MPI通信ライブラリ規格化最前線

東京大学
石川　裕

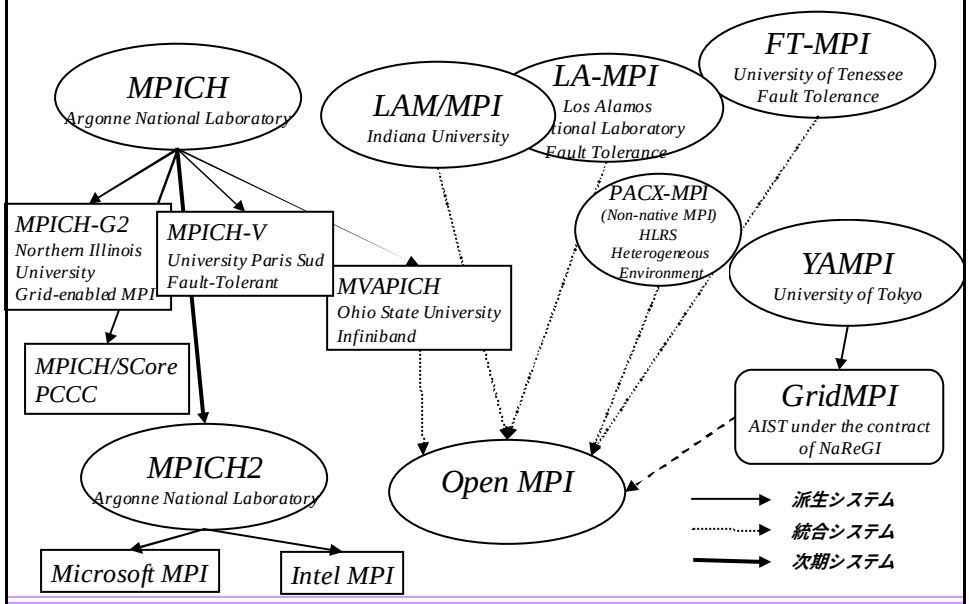
目次

- MPI通信ライブラリとは？
- MPI通信ライブラリ規格化の歴史
- MPI通信ライブラリ規格化の最近の動き

MPI通信ライブラリの概要

- 背景
  - 1990年初頭、各コンピュータベンダは独自の通信ライブラリを提供
  - ユーザプログラムのポータビリティ問題が深刻化
    - あるマシン上で開発したアプリケーションが他のベンダのマシンで動かず、プログラムを書き直さなければならない
  - 米国国立研究所、大学、並列計算機メーカが中心となり策定した通信ライブラリ
- 歴史
  - 1994年　MPI-1.0
  - 1995年　MPI-1.1
  - 1997年　MPI-2.0 & MPI-1.2

MPI通信ライブラリ実装いろいろ



MPI規格化の動き

- MPI-2.1
  - MPI-1.2とMPI-2.0は別のドキュメントとして提供されている。これらをマージして一つのドキュメントにすると共に間違いを修正する
- MPI-2.2
  - MPI-2.1に対するマイナーな変更を行う
  - 現存のプログラムが動かなくなるような変更は行わない
  - 大きな実装の変更を要求するような変更は行わない
- MPI-3
  - MPI-2.2に対して、プラットフォームおよびアプリケーションに対してより良い支援を行うために必要な変更を行う
  - 拡張は一貫性があること

MPI-2.1規格

- 2008年9月4日規格承認

  http://www.mpi-forum.org/
- MPI-2.1準拠処理系公開予定
  - 出典：MPI Forum

Library	Yes mm/yy
MPICH2	YES alpha
Open MPI (in V1.3)	YES upcoming
IBM	12/2008
Sun Microsystems	YES (Open MPI)
Microsoft	12/2008
Fujitsu	6/2009

Library	Yes mm/yy
Bull	
HP	12/2008
Cray	Q1 2009
NEC	YES
Intel	2009
SiCortex	YES

MPI-2.2規格化の現状

- Send Buffer Access
- Add const keyworkd to the C bindings
- to change MPAssertions to allow MPI implementation optimizations (MPI_INIT_ASSERTED)
- Add support for MPI_REQUEST_IGNORE, like MPI_STATUS_IGNORE
- MPI_Comm_set_info/get_info I settings per communicator
- Fix text about freeing error codes
- Ambiguity with MPI_TYPE_GET_CONTENTS
- Ambiguity with MPI_GRAPH_NEIGHBOR(_COUNT)
- MPI::Fint should be removed
- MPI_Request_free issues
- MPI::F_DOUBLE_COMPLEX MPI-2.1 Errata MPI::F_DOUBLE_COMPLEX (page 495 line 11)
- MPI-2.1 Errata MPI_MAX_OBJECT_NAME (page 237 lines 1+3, page 563, lines 18+20)
- Fix Scalability Issues in Graph Topology Interface
- Add a local Reduction Function
- Add a local Progression Function
- Add a callback function if a request completes
- Regular (non-vector) version of MPI_Reduce_scatter
- New binary operators (for segmented scans etc.) on (value,index) pair datatypes
- Extend predefined MPI_OP to user defined datatypes composed of a single, predefined type]
- Minor consistency issue: Same rule for MPI_IN_PLACE in MPI_Allreduce and MPI_Reduce_scatter?

MPI-2.2規格化の現状

- Clarification to intercomm MPI_Barrier
- Matched probe/receive
- Partial pack/unpack functionality
- Add MPI_IN_PLACE option to Alltoall
- Consistent Error Reporting Rules
- Error in example 4.18
- New Predefined Datatypes
- MPI_REPLACE in MPI_Accumulate
- Minor typo in description of MPI_Cart_shift
- Minor typo in description of MPI_Cart_rank
- Clarify who is client in external32 data representation advise to implementors
- Move the clarification of the Thread API to a more prominent place
- Support for large message counts
- Remove deprecated functions from the examples
- Small clarifications in chapter 10
- Ibsend and Irsend Advice_to Users misleading
- Inconsistent use of MPI_ANY_SOURCE in arguments
- Misleading rationale for MPI_Test

MPI-2.2規格化の現状

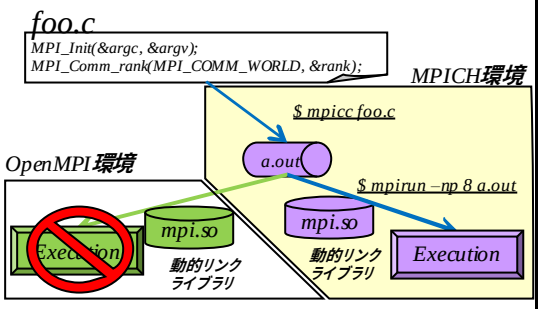
- Many C++ bindings are missing "const"
- MPI_GREQUEST_START function pointer args are missing "*"
- Consistency of function pointer typedef names
- Several text updates to Language Bindings chapter
- MPI_CANCEL argument is the wrong type
- Consistency of prefixes in Naming Conventions section
- Remove unnecessary/empty deprecated C++ bindings section
- Clarify semantics of one-sided semantics when changing synchronization mode
- Explicitly encourage routines for "good" one-sided memory for all memory regions
- Note that use of passive target communication for Fortran is not safe
- Dynamic Thread Levels
- Pre_MPI_Init_Behavior Pre-MPI_Init behavior clarification
- MPI-2.1 Change-Log: Versionnumber modified to 2.1
- Concurrent Use of MPI_Init() and MPI_Init_thread()
- Collective registration of datatypes and user-defined operations for one-sided communication

MPI-3規格化動向

- 以下のワーキンググループによる議論
  - Application Binary Interface
  - Collective Operations
  - Fault Tolerance
  - Fortran Bindings
  - Generalized Requests
  - MPI Sub-Setting
  - Point-To-Point Communications
  - Remote Memory Access
  - Tools Support (最近追加)
  - Miscellaneous (最近追加)
  - Hybrid Programming (最近追加)

MPI-3規格化動向：Application Binary Interface

- ABI (Application Binary Interface)とは、バイナリレベルでのデータ型、手続き呼び出し規約を意味する。現状のMPIでは、例えば、あるプログラムをMPICH2環境でコンパイルしたダイナミックリンク可能なプログラムは、OpenMPI環境では動作しない
- 企業からの要望が強い
  - MPIで記述されたアプリケーションを販売するとき、ユーザの利用環境を聞かないとバイナリが提供できない。多くのユーザは、どのMPI通信ライブラリを使っているか知らない
- 問題例：MPI_Communicatorの型
  - MPICH2の場合：int型(32bit)
  - OpenMPIの場合：address型(64bit)



MPI-3規格化動向: Collective Operations (1 / 2)

- Non-blocking Collectives
  - 1対1通信では、blocking/non-blockingがあるが、collective通信にはない
  - Non-blocking collective通信があれば、通信の隠ぺいが可能となる
  - 検討事項
    - Non-blockingとblocking collective通信を一緒に使うと問題が生じる

MPI_Ibarrier(..., &req)	*MPI_Bcast(…)*
MPI_Bcast(…)	*MPI_Ibarrier(…, &req)*
MPI_Wait(&req, …)	*MPI_Wait(&req, …)*

    - 一度に許すoutstanding数をどうするか
- Sparse/Toplogical Collectives
  - 部分的なall to all, reduce

MPI-3規格化動向: Collective Operations (2 / 2)

- Persistent Collectives
  - Point to point通信では、persistent communicationがあるが、collectivesにはない。
- MPI Plans
  - Non-blocking collectives, sparse collectives, persistent collectivesと拡張していくのではなく、これらcollectivesを効率よく実装できるようにするprimitivesを定義しようとする動きもある

Persistent communication使用例

```
MPI_Recv_init(…, &req[0]);
MPI_Send_init(…, &req[1]);
for(…) {
        ….:
        MPI_Startall(2, &req);
        MPI_Waitall(2, &req, …);
}
```

MPI Plans使用例

```
Void my_ibarrier(…)
{
   for (…) {
     MPI_Plan_send(…);
     MPI_Plan_recv(….);
   }
  MPI_Plan_init(…);
}
```


MPI-3規格化動向: Fault Tolerance

- ポータブルな耐故障機能を実装できるような支援機能を定義する
- 議論されている項目
  - Error Reporting Rules
    - エラーをどのように報告するか
  - Comm Integrity Validation
    - Collective operation時にエラーが発生した時に、consistentな状態かどうかを調べる
  - Transactional Messages
    - 通信エラーをどうモデル化してアプリケーションに伝えるか
  - Piggybacking for Point-to-Point Communication
    - 主データ交換にプラスアルファのデータを送れるようにしようとする提案
  - Fault handling = error handling
  - Quiescence Interface
    - システムレベルcheckpoint/restart機能と協調するためのAPIを定めようという提案
  - Process Creation and Management Extensions
    - Non-blockingプロセス生成とエラー管理
  - Fault Tolerant Master Worker jobs
    - MPI Spawnを使ったmaster-workerモデル型アプリケーションで、コミュニケータを共有しているときに、一つのプロセスの障害が全てのプロセスに波及することを避ける
    - 現状：[MPI_Abort] may abort all processes in MPI_COMM_WORLD (ignoring its comm argument). Additionally, it may abort connected processes as well, although it makes best attempt to abort only the processes in comm."

MPI-3規格化動向：その他

- Generalized Requests
  - Redefine the generalized requests interface. A more flexible interface between the user defined requests and the MPI library is required in order to allow the provider of the generalized request to integrate a progress function inside the MPI library. The ultimate goal is to allow the generalized requests progress to be done without a special test or wait function.
- MPI Sub-Setting
  - To establish a mechanism by which MPI implementations can provide support for a subset of the full MPI standard, maintaining full API and semantic compatibility with the complete MPI standard. This is aimed at allowing optimization opportunities such as for performance or resource foot-print.
- Point-To-Point Communications
  - To re-examine the MPI peer communication semantics and interface, and consider additions and/or changes needed to better support point-to-point data movement within MPI.
- Remote Memory Access
  - To re-examine the MPI RMA interface and consider additions and or changes needed to better support the one-sided programming model within MPI.

おわりに

- MPI-2.1の日本語化計画を始めています
  - http://www.il.is.s.u-tokyo.ac.jp/~ishikawa/mpi-j
- MPI-2.2規格化
  - Small changesとしているが、いろいろな提案が出てきている
  - 議論のうえ、MPI-3に移動する場合もある
- MPI-3規格化
  - あれもこれも新しい機能を追加すると、ドキュメントは倍に膨れ上がるかもしれない
  - MPI Spawnの拡張も議論されているが、MPI Spawnの実装は現状でも限定的かつ実装は後回しの感がある
- アッパコンパチビリティを保つことの限界
  - メッセージサイズが32ビットを超えることができないために、新たなる関数を提供するのか？
    - C言語は暗黙の型変換で逃げることができるが、Fortranは無理
- ポータブルかつシンプルな通信ライブラリAPIおよび実装が必要になったか？