Cluster Computing

MPI

Industrial Standard Message Passing

Cluster Computing

MPI Features

- Industrial Standard
  - Highly portable
  - Widely available
- SPMD programming model
  - Synchronous execution

Cluster Computing

MPI Outer scope

- int MPI_Init(int *argc, char ** argv)
- int MPI_Finalize()

Cluster Computing

Basic Communication with MPI

- Recieve
  - Blocking
    - int MPI_ Recv(void *buf, int count, MPI_Datatype,
      int source, int tag,
      MPI_ comm comm,
      MPI_ Status *status)
  - Nonblocking
    - int MPI_ Irecv(void *buf, int count, MPI_Datatype,
      int source, int tag,
      MPI_ comm comm,
      MPI_ Request *request)

Cluster Computing

Basic Communication with MPI

- Send
  - Standard
    - Blocking
      - int MPI_Send(void *buf, int count, MPI_ Datatype,
        int dest, int tag, MPI_Comm comm)
    - Nonblocking
      - int MPI_ISend(void *buf, int count, MPI_ Datatype,
        int dest, int tag, MPI_ Request *request)

Cluster Computing

Basic Communication with MPI

- Send
  - Ready
    - Blocking
      - int MPI_Rsend(void *buf, int count, MPI_ Datatype,
        int dest, int tag, MPI_Comm comm)
    - Nonblocking
      - int MPI_IrSend(void *buf, int count, MPI_ Datatype,
        int dest, int tag, MPI_Request *request)

Cluster Computing

Basic Communication with MPI

- Send
  - Synchronous
    - Blocking
      - int MPI_Ssend(void *buf, int count, MPI_ Datatype,
        int dest, int tag, MPI_comm comm)
    - Nonblocking
      - int MPI_Issend(void *buf, int count, MPI_ Datatype,
        int dest, int tag, MPI_Request *request)

Cluster Computing

Basic Communication with MPI

- Send
  - Buffered
    - Blocking
      - int MPI_Bsend(void *buf, int count, MPI_ Datatype,
        int dest, int tag, MPI_comm comm)
    - Nonblocking
      - int MPI_Ibsend(void *buf, int count, MPI_ Datatype,
        int dest, int tag, MPI_Request *request)

Cluster Computing

Finalizing nonblocking operations

- Wait for operation to complete
  - Int MPI_Wait(MPI_Request *request,
    MPI_Status *status)
- Test if the operation has completed
  - Int MPI_Test(MPI_Request *request,
    int *flag,
    MPI_Status *status)

Cluster Computing

Wait variations

- MPI_Waitall
- MPI_Waitany
- MPI_Waitsome

Cluster Computing

Peeking for messages

- Blocking test for a message
  - MPI_Probe(int source, int tag,
    MPI_Comm comm,
    MPI_Status *status);
- Nonblocking probe
  - MPI_Probe(int source, int tag,
    MPI_Comm comm,
    int *flag;
    MPI_Status *status);

Cluster Computing

Addressing in MPI

- All processes are named continously from 0 within one or more process groups
- Process address
  - (id, communicator)

Cluster Computing

Data Type Support in MPI

C Datatype	MPI Datatype
int	MPI_INT
long	MPI_LONG
float	MPI_FLOAT
double	MPI_DOUBLE
char	MPI_CHAR
(struct)	MPI_PACKED
	MPI_ BYTE

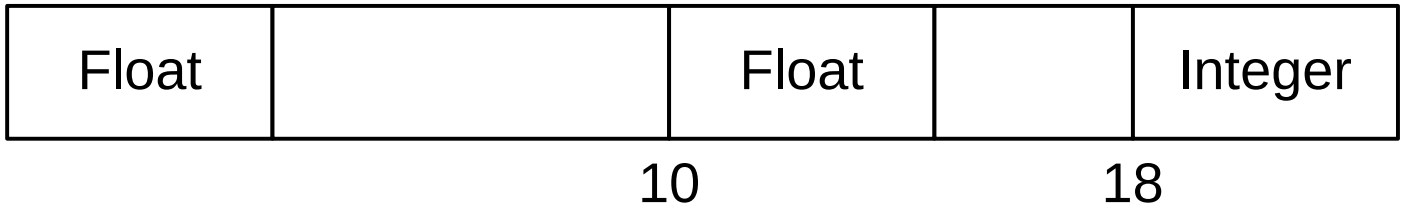
Cluster Computing

Application defined data-types

- MPI can handle userdefined datatypes
- A datatype is definied as
  - A vector of types
  - A vector on instances
  - A vector of displacements

Cluster Computing

Application defined data-types



(float, float, int)
(1,1,1)
(0,10,18)

Cluster Computing

Application defined data-types

```
int blk_ len[ 3] ={ 1,1,1};
MPI_Aint displ[ 3], start_ addr, addr;
MPI_Datatype typel[ 3]={ MPI_ FLOAT, MPI_ FLOAT, MPI_ INT};
displ[ 0] = 0;
MPI_Address (a,& start_ addr);
MPI_Address (b,& addr);
displ[1] = addr - start_ addr;
MPI_Address (n,& addr);
displ[2] = addr - start_ addr;
MPI_Type_struct (3, blk_ len, displ, typel, mesg_ ptr);
MPI_Type_commit (mesg_ ptr);
```

Cluster Computing

Communicator

- MPI processes belong to one or more communicators
- Addressing MPI processes always include a communicator and a absolute address within the communicator
- All processes initially belong to the communicator MPI_COMM_WORLD

Cluster Computing

Communicators

- int MPI_ Comm_ Size(MPI_ comm comm, int* size)
- int MPI_ Comm_ Rank(MPI_ comm comm, int* rank)

Cluster Computing

All for one and one for all

- Barriers
- Broadcasting
- Multicasting
- Scattering data
- Gathering data
- All to All
- Reductions
- Scanning

Cluster Computing

Barriers

- MPI provides a generel barrier mechanism
- int MPI_Barrier(MPI_COMM_WORLD);

+ Very easy to use

+ Can be optimized for the specific topology

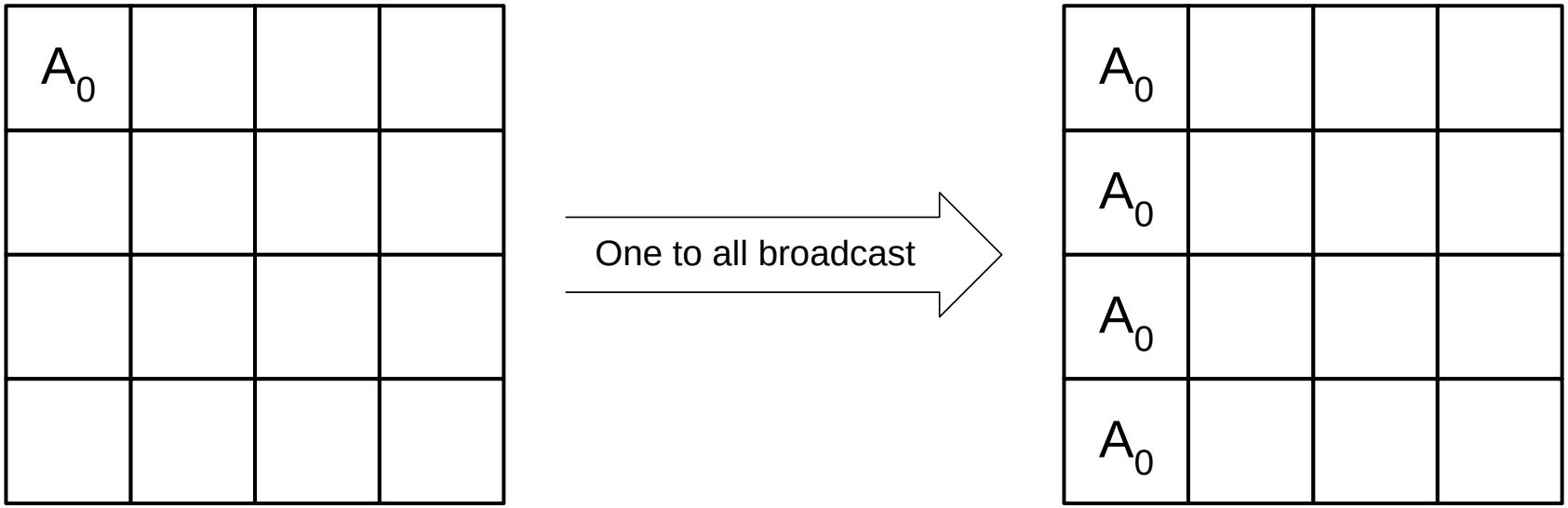
Cluster Computing

Broadcasting

- Send a message to all participants
- int MPI_Bcast(void *buf, int count,
  MPI_ Datatype datatype,
  int root,
  MPI_ comm comm)

Cluster Computing

Broadcast



Cluster Computing

All to All

- Efficient way to distribute partial results to all participants
- int MPI_Alltoall(void *sendbuf, int scount,
  MPI_Datatype
  sdatatype,
  void *recvbuf, int rcount,
  MPI_Datatype rdatatype,
  MPI_ comm comm)

Cluster Computing

All to All

Send Buffer

Data

Processor

A_0	B_0	C_0	D_0	E_0	F_0
A_1	B_1	C_1	D_1	E_1	F_1
A_2	B_2	C_2	D_2	E_2	F_2
A_3	B_3	C_3	D_3	E_3	F_3
A_4	B_4	C_4	D_4	E_4	F_4
A_5	B_5	C_5	D_5	E_5	F_5

Receive Buffer

Data

Processor

A_0	A_1	A_2	A_3	A_4	A_5
B_0	B_1	B_2	B_3	B_4	B_5
C_0	C_1	C_2	C_3	C_4	C_5
D_0	D_1	D_2	D_3	D_4	D_5
E_0	E_1	E_2	E_3	E_4	E_5
F_0	F_1	F_2	F_3	F_4	F_5

Cluster Computing

Multicasting

- Multicasting does not make sence in MPI
- Instead a processgroup can be created and broadcast within the group can be used

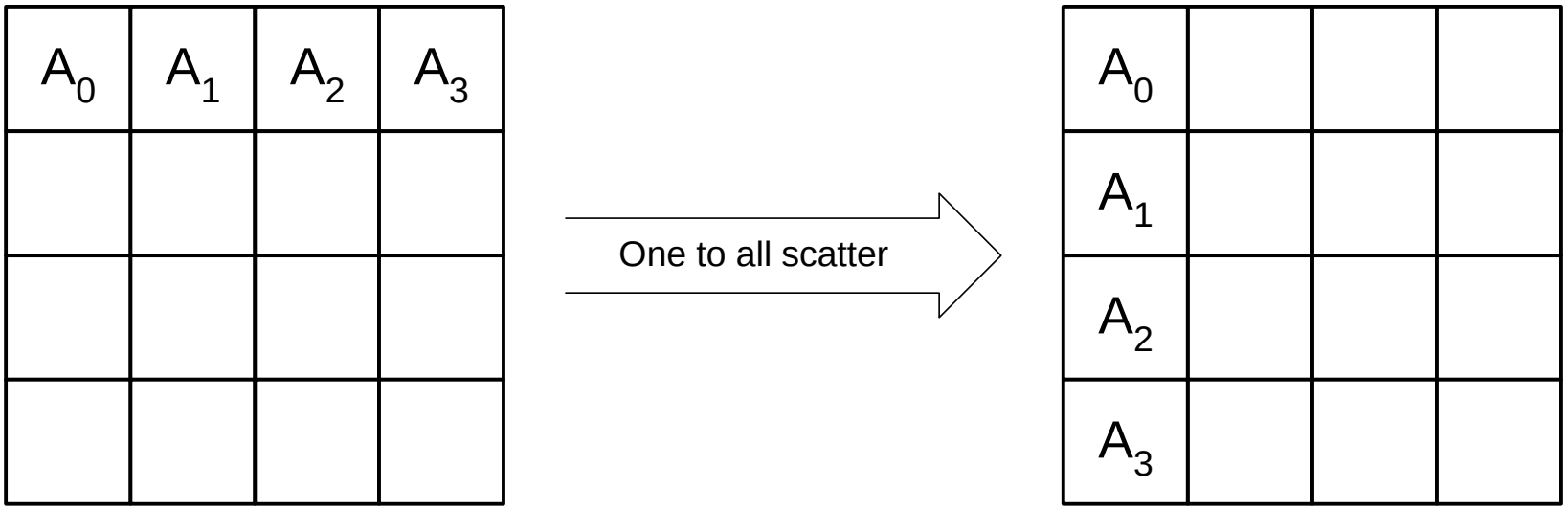
Cluster Computing

Scattering data

- Allows easy distribution of data amongst processors
- int MPI_Scatter(void *sendbuf, int scount,
  MPI_Datatype sdatatype,
  void *recvbuf, int rcount,
  MPI_Datatype rdatatype,
  int root,
  MPI_ comm comm)

Cluster Computing

Scattering data



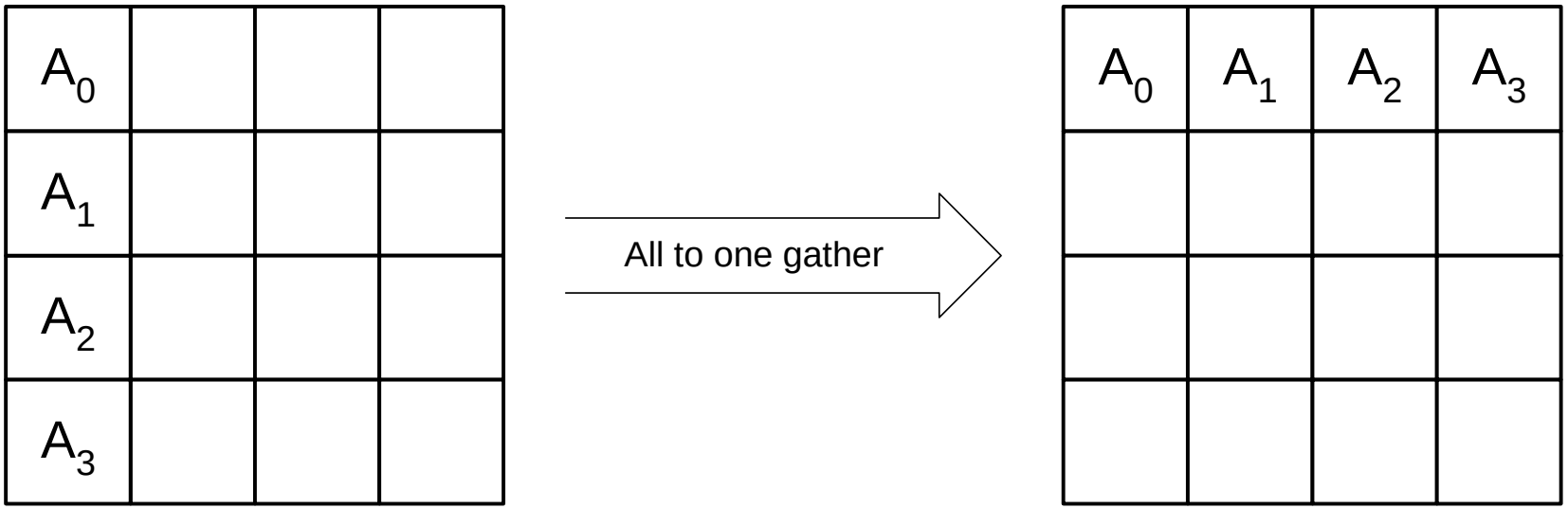
Cluster Computing

Gathering data

- Easy collection of results
- int MPI_Gather(void *sendbuf, int scount,
  MPI_Datatype
  sdatatype,
  void *recvbuf, int rcount,
  MPI_Datatype rdatatype,
  int root,
  MPI_ comm comm)

Cluster Computing

Gathering Data



Cluster Computing

Reductions

- $D = D_0 \oplus D_1 \oplus D_2 \oplus D_n$
- Builtin reductions
  - MPI_MAX
  - MPI_MIN
  - MPI_SUM
  - MPI_PROD

Cluster Computing

MPI_Reduce

- int MPI_ Reduce(void *sndbuf,
  void *recvbuf,
  int count,
  MPI_ Datatype datatype,
  MPI_ Op op,
  int root,
  MPI_ comm comm)
- Similar with MPI_Allreduce

Cluster Computing

Scanning

- $D_i = D_0 \oplus D_1 \oplus D_2 \oplus D_{i-1}$
- Same builtins as with Reductions
- int MPI_ Scan (void *sndbuf,
  void *recvbuf,
  int count,
  MPI_ Datatype datatype,
  MPI_ Op op,
  MPI_ comm comm)

Cluster Computing

Mapping topologies to MPI

- MPI applications all have a topology
  - Graph
  - Cartesian
  - No topology

Cluster Computing

Mapping topologies to MPI

- MPI_Graphcreate
- MPI_Cartcreate
  - MPI_Cartrank
  - MPI_Cartcoords

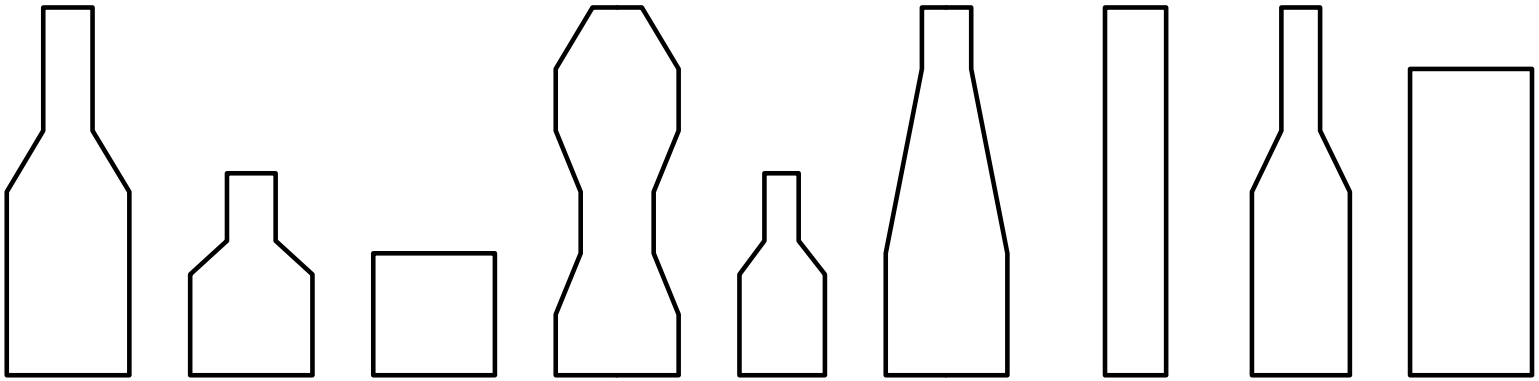
Cluster Computing

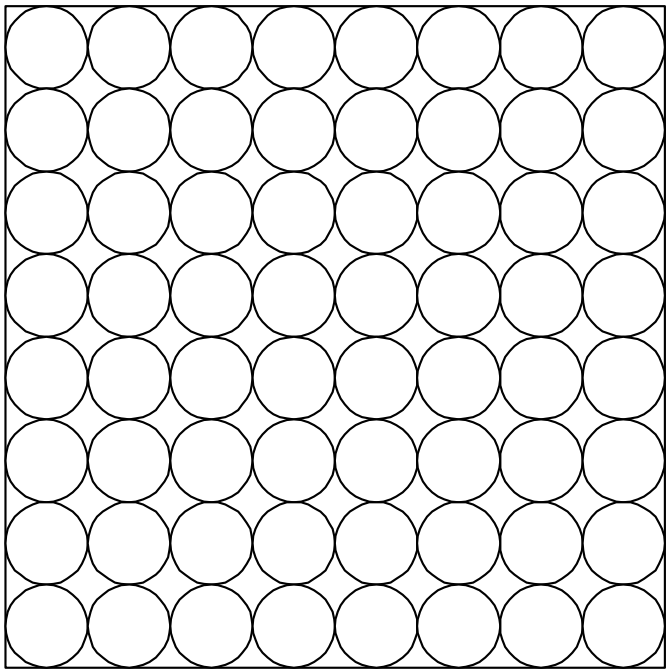
Example Neural Network training

- Training of large neural networks is very hard

Cluster Computing

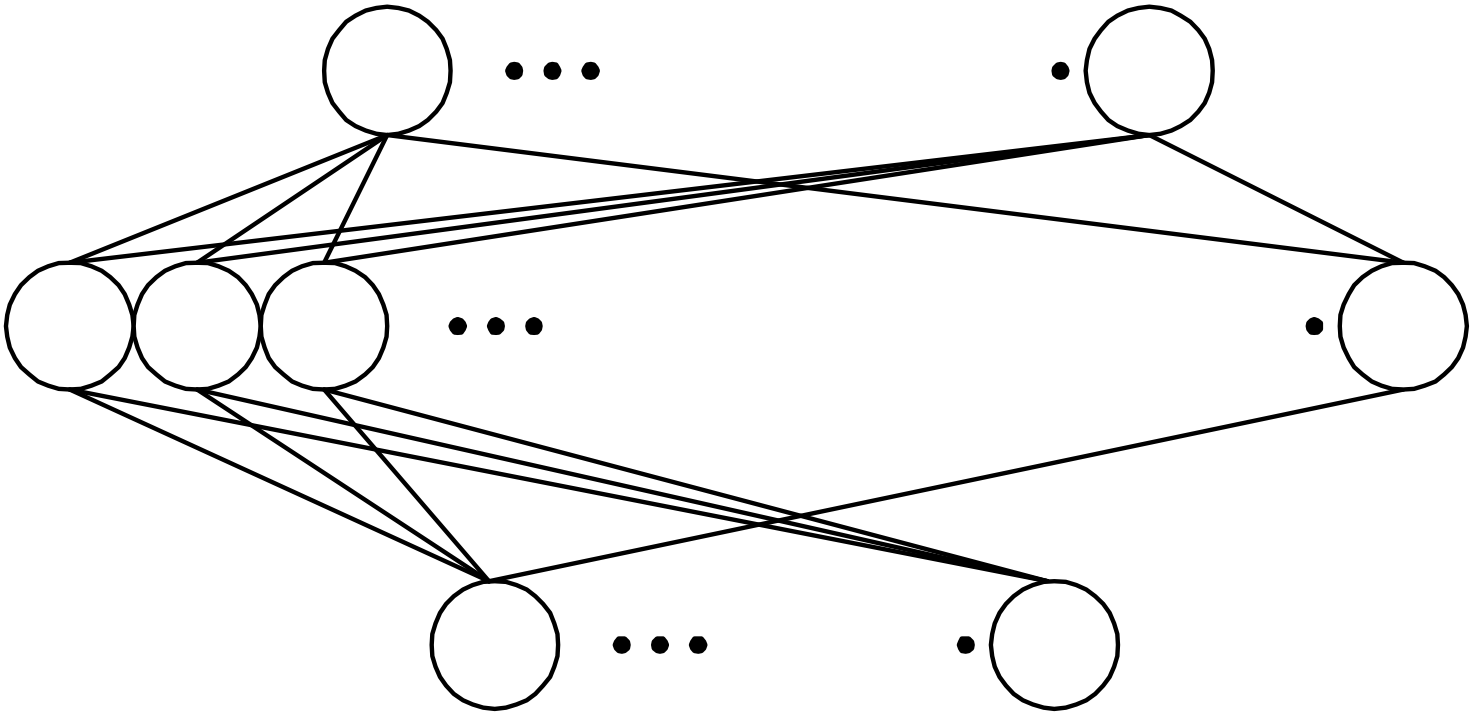
NN Training





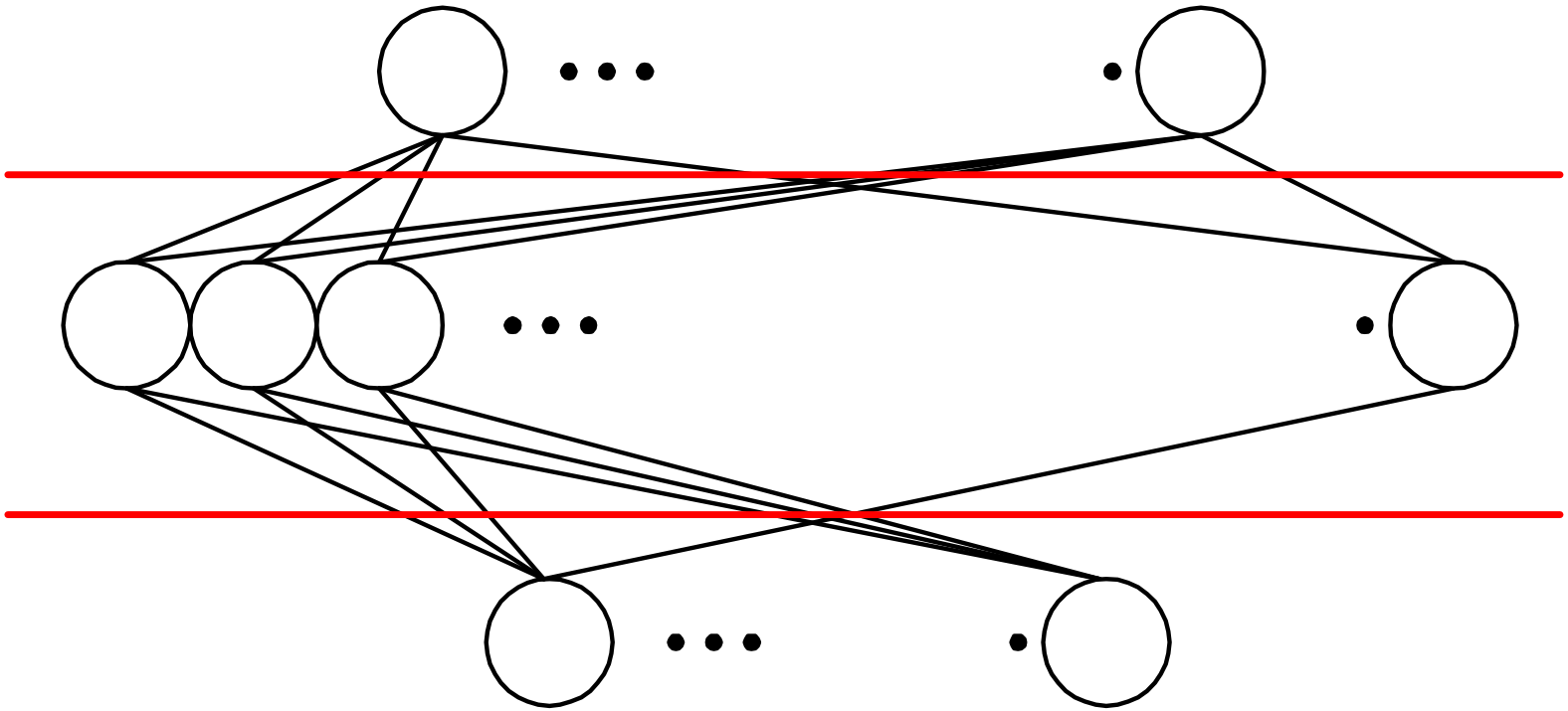
Cluster Computing

NN Training



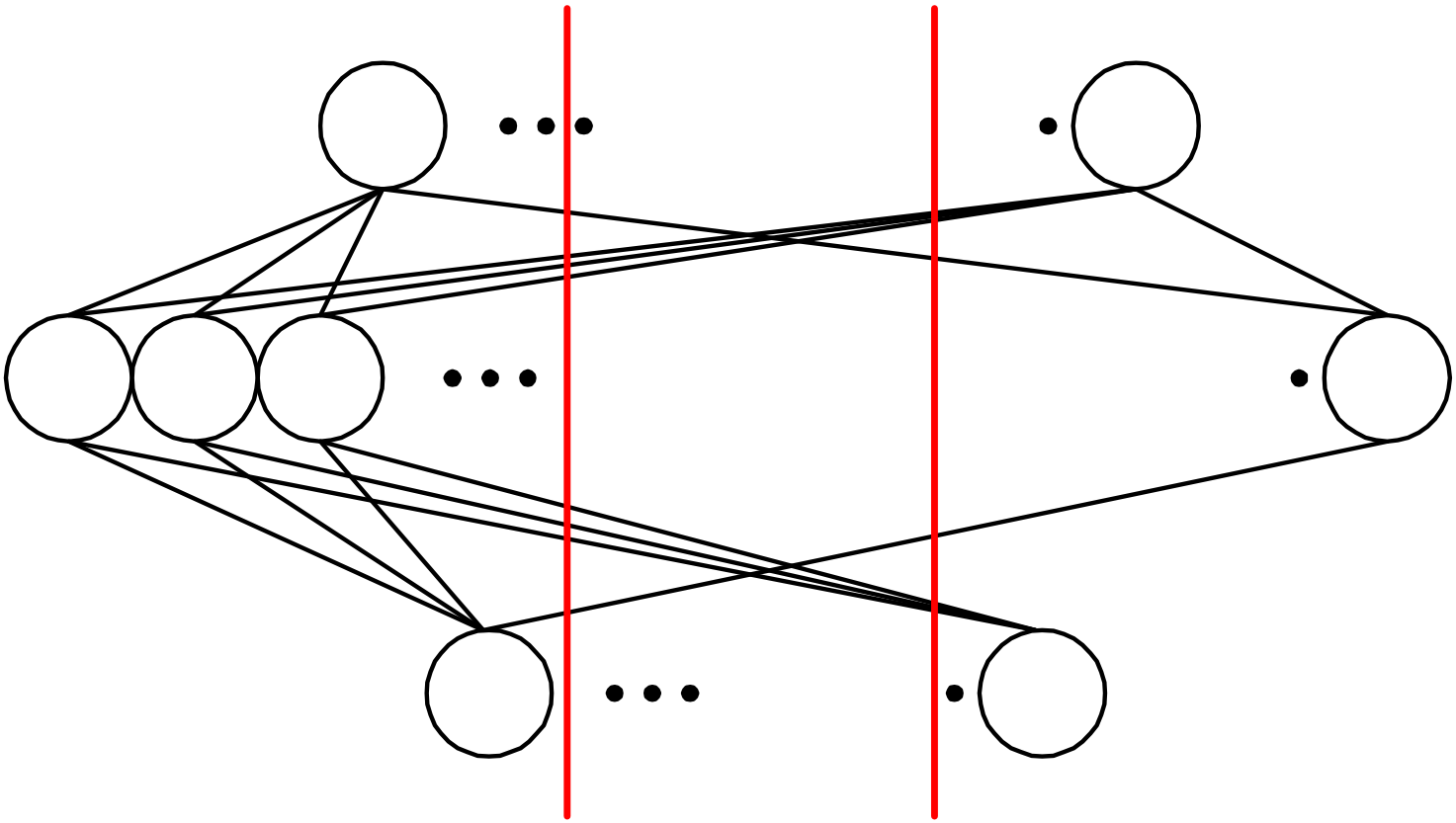
Cluster Computing

NN Training



Cluster Computing

NN Training



Cluster Computing

NN Training

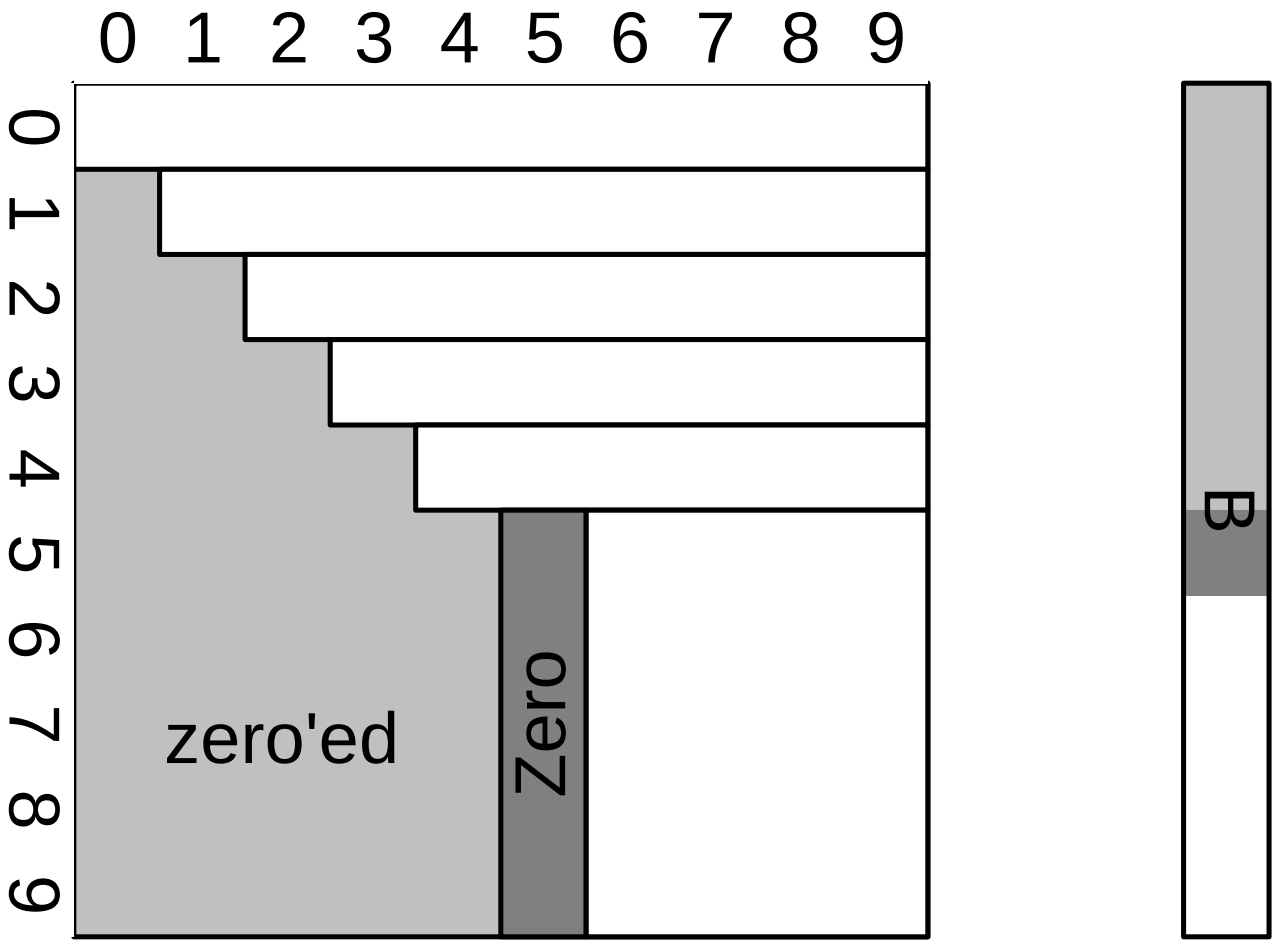
- Can be handled by an MPI_Alltoall after each itteration

Example Gaussian Elimination

- Solve *n* equations with *n* unknown
- Frequently used in scientific applications

Cluster Computing

Gaussian Elimination



Cluster Computing

Summary

- MPI is the definite industry leader in MPP programming
- Source code portability
- Loosely synchronous execution is very different from PVM