en fonction du facteur de redondance r dans le cas où $s = 16$, le réseau est constitué de 1000 noeuds et 250000 blocs stockés. | 126 |
| 11.2 | MTTF avec réparation avec $\tau = 1$ | 128 |

| | | |
|------|---|-----|
| 11.3 | Comparaison de la MTTF entre réplication, redondance et redondance répliquée | 130 |
| 12.1 | Temps pour l'exécution du programme <i>vide</i> sur une grappe de 200 noeuds. Avec le protocole d'exécution distante rsh/rshd pour les courbes de droite et avec ssh/sshd pour les courbes de gauche. | 137 |
| 12.2 | Evolution du nombre moyen du nombre de noeuds traversés par une requête pour atteindre un fichier dans un réseau de 10000 noeuds. | 140 |
| 14.1 | Topologie réelle et décrite par ENV. | 161 |
| 17.1 | Classification des métaheuristiques parallèles | 197 |
| 17.2 | Classification hiérarchique des recherches hybrides | 198 |
| 17.3 | Représentation du modèle parallèle coopératif hybride | 199 |
| 18.1 | Classification des métaheuristiques parallèles | 215 |
| 18.2 | Classification hiérarchique des recherches hybrides | 216 |
| 18.3 | Représentation du modèle parallèle coopératif hybride | 218 |
| 19.1 | Composants logiciels du projet GRID-TLSE. | 228 |
| 21.1 | Architecture de DarX | 255 |
| 21.2 | Vue de l'application par une tâche | 256 |
| 21.3 | Acheminement d'un message dans un groupe | 256 |
| 21.4 | Coût de réplication | 259 |
| 21.5 | Coût de la migration de serveur | 260 |
| 21.6 | Latence des communication en fonction du degré de réplication | 260 |
| 22.1 | Exemple de hiérarchie d'objets visuels pour représentation d'application Java répartie. | 273 |
| 22.2 | Architecture et pattern de l'application serveur de livre. | 274 |
| 22.3 | Diagramme espace-temps du pattern d'exécution. | 275 |
| 22.4 | Pattern de réception d'un message (niveau d'abstraction intergiciel). | 276 |
| 22.5 | Pattern de synchronisation sur un " <i>Java raw monitor</i> ". | 276 |
| 22.6 | Pattern d'interaction distante. | 277 |
| 22.7 | Analyse de la phase 2. | 277 |
| 22.8 | Utilisation des ressources systèmes par les JVMs. | 279 |
| 24.1 | Modélisation de la grille | 296 |
| 24.2 | Architecture de supervision de grille | 299 |
| 24.3 | La plate-forme DataGrid vue par MapCenter | 301 |
| 24.4 | MapCenter overview | 302 |
| 25.1 | APST models. | 310 |

| | | |
|------|--|-----|
| 25.2 | Computing platform model. | 314 |
| 25.3 | UMR dispatches the workload in rounds, where the chunk size is fixed within a round, and increases between rounds. | 316 |
| 26.1 | Graphe de tâches et graphe de la plateforme. | 329 |
| 27.1 | Représentation de la pénalité sur 2 processeurs | 340 |
| 27.2 | Structure dominante des ordonnancements pour 2 processeurs | 341 |
| 27.3 | Ordonnancement résolvant 3-PARTITION. | 342 |
| 27.4 | Contre-exemple d'ordonnancement non contigu optimal. | 344 |
| 27.5 | Principe de l'allocation en deux étagères S^1 et S^2 | 346 |
| 27.6 | Ordonnancement online par lot, avec release date | 349 |
| 27.7 | Principe de l'allocation dans le cas hiérarchique. | 351 |
| 28.1 | Résultats pour HMCT contre MCT sur 25 serveurs | 361 |
| 28.2 | Résultats pour MSF contre MCT sur 25 serveurs | 361 |
| 29.1 | Flot d'exécution d'un ISCS avec deux processeurs. | 366 |
| 29.2 | Flot d'exécution d'un ISCA avec deux processeurs. | 367 |
| 29.3 | Flot d'exécution d'un IACA avec deux processeurs. | 367 |
| 29.4 | Efficacités des algorithmes synchrone et asynchrone sur un système local homogène. | 370 |
| 30.1 | Le premier noeud est la source des tâches. | 376 |
| 30.2 | Représentation graphique d'un ordonnancement. | 377 |
| 30.3 | L'algorithme en pseudo-code. | 379 |
| 30.4 | On peut toujours éviter de croiser les communications. | 380 |
| 31.1 | Exécution pipelinée sur un nœud. | 390 |
| 31.2 | Exécution pipelinée avec taille variable sur 4 processeurs. | 392 |
| 31.3 | Temps d'exécution d'un pipeline sur 2 nœuds. | 393 |