省メモリ技術と動的最適化技術によるスケーラブル通信ライブラリの開発

九州大学情報基盤研究開発センター

南里　豪志

2012年11月2日

CREST「ポストペタスケール高性能計算に資するシステムソフトウェア技術の創出」
領域会議

研究のねらい

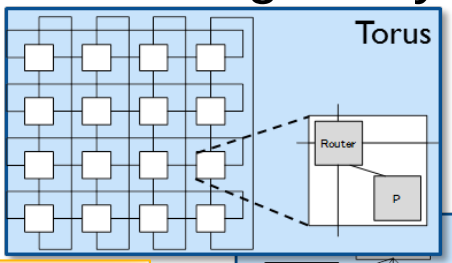
- エクサスケール計算環境での利用に耐える、

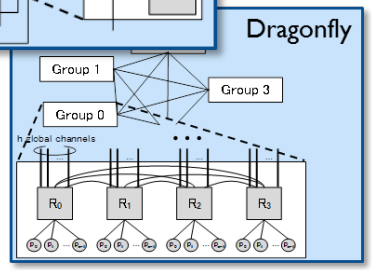
  「スケーラブルな通信ライブラリ実装技術」

  「スケーラブルな並列アプリケーション作成のための通信ライブラリ応用技術」

  の開発

- 想定する計算環境:
  数十万ノードの多次元トーラスもしくは high-radix網（Dragon-Fly等）
  - コア数: 数十コア～数百コア/ノード
  - メモリ: 数百GB/ノード
  - プログラミングモデル: MPI + スレッド or PGAS
  - プロセス数: ～数十プロセス



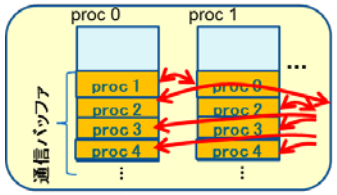


通信ライブラリへの性能要求 ⇒ 増

プロセス当たりメモリ量 ⇒ 減

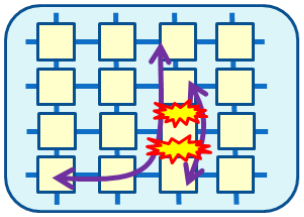
研究で取り組む課題

- 「数億プロセスに耐えるメモリ管理機構」
  - 従来：各プロセスで全プロセスの情報管理や通信相手毎のバッファ用意
    ⇒ 最悪の場合、プロセス当たり 100GB以上を使用
  - 想定計算環境では、プロセスあたりメモリ容量 1～10GB程度



各プロセス用に
通信バッファを用意

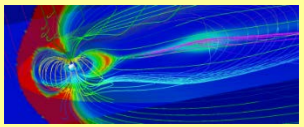
- 「高並列計算環境での通信ライブラリ動的最適化技術」
  - 従来：実行前の静的情報に基づく、人手によるチューニング
  - 想定計算環境では、
    i) プロセス数の増加に伴い最適化の探索空間が爆発的に拡大
    ii) プロセス配置などの実行時の状況による性能変動

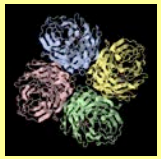

通信衝突

- 「アプリケーションと通信ライブラリの連携」
  - 従来：通信ライブラリ層に通信以外のアプリケーションの情報は伝わらない
  - 想定計算環境では、
    アプリケーションでの通信・計算のパターンに応じた実装の効率化が必要

ACEライブラリ

新規開発アプリケーション：
ACEライブラリのインタフェースを
活用したスケーラブルな実装





既存アプリケーション

既存ライブラリ
（MPI、ARMCI、GASNet、etc.）

ACEライブラリ

高レベルインタフェース：
計算＋通信の効率化

低レベルインタフェース：
隣接通信、集団通信の省メモリ・動的最適化

省メモリ通信プロトコル：
通信バッファを削減した基本通信

パケット制御インタフェース：
パケット送信間隔の制御

アプリケーション
グループ

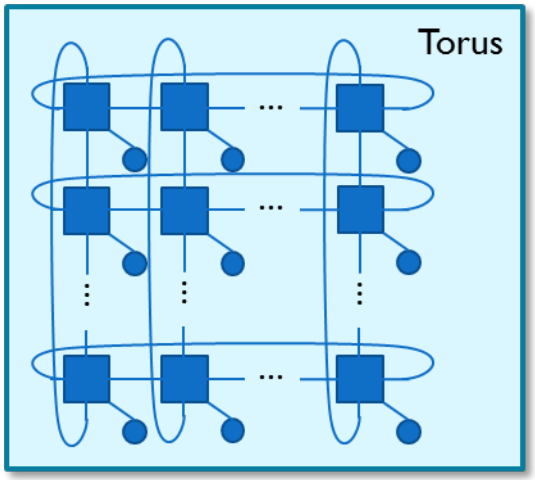
インタフェース
グループ

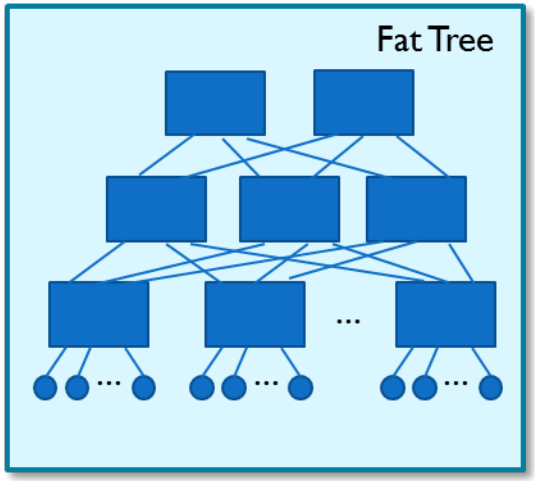
プロトコル
グループ

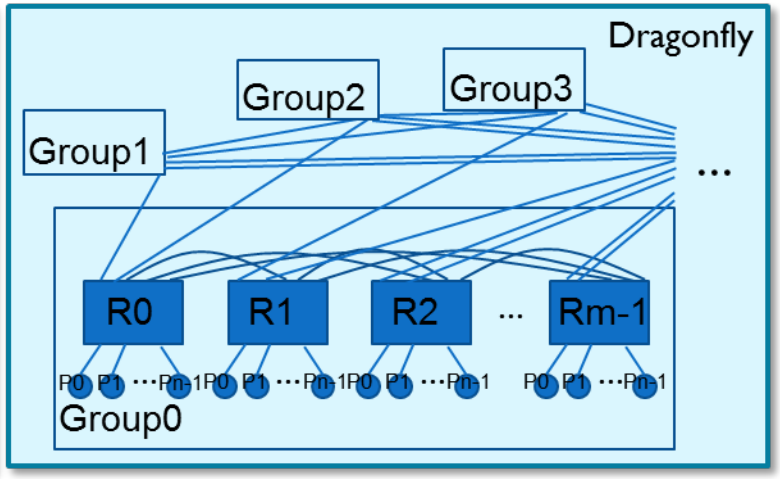
パケット制御
グループ

インタフェースグループ（九州大学　南里、森江、α）

- 研究の目標:
  スケーラブルな隣接通信及び集団通信のための省メモリアルゴリズム及び動的最適化技術
  - 実行時の状況に合わせて最適な実装を選択できる動的最適化機構
  - 使用するバッファ領域サイズがプロセス数に比例して増加しない省メモリ実装技術

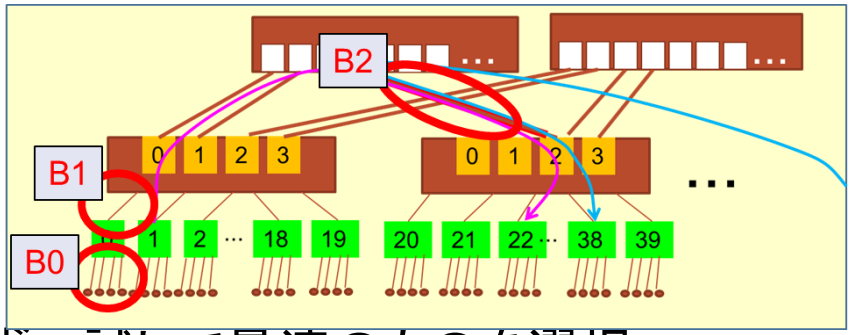






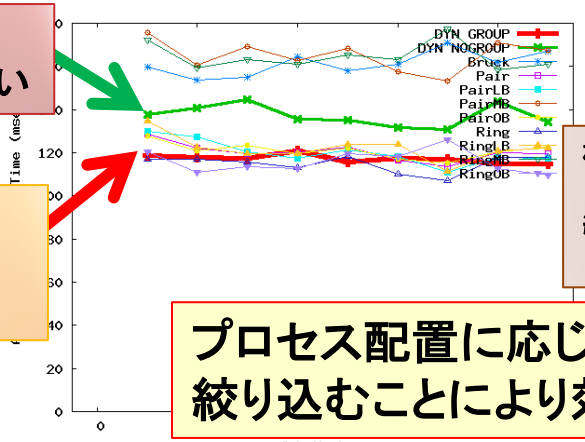
現在までの成果：集団通信アルゴリズムの動的選択技術

- Fat Tree網を対象とした集団通信アルゴリズム選択技術
  - プロセスの配置から、最も混雑すると予測される経路の平均バンド幅を予測
    ⇒ 各アルゴリズムの性能を予測
  - 予測結果に基づき、選択候補のアルゴリズムを絞り込み
  - 実行中に候補のアルゴリズムを一つずつ試して最速のものを選択



プロセス配置を考慮しない場合：
最速のアルゴリズムより大幅に遅い

プロセス配置を考慮した場合：
最速のアルゴリズムとほぼ同等の性能



横軸：ジョブ
（ジョブ毎にプロセス配置が変化）
縦軸：各アルゴリズムおよび提案技術の所要時間

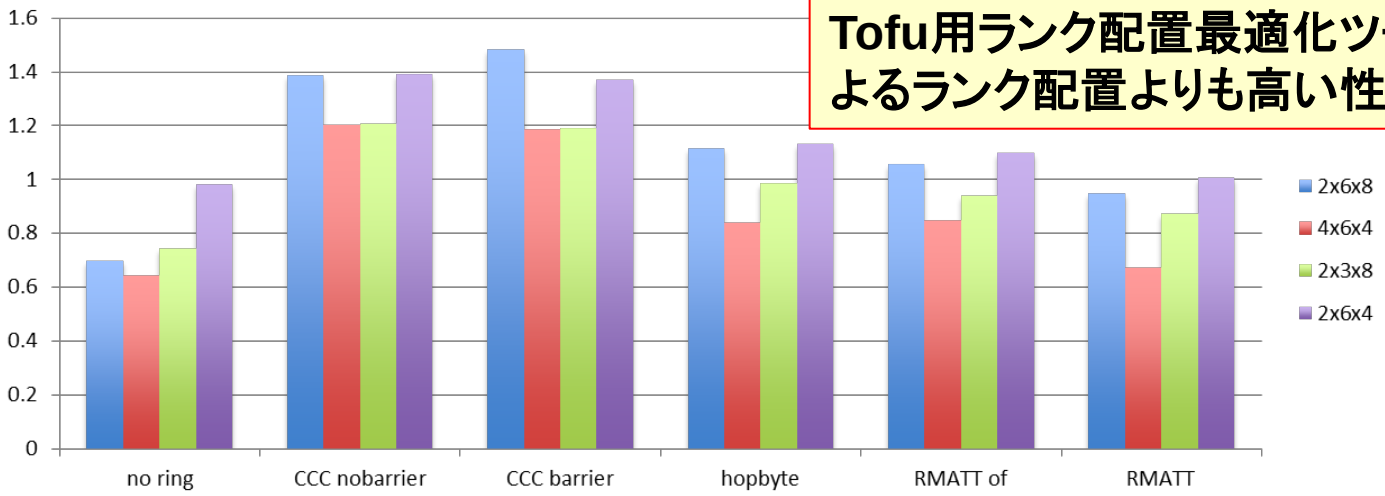
プロセス配置に応じてアルゴリズムを絞り込むことにより効率的な選択を実現

- Torus網におけるプロセス配置形状に応じた集団通信アルゴリズムの性能変動調査

現在までの成果：通信衝突を考慮したランク配置最適化

- 通信衝突によるペナルティ予測技術の提案およびそれにもとづいたランク配置最適化技術の開発
  - 通信間の依存関係より、同時に実行される通信群を抽出
  - 各通信群について、ランク配置による通信衝突ペナルティを予測
  - 全通信群での通信衝突ペナルティを目的関数としてランク配置最適化
    - simulated annealing
  - Tofuにおける効果：

ホップ数とバイト数による手法（hobbyte）やTofu用ランク配置最適化ツール(RMATT)によるランク配置よりも高い性能

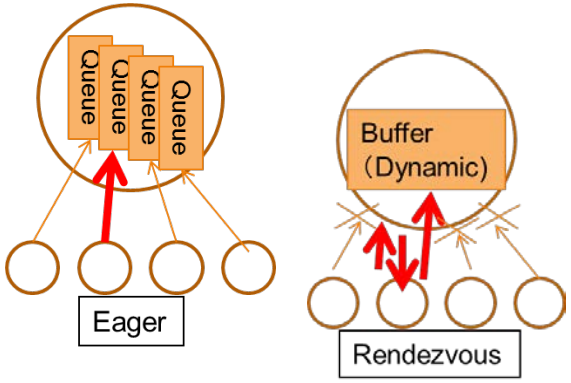


プロトコルグループ（富士通　住元、安島、三浦、秋元、岡本）

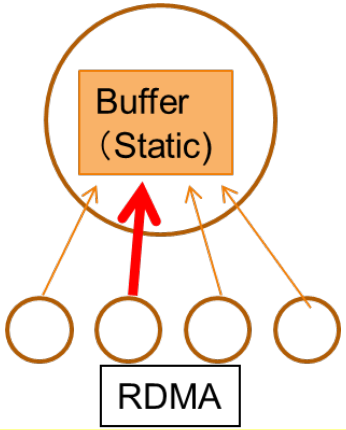
- 研究の目標:

通信バッファを削減した通信モデルにもとづいた通信プロトコル

- 数千万～数億プロセスに耐える省メモリ通信レイヤを実現するために、遠隔Atomic操作、Put/Get等の片側通信を利用し、通信資源の動的管理と低遅延を両立する技術を開発する



従来のプロトコル

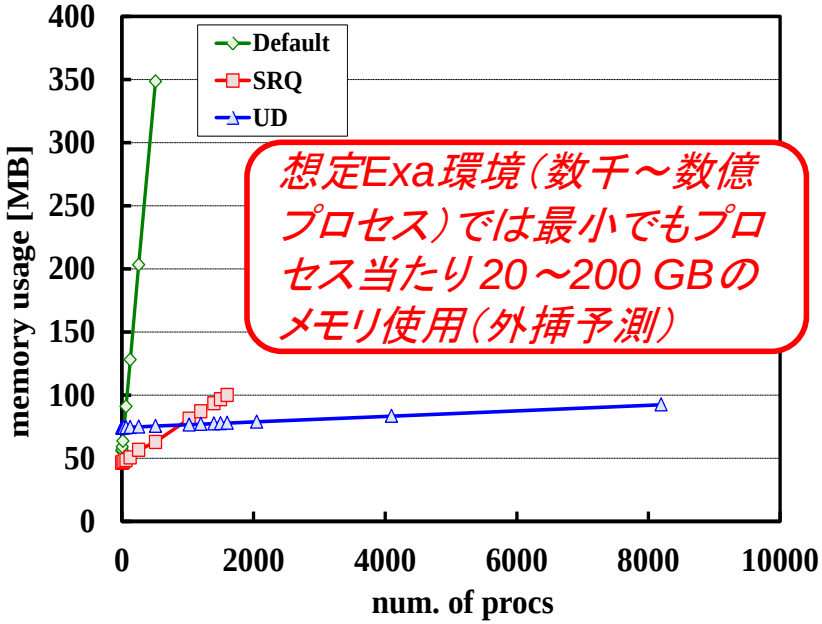


RDMA,Atomic操作による省メモリプロトコル

現在までの成果：基礎的な片側通信ライブラリの確立

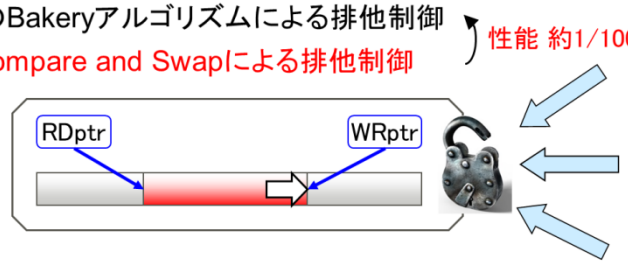
- 既存研究・公開MPIライブラリの評価を行い、本研究の重要性を再認識
  - 想定Exa環境ではライブラリのメモリ消費によりアプリ実行が艱難。削減が必須
- 遠隔Atomic操作と片側通信を用いたグローバルデータ構造の効率的な操作方法を検討・提案
  - 性能確保には通信ハードによる遠隔Atomic操作が必要（非サポートでは 1/1000倍）
  - リモート順序保証が有ることにより性能が更に3倍程度改善（リングバッファ方式時）

既存MPIのメモリ使用量測定結果

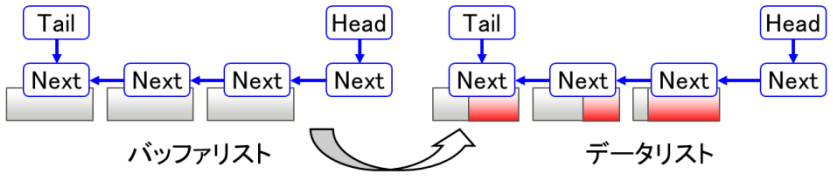


共有受信キューへのデータ送信操作方式検討

1. リングバッファ
   a. LamportのBakeryアルゴリズムによる排他制御
   b. Atomic Compare and Swapによる排他制御



2. バッファリスト ＋ データリスト
   a. Atomic Compare and Swapによるロックフリーキュー



現在までの成果：ライブラリの動的メモリ使用量測定

H24年度目標:

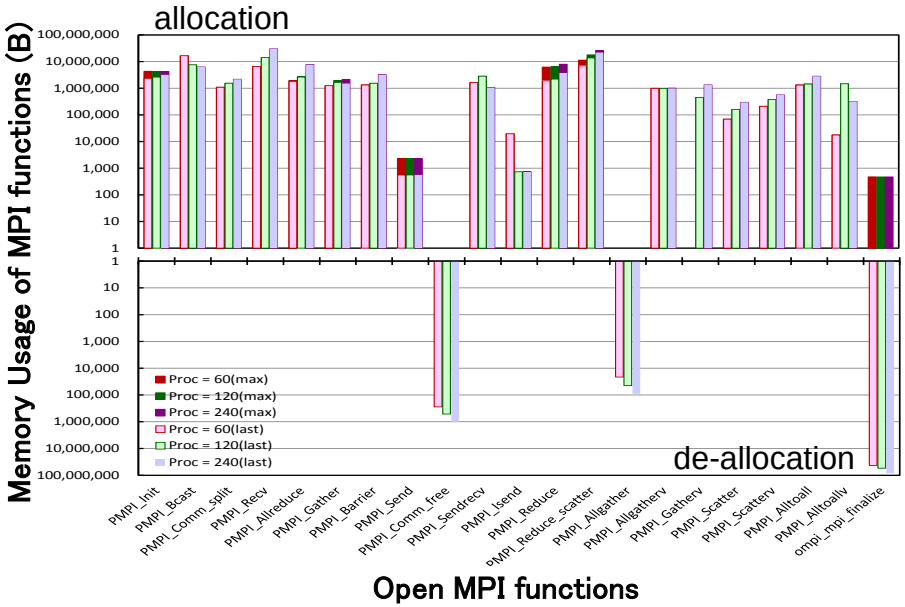
メッセージパッシング通信プロトコル低遅延・省メモリ化技術

- 既存/開発ライブラリの動的メモリ使用量の測定・分析手段を確立
  - アプリケーション実行中の動的メモリ使用量をライブラリ（通信）や関数別に分類集計・測定するツールを作成

```
---- Statistics of individual thread memory usage ----------
Thread ID: 3517
mem_size = 773,720, mem_max = 547,195,816
malloc: 970060, realloc: 518, memalign: 786, free: 967877

  ---- Statistics of individual library memory usage -------
  Library: /home/akimoto/OpenMPI/lib/libmpi.so.1
  mem_size = 791,680, mem_max = 43,866,296   MPI使用分
  malloc: 969573, realloc: 518, memalign: 786, free: 967395
  ------------------------------------------------------------
  Library: /lib64/libc.so.6                  アプリ使用分
  mem_size = 384, mem_max = 503,329,424
  malloc: 483, realloc: 0, memalign: 0, free: 477
  ------------------------------------------------------------
```

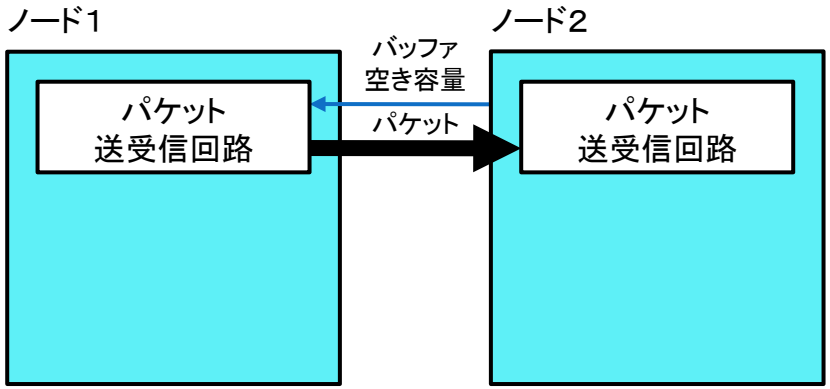
メインスレッドのライブラリ別のメモリ使用量（例）
（非測定アプリ: IMB-MPI1）

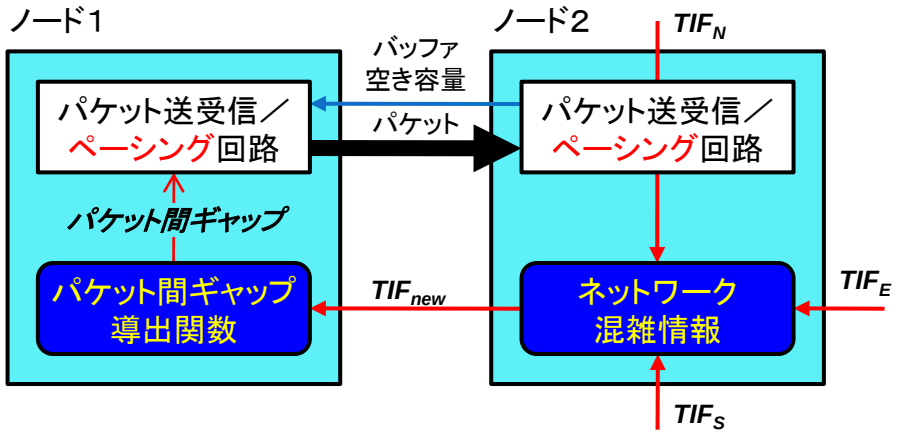


MPI関数別のメモリ資料量（例）
（非測定アプリ: IMB-MPI1）

パケット制御グループ（九州先端研/ISIT　柴村、薄田）

- 研究の目標：
  実行時の状況に応じてパケット送信間隔を動的に制御する通信最適化技術
  - 通信クリティカルパスのスループットを最大化し、通信混雑を緩和するパケットペーシング技術の開発
  - パケットペーシングによる集団通信や隣接通信の高速化技術の開発
  - 時々刻々と変化する通信状況に応じて、パケット間ギャップを自動的に変化させるパケットペーシングの自動化技術の開発

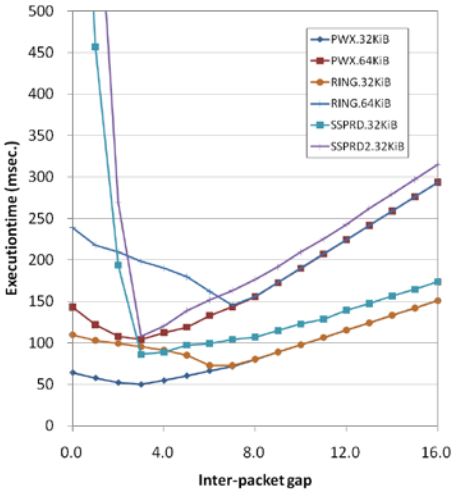




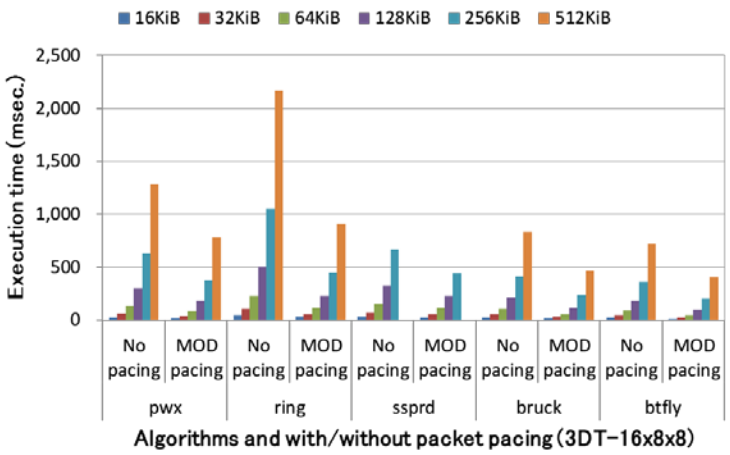
現在までの成果：パケットペーシングの基礎評価

- 既存の集団通信アルゴリズムを用いたパケットペーシングの基礎評価
  - 通信時間を最小化させるパケット間ギャップ値を調査
    - 集団通信アルゴリズム、ノード数、メッセージサイズ、パケット間ギャップ値を変え、インターコネクトシミュレータNSIMを用いて評価
    - 一般的な集団通信アルゴリズムに対するパケットペーシングの有効性を確認
  - パケット間ギャップ導出式を設計し、シミュレーションによって検証
    - 各アルゴリズムにおいてギャップ導出式に基づいたペーシングの効果を確認
    - ノード数やメッセージサイズの増加に対して速度向上率も増加（**ポストペタ級システムに有効**）

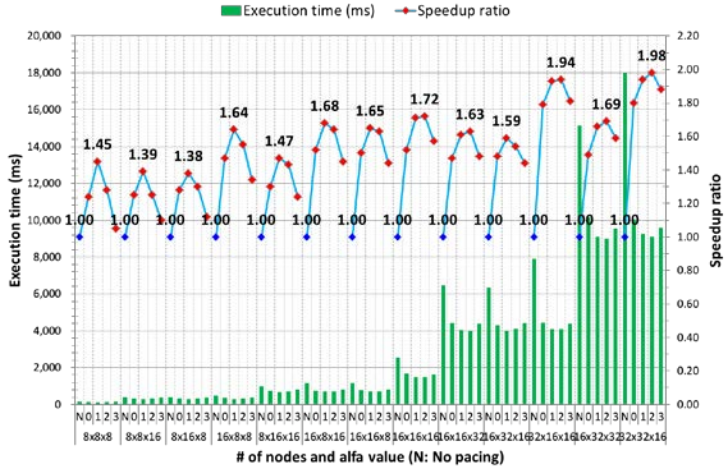
ギャップ値 対 実行時間



ギャップ導出式に基づいたペーシング



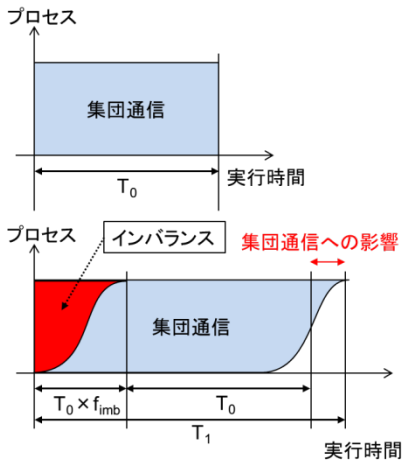
ノード数に応じたペーシングの効果



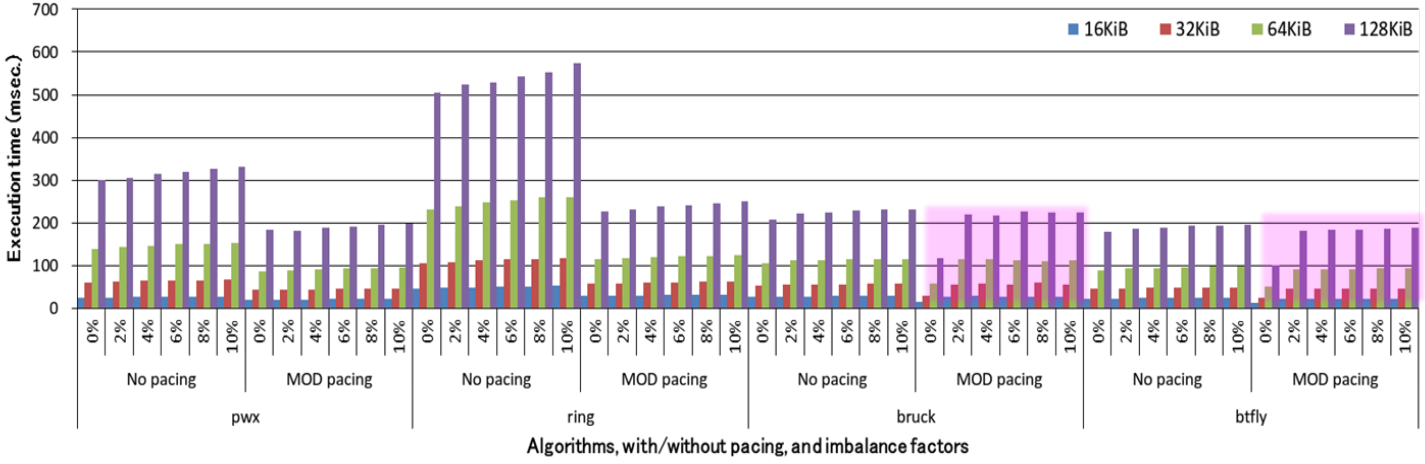
現在までの成果：パケットペーシングの基礎評価

- 集団通信に対するロード／ネットワークインバランスを考慮した評価
  - 通信開始時刻のインバランスが、パケットペーシングを用いた集団通信の実行に与える影響をシミュレーションによって評価
    - 集団通信開始前に各プロセスへn%のインバランス（無通信・無演算時間）を挿入
    - 集団通信アルゴリズム、トポロジ、メッセージサイズ、ノード数、に応じて「インバランスへの感受性」が異なることがわかった
    - インバランスによってペーシングの効果を損なう場合がある

インバランスの挿入



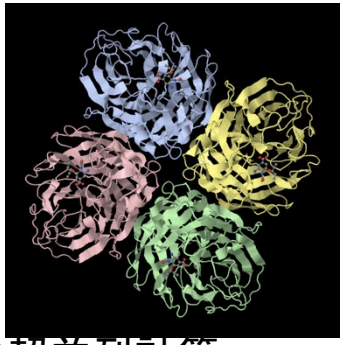
ペーシングを適用した集団通信に対するインバランスの影響



アプリケーショングループ（九州大学　高見、稲富、深沢、本田）

- 研究の目標：
  スケーラブルな並列アプリケーション作成のための通信ライブラリ応用技術
  - 高並列で実行できる実アプリケーションのチューニング・実装経験から問題点の洗い出し
    - OpenFMOプログラム
    - 電磁流体プログラム
      etc.
  - 現状の大規模並列計算機での詳細な性能測定結果と、性能評価モデルの比較
  - 代表的な計算・通信のパターンを抽出し、上位ライブラリとして設計
  - 新規ライブラリを利用する場合の性能評価とエクサスケールでの性能予測

現在までの成果
OpenFMOプログラム



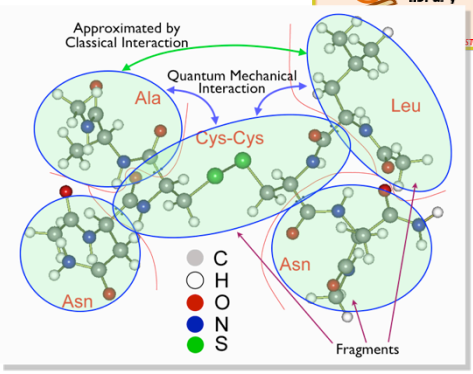
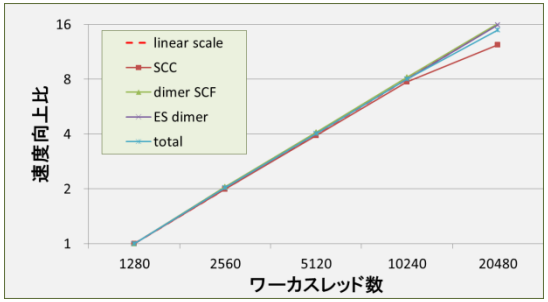


Fig. Flow of the FMO method

- プログラムの特徴:
  - 大規模分子の量子化学計算を
    Many-Task Computingの形で実現
  - 複数ノードのグループ化による効率的で堅牢な超並列計算
- 上位ACEライブラリとしての研究・開発項目:
  - 動的負荷分散の効率的な支援
  - 効率的な共有記憶機構
- 高並列計算時の性能解析

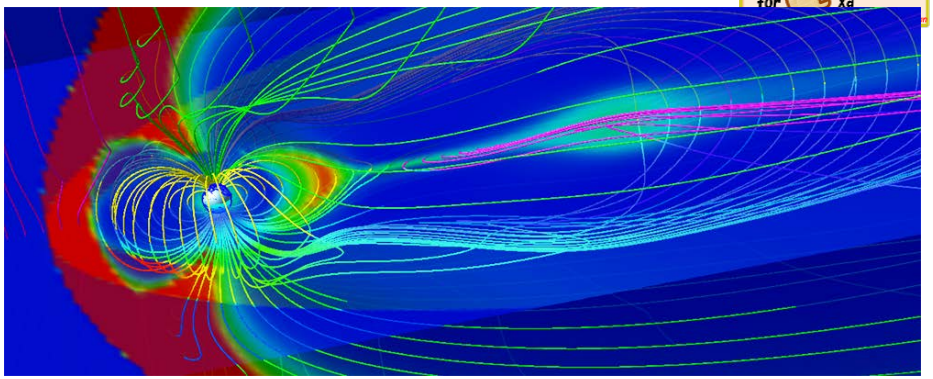


- 動的負荷分散技術の導入と効果の計測:

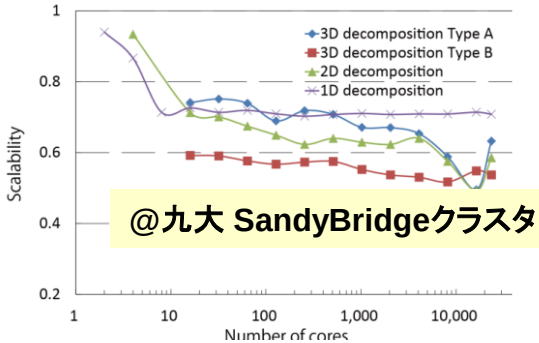
動的負荷分散導入前

動的負荷分散導入後

現在までの成果：電磁流体プログラム



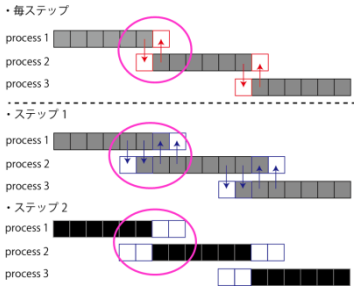
- プログラムの特徴:
  - 惑星磁気圏のMHDシミュレーション
  - 1～3次元領域分割による大規模並列計算
  - 比較的小規模な袖領域のみの隣接ノード間通信
- 上位ACEライブラリとしての研究項目:
  - 袖領域のPack/Unpack機能
  - 非同期通信の利用による隣接通信と計算のオーバーラップ
- 異なるアーキテクチャの計算機での実効性能比較

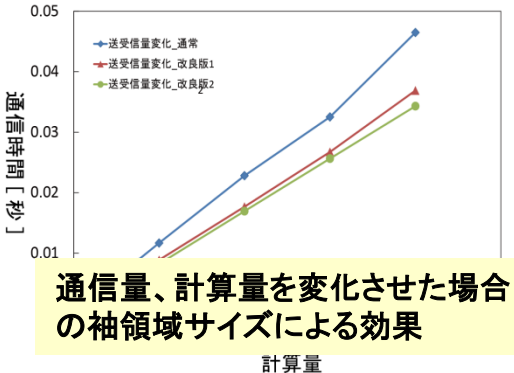


@九大 SandyBridgeクラスタ

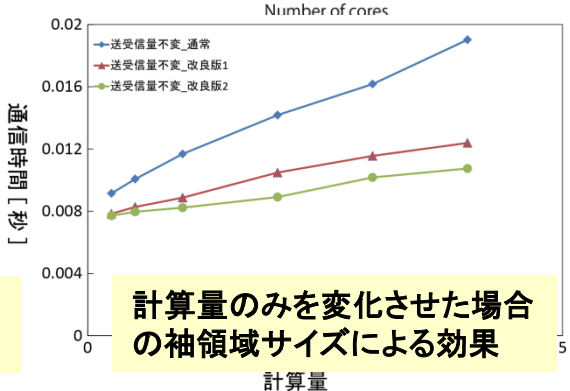
@九大 FX10

- 袖領域サイズの変化による通信効率の計測





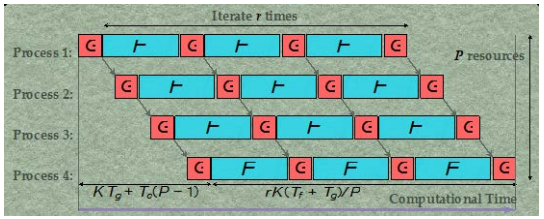
通信量、計算量を変化させた場合の袖領域サイズによる効果

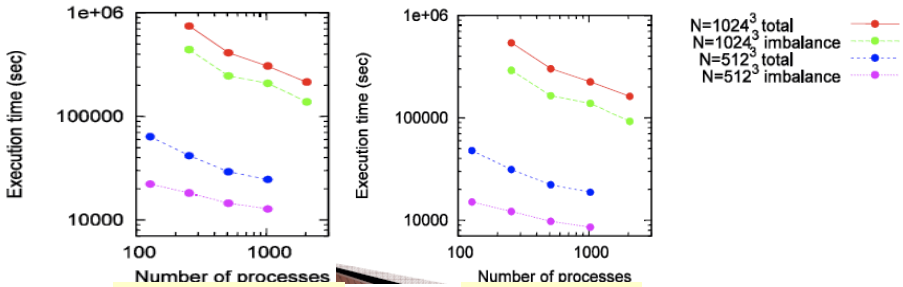


計算量のみを変化させた場合の袖領域サイズによる効果

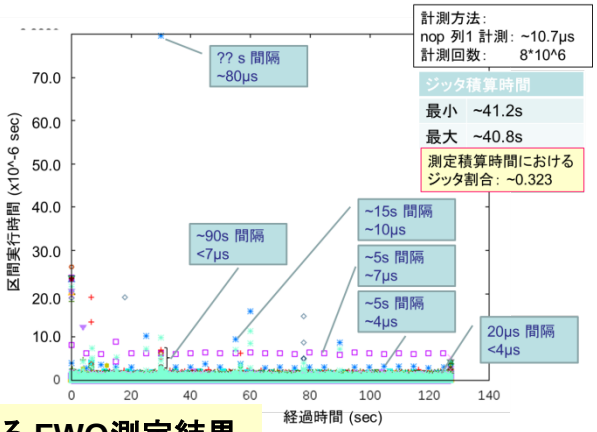
現在までの成果：既存並列アプリケーションの調査

- 依存関係のある計算の並列化(Parareal法)
  - 空間並列度に限界のある計算を時間方向に並列化
  - パイプライン的な並列化パターン
  - Aceライブラリとしての研究・開発項目
    - パイプライン並列化、グループ間通信、連成計算パターンのサポート
- N体問題プログラム
  - 適応的な負荷分散技術を用いた実装におけるスケーラビリティの調査
- 既存アプリケーションの調査:
  - 超並列計算の現状と将来に向けての見通し
    - SDHPC「計算科学研究ロードマップ白書」、IESP - White Papers
  - 既存高並列アプリケーションにおける重要な計算パターン:
    - 線形計算ライブラリの性能に依存するもの(FFT, dgemm, 等)
    - 領域分割による計算(1～3次元分割、袖領域の隣接通信)
    - Master-Worker的な実行パターン(Many-Task Computing)
- 高並列計算環境に向けた性能解析・予測技術
  - ジッタによる影響の解析技術
    - ジッタ測定、ハードウェアカウンタ測定





Fujitsu FX10　　Cray XE6



FX10 1ノードにおける FWQ測定結果

全体スケジュール

要素技術研究開発

- 隣接通信・集団通信動的最適化
- 省メモリメッセージパッシング通信プロトコル
- パターン通信向けパケットペーシング
- アプリケーションの通信隠蔽、動的負荷分散
- アプリケーション調査

- 隣接通信・集団通信省メモリ化、非ブロッキング通信動的最適化
- PGASライブラリの低遅延化省メモリ化
- パケット送信間隔の動的最適化
- 上位レベルインタフェースの開発

ACEライブラリ構築

- 各基本通信インタフェースの構築
- 各動的最適化技術の統合
- 省メモリプロトコルを活用した実装技術
- 上位レベルインタフェースの構築

公開・評価・フィードバック

- ACEライブラリを活用したアプリケーション作成
- 省メモリ性、動的最適化性能の検証
- Exaスケールでの性能予測
- フィードバックに基づいた改良、機能拡張

2011 2012 2013 2014 2015 2016 年度