Projet Arimaa – CR2 : Mercredi 24 Septembre

Présents : Toute l'équipe, Christian Raymond, et Nikos Parlavantzas

Ordre du jour :

1. Validation des technologies choisies
2. Étude de l'état de l'art
3. Orientation du cahier des charges

Information échangées

- Présentation de thèses et travaux sur la parallélisation du jeu Arimaa par Benoît :
    - Tout raisonnement trop lié au jeu ne pourra pas se généraliser. Cela ne sert à rien de commencer à étudier des cas qui ne sont relatifs qu'à Arimaa
    - Graphique très intéressant montrant la baisse de l'efficacité de l'algorithme utilisé par un thésard, géré par 4 threads quand le temps augmente. "Additionnal time for thinking worths the less the time engine has".
- Travail sur Arimaa simplifié, pour commencer, on sauvegardera des algorithmes de mémoire avec des tableaux simples en 2D remplis de 0 et 1.
- Validation du C++, de Grid5000, et de OpenMP.
- Ordre du jour des prochaines séances :
    - s'intéresser au contenu du prochain rapport (présentation algorithme, stéréotype de parallélisation)
    - continuer à s'intéresser à l'état de l'art, faire une étude plus poussée de la bibliographie, l'implémentation de la méthode de Montecarlo – Algo Min-Max
- Discuter des 3 stratégies de la parallélisation : (définitions à approfondir)
    - Root parallel : 1 thread, 1 arbre, fusion des statistiques de la première ligne et continue. Ne calcule qu'un nœud à la fois, puis recommence le calcul sur toutes les branches
    - Leaf parallel : plus simple pour les communications machines, on fusionne les résultats trouvés pour créer un nouveau nœud "moyen"
    - Tree parallel : développe les feuilles de façon parallèles (autant de thread que de feuilles)
- Problème dans un certain cas. Lors d'une surcharge de statistiques, il est possible de perdre quelques unes de ces statistiques. On utiliserait un mutex pour protéger et enregistrer ces statistiques. Le problème de cette méthode est le coût qui paraîtrait trop élevé. Ainsi, on ignorera le problème au vu de toutes les autres statistiques que nous avons.
- Faire une pré-étude des problèmes dans le premier rapport, décrire le projet, et les résoudre dans le deuxième rapport.
- Dans tous les cas, l'étude se fait sur plusieurs postes dès le départ.
- Répartition des rôles faite (temporaire avec rotations toutes les deux semaines à priori en fonction des charges de travail – les noms des postes ne sont pas définitifs) :
    - Rédacteur en chef : il gère et oriente le groupe, recherche des axes pour le compte-rendu d'octobre (Dan pour le moment)
    - Responsable réunion : il gère les réunions, le matériel, la relation avec les enseignants, il oriente le debriefing, et sait de quoi tout parle. (Dan pour le moment)
    - Responsable parallélisation : il approfondit les notions des 3 parallélismes de Christian Raymond (Tree/Leaf/Root), recherche de bibliographe et comparatifs des différentes méthodes (Benoît et Mikaïl pour le moment)
    - Responsable utilitaires: il gère l'UML (si le temps le permet) et Microsoft Project, travail sur le suivi des tâches, les deadlines et la répartition des ressources (Baptiste pour le moment)
    - Responsable application : il complète la version du jeu d'Arimaa et cherche à savoir

si la nouvelle version de C++ 2011 est stable pour l'utiliser dans le projet.

- Écriture des compte-rendus en Latex pour la forme et se familiariser avec le format.

Planification

Tâche	Responsable	Deadline
État de l'art, thèses, comparatifs,bibliographie	Benoît,Mikaïl	01/10 extendable
Établir le cahier des charges/compte-rendu	Dan	01/10 Octobre
Compléter le jeu, et la compatibilité C++ 2011	Gabriel	01/10 extendable
Tutoriel MS Project + UML si possible	Baptiste	01/10 extendable

Date de la prochaine réunion : 01/10

Documents additionnels :