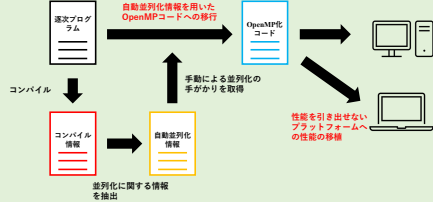


背景

- ▶ **プログラム高速化のためのスレッド並列化の需要増加**
 - ▶ さまざまな分野において高性能計算の需要が増加
 - ▶ プロセス内で複数のスレッドを起動することで並列化処理

- ▶ **様々なプラットフォーム向けにコンパイラが開発**
 - ▶ 自動並列化機能によって人的コストを抑えつつ並列化
 - ▶ コードは特定のコンパイラによって最適に並列化
- 課題：コンパイラの自動並列化機能は別のコンパイラシステムを想定したコードを最適に並列化することができない**

既存手法：単体のコンパイラから取得した並列化情報を元にOpenMPコード生成



- ▶ 逐次プログラムをコンパイルする際に自動並列化し、コンパイル情報を取得
- ▶ 得られた並列化の手がかりを用い、特定のコード向けに想定されていないプラットフォームでも使用できるようにOpenMP化

▶ **性能を他のプラットフォームに移植可能**
 依然として存在する問題点：単体のコンパイラの並列化性能しか移植することができず不十分

目的と提案手法

- ▶ **研究の目的**
 - ▶ 複数のコンパイラから得られた自動並列化情報を集合知として合わせ、並列化効果を最大化する

並列化箇所	コンパイラ1	コンパイラ2	コンパイラ3	マージ
1	並列化	並列化		並列化
2			並列化	並列化
3		並列化		並列化
4			並列化	並列化
5		並列化		並列化

- ▶ **提案手法：複数のコンパイラから取得した並列化箇所を適切に組み合わせOpenMPコード生成**
 - 方針1：可能な限り多くの箇所を並列化
 - 方針2：ループネストの内側と外側で並列化箇所の判断が分かれた際は外側のループを並列化
- ▶ 特定コンパイラが並列化を不得手とするプログラム構造を別のコンパイラの並列化情報によって並列化

▶ **高い並列化性能を実現**

評価

- ▶ **評価環境**
 - ▶ 使用ベンチマーク：PolyBench ataxベンチマーク

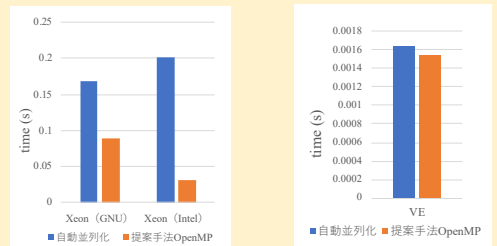
プロセッサ			コンパイラ	
種類	製品名	コア数	種類	バージョン
VH	Xeon Gold 6226	12	GNU	11.2.1
VE	Type 20B	8	Intel	2021.7.0
			NEC	3.5.1

並列化結果（表と提案手法の方針の対応関係）

コードの役割	行	GNU	Intel	NEC	提案手法
配列の初期化	29	並列化	並列化		並列化
	31	並列化	並列化	並列化	並列化
	68			並列化	並列化
メインの計算カーネル	71		並列化		
	73		並列化		

- ▶ 並列化情報が異なる箇所
 - {方針1} 29行目：NECでは並列化していないGNUとIntelを優先
 - {方針2} 68行, 71行, 73行目：最も外側ループの68行を優先

- ▶ **並列化した場合の性能（グラフとグラフからの考察）**



- ▶ Xeon (GNU), Xeon (Intel), VE (NEC) のいずれの場合も実行時間減少
- ▶ Xeon (GNU), Xeon (Intel) の実行時間減少率が大きい
 - ▶ NECコンパイラから得られたメインの計算カーネルの並列化情報の恩恵

▶ **提案手法の効果が表れている**

今後の研究計画と結論

- ▶ **研究計画**
 - ▶ 提案モデルの自動システム化
 - ▶ 並列化に有効なプロファイル情報を分析
 - ▶ 提案手法が有効なベンチマークを調査し特徴を分析

- ▶ **結論**
 - ▶ PolyBench ataxベンチマークにおいて複数の自動並列化情報を用いて性能向上
 - ▶ NECコンパイラの並列化情報が性能向上に大きく寄与