λコンピューティング環境における OpenMPライブラリの設計と実装

大阪大学 大学院情報科学研究科
村田研究室 博士前期課程2年
合田 圭吾

発表内容

- 研究背景と目的
- AWG-STARシステムの概要
- 並列プログラミングAPI　OpenMPの概要
- AWG-STARシステム上でのOpenMPの設計
- ベンチマークによる性能評価
- まとめ

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
  - 複数の計算機やストレージなどをネットワークを介して接続し，1台の仮想計算機として機能させる技術
- TCP/IPを用いた通信の問題
  - パケット処理による遅延や輻輳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

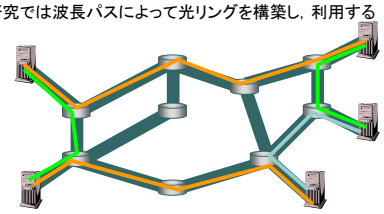
高速・高信頼な通信パイプをユーザに提供する新たな技術が必要

λコンピューティング環境の提案

- ノード計算機，ルータ間を光ファイバで接続
- 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

⇒ 高速・高信頼な通信を実現

- 本研究では波長パスによって光リングを構築し，利用する

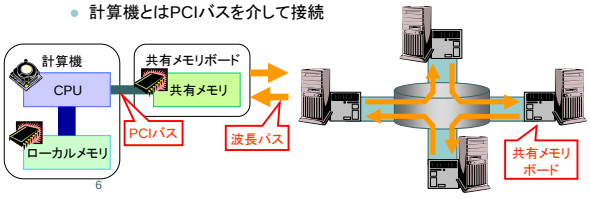


研究の目的

- 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
  - λコンピューティング環境上に共有メモリシステムを展開
    ⇒ AWG-STAR システム を利用
- AWG-STAR システム上にOpenMPを用いた分散並列計算環境を構築
  - 共有メモリシステムの性能をベンチマークアプリケーションを用いた実験によって明らかにする

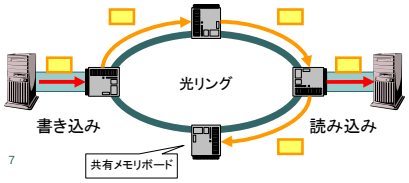
AWG-STAR システム

- 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同一に保たれる
    - CPU からはローカルメモリと同様にしてアクセスすることが可能
  - 計算機とはPCIバスを介して接続



AWG-STAR システムにおけるデータ共有

- 共有メモリボード上のメモリへのアクセスを通じてノード計算機間でデータを共有
  - 共有メモリへ書き込んだ内容は光リングを通じて他ノード計算機へ反映される
    - 反映処理はCPUとは独立して共有メモリボードが行う
  - 共有データの取得は共有メモリからの読み込みにより実現



7

OpenMPの概要

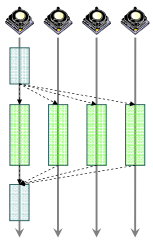
- 並列プログラミングAPIの標準規格
  - C言語（およびFortran）の拡張
  - ソースコード中にOpenMPディレクティブと呼ばれる指示文を記述することでプログラミング
  - 並列計算プログラムの動作の抽象的な記述のみでプログラミング可能
    - 並列プログラミングの専門的な知識のない人間でも比較的容易に習得可能

8

OpenMPによるプログラミング

- OpenMP ソースコード
  - 通常のプログラム中の並列処理可能な部分をディレクティブで指定する

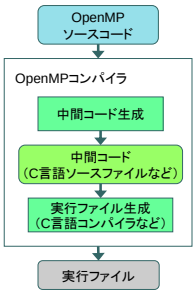
⇒ 指定したコードブロックが並列実行される



```
w = 1.0 / N;                                          逐次実行部
pi = 0.0;
#pragma omp parallel for private(i, local) reduction(+:pi)
for (i = 0; i < N; i++) {                             並列実行部
    local = (i + 0.5) * w;
    pi += 4.0 / (1.0 + local * local);
}
pi *= w;                                              逐次実行部
```

9

OpenMPの実現



- OpenMP はコンパイラによって実現される
  - ディレクティブを解釈し，スレッドライブラリの呼び出しなど並列処理に必要な処理を自動的にプログラム中に埋め込む
  - 生成されるコードは対象OSやハードウェア構成ごとに異なる
    - ⇒ 各環境ごとに専用のコンパイラが必要
- OpenMP コンパイラの動作
  - 中間コード生成
    - OpenMPソースコードから，スレッドライブラリ呼び出しなどのコードを埋め込んだプログラムコードを生成する
  - 実行ファイル生成
    - 生成された中間コードから実行ファイルを作成する

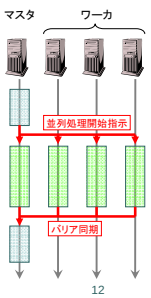
10

AWG-STARシステム上でのOpenMPの設計と実装

- AWG-STARシステム上にOpenMPを実現
  ⇒ AWG-STAR環境専用コンパイラの実装が必要
- OpenMPの実装に必要となる要素
  - 並列実行部の複数ノード計算機による並列処理
  - ノード計算機間でのデータ共有
  - ロック,バリア同期機構
- 既存のOpenMPコンパイラ OMPi をベースに開発
  - 生成される中間コードをAWG-STAR向けに改変

11

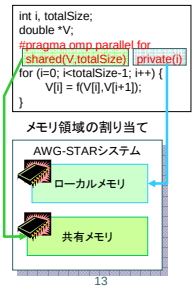
並列実行部の実行方法の設計



- マスタ・ワーカ方式
  - 全てのノードは同じ実行ファイルを実行する
  - ワーカノードはプログラムが起動されるとすぐに待機状態となってマスタノードからの指示を待つ
  - マスタは並列処理の準備ができるとワーカノードに処理の開始を指示する
    - ワーカノードに実行すべきプログラムコードのアドレスを与える
- プログラムの起動以降はノード間の通信には光リングのみを使う
  - マスタノードからワーカノードへの指示を低遅延で可能にする

12

データ共有の設計



- ノード計算機間でのデータ共有
  - プログラム中の変数を複数のノード計算機から読み書きできるようにする
- AWG-STARシステムの共有メモリの利用
  - 共有する必要のある変数のメモリ領域は共有メモリ領域に割り当てる
  - 共有する必要のない変数はローカルメモリに割り当てる
    - ローカルメモリの方がアクセスが高速
    - 共有するかどうかはOpenMPディレクティブのパラメータなどから判別可能

13

ロック，バリア同期の設計

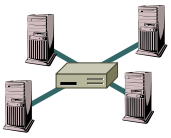
- ロック，バリア同期
  - 共有データアクセスに対する排他制御，計算の実行タイミングの同期を提供する機能
  - AWG-STARシステムの光リングを用いた専用のロックおよびバリア同期機構を用意[8]
    - 光リングを利用することにより低遅延でノード計算機間の同期を実現
  - コンパイラが生成した中間コードからこのAWG-STARシステム専用同期機構を呼び出すようにする

[8] 井本 舞，合田 圭吾，馬場 健一，村田 正幸：“λコンピューティング環境におけるOpenMPライブラリのためのデータ共有機構の設計”，電子情報通信学会技術研究報告(PN2006-28)，pp.19-24 (2006).

14

ベンチマークによる性能評価

- 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: 書き込み60MB/s 読み込み67MB/s



- 性能比較対象：SCore
  - Linux用クラスタミドルウェア
  - 専用のOpenMP実装Omni/SCASHを備える
    - Gigabit Ethernet + ソフトウェア分散共有メモリ
      - 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

15

評価に用いるベンチマーク

- 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)

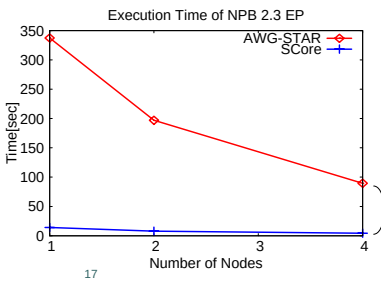
これら2つのベンチマーク結果について説明

- 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

16

実験結果

- NPB 2.3 EP



- 比較対象のSCoreでの計算時間がAWG-STAR よりも短くなっている
- AWG-STARにおいてもノード数が増加するに従って計算時間が短縮されており，並列化による性能向上が確認できる

約20倍の性能差

17

実験結果

- NPB 2.3 BT
  - NPB 2.3 EP と同じような傾向
    - AWG-STAR の計算時間はSCoreより長い
      - AWG-STAR，SCore間の性能差はEPと比較して拡大
    - AWG-STARではノード数を増やすに従って計算時間が短縮される

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

約40倍の性能差

18

考察

- 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
  - AWG-STARシステムの共有メモリボード上のメモリへはPCIバスを経由してアクセス
    - 高速にアクセスできないため性能上のボトルネックに
    - 共有メモリアクセスのボトルネックを解消することで性能が向上する可能性がある

19

まとめ

- AWG-STARシステム上に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
- 今後の課題
  - 現状OpenMPの仕様の全ての機能を実装できていない
  - 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

20