

1 Introduction

1.1 Contexte de la thèse

- contexte général (Laboratoire, équipe)
- Traitement d'images embarqué
- nécessité de code legacy
- Contraintes de traitement temps réel
- Calcul parallèle
- Projets : Ocelle, Teraops.

1.2 Architectures Parallèles

- Les niveaux de parallélisme
- les différents types d'architectures : Multi-core, GPU, modèle mémoire(partagée, distribuée), SIMD, SWAR
- L'architecture du Cell

1.3 Modèles de Programmation et Outils de Parallélisation

- Présentation des différents modèles de programmation (Shared Memory, Message Passing, Stream processing)
- OpenMP (Compilateur XLC, barcelona CellSS) + MPI (MPI en général, MPI microtask IBM) + Stream Processing (RapidMind, Sequoia)
- vectorisation (gcc, xlc) difficultés sur le Cell
- Outils de haut-niveau SPEARS, Gedae
- validation outils (quand c'est possible) avec code simple (addition matricielle).

2 Optimisation des transferts et du contrôle

- Tiling 1D, 2D
- bord, chaînage d'opérateur, fusion.
- double buffering.
- Outils : Cell_MPI, dépôt APP(peut être dans le contexte aussi)

3 Parallélisation de Code de Traitement d'Images

3.1 Algorithmes Régulier

3.1.1 Harris points d'intérêt

- Présentation de l'algorithme
- Algorithme représentatif du traitement d'images bas niveau.

- difficultés liées à l'implémentation, adaptation aux contraintes du Cell, jeux d'instruction(floats) , DMA....
- Modèle de parallélisation
- Mesure de performances
- Modèle analytique pour la prédiction de performances

3.1.2 Un autre algo régulier (Sigma Delta, ..)

- mémorisation dans SPE ou pas.

3.1.3 Un autre algo irrégulier ECC

- load balancing

4 Conclusion