Message Passing Interface
Lecture 1

Manfred Liebmann
Technische Universität München
Chair of Optimal Control
Center for Mathematical Sciences, M17
manfred.liebmann@tum.de

November 17, 2015

Message Passing Interface (MPI)

The Message Passing Interface Standard (MPI) is a message passing library standard based on the consensus of the MPI Forum, which has over 40 participating organizations, including vendors, researchers, software library developers, and users. The goal of the Message Passing Interface is to establish a portable, efficient, and flexible standard for message passing that will be widely used for writing message passing programs. As such, MPI is the first standardized, vendor independent, message passing library. The advantages of developing message passing software using MPI closely match the design goals of portability, efficiency, and flexibility. MPI is not an IEEE or ISO standard, but has in fact, become the "industry standard" for writing message passing programs on HPC platforms.

Message Passing Interface Tutorial:

`https://computing.llnl.gov/tutorials/mpi/`

Message Passing Interface Forum

The MPI Forum is an open group with representatives from many organizations that define and maintain the MPI standard.

`http://www.mpi-forum.org`

MPI Documents: MPI-3.1 was approved by the MPI Forum on June 4, 2015.

`http://www.mpi-forum.org/docs/docs.html`

Point-to-Point Communication

Sending and receiving of messages by processes is the basic MPI communication mechanism. The basic point-to-point communication operations are **send** and **receive**.

```
#include "mpi.h"
int main( int argc, char *argv[])
{
  char message[20];
  int myrank;
  MPI_Status status;
  MPI_Init( &argc, &argv );
  MPI_Comm_rank( MPI_COMM_WORLD, &myrank );
  if (myrank == 0)    /* code for process zero */
  {
      strcpy(message,"Hello, there");
      MPI_Send(message, strlen(message)+1, MPI_CHAR, 1, 99, MPI_COMM_WORLD);
  }
  else if (myrank == 1)  /* code for process one */
  {
      MPI_Recv(message, 20, MPI_CHAR, 0, 99, MPI_COMM_WORLD, &status);
      printf("received :%s:\n", message);
  }
  MPI_Finalize();
  return 0;
}
```

Build Scripts and Program Execution

Building MPI programs requires special build scripts for C/C++ programming language.

```
mpicc example1.c -o example1
```

```
mpicxx example1.cpp -o example1
```

MPI programs are executed with the `mpirun` command. The `-np` option defines the number of processes.

```
mpirun -np 2 ./example1
```

MPI library Implementations: OpenMPI, MVAPICH

MPI Collective Communication Routines

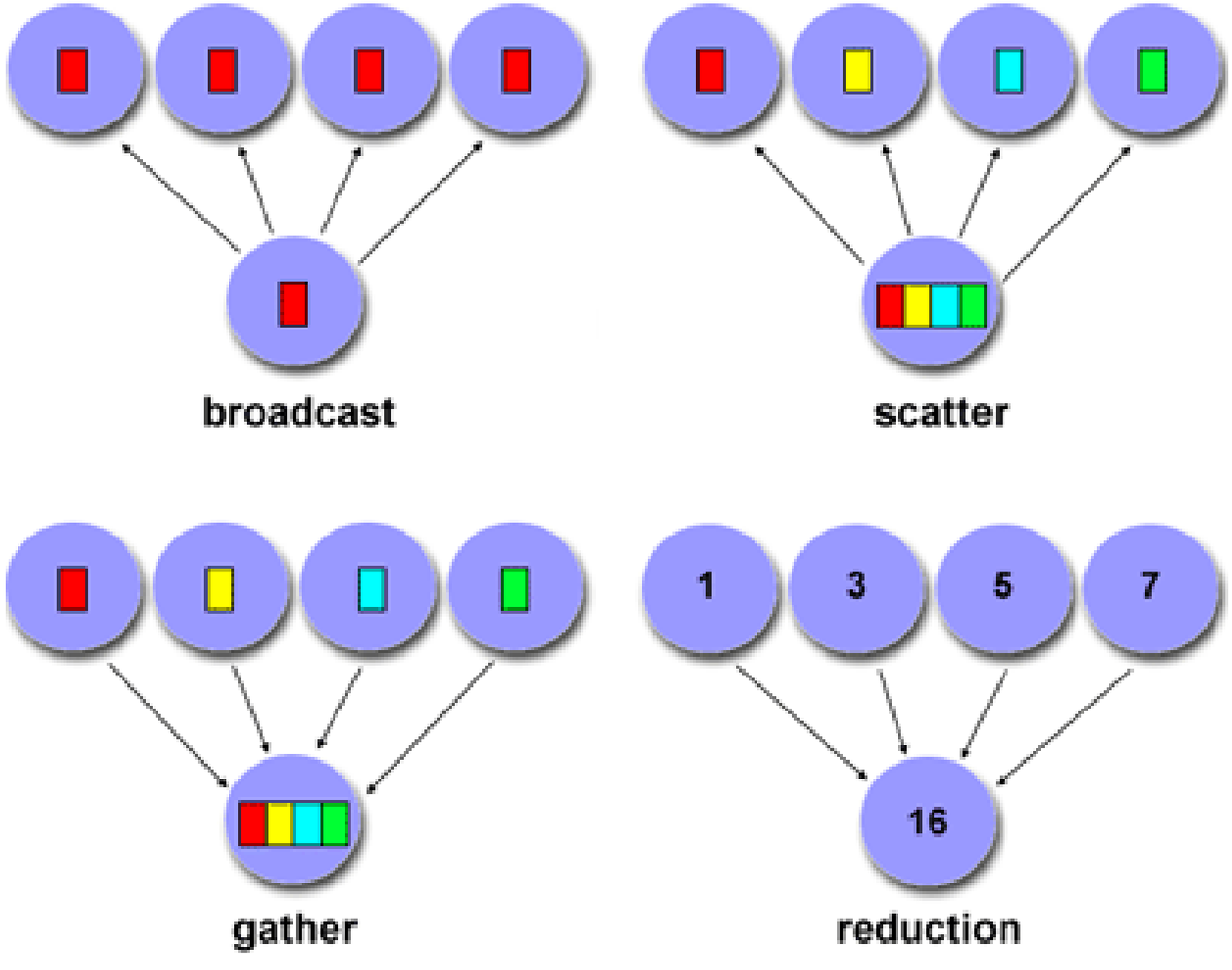

Figure 1: Collective communication routines: broadcast, scatter, gather, reduction

MPI Communicators and Groups

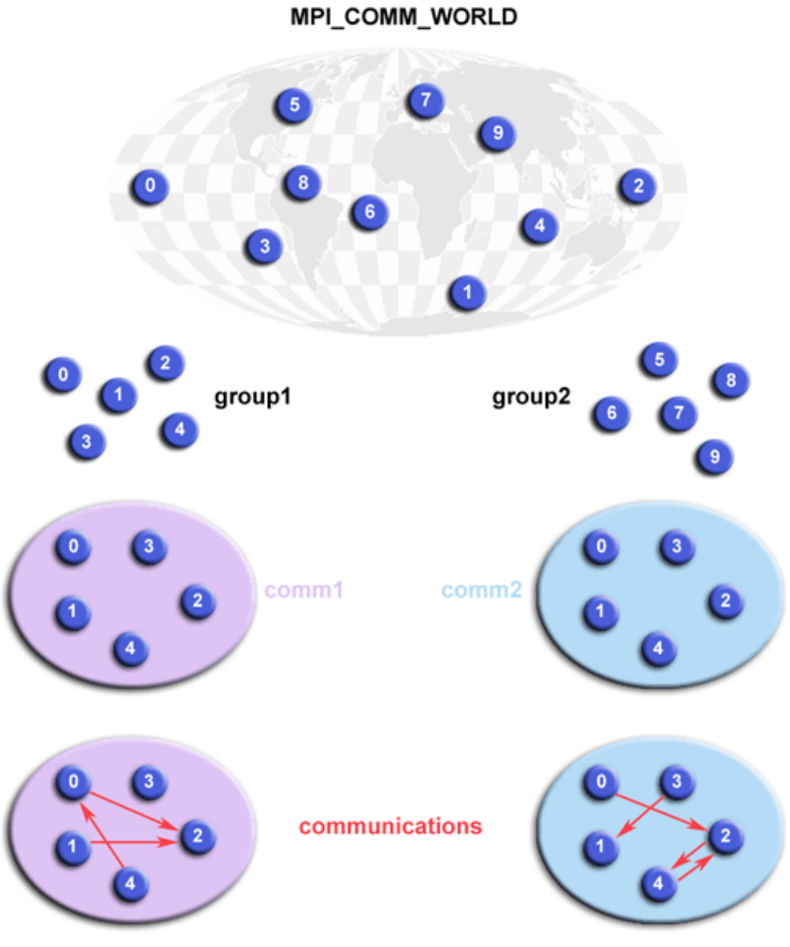

Figure 2: MPI communicators and groups

Reading List

W. Gropp: *Using MPI: Portable Parallel Programming with the Message-Passing Interface*, MIT Press 2014