Advanced Network Architecture research group

λコンピューティング環境構築のための Globus Toolkit を用いた MPI ライブラリの実装と評価

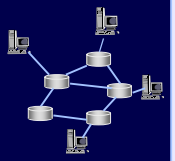
大阪大学 大学院情報科学研究科
井本 舞

発表内容

- 研究の背景
- 研究の目的
- 共有メモリシステムを利用した MPI ライブラリの実装方法
- 性能評価
- まとめと今後の課題

研究の背景

- グリッドコンピューティング
  - ネットワークを介して複数の計算機を接続し、計算資源、ストレージを共有
    - 広域で大規模な計算
    - 大容量データの転送
- 通信オーバーヘッドが問題
  - TCP/IP が通信に使われる
    - パケット処理によるオーバーヘッド
    - パケットロスによる再送遅延

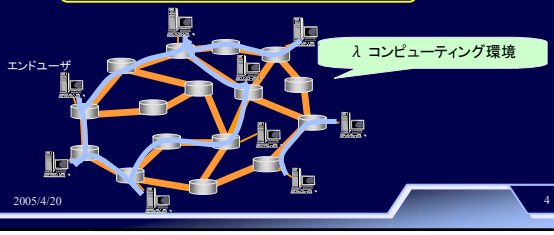


高速かつ、高信頼な通信パイプをエンドユーザに提供する新たな技術が必要

λコンピューティング環境の提案

- 計算機、ルータを光ファイバで接続する
- 波長パスを張り、波長パスを通信の最小粒度とする

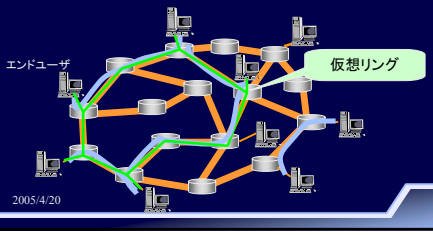
パケット処理のオーバーヘッドを削除



λコンピューティング環境の提案

- 波長パスを利用して仮想リングを構成する

仮想光リングを構成することで
光リングを専用の高速通信路として利用可能



Globus Toolkit による グリッド環境構築

- Globus Toolkit によってグリッド環境を構築
  - グリッド環境構築のためのミドルウェア
  - 通信、認証、ジョブ管理などを行う
  - ユーザには、各計算機の実装に依存しないインターフェースを提供する
- Globus Toolkit の下位層である通信には、λコンピューティング環境を使用
  - 高速な通信が可能となる

λコンピューティング環境に Globus Toolkit を導入することにより、ユーザはλコンピューティング環境を意識することなく高速な分散計算環境を使うことができる

研究の目的

λコンピューティング環境において並列計算を行うことを対象とする

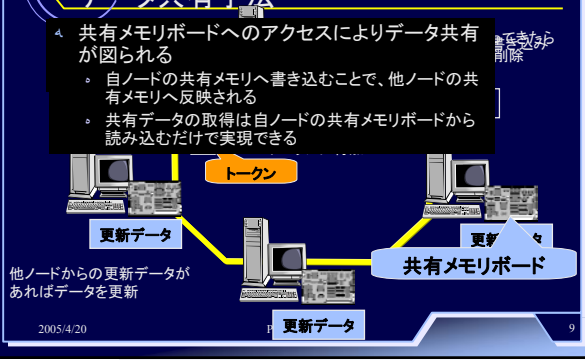
- λコンピューティング環境に Globus Toolkit を導入
- λコンピューティング環境における MPI ライブラリの実装と評価
  - NTT フォトニクス研究所が開発した AWG-STAR システムを利用

AWG-STAR システム

- 各ノードは波長ルータ（AWG）に接続し光リングネットワークを構成
- 各ノードは共有メモリボードを搭載
  - 各共有メモリボードは同じデータを保持

AWG-STARにおけるデータ共有手法

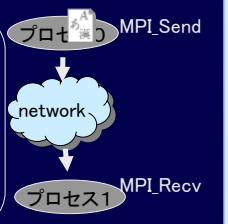
- 共有メモリボードへのアクセスによりデータ共有が図られる
  - 自ノードの共有メモリへ書き込むことで、他ノードの共有メモリへ反映される
  - 共有データの取得は自ノードの共有メモリボードから読み込むだけで実現できる



MPI (Message Passing Interface)

- 並列計算ではメッセージ交換を行いながら計算を進める
- MPI はプロセス間でメッセージを交換するための仕様

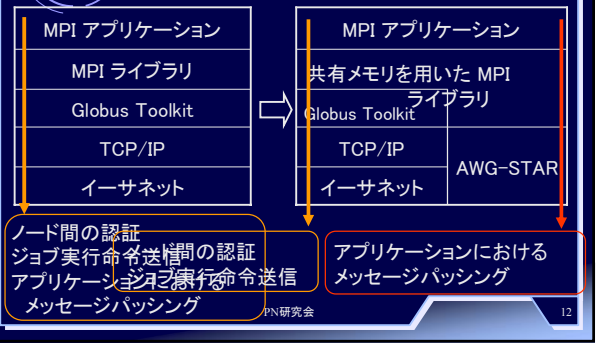
```
If(my_rank == 0){
  sprintf(message,"%s", Hello);
  MPI_Send ( 送信先ランク：1
             メッセージ：message);
}
 else if (my_rank == 1){
  MPI_Recv ( 送信元ランク：0
             メッセージ：message);
}
```



MPI ライブラリの実装方法

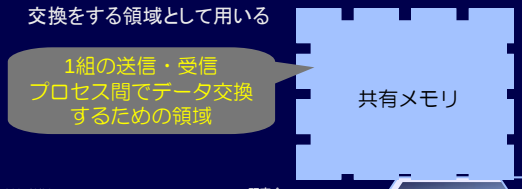
- すべてのライブラリをスクラッチから作成する方法
- 既にあるライブラリを基にして作成する方法
  - Globus Toolkit 上で動作する MPICH-G2 の手法を 利用
    - 認証、ジョブ命令送信などは MPICH-G2 と同じ
    - 通信を TCP/IP から AWG-STAR を用いたものに変更

λコンピューティング環境における並列計算



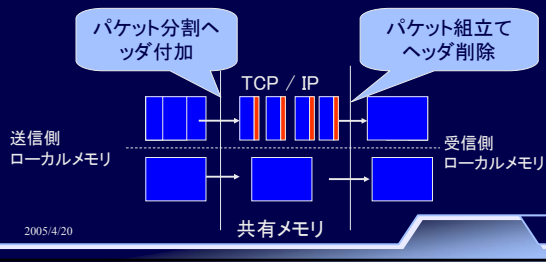
共有メモリを用いた MPI ライブラリの実装

- 共有メモリ上で動的にメモリを割り当てることが できない
  - プロセスの個数をnとすると、共有メモリを$n \times n$に分割
  - 一つの領域を一組の送信/受信プロセス間でのデータ交換をする領域として用いる



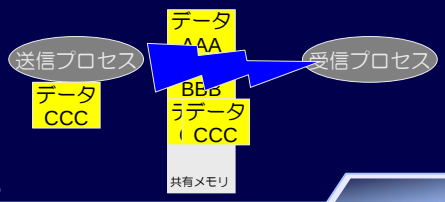
共有メモリを用いた メッセージパッシング方法

- 共有メモリ上でデータを交換する
  - データをまるごと共有メモリに書き込むことが可能



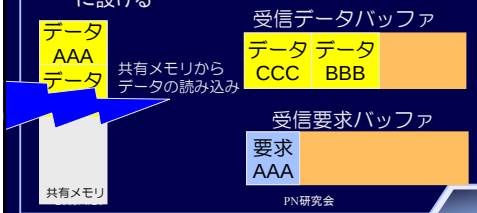
メッセージ送信側の動作

- 送信関数が呼ばれると、送信データを共有メモリに書きこむ
- 送信データを書き込んだ後、受信プロセスにシグナルを送る
  - シグナルは AWG-STAR が提供する機能
  - 任意のプロセスに送信できる



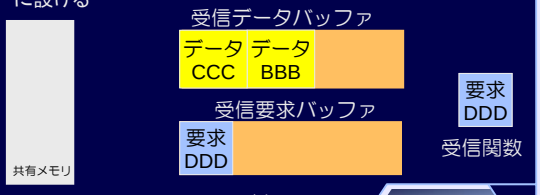
メッセージ受信側の動作

- 受信関数が呼ばれるタイミングと、データを受信 するタイミングが異なることを考慮
  - 受信データバッファと受信要求バッファをローカル メモリ上に設ける



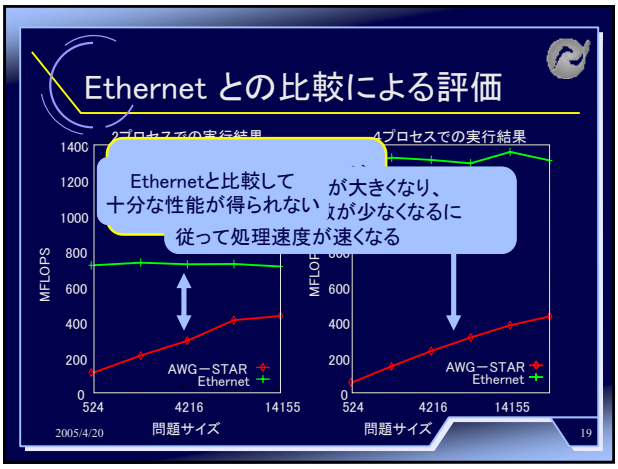
メッセージ受信側の動作

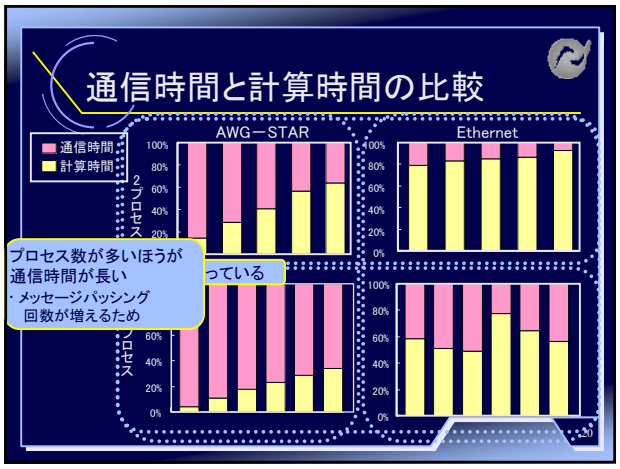
- 受信関数が呼ばれるタイミングと、データを受信 するタイミングが異なることを考慮
  - 受信データバッファと受信要求バッファをローカル メモリ上に設ける

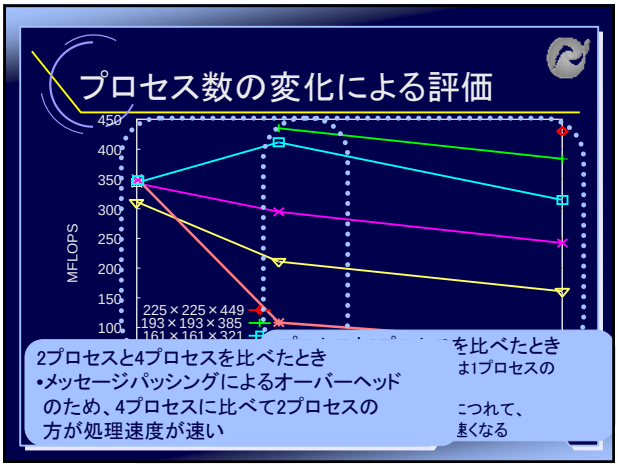


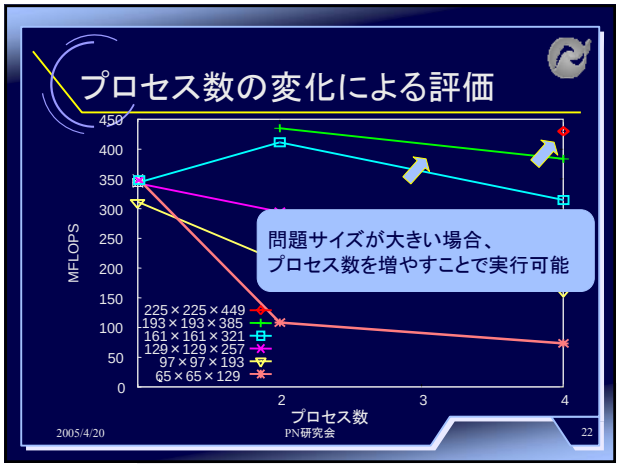
アプリケーションによる評価

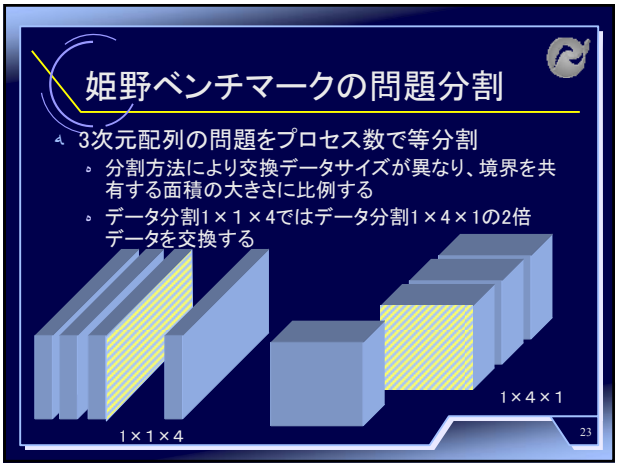
- 実験環境：ノード計算機最大4台で実行
  - 1ノード計算機上で1プロセス
- 評価アプリケーション：姫野ベンチマーク
  - ポアソン方程式をヤコビ法で解くときの処理速度を 計測するベンチマーク
  - 3次元配列の問題をプロセス数で等分割して並列計算
  - データ交換に MPI を利用
  - 様々な問題サイズを設定できる
    - 問題サイズと交換するデータサイズが比例
    - 問題サイズとデータ交換の回数が反比例

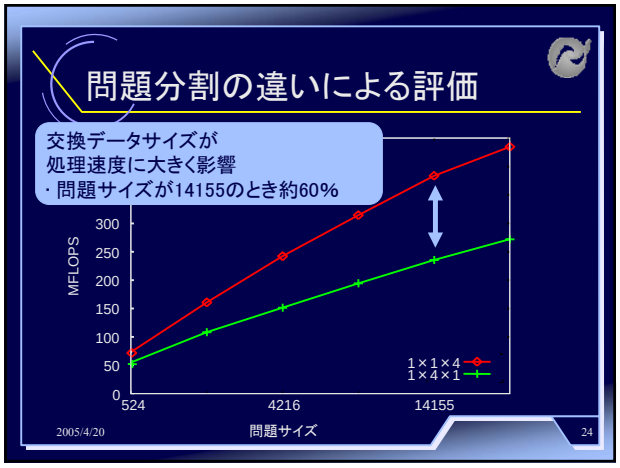

Ethernet との比較による評価
4プロセスでの実行結果
Ethernetと比較して
十分な性能が得られない
が大きくなり、
が少なくなるに
従って処理速度が速くなる
MFLOPS
AWG－STAR
Ethernet
524
4216
14155
問題サイズ
2005/4/20
19

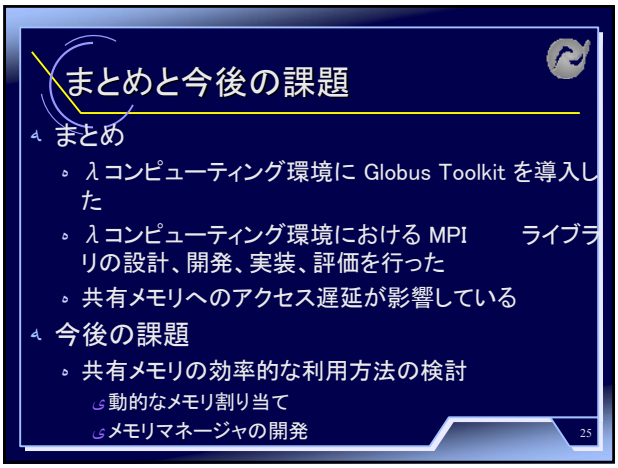

通信時間と計算時間の比較
通信時間
計算時間
AWG－STAR
Ethernet
2プロセス
プロセス数が多いほうが
通信時間が長い
・メッセージパッシング
回数が増えるため


プロセス数の変化による評価
MFLOPS
225×225×449
193×193×385
161×161×321
2プロセスと4プロセスを比べたとき
•メッセージパッシングによるオーバーヘッド
のため、4プロセスに比べて2プロセスの
方が処理速度が速い


プロセス数の変化による評価
MFLOPS
225×225×449
193×193×385
161×161×321
129×129×257
97×97×193
65×65×129
問題サイズが大きい場合、
プロセス数を増やすことで実行可能
プロセス数
2005/4/20
PN研究会
22


姫野ベンチマークの問題分割
3次元配列の問題をプロセス数で等分割
分割方法により交換データサイズが異なり、境界を共有する面積の大きさに比例する
データ分割1×1×4ではデータ分割1×4×1の2倍データを交換する
1×4×1
1×1×4
23


問題分割の違いによる評価
交換データサイズが
処理速度に大きく影響
・問題サイズが14155のとき約60%
MFLOPS
1×1×4
1×4×1
524
4216
14155
問題サイズ
2005/4/20
24


まとめと今後の課題
まとめ
λコンピューティング環境に Globus Toolkit を導入した
λコンピューティング環境における MPI ライブラリの設計、開発、実装、評価を行った
共有メモリへのアクセス遅延が影響している
今後の課題
共有メモリの効率的な利用方法の検討
動的なメモリ割り当て
メモリマネージャの開発
25