FUJITSU

MPI-Adapter チュートリアル

富士通研究所
住元　真司

PC Cluster Consortium

FUJITSU

発表の概要

- MPI-Adapterとは何か？
- MPI-Adapterの概要
- MPI-Adapterの仕組み
- MPI-Adapterの利用
- デモンストレーション

FUJITSU

MPI-Adapterとは何か？

FUJITSU

クラスタを使っていて こんなことで困ったことはありませんか？

- 複数クラスタでのプログラム開発
  - 実行するためにソース転送、コンパイルする手間
  - クラスタ毎に性能が違って、それが環境の問題なのか、プログラムの問題なのかわからない。
- オープンソースアプリの利用
  - ソースをダウンロードしてコンパイルするのが面倒
- 昔のクラスタ計算環境で作成したバイナリ利用
  - 実行バイナリはあるが、実行環境もコンパイル環境もない。
- マルチコアノートＰＣ上での開発環境
  - 移動中にプログラム開発をやって、実行だけはリモートで大きなクラスタを使いたい。

このような環境でもＰＣクラスタを便利に使いたい

FUJITSU

MPI-Adapterとは？

- PCクラスタをより便利にするプログラム
  - MPIアプリケーションがコンパイルされた環境と、実行されるクラスタ実行環境との違いを吸収するアダプタです
  - 大学センターでも、個人環境でも手軽にユーザが使えることをめざしました
- 動作条件があります
  - 実行プログラムが動的リンクのMPIライブラリを使っていること
  - 各サイトに固有な動的リンクライブラリは個別に準備する必要あり

MPI-Adapterを使うとPCクラスタの利用法はこう変わります。

- 複数のクラスタでプログラム開発
  - 同じ実行バイナリを各クラスタに転送して実行可能
  - 実行バイナリ部分はまったく同じのため、意図的にＭＰＩランタイムを変更して性能差をみることができます。
- コンパイルするのが面倒
  - 単一の実行バイナリをダウンロードし、複数の異なる実行環境のクラスタで実行できます。
- 昔のアプリケーションを実行したい。
  - MPIライブラリを最新のクラスタ環境に置き換えての実行が可能になります。
- マルチコアノートＰＣ上での開発環境
  - 実行バイナリだけ転送して実行可能になります

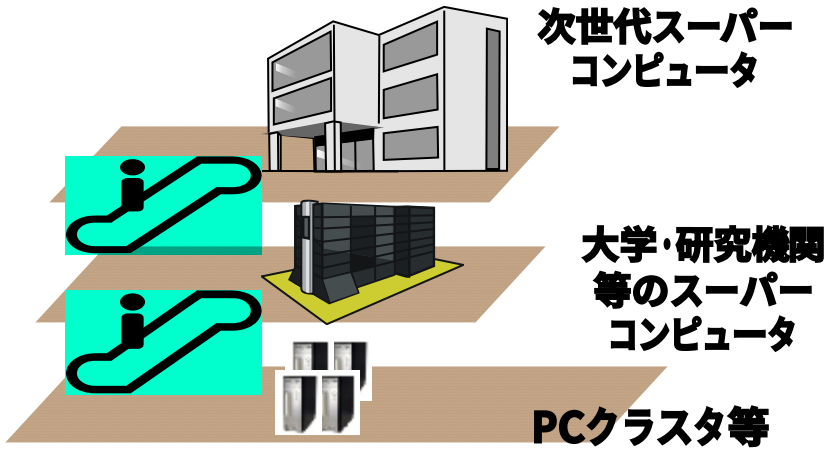
FUJITSU

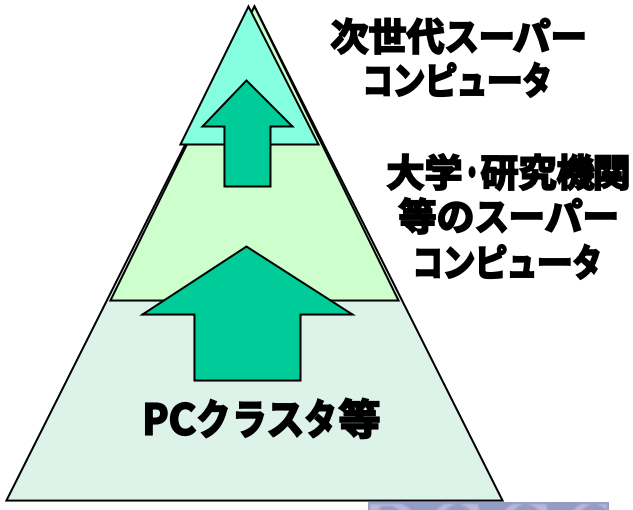
MPI-Adapterの概要

eSCienceプロジェクトのめざすもの

- 現状： PCクラスタの利用裾野の広がり→ 研究室単位での導入
  - さまざまな並列アプリの研究開発が実行されている。
  - しかし、開発されたアプリの可能性が導入クラスタの規模レベルに留まることが多い。

- 目標： 大学、研究機関の研究による並列アプリをより実用的、より大規模に引き上げ、ペタフロップス級の超大規模アプリの開拓





FUJITSU

eSCienceプロジェクト(Runtime)の課題

- PCクラスタ毎に実行環境が異なり、ユーザがクラスタ毎の実行環境を意識する必要あり。
  - プログラムの実行の仕方：
    - TSS実行コマンドの違い、バッチスクリプトの記述などの変更が必要
  - 実行プログラムに必要なファイル群へのアクセス：
    - 必要なファイル群を個別に手動での転送が必要
  - 実行プログラムのコンパイル：
    - ソースファイルからの再コンパイルが必要
- MPI-Adapter：　ユーザに実行プログラムの再コンパイル不要な環境を提供するために開発

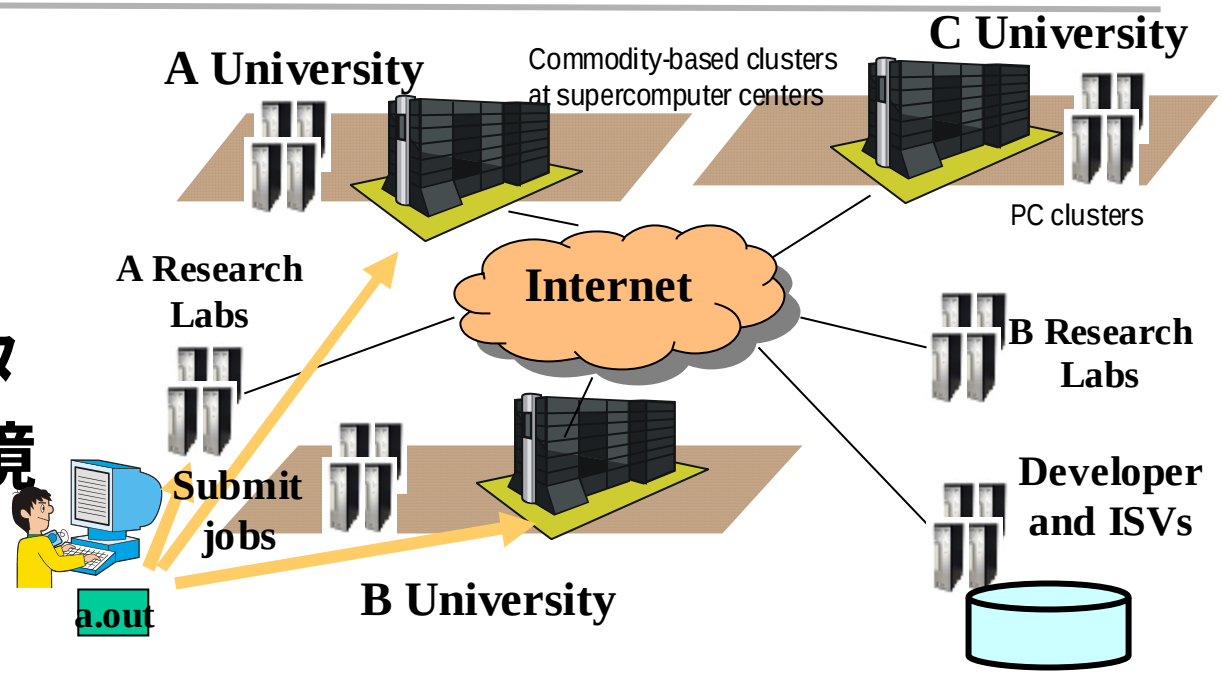
FUJITSU

MPI-Adapter開発のMotivation

- ＰＣクラスタの広がり：
  - 計算機センタ（RICC, T2K,TSUBAME）、研究室クラスタ
- 利用方法が拡大：
  - インターネットから様々なクラスタを選択利用
- PCクラスタ毎のMPI実行環境の違い
  - 実行環境毎にソースからのコンパイルが必要
  - 例： T2Kマシンはハードウェア構成がほぼ同等にも関わらず、相互にMPIプログラムを実行する場合再コンパイルが必要
- もっと簡単に、複数のＰＣクラスタを自由自在に使えないか？

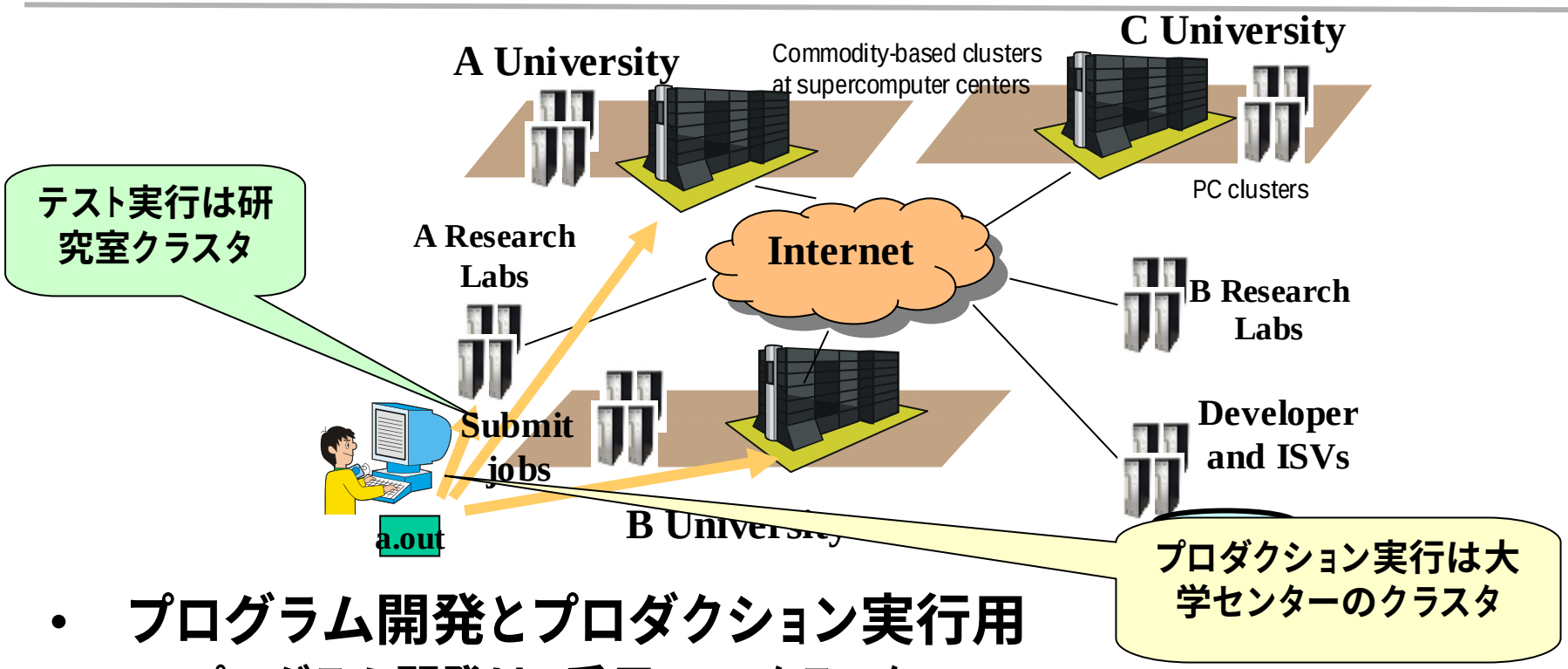
MPI-Adapterの目指すもの：シームレスな MPI 実行環境の提供

- **目指すもの：PCクラスタで作成のMPIバイナリは、他のどのPCクラスタでも実行可能な環境を実現する**



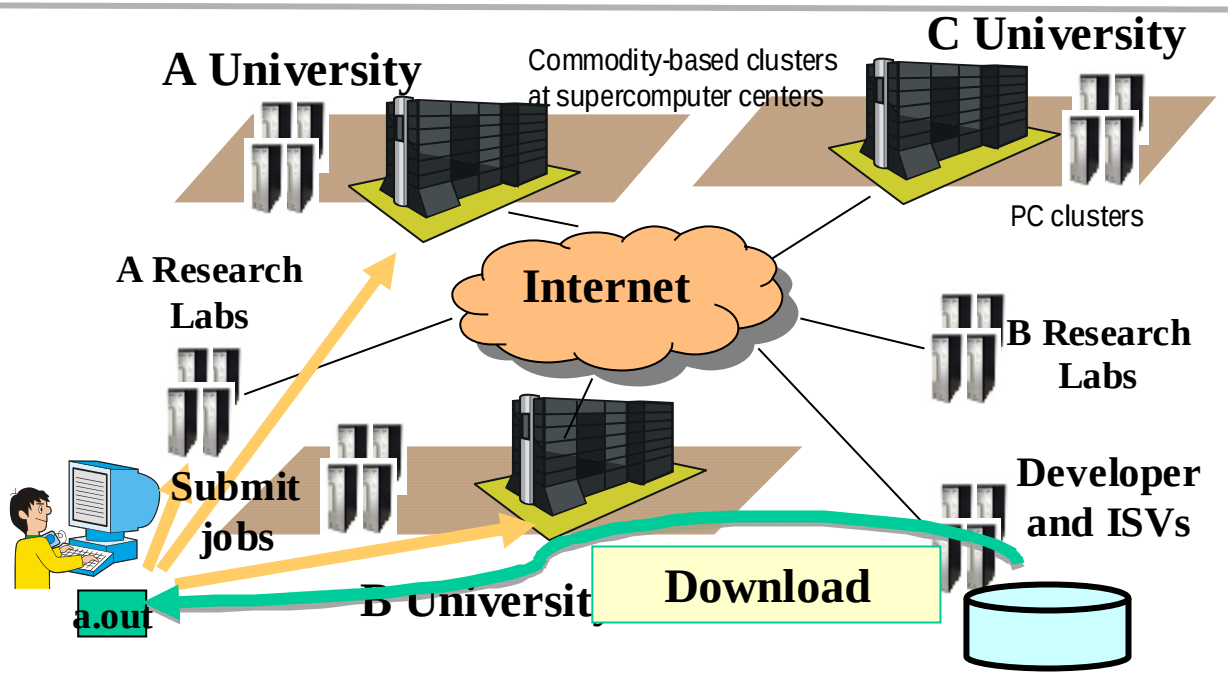
- Use Cases:
  1. **用途によるクラスタの使い分け：プログラム開発用とプロダクション実行用**
  2. **バイナリ配布：　ISV向け、コンパイル不要**
  3. **クラスタ環境の変更：　機能変更、性能向上**

Use Case 1:用途によるクラスタの使い分け



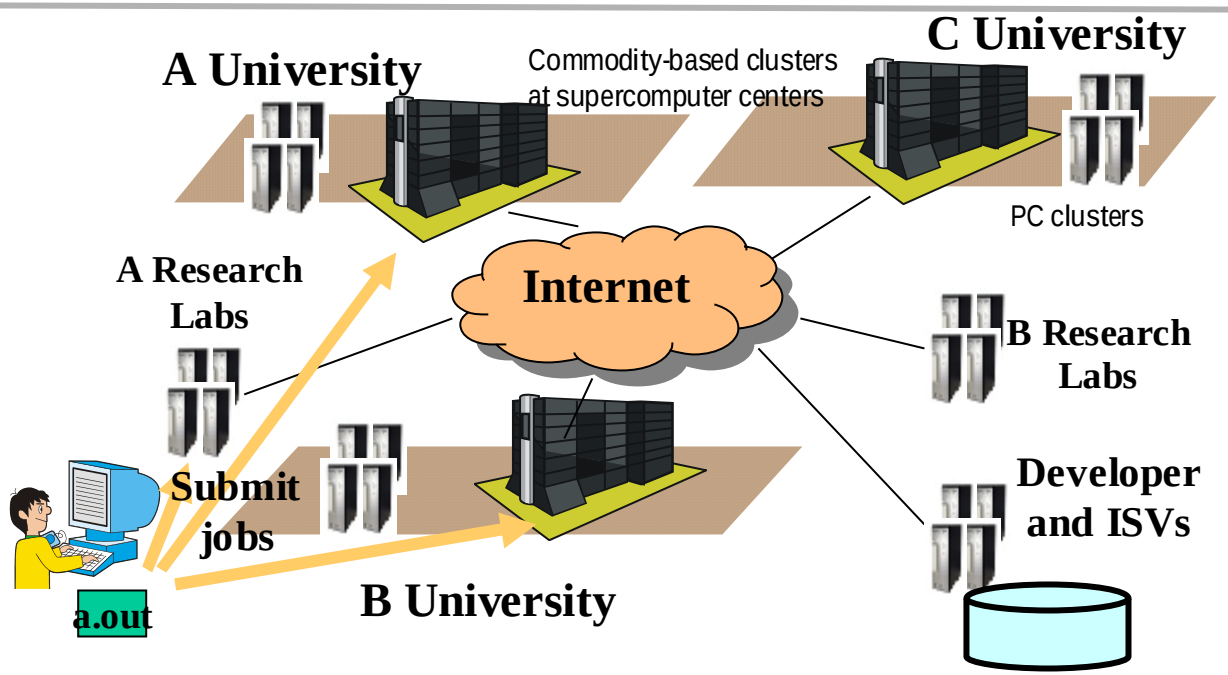
- プログラム開発とプロダクション実行用
  - プログラム開発は、手元のPCクラスタ
  - プロダクション実行用はセンターのPCクラスタ
- 複数のクラスタのシームレスな相互利用：
  - プログラム実行規模と計算リソース、システムの空き具合、プログラム実行時の課金状況により実行するPCクラスタを選択

Use Case 2: バイナリ配布



- 開発したプログラムをMPI実行バイナリとして配布することにより、利用ユーザの利便性を図る
- ISVは一つのMPI実行バイナリの提供： 利用者側の実行環境に合わせてISVプログラムを実行

Use Case 3: クラスタ環境の変更



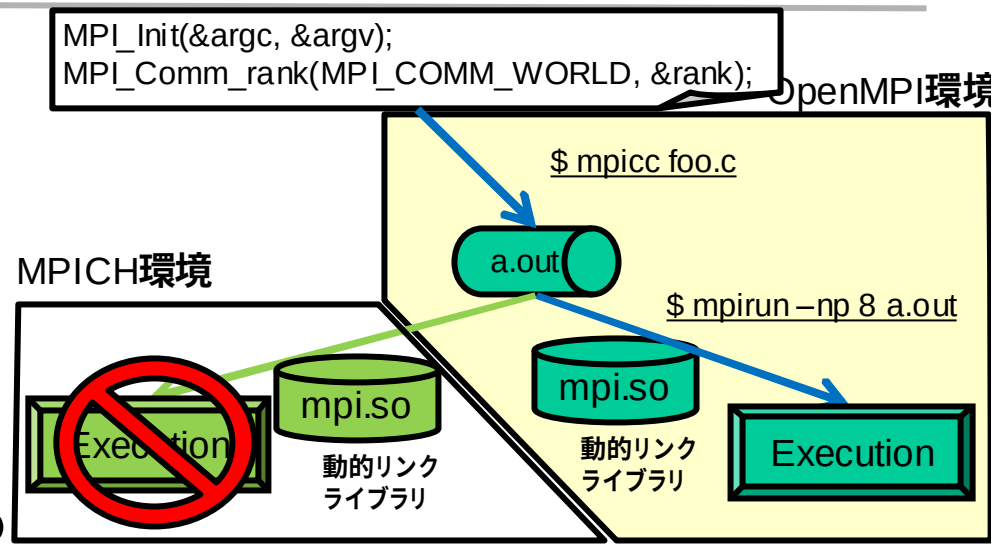
- MPIの機能毎に自由な置き換え（組み合わせ）が可能になる
  - MPIランタイムの個別ユーザの変更は一部の特権ユーザ以外容易ではない、特にプロダクション実行用のクラスタは困難
  - MPIの集合通信のアルゴリズム変更（関数毎のチューニング）
  - デバッグ用のプロファイラの動的挿入、ログ情報の動的挿入を、プログラム本体の再コンパイルなく実施可能

FUJITSU

MPI-Adapterの仕組み

MPI-Adapter実現の課題

- MPI規格書には、ABI (Application Binary Interface) が未定義
- このため、異なる実行環境では正常に動作しない。
- このような場合は、再コンパイルが必要になる。



- 実行可能とするためには、ABIを一致させる必要がある。

FUJITSU

ABI違いの原因：MPI定義の型定義

- MPIの仕様書にはいくつかの型が定義
- MPIの実装時にどのように型を実現するかは実装依存
  - 例： 数値での実装、構造体での実装
- この実装の違いが、バイナリの可搬性の障害になっている

用途	型 (値例)
Communicator	MPI_Comm (MPI_COMM_WORLD)
Group	MPI_Group
Request	MPI_Request
Status	MPI_Status
Data type	MPI_Datatype (MPI_Int,)
Operation	MPI_Op
Window	MPI_Win
File	MPI_File
Info	MPI_Info
Pointerの差分	MPI_Aint
Offset	MPI_Offset
Error Handler	MPI_Errorhandler

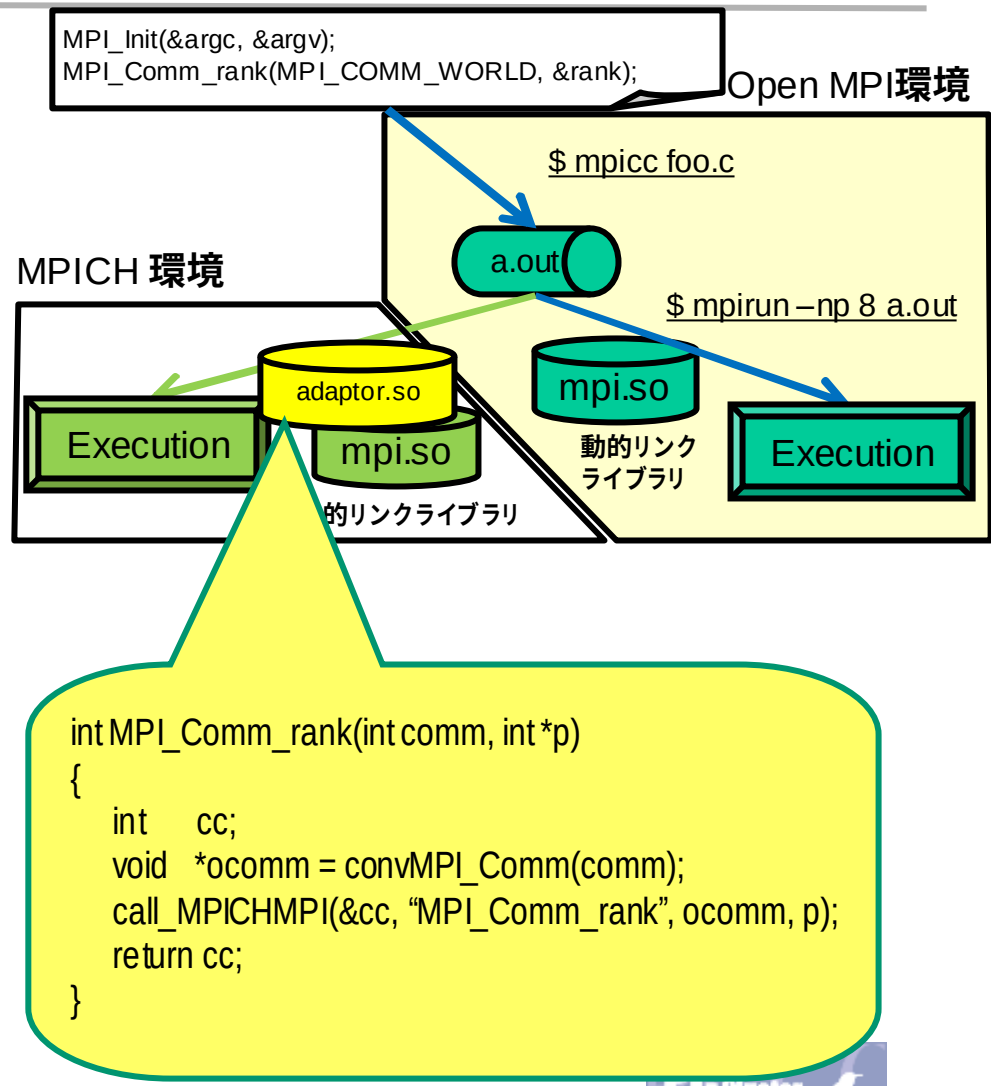
ABI違いの原因：MPI定義の型定義 OpenMPIとMPICH2の実装例

	用途	MPICH2	OpenMPI
C言語	MPI_COMM_WORLD	0x44000000	&ompi_mpi_comm_world
	MPI_INT	0x4c000405	&ompi_mpi_int
	MPI_INTEGER	0x4c00041b	&ompi_mpi_integer
	MPI_SUCCESS	0	0
	MPI_ERR_TRUNCATE	14	15
Fortran言語	MPI_COMM_WORLD	0x44000000	0
	MPI_INTEGER	0x4c00041b	7
	MPI_SUCCESS	0	0
	MPI_ERR_TRUNCATE	14	15

- OpenMPIとMPICH2ではバイナリ可搬性がない
  - MPICH2は32bit INTを用いた実装
  - OpenMPIは構造体を主体とした実装
- Fortranは32bit INTを用いた実装

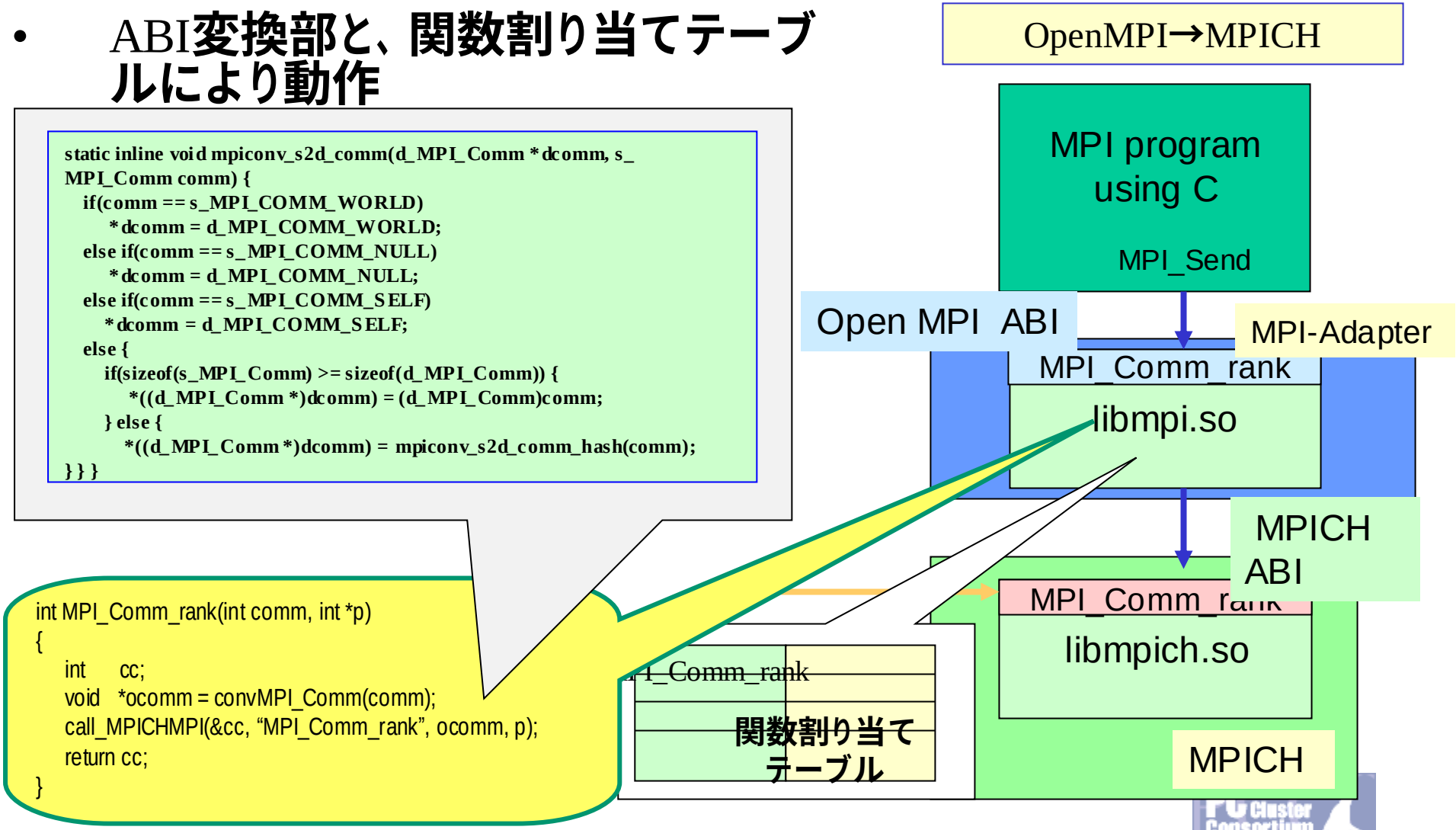
MPI-Adapterのアプローチ

- MPI-Adapterを2つの異なるMPI実装の間に挿入しABIの整合性を確保
  - ユーザはMPI実行環境の違いを意識しないで、異なるクラスタ上でのプログラム実行可能



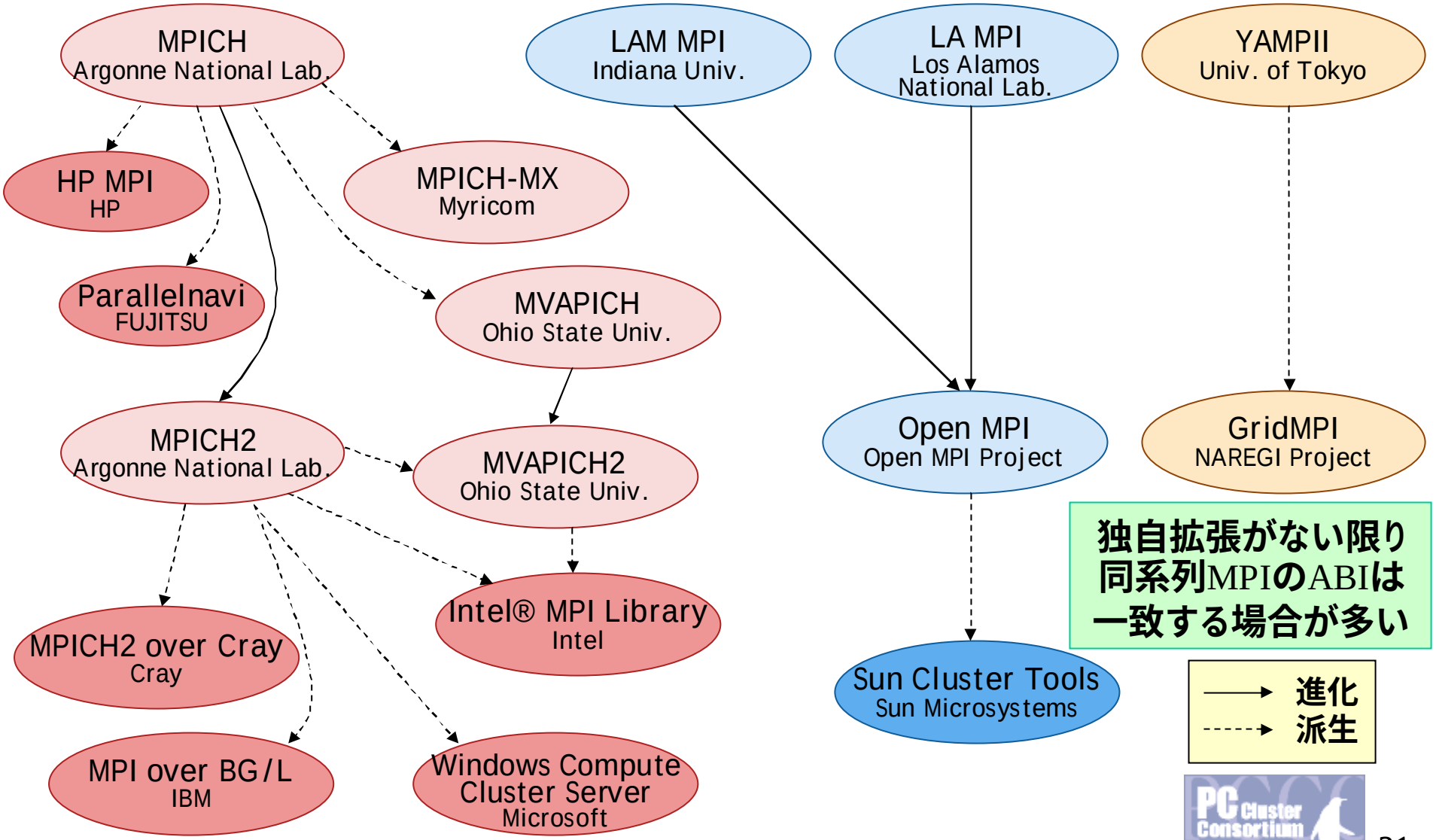
MPI-Adapterの動作内部構造

- ABI変換部と、関数割り当てテーブルにより動作



FUJITSU

ABI互換の適用性： MPIライブラリの系譜



FUJITSU

MPI-Adapterの利用

FUJITSU

MPI-Adapterの利用

- SCore 7.0Beta5には、MPI-Adapterのαバージョンが含まれています。
- MPI-Adapterの作り方
  - ターゲットのMPIランタイムを準備する
  - 設定ファイルを書く
  - コンパイル、インストールする
- MPI-Adapterの利用法
  - コマンドの説明
  - 実行例
  - デモンストレーション

MPI-Adapterの作り方 (1)
ターゲットのMPIランタイムの準備

- ターゲットのMPIランタイムを準備する
  - コンパイル、インストールして、ライブラリを作成
  - SCoreの場合はMPICH2-SCoreのほか、CentOSの以下が利用可能
    - MVAPICH2, Open MPI
  - 標準のインストール先

MVAPICH2	/usr/lib64/mvapich2/ 下
Open MPI	/usr/lib64/openmpi/ 下

MPI-Adapterの作り方（2）
設定ファイルを書く

- MPI-Adapterのディレクトリ下の設定ファイルを必要に応じて修正する
  - 設定ファイル：

./score7-src/eScience/Xruntime/mpi-adapters/ mpi.x86_64-rhel5-linux2_6.conf.tmpl

mpiname	type	install_root
コマンド指定に利用	ビルド時の属性	インストール先ディレクトリ
mpich_score	mpich_score	@PCCC_INSTALL_ROOT@
mvapich	mpich	/usr/lib64/mvapich2/*
ompi	ompi	/usr/lib64/openmpi/*

Type（ビルド時の属性）：mpich_score, mpich, ompi, hpmpiが指定可能

MPI-Adapterの作り方 (3)
コンパイル、インストールする (1)

- mpi-adapterディレクトリ下でsmakeコマンドを実行
  - SCoreの場合は/opt/score/下にインストールされる

```
s-sumi@mpi-adapter#  smake distclean all install
```

- インストール先
  - コマンド:　　/opt/score/bin/mpi-adapter 下
  - ライブラリ:　/opt/score/lib/mpi-adapter　下
- コンパイル＆インストール後:

```
[s-sumi@tdev000 mpi-adapters]$ cd /opt/score/
[s-sumi@tdev000 score]$ ls bin/mpi-adapter
bin/mpi-adapter
[s-sumi@tdev000 score]$ ls lib/mpi-adapters/obj.x86_64-rhel5-linux2_6/
mpich_score-mvapich  mvapich-mpich_score  ompi-mpich_score
mpich_score-ompi     mvapich-ompi         ompi-mvapich
[s-sumi@tdev000 score]$
```

MPI-Adapterの作り方 (3)
コンパイル、インストールする (2)

- ライブラリのディレクトリの中身

```
[s-sumi@tdev000 obj.x86_64-rhel5-linux2_6]$ ls ompi-mpich_score/
dummy          libmpi_f77.so.0.0    libmpi.so.0      rpath
libmpi_f77.so    libmpi_f77.so.0.0.0  libmpi.so.0.0
libmpi_f77.so.0  libmpi.so          libmpi.so.0.0.0
[s-sumi@tdev000 obj.x86_64-rhel5-linux2_6]$ ls ompi-mpich_score/dummy/
libdummy.so  libopen-pal.so  libopen-pal.so.0  libopen-rte.so  libopen-rte.so.0
[s-sumi@tdev000 obj.x86_64-rhel5-linux2_6]$
```

置き換え用ライブラリ

ダミー用ライブラリ

- きちんと作られているか確認

FUJITSU

MPI-Adapterの利用法：基本

- **利用の基本スタイル**

```
% mpirun –np 4 mpi-adapter [options] mpi-bin.exe
```

Options:

-s: 実行するプログラムのmpiname (スライド24)

-d: 実行環境(mpirun)のmpiname (スライド24)

例：ompi, mvapich, mpich_score

デフォールトは, -s ompi, -d mpich_score

FUJITSU

MPI-Adapterの利用法：

- Open MPIバイナリをMPICH2/SCoreで実行

```
% mpirun –np 4 mpi-adapter ompi.exe
% mpirun –np 4 mpi-adapter –s ompi ompi.exe
% mpirun –np 4 mpi-adapter –s ompi –d mpich_score ompi.exe
```

- Open MPIバイナリをMPICH2で実行

```
% /opt/MPICH2/bin/mpirun –np 4 mpi-adapter –d mpich2 ompi.exe
% /opt/MPICH2/bin/mpirun –np 4 mpi-adapter –s ompi –d mpich2 ompi.exe
```

FUJITSU

デモンストレーション

FUJITSU

MPI-Adapter デモンストレーション

- Vmware環境を使ったMPI-Adapterデモ
  - Intel Core2 Duo(2 core), Cent OS 5.4, SCore7
  - MPI Runtimes: MPICH2, Open MPI, HP MPI MPICH2/SCore
- NAS Parallel ベンチマークバイナリ
  - MPICH2, Open MPI, MPICH2/SCore, HP MPI
- デモンストレーション
  - mpirun でプログラム実行
    MPI-Adapterを使う場合、使わない場合

FUJITSU

まとめ

- MPIバイナリのABI互換性を実現するMPI-Adapter
  - 仕組み、概要について説明
  - PCクラスタ利用をより便利にするツール
  - 使い方と簡単なデモ
  - さまざまな、応用が可能で適用範囲が広がる
- 実績としては、MPICH系(MPICH2, MVAPICH)、Open MPI、HP MPIで稼働確認済み
- 今後の予定
  - MPIのテストスーツによる網羅的な品質向上
  - 利用環境の充実
    - MPIxxのバージョン毎の違いの吸収(ex. Open MPI)
    - MPIバイナリの自動検出
  - 実運用で実績を積む
- 謝辞：研究は、eScienceプロジェクトの一部として実施された

FUJITSU

参考文献

- **並列プログラムの実行可搬性を実現するMPI通信ライブラリの設計**
  (**情報処理学会** HOKKE 2009, 2009-ARC-182(20) 2009-HPC-119(20), pp.115-120, 2009.)
  住元真司, 中島耕太, 成瀬彰, 久門耕一(富士通研), 安井隆(日立), 鴨志田良和, 松葉浩也, 堀敦史, 石川裕(東大)
- **シームレスなMPI環境を実現するMPI-Adapterの設計と性能評価** (**情報処理学会** SWoPP 2009、2009-HPC-121(12), pp. 1-8, 2009)
  住元真司, 中島耕太, 成瀬彰, 久門耕一(富士通研), 安井隆(日立), 鴨志田良和, 松葉浩也, 堀敦史, 石川裕(東大)
- The Design of Seamless MPI Computing Environment for Commodity-based Clusters (Euro PVM/MPI 2009, Lecture Notes in Computer Science. Springer Berlin / Heidelberg, Volume 5759/2009, ISBN 978-3-642-03769-6, Page 9-19)
  Shinji Sumimoto, Kohta Nakashima, Akira Naruse, Kouichi Kumon, Takashi Yasui, Yoshikazu Kamoshida, Hiroya Matsuba, Atsushi Hori and Yutaka Ishikawa (2009 Best Papers)

FUJITSU

Thank you.

FUJITSU

付録： MPI-Adapterの技術概要

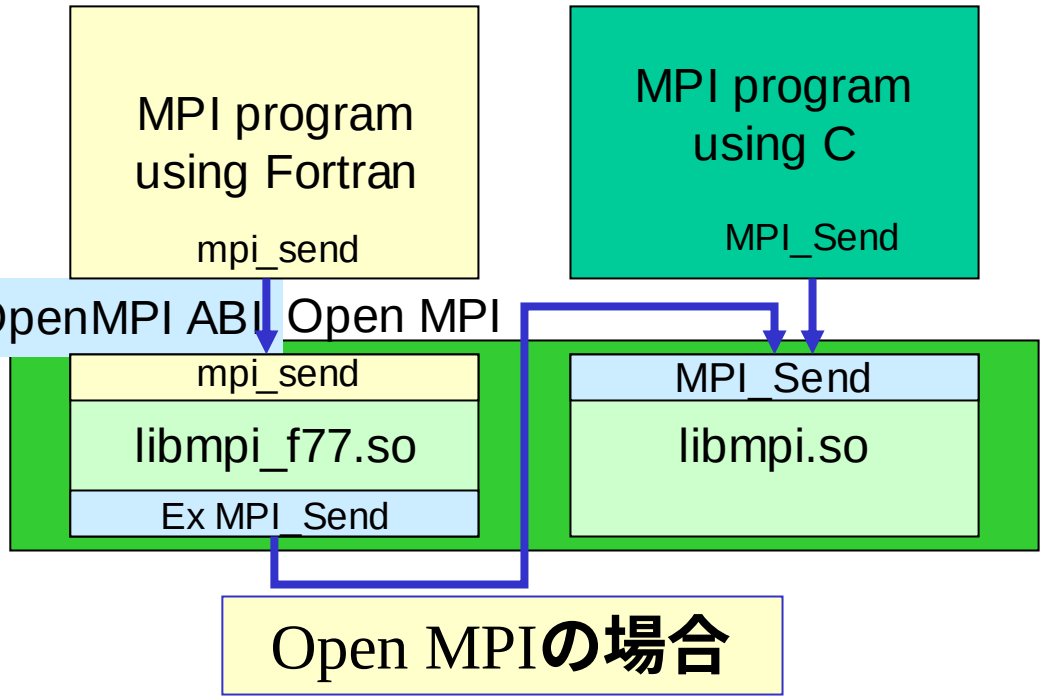
FUJITSU

MPI-Adapterの概要

- 簡単な設定で利用可能
- MPI-Adapterのオーバヘッドは無視できる程度
- 多くのMPIランタイムに適用可能
  - 現状、Open MPI, HP MPI→ MPICH, MPICH2での動作を確認
  - ターゲットのMPIランタイムのヘッダより自動的にABI情報を抽出

MPIライブラリの実装構造例：Open MPIの場合

- C用のMPIライブラリ:
  MPIライブラリ本体を実装
- Fortran用MPIライブラリ:
  Fortran→Cコンバータ
  (F2C) による実装が多い
- MPICHの場合
  - libfmpich.so (Fortran)
  - libmpich.so (C)

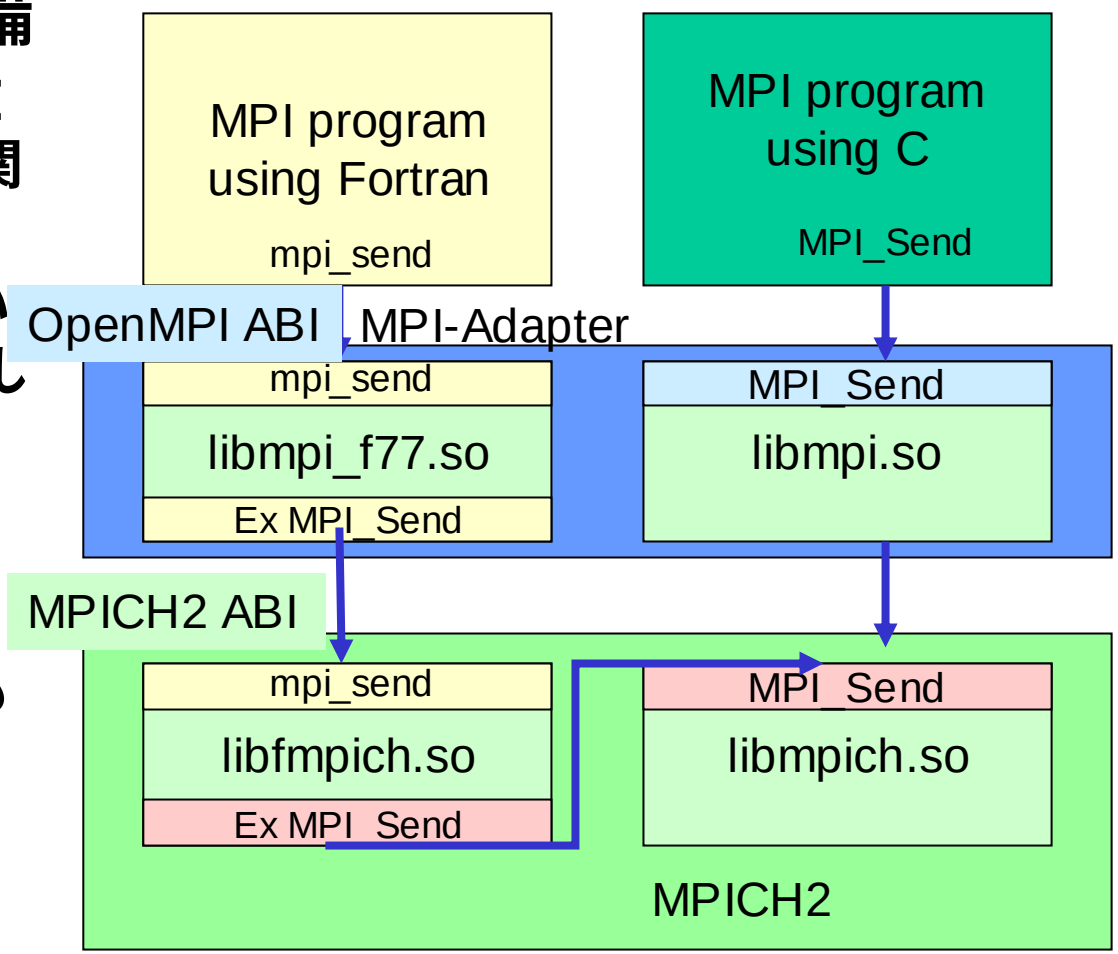


Open MPIの場合

C++の実装も同様に
C++→C変換

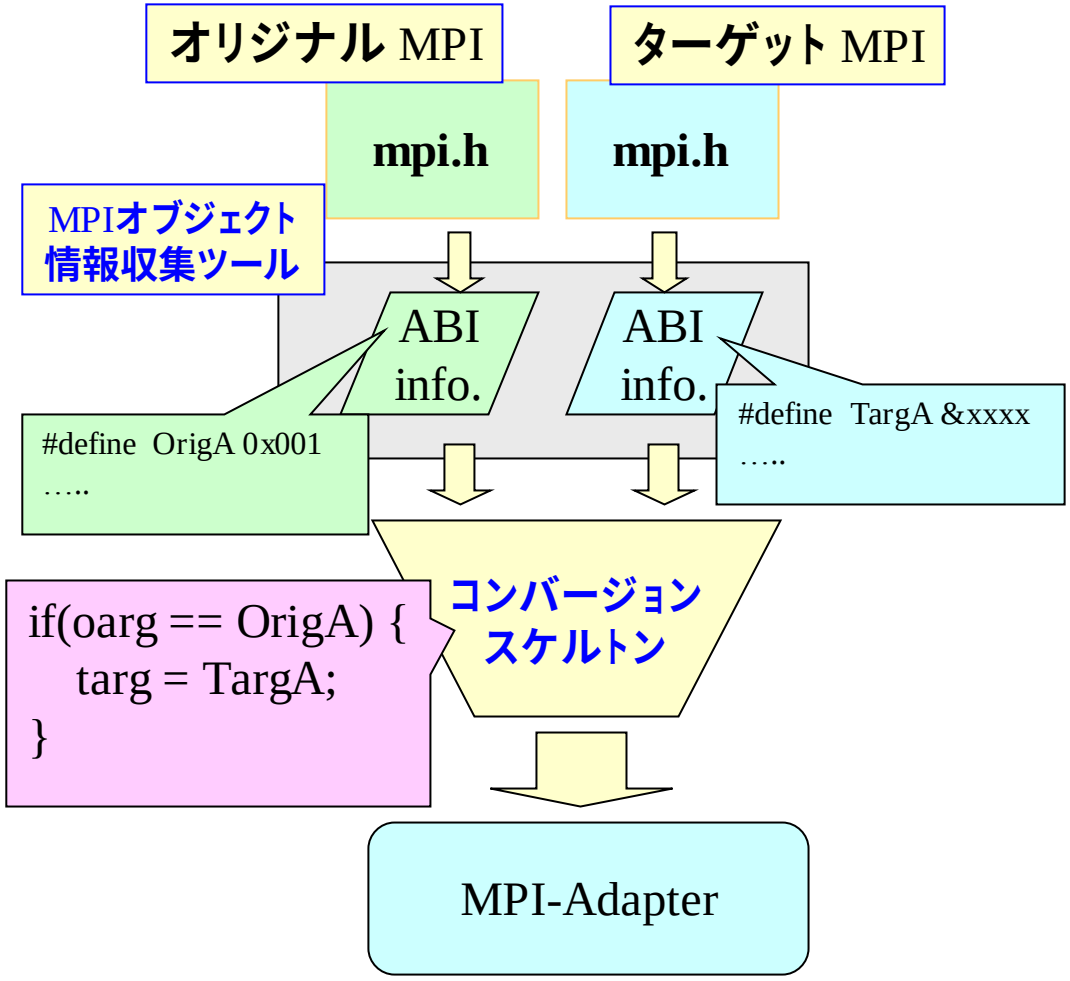
MPI-Adapterの実装と動作

- **変換手順：libmpi.so.0 準備**
  - MPI_Init時にターゲットMPIのライブラリをdlopen()して関数変換テーブルを作成
  - 関数実行時に型変換を行い、関数テーブル経由で呼出し
  - 返値も変換して返す
- **Miscライブラリ対応**
  - OpenMPIのmiscライブラリもダミーを準備 (libopen-rte.so.0, libopen-pal.so.0)



複数のMPIランタイムからの MPI ABI変換情報の自動抽出

- MPIオブジェクト情報収集ツールによりMPIヘッダ(mpi.h, mpif.h)からABI情報を抽出する
  - 一つのMPIランタイムにつき一つのABI情報
- 収集したABI情報から2つを選択してMPI コンバージョンスケルトンに導入することによりMPI-Adapterを生成.



MPI-Adapterの生成

FUJITSU

MPI-Adapter評価

- MPI-Adapter挿入オーバヘッド評価
  - MPI通信遅延（共有メモリ利用）
  - NAS並列ベンチマーク（Fortran, Class A,B,C）

- 実行環境
  - RX200S2クラスタ(Xeon 3.8GHz, SCore7.0）
  - ネットワーク：Intel E1000 NIC,
    Netgear 48Port Switch
  - 利用MPI MPICH2/SCore w/ PMX/Etherhxb,shmem

FUJITSU

MPI遅延によるオーバヘッド評価

MPI-Pingpong(mpi_rtt)プログラムによる測定, PMX/Shmem利用

usec	Fortran	C	Overhead (/MPI call)
Open MPI+ MPI-Adaptor	3.154	3.065	0.082(0.022)
MPICH2/SCore	3.103	3.055	0.048(0.012)
Overhead(/MPI)	0.051(0.013)	0.010(0.0025)	0.034(0.0085)

単位：usec

- Fortran to C ABIトランスレータのオーバヘッド
  - MPICH2=0.012usec, Open MPI=0.022usec
- MPI-Adapterのオーバヘッド(Open MPI → MPICH2)
  - Fortran（INT to INT）=0.013usec, C (Pointer to INT)=0.0025usec
- 総じて、挿入オーバヘッドは20ns程度と極めて小さい

FUJITSU

NPBによる評価(性能劣化率)

16x1の結果、Fortranはgfortran

16 node	BT	CG	FT	LU	MG	SP	IS
Class A	-0.1%	3.4%	0.2%	0.4%	-3.7%	0.0%	-0.4%
Class B	-2.2%	3.7%	2.9%	-0.5%	-0.8%	1.6%	4.8%
Class C	0.3%	1.2%	0.4%	0.1%	-5.0%	0.6%	0.3%

- C版のMPI-Adapterと同様の傾向
  - 最大で3.7%のオーバヘッド
  - MPI-Adapter挿入時の方が速い場合あり(-5.0%)
- 性能の違いは直接的なものではなく、間接的なもの

性能向上