

MPI 常用例程

初等例程

函数名	参数	动作
<code>int MPI_Init(int *argc, char **argv[])</code>	来自 <code>main()</code> 的参数 来自 <code>main()</code> 的参数	初始化 MPI 环境
<code>int MPI_Finalize(void)</code>		终止 MPI 执行环境
<code>int MPI_Comm_rank(MPI_Comm comm, int *rank)</code>	通信子 序号 (返回的)	确定在通信子中进程的序号
<code>int MPI_Comm_size(MPI_Comm comm, int *size)</code>	通信子 组的大小 (返回的)	确定与通信子关联的组的大小
<code>double MPI_Wtime(void)</code>		以秒为单位返回从过去某点开始的执行时间

点对点消息传递

函数名	参数	动作
<pre>int MPI_Send(void *buf, int count, MPI_Datatype datatype, int dest, int tag, MPI_Comm comm)</pre>	发送缓冲区 缓冲区内的项数 项的数据类型 目的进程序号 消息标识 通信子	发送消息（锁定） MPI_Datatype 与 C 对应 MPI_CHAR signed char MPI_INT signed int MPI_FLOAT float
<pre>int MPI_Recv(void *buf, int count, MPI_Datatype datatype, int source, int tag, MPI_Comm comm, MPI_Status *status)</pre>	接收缓冲区（已装载） 缓冲区内最大项数 项的数据类型 源进程序号 消息标识 通信子 状态（返回的）	接收消息（锁定）
<pre>int MPI_Isend(void *buf, int count, MPI_Datatype datatype, int dest, int tag, MPI_Comm comm, MPI_Request *request)</pre>	发送缓冲区 缓冲区内消息项数 消息项的数据类型 目的进程序号 消息标识 通信子 请求句柄（返回的）	启动非锁定发送 相关的例程： MPI_Ibsend() 启动缓冲 MPI_Irsend() 启动就绪 MPI_Issend() 启动同步
<pre>int MPI_Irecv(void *buf, int count, MPI_Datatype datatype, int source, int tag, MPI_Comm comm, MPI_Request *request)</pre>	接收缓冲区 缓冲区内消息项数 消息项的数据类型 源进程序号 消息标识 通信子 请求句柄（返回的）	启动非锁定接收
<pre>int MPI_Wait(MPI_Request *request, MPI_Status *status)</pre>	请求句柄 状态（如果等待，则同 MPI_recv() 的返回状态）	等待 MPI 发送或接收结束，然后返回
<pre>int MPI_Test(MPI_Request *request, int *flag, MPI_Status *status)</pre>	请求句柄 操作结束则返回真（返回的） 状态（返回的）	测试非锁定操作是否结束
<pre>int MPI_Probe(int source, int tag, MPI_Comm comm, MPI_Status *status)</pre>	源进程序号 消息标识 通信子 状态（返回的）	对消息进行锁定测试（不接收消息）
<pre>int MPI_Iprobe(int source, int tag, MPI_Comm comm, int *flag, MPI_Status *status)</pre>	源进程序号 消息标识 通信子 有消息则为真（返回的） 状态（返回的）	对消息进行非锁定测试（不接收消息）

组例程

函数名	参数	动作
<code>int MPI_Barrier(MPI_Comm comm)</code>	通信子	锁定进程直到所有进程已调用该进程
<code>int MPI_Bcast(void *buf, int count, MPI_Datatype, int root, MPI_Comm comm)</code>	接收缓冲区（已装载） 缓冲区内最大项数 项的数据类型 根进程序号 通信子	从根进程向 comm 中的所有进程和自身广播消息
<code>int MPI_ALLtoall(void *sendbuf int sendcount, MPI_Datatype, sendtype, void *recvbuf, int recvcount, MPI_Datatype recvtype, MPI_Comm comm)</code>	发送缓冲区 发送缓冲区内元素数 发送元素的数据类型 接收缓冲区（已装载） 每次接收的元素数 接收元素的数据类型 通信子	从所有进程向所有进程发送消息 相关例程： MPI_ALLtoallv() 带偏移
<code>MPI_Gather(void *sendbuf, int sendcount, MPI_Datatype sendtype, void *recvbuf, int recvcount, MPI_Datatype recvtype, int root, MPI_Comm comm)</code>	发送缓冲区 发送缓冲区内元素数 发送元素的数据类型 接收缓冲区（已装载） 每次接收的元素数 接收元素的数据类型 接收进程序号 通信子	汇集进程组的各个值 相关例程： MPI_Allgather() 并向所有进程分发 MPI_Gatherv() 汇集到指定单元中，在 recvcount 后附加参数 MPI_Allgatherv() 汇集到指定单元中并向所有进程分发，在 recvcount 后附加参数
<code>MPI_Scatter(void *sendbuf, int sendcount, MPI_Datatype sendtype, void *recvbuf, int recvcount, MPI_Datatype recvtype, int root, MPI_Comm comm)</code>	发送缓冲区 发送缓冲区内元素数 发送元素的数据类型 接收缓冲区（已装载） 每次接收的元素数 接收元素的数据类型 接收进程序号 通信子	从根进程部分地散播缓冲区中的值到进程组 相关例程： MPI_Scatterv() 从散播根进程按特定部分散播缓冲区中的值到进程组 MPI_Reduce_scatter() 规约值并散播结果
<code>int MPI_Reduce(void *sendbuf, void *recvbuf, int count, MPI_Datatype recvtype, MPI_OP op, int root, MPI_Comm comm)</code>	发送缓冲区地址 接收缓冲区地址 发送缓冲区内元素数 发送元素的数据类型 规约操作 MPI_MAX 最大 MPI_MIN 最小 MPI_SUM 求和 MPI_PROD 求积 存放结果的根进程号 通信子	将所有进程值规约为一个值 相关例程： MPI_ALLreduce() 规约成单个值并返回给所有进程