

Compilation polyédrale

Félix-Antoine Ouellet

Université de Sherbrooke

4 décembre 2014

- 1 Motivation
- 2 Compilation traditionnelle
- 3 Approche polyédrale
- 4 Parallélisation automatique
- 5 Conclusion

Plan

- 1 Motivation
- 2 Compilation traditionnelle
- 3 Approche polyédrale
- 4 Parallélisation automatique
- 5 Conclusion

L'ère du parallélisme

Problèmes courants

Optimisations des boucles

Problèmes courants

Parallélisation d'applications existantes

Problèmes courants

Rendre le parallélisme accessible

Plan

- 1 Motivation
- 2 Compilation traditionnelle
 - Bases de la compilation
 - Processus de compilation
 - Analyse de dépendences
- 3 Approche polyédrale
- 4 Parallélisation automatique
- 5 Conclusion

Notions importantes

- Transforme un programme écrit dans un langage (de haut niveau) en un programme écrit dans un autre langage (de bas niveau).
- Maintient la sémantique du programme original.

Architecture usuelle

Étape 1 - AST

Étape 2 - CFG

Représentation intermédiaire

POURQUOI ??



Importance des dépendences

Représentation

Tests de dépendences

Plan

- 1 Motivation
- 2 Compilation traditionnelle
- 3 Approche polyédrale**
 - Représentation
 - Optimisations
 - Limitations
- 4 Parallélisation automatique
- 5 Conclusion

Représentation

Limitations

- Accès non affines
- Boucles irrégulières

Plan

- 1 Motivation
- 2 Compilation traditionnelle
- 3 Approche polyédrale
- 4 Parallélisation automatique
 - Schedule
 - Support présent
- 5 Conclusion

Intuition

Support présent

- GCC (Graphite)
- LLVM (Polly)
- Langages expérimentaux (X10)
- Plateformes expérimentales (PLUTO)

Plan

- 1 Motivation
- 2 Compilation traditionnelle
- 3 Approche polyédrale
- 4 Parallélisation automatique
- 5 Conclusion

Conclusion