Основи на MPI

Увод в паралелното програмиране #3

Материалите са разработени в рамките на договор ДО-02-136/2008 с Фонд Научни Изследвания
Проект: IRC-CoSiM Ръководител на проекта: проф. дфзн Ана Пройкова, ФзФ на СУ
http://www.irc-cosim.uni-sofia.bg/

Интерфейсът MPI

Преносимост

- Абстрактни алгоритми
- Конкретни програми реализаци (изходен код)
- Единен абстрактен механизъм за предаване на съобщения
- Абстракциите скриват особеностите на конкретните плаформи – middleware

MPI - история

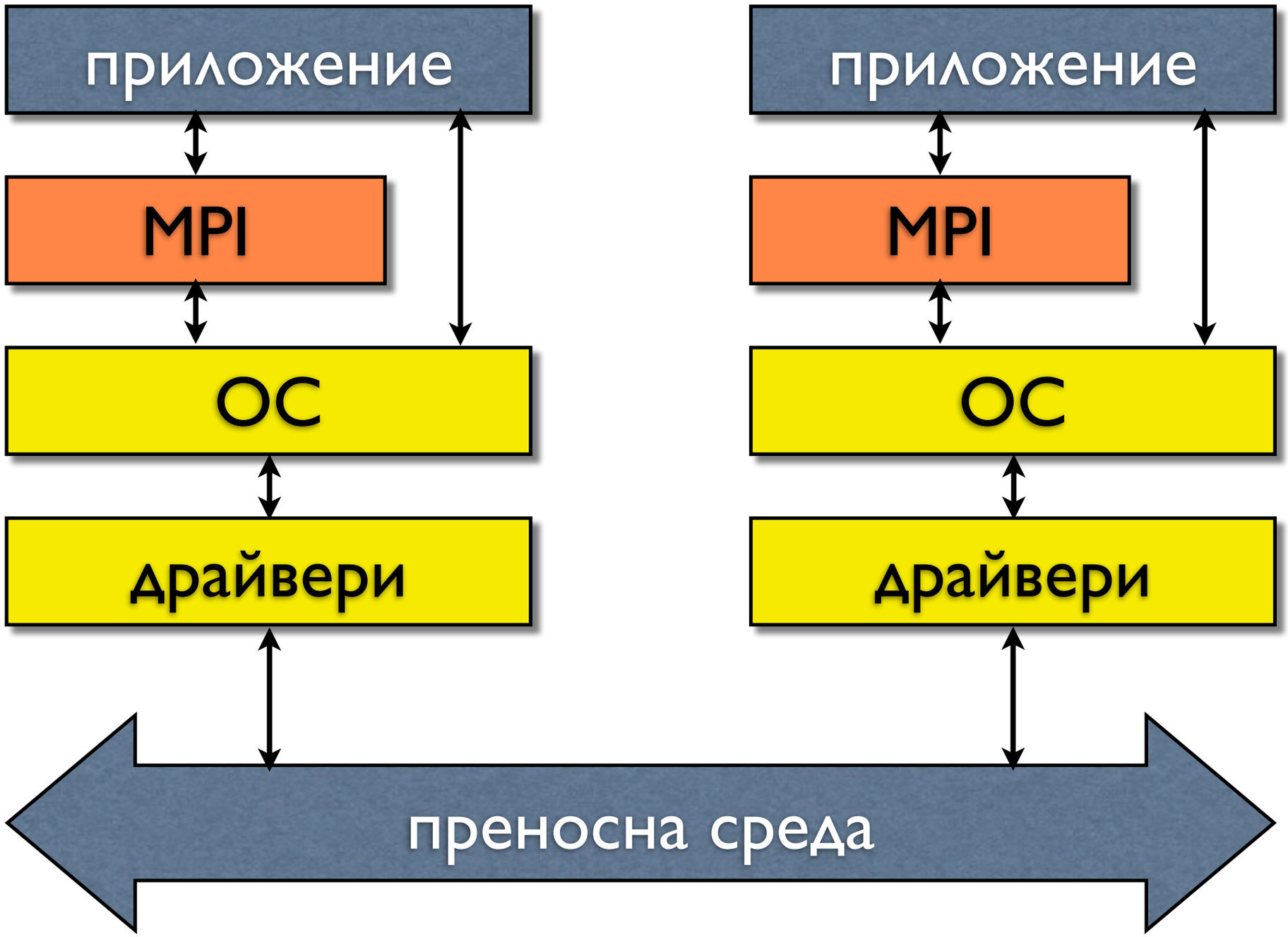
- MPI == Message Passing Interface
- Първият опит за стандартизиране на предаването на съобщения с участието на всички главни играчи в HPC

- MPI форум (1994)
  - 60 човека от 40 огранизации по цял свят
  - две години предложения и обсъждания
  - първа версия на стандарта
- MPI-2 (1996)

MPI - цели и обхвати

- Главни цели:
  - преносимост на изходния код
  - ефикасна реализация върху множество хардуерни платформи и среди
- Поддръжка на хетерогенни среди
- Стандартът не покрива явно:
  - начално зареждане на процесите
  - стартиране на нови процеси по време на изпълнение (стандарт в MPI-2)
  - отстраняване на грешки
  - паралелен вход/изход (стандарт в MPI-2)

MPI като middleware



Библиотеката MPI

- Широк набор функции (MPI 1.2):
  - комуникация
  - създаване и преобразуване на платформенонезависими типове
  - виртуални топологии
  - глобални редукции
  - синхронизация
  - точки за прикачване на външни програми (hooks)
- Еднотипен интерфейс към Fortran и C; интерфейс с класове за C++
- MPI-2 добавя:
  - паралелен вход/изход
  - динамичен контрол на броя MPI процеси

Реализации на MPI

- Един стандарт – множество реализации
- Съвместимост между реализациите:
  - еднаква семантика и прототип на функциите
  - няма гаранции за двоична съвместимост
- Оптимизирани реализации за конкретни HPC платформи
- Многоплатформени реализации:
  - LAM, Open MPI, MPICH и др.

Open MPI

- Наследник на LAM
- Свободна реализация с отворен код (нов BSD лиценз) за:
  - мрежи от работни станции (NoW)
  - клъстери
  - специализирани архитектури (напр. Cray)
- Пълна поддръжка на MPI-2
- Модулна архитектура (MCA)
- Интеграция с DRM (PBS, SGE и др.)
- http://www.open-mpi.org

Open MPI

- Набор от кооперирващи процеси под управление на OpenRTE (Open Run-Time Environment)
- Процесите комуникират директно чрез btl модули
- OpenRTE управлява зареждането на процесите
- Вселена от машини и процеси (OpenRTE universe)

OpenRTE

- Модулна поддържаща платформа за изпълнение на паралелни програми
- Open MPI програмите се изпълняват през mpirun
- Файл с изпълнителни възли (hostsfile)
- Модули за отдалечено стартиране:
  - rsh/ssh
  - интеграция с DRM

Open MPI на PHYSON

- Множество модули за различните версии:
  - **openmpi/1.3-gcc-4.2**
  - openmpi/1.3-intel-11.0
  - openmpi/1.4.3-gcc-4.2
  - openmpi/1.4.3-intel-11.1
- Интегриран със Sun Grid Engine:
  - паралелни среди ompi, ompix8, ompisn
  - mpirun -np $NSLOTS ...

Open MPI на PHYSON

Примерна задача за SGE:

```
#$ -cwd
#$ -pe ompix8 16

module load openmpi
mpirun -np $NSLOTS mpiapp.x
```

Основи на MPI

Манипулатори

- Базов обект в MPI
- Обозначава някой от множеството вътрешни обекти, поддържани от MPI
- Прости типове – подлежат на копиране чрез присвояване
- Основни типове манипулатори:
  - комуникатори
  - типове данни
  - оператори на редукция
  - паралелен вход/изход (MPI-2)
- Константи за предефинираните манипулатори

Типове данни

- Съобщенията в MPI са строго типизирани
- Изпълнението върху хетерогенни архитектури налага някои ограничения и изискване за преобразуване на представянето на данните
- MPI предлага предефинирани типове за всички прости типове от C и Fortran
- Възможност за комбиниране на типовете в производни типове – матрици, подматрици, структури и др.
- Произволно влагане на типове за сложни обекти

Типове данни от C

Манипулатор на тип в MPI	Тип данни от C
MPI_CHAR	signed char
MPI_SHORT	signed short
MPI_INT	signed int
MPI_LONG	signed long
MPI_UNSIGNED_CHAR	unsigned char
MPI_UNSIGNED_SHORT	unsigned short
MPI_UNSIGNED	unsigned int
MPI_UNSIGNED_LONG	unsigned long
MPI_FLOAT	float
MPI_DOUBLE	double
MPI_LONG_DOUBLE	long double
MPI_BYTE	1 байт
MPI_PACKED	

Типове данни от Fortran

Манипулатор на тип в MPI	Тип данни от Fortran
MPI_INTEGER	INTEGER
MPI_REAL	REAL
MPI_DOUBLE_PRECISION	DOUBLE PRECISION
MPI_COMPLEX	COMPLEX
MPI_LOGICAL	LOGICAL
MPI_CHARACTER	CHARACTER*1
MPI_BYTE	1 байт
MPI_PACKED	

Конвенции на интерфейса

- Конвенция на именуване:
  - C: *MPI_First_word_capitalised_others_not()*
  - FORTRAN 77: *MPI_ALL_UPPERCASE()*
  - Fortran 90+: *MpI_CaSe_does_NOT_mAttEr()*
  - константи във всички езици: *MPI_CONSTANTS_UPPERCASE*
- Конвенция на извикванията:
  - C: функции, които връщат int код за грешка
  - Fortran: подпрограми, чийто последен аргумент е INTEGER код за грешка
  - *MPI_Wtime()* и *MPI_Wtick()* - връщат double (C) или DOUBLE PRECISION (Fortran) и нямат аргументи и в двата езика

Включване на MPI в програмите

- Заглавен файл за C

```
#include <mpi.h>
```

- Файл за включване за FORTRAN 77

```
IMPLICIT NONE
INCLUDE ‘mpif.h’
```

- Модул за Fortran 90+

```
use mpi
implicit none
```

Инициализация

- Първоначална инициализация на MPI с MPI_Init
  - C: *int MPI_Init(int *pargc, char ***pargv);*
  - Fortran: *SUBROUTINE MPI_INIT(ierr)*
- Много малко MPI функции работят преди инициализацията
- Всички участващи процеси трябва да инициализират собственото копие на библиотеката!

Финализиране

- Всички процеси трябва да финализират изпълнението си посредством извикване на MPI_Finalize
  - C: *int MPI_Finalize();*
  - FORTRAN: *SUBROUTINE MPI_FINALIZE(ierr)*
- Финализирането позволява на MPI да изчисти всички буфери и да освободи заетите ресурси
- Никакви следващи MPI извиквания не са позволени след финализирането

Идентичност на процеса

- Набор от комуникиращи процеси – комуникатор
- Всеки процес има уникален адрес в комуникатора – ранг
- Брой на процесите:
  - *int MPI_Comm_size(MPI_Comm comm, int *size);*
  - *SUBROUTINE MPI_COMM_SIZE(comm, size, ierr)*
- Ранг на конкретния процес:
  - *int MPI_Comm_rank(MPI_Comm comm, int *rank);*
  - *SUBROUTINE MPI_COMM_RANK(comm, rank, ierr)*

Помогни си сам

- Документация:
  - man страници на Open MPI на PHYSON
    man MPI_What_function
  - man страници на други MPI реализации
  - документация на конкретна MPI реализация
  - Интернет търсачки
- Книги:
  - “Parallel Programming with MPI” (Morgan Kaufmann, ISBN 978-1558603394)
  - “Using MPI” (The MIT Press, ISBN 978-0262571326)
- Други курсове и самоучители