FUJITSU

MPI-Adapter for Portable MPI Computing Environment

Shinji Sumimoto, Kohta Nakashima, Akira Naruse, Kouichi Kumon
(Fujitsu Laboratories Ltd.), Takashi Yasui (Hitachi Ltd.),
Yoshikazu Kamoshida, Hiroya Matsuba, Atsushi Hori,
Yutaka Ishikawa (The University of Tokyo)

FUJITSU

Outline of This Presentation

- Background
- Portable MPI Computing Environment
  - Issues of keeping MPI ABI compatibility
  - MPI-Adapter Approach
- Current Status
- MPI-Adapter Demonstration

FUJITSU

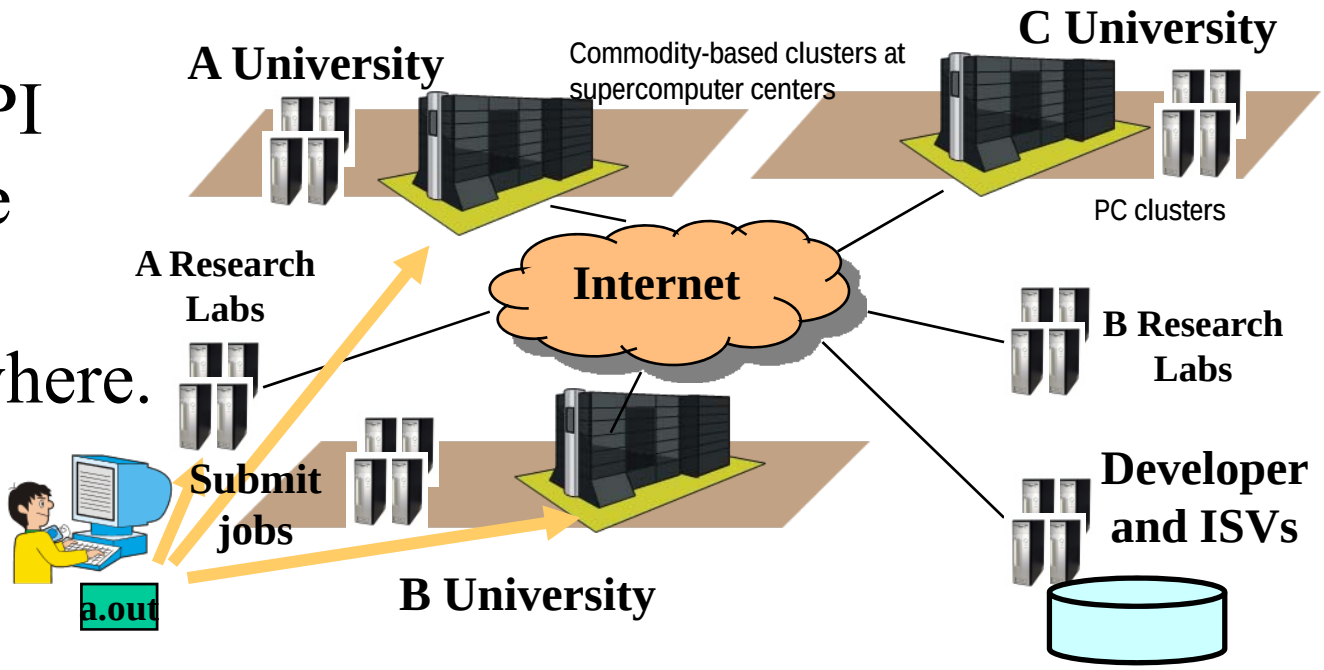
Background

- Commodity-based clusters are widely used for high – performance computing.
  - RICC, Tsubame, T2K etc. in JAPAN
- Users can use several clusters through the Internet.
- However, users must re-compile their program even if using PC clusters (x86 and Linux).
  - This limitation does little to expand PC cluster use.

- ABI compatibilities should be realized on PC Clusters.
  - Portable MPI Computing Environment

Portable MPI Computing Environment

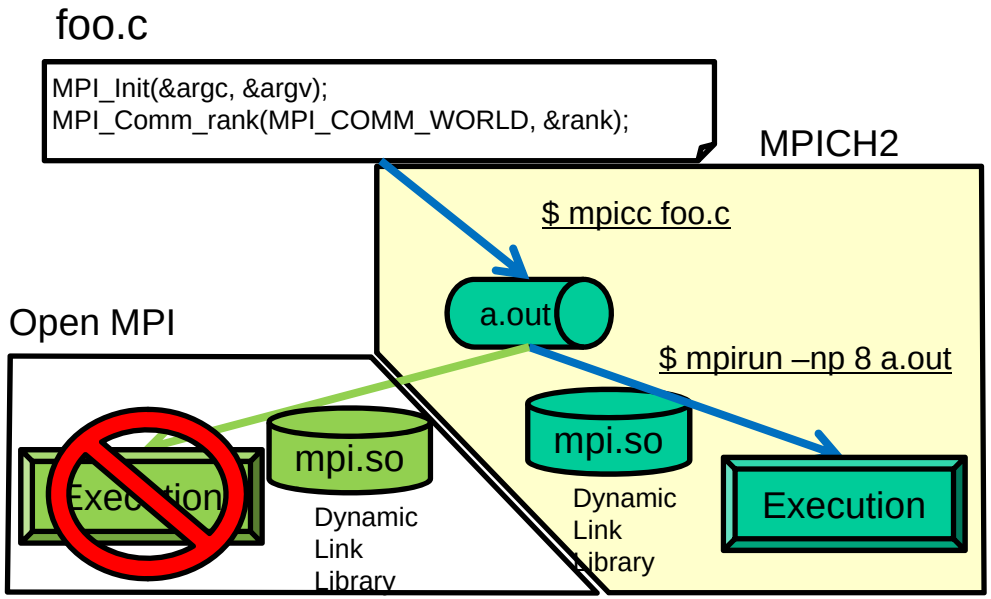
- Goal: Same MPI binaries are able to run on PC Clusters everywhere.



- Use Cases:
  - Selecting Clusters: for development and production
  - Binary Distribution: for ISV and developper
  - Changing Runtime Environment: for Functionality and performance issues

Issue of Portable MPI Computing Environment

- MPI standard does not define MPI application binary interface (ABI)
  - Ex. MPI_Comm type
    Open MPI: address type
    MPICH2: 32 bit integer



- Issue: Providing a mechanism to keep ABI among PC clusters

FUJITSU

Objects and Type Definitions on MPI Standard

- MPI standard defines several MPI objects and type definitions.
- Implementation of them depends on MPI runtime.
  - The differences are the reason of lack of ABI compatibility.

Objects	Types（a pre-defined value）
Communicator	MPI_Comm (MPI_COMM_WORLD)
Group	MPI_Group
Request	MPI_Request
Status	MPI_Status
Data type	MPI_Datatype (MPI_Int,)
Operation	MPI_Op (MPI_MAX)
Window	MPI_Win
File	MPI_File
Info	MPI_Info
Pointer diffs.	MPI_Aint
Offset	MPI_Offset
Error Handler	MPI_Errorhandler

Differences of Pre-defined Values between MPICH2 and Open MPI

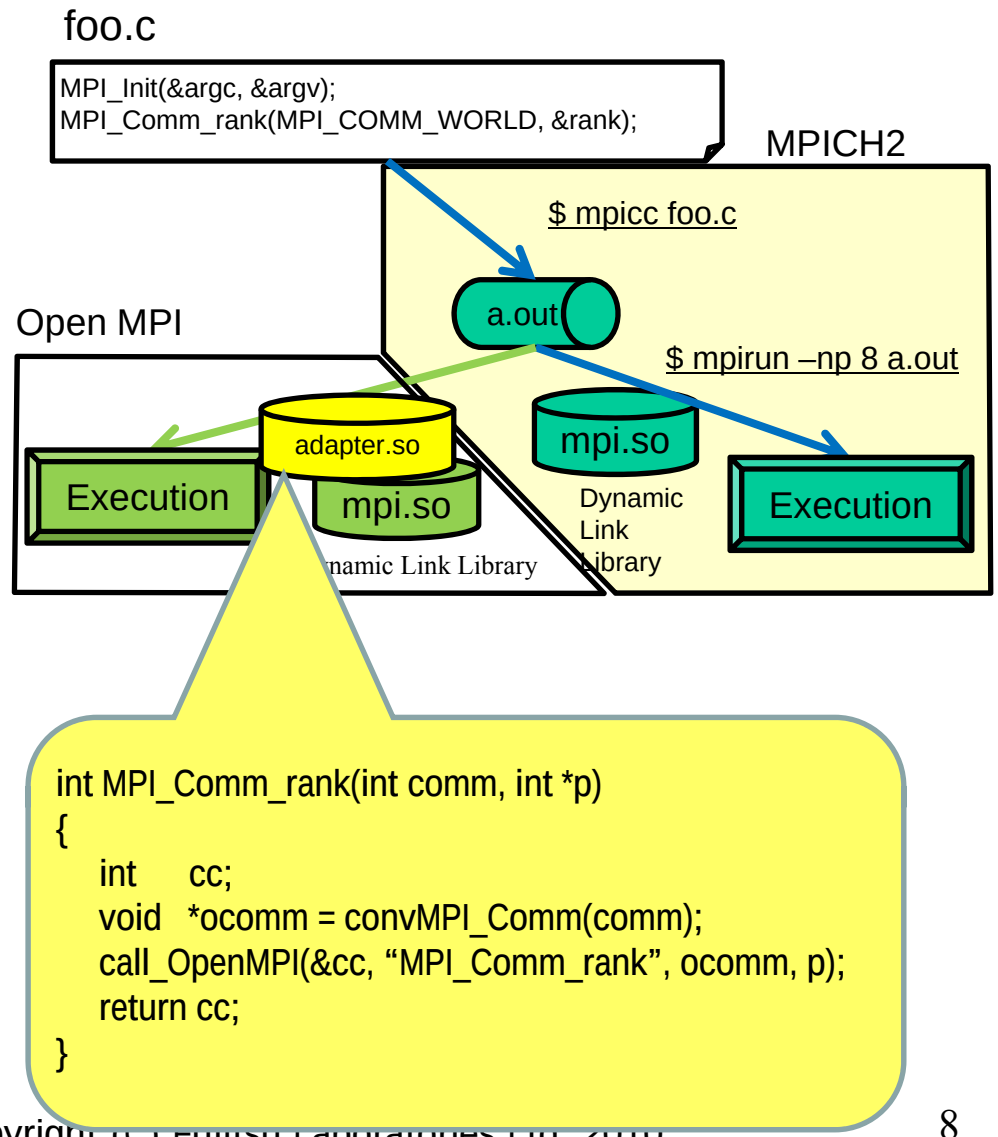
FUJITSU

	Pre-defined Values	MPICH2	Open MPI
C Lnag.	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	MPI_COMM_WORLD	0x44000000	0
	MPI_INTEGER	0x4c00041b	7
	MPI_SUCCESS	0	0
	MPI_ERR_TRUNCATE	14	15

- No ABI compatibility between MPICH2 and Open MPI
  - MPICH2: 32bit INT based implementation
  - Open MPI: Structure based implementation
- In Fortran implementation, 32 bit implementation, but values are different between MPICH2 and Open MPI

Realizing MPI ABI Compatibility

- Our Approach: MPI-Adapter Translation
  - Inserting MPI-Adapter between two different MPI distributions.
  - Dynamic Link Library Based
  - No need to modify Application Binaries and MPI Runtime Libraries



Differences of MPI Implementations

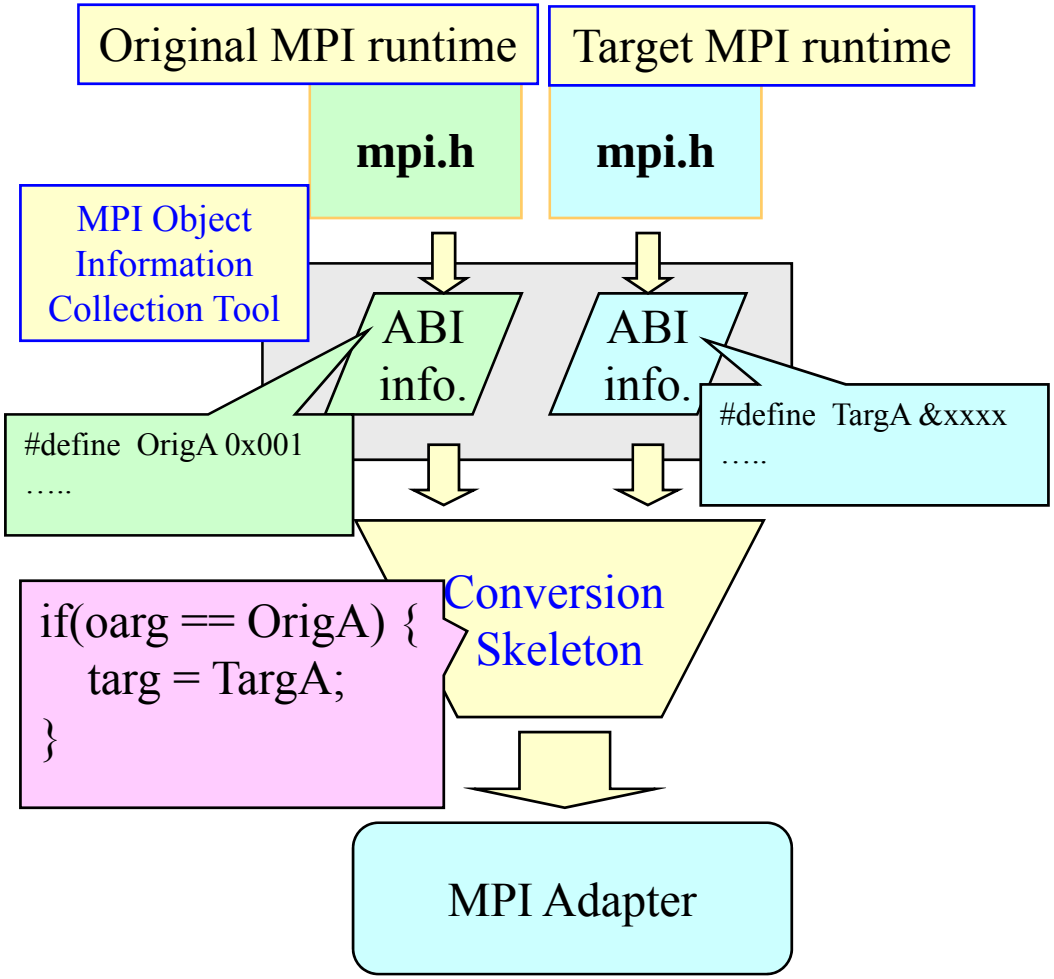
Survey of ABI Working Group (MPI Forum)

	Differences
Intel MPI	MPICH2 based (Integer)
MS MPI	MPICH2 based (Integer)
HP MPI	Original (Structure Based)
LAMPI	Original (Integer and Structure)
NEC MPI	Original? (Integer)

- Two Groups: Integer, Structure or Combination of Integer and Structure Based Implementation

How to Translate MPI ABI among several MPI Implementations Automatically?

- Getting ABI information from MPI headers (mpi.h, mpif.h) by using MPI Object Information Collection Tool
- Selecting two MPI ABI information and building MPI-Adapter by using Conversion Skeleton.
  - One ABI info. for one MPI implementation.



Making Process Flow of MPI Object Information

FUJITSU

Current Status of MPI-Adapter

- Developed a Tool for making ABI information and MPI-Adapter from MPI runtime automatically
- MPI-Adapter works well on several MPI runtimes:
  - MPICH2 based: MPICH2, MPICH2/SCore, MPICH2-MX, MVAPICH
  - Open MPI, HP MPI
- Test Status:
  - Basic MPI Functions are tested, not whole of MPI2 functions.
    - Intel MPI Benchmarks (IMB), NAS Parallel Benchmarks.
  - MPI-Adapter works well on several clusters in Fujitsu Labs and T2K Todai Cluster.

Some Cluster Environments using MPI-Adapter Portability Testing

	Distribution (Kernel) MPI	Glibc	GCC PE
Flab Cluster 1 RX200(Xeon)	CentOS 5.2 (2.6.18-8) MPICH2/SCore, Open MPI	2.5.12	4.1.1-52 16
Flab Cluster 2 HX600(Opteron)	CentOS 5.2 (2.6.18-92) MVAPICH, Open MPI	2.5-24	4.1.2-42 64
Flab PC Phenom	CentOS 5.3 (2.6.18-164) Open MPI, MPICH2	2.5-34	4.1.2-44 4
Flab PC2 Opteron	FedoraCore 11 (2.6.30-10) MVAPICH2, MPICH2	2.10-2	4.4.1-2 4
T2K Todai HA800	RedHat EL 5.1 (2.6.18-53) MPICH2-MX, HP MPI	2.5.24	4.1.2-14 256

- MPI-Adapter works well among these clusters

MPI-Adapter Overhead Evaluation on Fujitsu RX200 Cluster

FUJITSU

Using MPI-Pingpong(mpi_rtt) Program on PMX/Shmem

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)

Unit : usec

- Fortran to C ABI Translation Overhead
  - MPICH2=0.012usec, Open MPI=0.022usec
- MPI-Adapter Overhead (Open MPI → MPICH2)
  - Fortran（INT to INT）=0.013usec, C (Pointer to INT)=0.0025usec
- Overhead of inserting MPI-Adapter is quite small

Performance Difference using MPI-Adapter on MPICH2-MX Runtime at T2K-Todai Cluster

FUJITSU

256 PE, Fortran=gfortran

Class C	BT	CG	FT	LU	MG	SP
Open MPI	0.5%	0.3%	1.0%	2.3%	0.8%	-1.3%
HP MPI	0.5%	0.6%	0.2%	-0.3%	2.7%	-1.1%

Performance UP

- Open MPI binaries were compiled on Flab Cluster 1 and Copied to T2K-Todai Cluster.
- Performance Difference: Less than 2.7%

FUJITSU

MPI-Adapter Demonstrations

- Demonstration on VMware environment
  - Intel Core2 Duo(2 core), Cent OS 5.4, SCore7
  - MPI Runtimes: MPICH2, Open MPI, MPICH2/SCore
- Pre-build NAS Parallel benchmark Binaries
  - MPICH2, Open MPI, MPICH2/SCore, HP MPI
- Demonstration
  - Run mpirun program w/ (w/o) inserting MPI-Adapter

FUJITSU

Summary

- MPI-Adapter for Portable MPI Computing Environment.
  - Keeping MPI ABI compatibility by MPI ABI translator.
  - Implemented and Evaluated on T2K-Todai Cluster and several Fujitsu Clusters
    - Overhead of inserting MPI-Adapter is negligible
    - Works well among MPICH2/SCore, MPICH2, Open MPI, HP MPI runtimes
- Future Work
  - Tested among Three T2K Clusters (Tsukuba, Todai, and Kyoto), and entire MPI functions using MPI test suites.
  - Other Usage: Profiler Interface….
- Acknowledgement: This research was partially supported by the eScience project of the MEXT, Japan.

FUJITSU

Thank You.