

XcalableMP の現状と課題

佐藤 三久

理研 AICS・筑波大学

XcalableMP

- 分散メモリ環境を対象とした指示文ベースの
並列言語
- 次世代並列プログラミング言語検討委員会
→ 当部会において仕様を検討、提案。
- 2つの並列プログラミングモデルをサポート
 - グローバルビューモデルによる定型的な並列化
 - ローカルビューモデルによる自由度の高い並列化

XcalableMP as evolutionary approach

- **We focus on migration from existing codes.**
 - Directive-based approach to enable parallelization by adding directives/pragma.
 - Also, should be from MPI code. Coarray may replace MPI.
- **Learn from the past**
 - Global View for data-parallel apps. Japanese community had experience of HPF for Global-view model.
- **Specification designed by community**
 - Spec WG is organized under the PC Cluster Consortium, Japan
- **Design based on PGAS model and Coarray (From CAF)**
 - PGAS is an emerging programming model for exascale!
- **Used as a research vehicle for programming lang/model research.**
 - XMP 2.0 for multitasking.
 - Extension to accelerator (XACC)

関連したプロジェクト

- E-scienceプロジェクト検討 (2008)
- PC Cluster コンソーシアム XcalableMP 規格部会での議論
- 日仏 FP3Cプロジェクト
- 理研計算科学研究機構で、京コンピュータ向けに開発
- ポストペタCREST (研究代表者: 朴@筑波大、分担 村井)
 - XcalableACC (XMP+OpenACC) for accelerated Cluster
- SPPEXA (研究代表者: 朴@筑波大、分担 村井)
 - ドイツ アーヘン大学、フランス MDLSとのXMPのプログラム論理検証技術
- FS2020 プロジェクト
 - ポスト京向けにXMP2.0 を検討、Fortran 2008, C++

XcalableMPの現況

- PCクラスタコンソーシアムで規格を議論
 - 2017/April Version 1.3仕様を公開。
 - Version 1の系統はほぼ「収束」
 - 2015年より、次期仕様「XMP2.0」の検討を開始。
 - PGAS + Multitasking for Multicore
 - Code transformation for Optimization
 - (Accelerator)

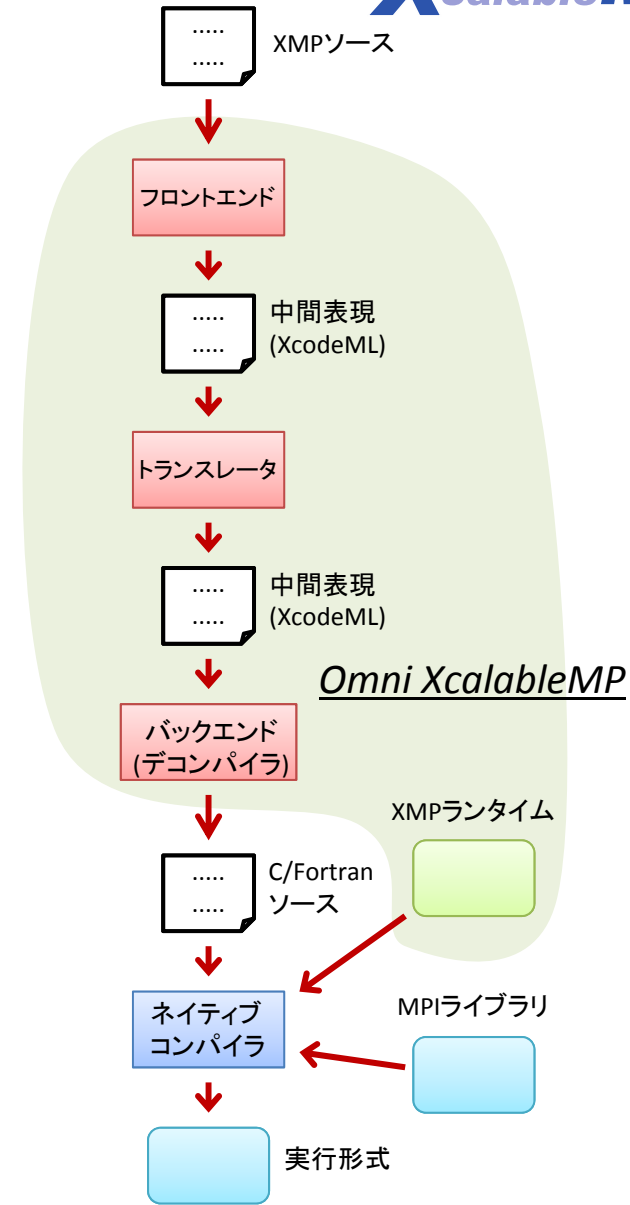
www.xcalablemp.org

- 理研・筑波大で、レファレンス実装
 - Omni XMP コンパイラ

omni-compiler.org

Omni XcalableMP

- 理研AICSと筑波大で開発中のXMP処理系
 - XMP/C
 - XMP/Fortran
- オープンソース
- トランスレータ + ランタイム(MPIベース)
- OpenACC、XcalableACC対応
 - OpenMP 4 を計画
- Front-end
 - Fortran 2008: 今年度、ほぼ終了
 - C++: LLVM clangベースで開発中



XcalableMPプロジェクト (&規格部会)の これからの方向性

- 言語、コンパイラ研究とともに、使ってもらうことが大切
 - 多様な使い方を提案する必要がある。
- プログラムの最適化・高速化に貢献できるモデル・仕組みを提案
 - MPIよりも速いプログラムを！
 - PGASとマルチタスクの統合のモデル
- 「規格化」することは重要だが、XMP 2.0ではもう少し、研究指向もありか!?
 - GPU/FPGAも！

現在のメニーコア・プログラミングの問題点

- 今のところ、MPI+OpenMPがstandard
 - ノード上に複数プロセス
 - (flat-MPIのほうが速い?)
- コアが多いところでは、taskモデルで書いたほうが効率のいいプログラムがある。
 - Global syncよりも効率的(な、ときがある) 特に、大規模では。
 - 計算と通信のオーバーラップが可能(であるはず)
- MPIのMPI_THREAD_Multipleが速くない。
 - “End point per process”の問題?
- PGAS(one-sided comm.)がsimpleなはず! ?
- ノード間のタスクモデルをどう書くか。
 - OpenMP 4の拡張は、簡単ではない。