

# 高性能计算与云计算课程实验



授课老师：何克晶 副教授

# MPI

- \* 什么是MPI (Message Passing Interface)
- \* 是函数库规范，而不是并行语言；操作如同库函数调用
- \* 是一种标准和规范，而非某个对它的具体实现（MPICH等），与编程语言无关
- \* 是一种消息传递编程模型，并成为这类编程模型的代表

# MPI几个重要概念

- \* 进程组 (process group) 指MPI程序的全部进程集合的一个有序子集, 进程组中每个进程被赋予一个在该组中唯一的序号(rank), 用于在该组中标识该进程。序号的取值范围是[0,进程数-1]
- \* 通信器/通信子 (communicator)
- \* 基于某个进程组来定义, 且在该进程组, 进程间可以相互通信
- \* 任何MPI通信函数都必须通过通信器进行
- \* MPI系统提供省缺的通信器MPI\_COMM\_WORLD, 所有启动的MPI进程通过调用函数MPI\_Init()包含在该通信器内; 各进程通过函数MPI\_Comm\_size()获取通信器包含的(初始启动)的MPI进程个数
- \* 域内通信器 (同一进程组内的通信) 和域间通信器 (不同进程组进程间的通信, 只具备点对点通信)

# MPI几个重要概念

- \* 进程序号 (rank) 用来在一个进程组或通信器中标识一个进程
  - \* MPI 程序中的进程由进程组或通信器内的序号唯一确定, 序号相对于进程组或通信器而言 (假设 $np$ 个进程, 标号 $0 \dots np-1$ )
  - \* 同一个进程在不同的进程组或通信器中可以有不同的序号, 进程的序号是在进程组或通信器被创建时赋予的
  - \* MPI 系统提供了一个特殊的进程序号MPI PROC NULL, 它代表空进程(不存在的进程), 与MPI\_PROC\_NULL间的通信实际上没有任何作用
- 
- \* 消息 (message)
  - \* 分为数据 (data) 和包装 (envelope) 两个部分
  - \* 包装由接收进程序号/发送进程序号、消息标号和通信器三部分组成; 数据包含用户将要传递的内容

# MPI数据类型

MPI Datatype	C Datatype
MPI_CHAR	Signed char
MPI_SHORT	Signed short int
MPI_INT	Signed int
MPI_LONG	Signed long int
MPI_UNSIGNED_CHAR	Unsigned char
MPI_UNSIGNED_SHORT	Unsigned short int
MPI_UNSIGNED	Unsigned int
MPI_UNSIGNED_LONG	Unsigned long int
MPI_FLOAT	Float
MPI_DOUBLE	Double
MPI_LONG_DOUBLE	Long double
MPI_BYTE	
MPI_PACKED	

# MPI程序基本结构

MPI include file

变量定义

MPI 环境初始化

执行程序  
进程间通信

退出 MPI 环境

```
#include <mpi.h>
void main (int argc, char *argv[])
{
    int np, rank, ierr;
    ierr = MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &np);
    /*      Do Some Works          */
    ierr = MPI_Finalize();
}
```



Thanks !