Design and Implementation of OpenMP Compiler for λ Computing Environment

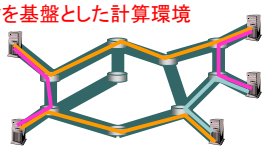
大阪大学 大学院情報科学研究科
情報ネットワーク学専攻 村田研究室
合田 圭吾

背景

- グリッドコンピューティング技術への期待の高まり
- TCP/IPは分散並列計算には不向き
  - パケット処理による遅延などオーバヘッドが発生

  ⇒大量のデータ交換を行う大規模計算への応用では十分な計算性能を達成することは困難
- λコンピューティング環境を提案
  - 高速・高信頼なネットワークを基盤とした計算環境

  ⇒ノード計算機間に波長パスを設定し，専用の通信路として用いる



2

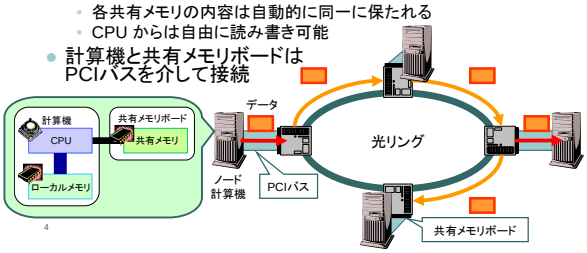
研究の目的

- λコンピューティング環境上で高速な分散並列計算を実現
  - AWG-STARシステムを利用し，λコンピューティング環境を構築
  - OpenMPを用いた分散並列計算環境を実現
    - AWG-STAR システム用のOpenMPコンパイラを設計・実装する

3

AWG-STAR システム

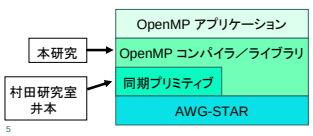
- NTT フォトニクス研究所により開発
- 各ノード計算機は共有メモリボードを搭載し，ノード計算機間に設定した波長パスによって光リングを形成
  - ボード上に共有メモリを搭載
    - 各共有メモリの内容は自動的に同一に保たれる
    - CPU からは自由に読み書き可能
  - 計算機と共有メモリボードはPCIバスを介して接続



4

OpenMP

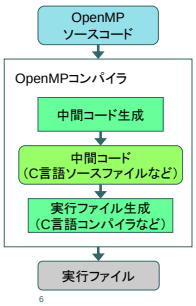
- 並列プログラミングAPIの標準規格
  - C言語（およびFortran）の拡張
- ソースコード中にOpenMPディレクティブと呼ばれる指示文を記述することでプログラミング
  - 比較的容易に並列プログラムを開発できる



OpenMPを用いたAWG-STAR上のプログラミング環境

5

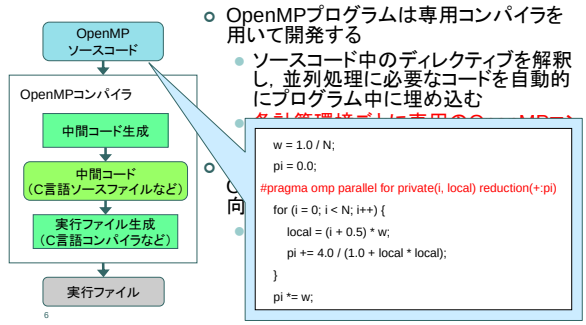
OpenMPコンパイラ



- OpenMPプログラムは専用コンパイラを用いて開発する
  - ソースコード中のディレクティブを解釈し，並列処理に必要なコードを自動的にプログラム中に埋め込む
  - 各計算環境ごとに専用のOpenMPコンパイラが必要
- 本研究では既存のOpenMPコンパイラOMPiに基づいてAWG-STARシステム向けのコンパイラを実装する
  - OMPiは通常のマルチプロセッサ計算機向けであるため，AWG-STAR環境向けに修正を行う

6

OpenMPコンパイラ



- OpenMPプログラムは専用コンパイラを用いて開発する
  - ソースコード中のディレクティブを解釈し，並列処理に必要なコードを自動的にプログラム中に埋め込む
  - 各計算環境ごとに専用のOpenMPコン
- 本研究では既存のOpenMPコンパイラOMPiに基づいてAWG-STARシステム向けのコンパイラを実装する
  - OMPiは通常のマルチプロセッサ計算機向けであるため，AWG-STAR環境向けに修正を行う

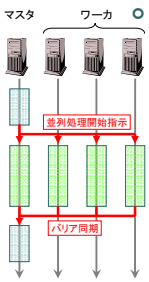
6

OpenMPコンパイラ

- OpenMPプログラムは専用コンパイラを用いて開発する
  - ソースコード中のディレクティブを解釈し，並列処理に必要なコードを自動的にプログラム中に埋め込む
  - 各計算環境ごとに専用のOpenMPコンパイラが必要
- 本研究では既存のOpenMPコンパイラOMPiに基づいてAWG-STARシステム向けのコンパイラを実装する
  - OMPiは通常のマルチプロセッサ計算機向けであるため，AWG-STAR環境向けに修正を行う

6

AWG-STAR向け OpenMPコンパイラの設計

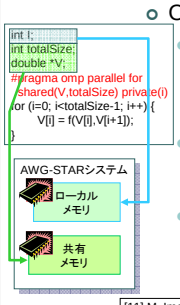


- OpenMPの実装に必要となる要素
  - 複数ノード計算機による並列処理
    - ノード計算機の1台をマスタとして設定
    - マスタがその他のノード計算機に光リング経由で指示を出して並列処理を制御する
  - ノード計算機間でのデータ共有
    - プログラム中で用いる共有データをAWG-STARシステムの共有メモリ上に置いて計算する
  - ロック,バリア同期機構
    - AWG-STARの提供する共有メモリ・シグナリング機構を利用した独自のロック，バリア同期ライブラリを作成し[11]，利用する

[11] M. Imoto, "Design and implementation of synchronization primitives for λ computing environment," Master's thesis, Graduate School of Information Science and Technology, Osaka University, Feb. 2007.

7

AWG-STAR向け OpenMPコンパイラの設計



- OpenMPの実装に必要となる要素
  - 複数ノード計算機による並列処理
    - ノード計算機の1台をマスタとして設定
    - マスタがその他のノード計算機に光リング経由で指示を出して並列処理を制御する
  - ノード計算機間でのデータ共有
    - プログラム中で用いる共有データをAWG-STARシステムの共有メモリ上に置いて計算する
  - ロック,バリア同期機構
    - AWG-STARの提供する共有メモリ・シグナリング機構を利用した独自のロック，バリア同期ライブラリを作成し[11]，利用する

[11] M. Imoto, "Design and implementation of synchronization primitives for λ computing environment," Master's thesis, Graduate School of Information Science and Technology, Osaka University, Feb. 2007.

7

計算環境の性能評価

- 実験環境
  - CPU: Intel Xeon (3.06GHz)×1
  - OS: Red Hat Linux 7.2
  - コンパイラ: GCC 2.96
  - ノード数: 4台
  - AWG-STAR 共有メモリボード
    - 光インターフェース速度: 2.152Gbps
    - 共有メモリアクセス最大速度：書き込み64MB/s 読み込み80MB/s
- 性能比較対象：SCore
  - Gigabit Ethernet環境で並列計算を実現するミドルウェア
  - 専用のOpenMPコンパイラOmni/SCASHを備える
    無償配布されていることからOpenMP関連の研究で性能比較実験によく利用される

8

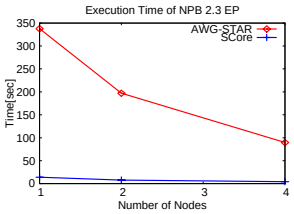
評価用ベンチマーク

- NPB (NAS Parallel Benchmark) 2.3
  - EP
    - 問題サイズ：Class W (2^{25})
  - BT
    - 問題サイズ：Class W (24×24×24)
- 姫野ベンチマーク
  - 問題サイズ：XS (128×64×64)
- ベンチマークの特性
  - 共有データへのアクセス頻度**低**　：EP
  - 〃　**やや高**：BT，姫野

9

実験結果

- NPB 2.3 EP



- NPB 2.3 BT

ノード数	NPB 2.3 BT 実行時間（秒）	
	AWG-STAR	SCore
1	47930	25
2	27322	357
4	17172	430

- AWG-STARシステムの性能がSCoreに及ばない
  - ⇒ 共有メモリアクセス性能に問題
  - NPB EP では4ノードで約20倍の性能差→ NPB BT では約40倍の性能差
  - 共有データへのアクセス頻度の高いベンチマークで性能差が広がる

10

AWG-STARシステムの共有メモリ

- 共有メモリへのアクセスは PCI バス経由
  - アクセス速度がローカルメモリに比べて低速
    - 平均速度：1MB/s　（ローカルメモリ：2GB/s）

⇒計算性能を大幅に低下させる

- 現状ではアプリケーションの計算時間の大部分を共有メモリへのアクセスが占める
  - 4ノードの場合：約96%
    - 共有メモリへの読み書きに要するアクセス時間とプログラム中での共有メモリアクセス回数から算出
- 現在次期AWG-STARシステムを開発中
  - 共有メモリアクセス速度をローカルメモリと同等に改善予定
    - 計算性能を大幅に改善する可能性がある

11

まとめ

- AWG-STARシステム上に並列プログラミング用 API OpenMPを実装
- ベンチマークで性能を評価
  - 現状では既存の計算環境と同等の性能は達成できなかった
  - 共有メモリの性能向上で大幅に改善する可能性がある
- 今後の課題
  - 共有メモリの性能を向上させた次期AWG-STARシステムでの実装
  - OpenMPの仕様の全ての機能を実装

12