



XcalableMP(XMP) の開発状況

佐藤三久

筑波大学・理研AICS

“Petascale” 並列プログラミング WG (XcalableMP WG)



■ 目的

- “標準的な”並列プログラミングのためのペタスケールを目指した並列プログラミング言語の仕様を策定する
- “標準化”を目指して、“world-wide” communityに提案する。

■ Members（発足当時）

- Academia: M. Sato, T. Boku (compiler and system, U. Tsukuba), K. Nakajima (app. and programming, U. Tokyo), Nanri (system, Kyusyu U.), Okabe, Yasugi (HPF, Kyoto U.)
- Research Lab.: Watanabe and Yokokawa (RIKEN), Sakagami (app. and HPF, NIFS), Matsuo (app., JAXA), Uehara (app., JAMSTEC/ES)
- Industries: Iwashita and Hotta (HPF and XPFortran, Fujitsu), Murai and Seo (HPF, NEC), Anzaki and Negishi (Hitachi)

■ 2007年12月にkick-off, 現在、e-scienceプロジェクトの並列プログラミング検討委員会に移行

■ メーカーからのコメント・要望（活動開始時）

- 科学技術アプリケーション向けだけでなく、組み込みのマルチコアでも使えるようなものにするべき。
- 国内の標準化だけでなく、world-wideな標準を目指す戦略を持つべき
- 新しいものをつくるのであれば、既存の並列言語 (HPF やXPFortranなど)からの移行パスを考えてほしい

「シームレス高生産・高性能プログラミング環境」(代表 東京大学 石川裕、H20-23, 3.5年)

「並列アプリケーション生産性拡大のための道具」の開発

PCクラスタから大学情報基盤センター等に設置されているスパコンまで、ユーザに対するシームレスなプログラミング環境を提供

- 高性能並列プログラミング言語処理系
 - 逐次プログラムからシームレスに並列化および高性能化を支援する並列実行モデルの確立とそれに基づく並列言語コンパイラの開発
- 高生産並列スクリプト言語
 - 最適パラメータ探索など粗粒度の大規模な階層的並列処理を、簡便かつ柔軟に記述可能で処理効率に優れたスクリプト言語とその処理系の開発
- 高効率・高可搬性ライブラリ開発
 - 自動チューニング(AT)機構を含む数値計算ライブラリ開発
 - PCクラスタでも基盤センタースパコン(1万規模CPU)でも単一実行時環境を提供するSingle Runtime Environment Image環境の提供

高性能並列プログラミング言語処理系開発

筑波大
学
東京大
学

次世代並列プログラミング言語仕様検討会
(主査：佐藤三久@筑波大)
NEC、富士通、日立、JAXA、JAMSTEC、核融合研、筑波大、東大、京大、九大

高生産並列スクリプト言語の開発

京都大
学

富士通研究所

高効率・高可搬性ライブラリ開発

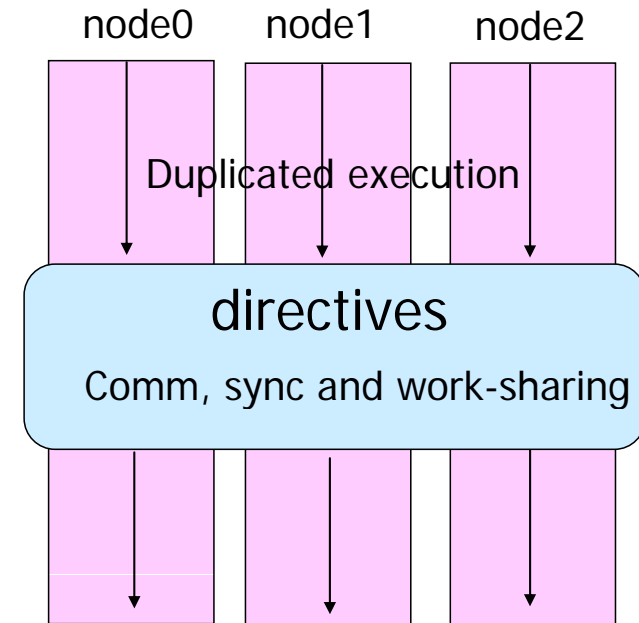
東京大学

富士通研究所

日立中央研究所

XcalableMP : directive-based language eXtension for Scalable and performance-tunable Parallel Programming

- **Directive-based language extensions** for familiar languages F90/C/C++
 - コードの書き換えや教育のコストを抑えること
- **“Scalable” for Distributed Memory Programming**
 - SPMDが基本的な実行モデル。
 - MPIのように、各ノードでスレッドが独立に実行を開始する。
 - 指示文(directive)がなければ、重複実行
 - タスク並列のためのMIMD実行も
- **“performance -aware” for explicit communication and synchronization.**
 - 指示文を実行するときに、Work-sharing や通信・同期がおきる。
 - すべての同期・通信操作は、指示文によって起きる。HPFと異なり、パフォーマンスのチューニングがわかりやすくなる。

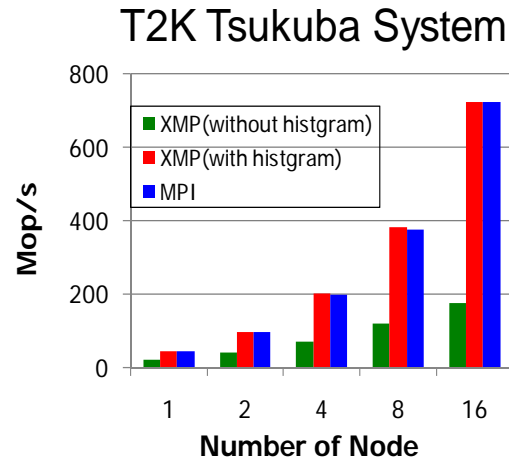


Status of XcalableMP

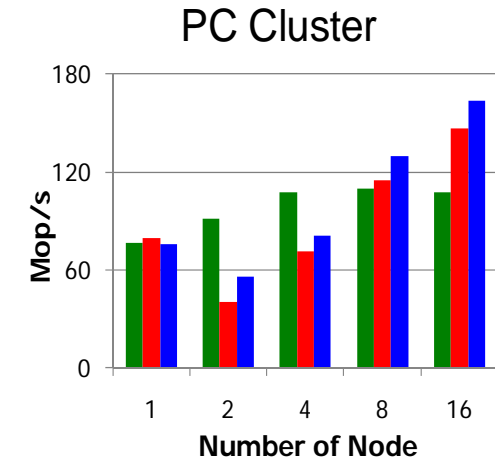


- Status of XcalableMP WG
 - Discussion in monthly Meetings and ML
 - XMP Spec Version 1.0 is available at XMP site.
 - XMP-IO and multicore extension are under discussion.
- Compiler & tools
 - XMP prototype compiler (xmpcc version 0.5) for C is available from U. of Tsukuba. Open-source, C to C source compiler with the runtime using MPI
 - XMP for Fortran 90 is under development. (これについてはあとで)
- Codes and Benchmarks
 - NPB/XMP, HPCC benchmarks, Jacobi ..
 - [Honorable Mention in SC10/SC09 HPCC Class2](#)
- Platforms supported
 - Linux Cluster, Cray XT5 ...
 - Any systems running MPI
 - The current runtime system designed on top of MPI

NPB IS performance

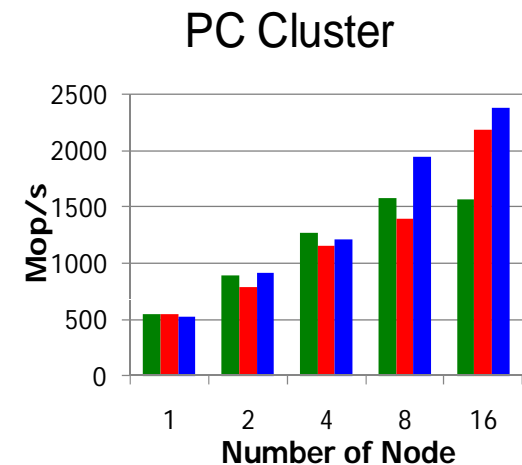
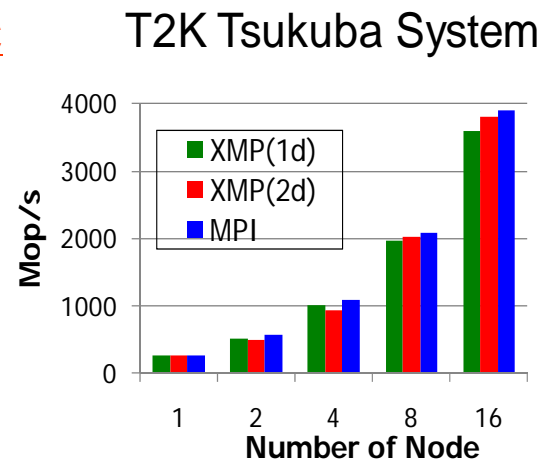


- Coarray is used
- Performance comparable to MPI



- Two-dimensional Parallelization
- Performance comparable to MPI

NPB CG performance



並列プログラミング言語XcalableMP(XMP) 規格部会 (PCクラスタコンソーシアム)



■ 並列プログラミング言語XcalableMPとは

- XcalableMP言語仕様検討委員会(大学、研究機関、メーカーの有志がメンバ)で、仕様を検討した並列プログラミング言語
- 分散メモリの並列システム(PCクラスタ)を対象、PGASモデル、HPFの経験を反映
- ベース言語C/Fortranを指示文で拡張(既存プログラムの利用が容易)
- 現在、仕様v1.0を決定、公開中 <http://www.xcalablemp.org/>
- 「e-サイエンス実現のためのシステム統合・連携ソフトウェアの研究開発」(代表:東大 石川裕、H23年度終了)において、筑波大を中心にレファレンス実装

■ 部会の活動、計画概要

- 部会長:佐藤三久、副部会長:岩下英俊 (富士通(株))、林 康晴(日本電気(株))
- 1)並列プログラミング言語XcalableMPの言語仕様(v1.0以降)の検討・決定
- 2)並列プログラミング言語XcalableMPの普及活動(講習会、ワークショップ、コンテストの開催等)
- 3)PCクラスタ向け並列プログラミング言語の動向の調査

■ 参加対象者

- 並列プログラミング言語の開発に関心のある会社、並列言語を利用に関心のあるHPC研究者および会社、並列言語・プログラミングに関する研究者
- MLに登録

■ 仕様決定のプロセスについて(第1回での結論)

- 規格について検討を行う部会や技術討論会の参加者については、会員であるかどうかは問わない。
- 議論の結果は、仕様書に反映させ、それを案として提出

- 最終的な規格決定について、投票で決定
- 投票の票数については、正会員に限り、口数を上限に、過去一年の部会の参加回数に等しい票を持つ。
- 過半数の賛成をもって、決定

- 1. Coarray (version 1.0では入れたが、まだいろいろと不明な点がある)
 - 宣言、割付け、スコープ
 - 特に、全体でないノード集合で宣言される場合の扱い
- 2. Interface for MPI (proposal in ver 1.0)
- 3. Directives for Thread Parallelism (proposal in ver 1.0)
- 4. Interface to MPI Numerical Lib. (proposal in ver 1.0)
 - Xmp_desc_ofの使い方
- 5. XMP-IO (proposal in ver 1.0)
 - アトミックモード？

課題(2/2)



- 6. XMP-collective 組込み関数
 - 配列処理関数
- 7. メモリレイアウト
- 8. アクセラレータ拡張
 - 筑波大提案(xmp-dev)を含む
- 9. タスク並列 dynamic task
- 10. 再分散機能 re-distribution

■ 現状

- ほとんどのクラスタがいまや、マルチコアノード(SMPノード)
- 小規模では格コアにMPIを走らせるflat MPIでいいが、大規模ではMPI数を減らすためにOpenMPとのハイブリッドになっている。
 - ハイブリッドにすると(時には)性能向上も。メモリ節約も。
- しかし、ハイブリッドはプログラミングのコストが高い。

■ 2つの方法

- OpenMPをexplicitに混ぜて書く方法
- loop directiveから、implicitにマルチスレッド・コード(OpenMP)を出す方法 → explicitに書くことになった

マルチコア対応



- デフォルトでは、シングルスレッドで実行
- マルチスレッド実行する場合は、thread(=スレッド数)を指示
 - OpenMPでいろいろなものを指定するのは面倒なので、auto scopingも検討

```
#pragma xmp loop (i) on ...  
for( ... i ...){  
#pragma omp for  
  for( ... j ...){  
  ....  
  }  
}
```

```
#pragma xmp loop (i) on ... threads ... openmpの指示行  
for( ... i ...){  
  ....  
}
```

- Design
 - Provide efficient IO for global distributed arrays directly from lang.
 - Mapping to MPI-IO for efficiency
 - Provide compatible IO mode for sequential program exec.

- IO modes
 - (native local IO)
 - Global collective IO (for global distributed arrays)
 - Global atomic IO
 - Single IO to compatible IO to seq. exec

並列ライブラリインタフェース

- すべてをXMPで書くことは現実的ではない。他のプログラミングモデルとのインタフェースが重要
- MPIをXMPから呼び出すインタフェース
- (MPIからXMPを呼び出すインタフェース)
- XMPから、MPIで記述された並列ライブラリを呼び出す方法
 - 現在、Scalapackを検討中
 - XMPの分散配列記述から、Scalapackのディスクリプタを作る
 - XMPで配列を設定、ライブラリを呼び出す
 - その場合、直によびだすか、wrapperをつくるか。

GPU/Manycore extension (XMP-dev)



- 別のメモリを持つ演算加速装置が対象
 - メモリをどのように扱うかが問題
 - 並列演算はOpenMP等でも行ける
- Device指示文
 - Offloadする部分を指定
 - ほぼ同じ指示文を指定できる(但し、どの程度のことができるかはそのdeviceによる)
 - GPU間の直接の通信を記述ができる。
- Gmove指示文で、GPU/host間のデータ通信を記述
- OpenACCとの関係

```
#pragma xmp nodes p(10)
#pragma xmp template t(100)
#pragma xmp distribute t(block) on p

double A[100];
double G_A[100];
#pragma xmp align to t: A, G_A
#pragma device(gpu) allocate(G_A)
#pragma shadow G_A[1:1]

#pragma xmp gmove out
    G_A[:] = A[:]    // host->GPU

#pragma xmp device(gpu1)
{
#pragma xmp loop on t(i)
    for(...) G_A[i] = ...
#pragma xmp reflect G_A
}

#pragma xmp gmove in
    A[:] = G_A[:]    // GPU->host
```

他にも



- Performance tools interface
- Fault resilience / Fault tolerance
- ...

部会の進め方 (2/2)



- ML, プロジェクト管理、普及...
 - xmp-wg@ccs.tsukuba.ac.jp
 - Xmp-dev@ccs.tsukuba.ac.jp
 - agendaは、Tracを立ち上げて管理する予定
 - ソースの管理 sourceforge.netに移行？
 - 国際展開 - external reviewを頼む予定

- 開発体制について
 - 現在、筑波大にて開発を進めている
 - 開発の主体を、AICSにもっていきたい
 - アクセラレータ拡張は、筑波大(XMP-dev)
 - メーカーのサポートを期待

おわりに



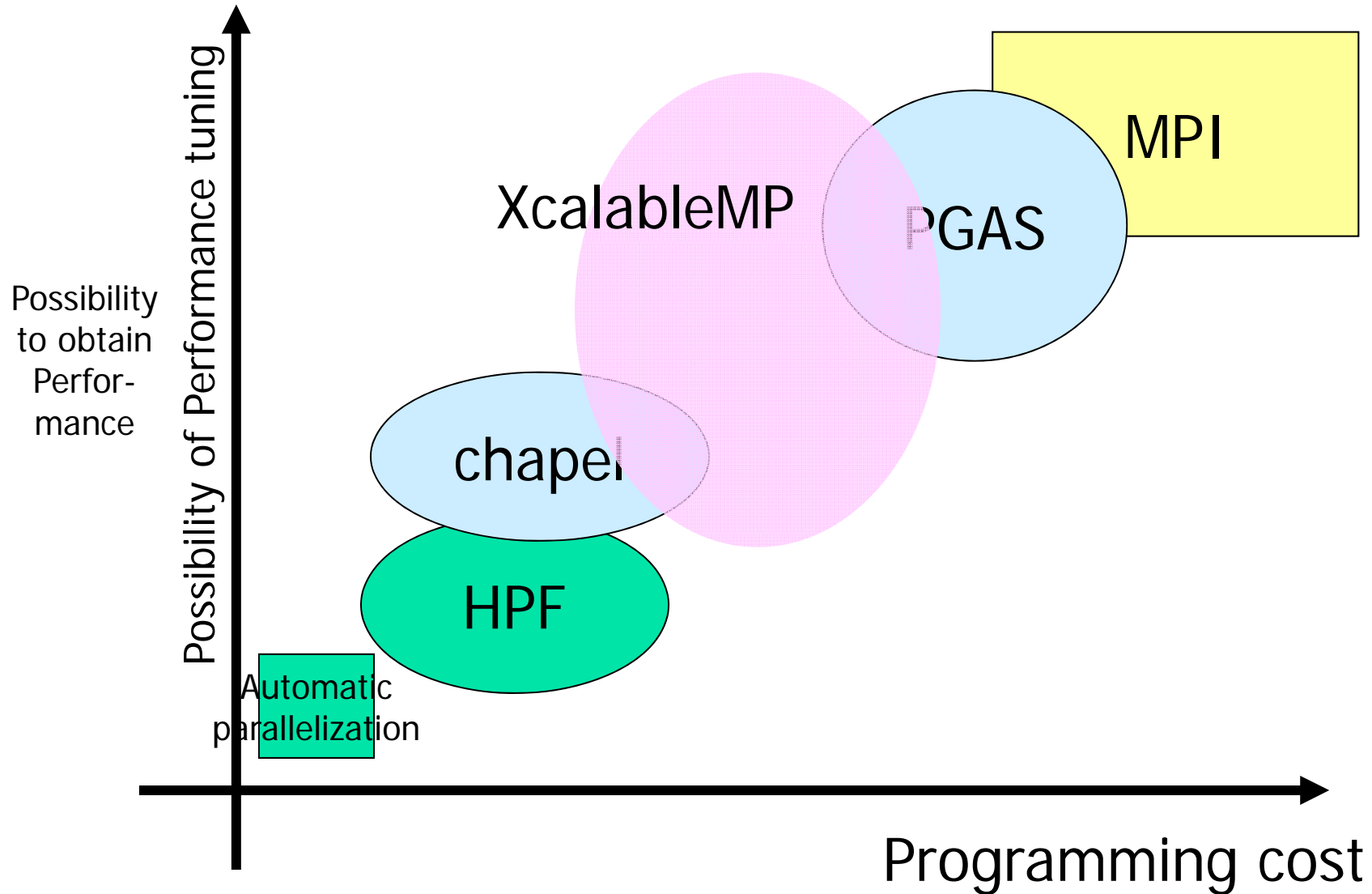
■ XMPを使うメリットは？

- プログラムが(MPIと比べて)論理的に、簡単かける(はず)
- 既存の言語C, Fortranから使える
- Multi-node GPUに対応
- マルチコア化が進むと、MPI-OpenMPは限界がある(とおもう)

■ XMPは主流になれるのか？

- 少なくとも、PGASはこの数年のトレンド。XMPは、CAFをサブセットとして含んでいる
- HPFの経験がある(はず) HPFではある程度のプログラムはかけていた(はず)
- GPUについては、わからない
- すくなくとも、5年は開発・保守を続ける(つもり)
- ポイントは、メーカーがついてくるか。現在のところ、富士通とCray...

Target area of XcalableMP



PGAS言語の利点・欠点

- MPIとHPFの中間に位置する
 - わかり易いモデル
 - 比較的、プログラミングが簡単、MPIほど面倒ではない
 - ユーザから見えるプログラミングモデル。通信、データの配置、計算の割り当てを制御できる
 - On-sided communicationはスケーラブルであることが期待されている
 - MPIなみのtuningもできる。
 - プログラムとしてpack/unpackをかいてもいい。

- 欠点
 - 並列のために言語を拡張しており、逐次には戻れない。(OpenMPのようにincrementalではない)
 - やはり、制御しなくてはならないか。
 - 性能は？

おわりに



■ XMPのこれから

- PAGESコミュニティでは認知されつつある。京関連で、いろいろな広報活動はしている。
- MPIとどう付き合うか
- XMPから、MPIライブラリのインタフェース
- Hybrid programming (eg. UPC and MPI, ...)

■ 進捗・計画

- XMP/Fortranはあと1か月ぐらい (計画の半年遅れ)
- XMP/Cは一応使えているので、使ってみてください。
- もちろん、京でも使えるようにします
- ISC 2012で、Omni XMP compiler 1.0 リリースしたい。
- SC2012のHPCC Class2に、京での結果をsubmitする計画
- デモ