Cluster Computing

Weijie Zhao

03/05/2023

Message Passing Interface (MPI)

- Introduced in early 90’s
- Each process may have multiple threads
- Each process has its own address space
- Inter-process communication

MPI Example

```
#include <stdio.h>
#include <mpi.h>
int main(int argc, char *argv[])
{
    MPI_Init(&argc, &argv);
    printf(“hello world!\n");
    MPI_Finalize();
    return 0;
}
```

MPI Example

```
#include <stdio.h>
#include <mpi.h>
int main(int argc, char *argv[])
{
    int rank, size;
    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &size);
    printf(“hello world from %d of %d!\n", rank, size);
    MPI_Finalize();
    return 0;
}
```

MPI Communications

```
int MPI_Send(
    void* data,
    int count,
    MPI_Datatype datatype,
    int destination,
    int tag,
    MPI_Comm communicator)
int MPI_Recv(
    void* data,
    int count,
    MPI_Datatype datatype,
    int source,
    int tag,
    MPI_Comm communicator,
    MPI_Status* status)
```

MPI Communications

```
int MPI_Probe(
    int source,
    int tag,
    MPI_Comm comm,
    MPI_Status* status)
int MPI_Get_count(
    MPI_Status* status,
    MPI_Datatype datatype,
    int* count)
```

MPI Communications

```
int MPI_Isend(
        const void *buf,
        int count,
        MPI_Datatype datatype,
        int dest,
        int tag,
        MPI_Comm comm,
        MPI_Request *request)
```

MPI Communications

```
int MPI_Wait(
        MPI_Request *request,
        MPI_Status *status)

int MPI_Test(
        MPI_Request *request,
        int *flag,
        MPI_Status *status)
```

Communicator

```
int MPI_Comm_split(
        MPI_Comm comm,
        int color,
        int key,
        MPI_Comm * newcomm)

int MPI_Comm_free(MPI_Comm *comm)
```

Compilation and Execution

- MPICH, OpenMPI
- mpicc, mpiCC, mpic++
- mpiexec, mpirun
- mpiexec -np 4 ./a.out
- mpiCC a.cc --showme

- SLURM
  - sbatch
  - srun

MPI Configuration

- For each node, create a user that can ssh to all other nodes
- Install MPICH/OpenMPI
- mpirun -np 4 --hostfile myhost_file ./a.out
  - node1 slots=2 max_slots=10
  - node2 slots=2 max_slots=10

- mpirun -np 4 --hostfile myhost_file --byslot ./a.out
- mpirun -np 4 --hostfile myhost_file --bynode ./a.out

MPI Collective Communications

- MPI_Barrier(MPI_Comm communicator)
- MPI_Bcast(void* data, int count, MPI_Datatype datatype, int root,
  MPI_Comm communicator)
- MPI_Reduce(const void *sendbuf, void *recvbuf, int count,
  MPI_Datatype datatype, MPI_Op op, int root, MPI_Comm comm)
  - MPI_MIN, MPI_MAX, MPI_MINLOC, MPI_MAXLOC, MPI_BOR, MPI_BXOR, MPI_LOR, MPI_LXOR, MPI_BAND, MPI_LAND, MPI_SUM and MPI_PROD
- MPI_Allreduce(const void* send_buffer, void* receive_buffer, int count,
  MPI_Datatype datatype, MPI_Op operation,
  MPI_Comm communicator)

Cluster Computing

- MPI
  - Inter-node communication
  - High-performance computing
- Node failure
  - Broken hardware
  - Software bugs
  - Insufficient resources
- Node failure happens commonly for clusters with 1,000+ nodes
  - $(1 - p)^{1000}$

Cluster Computing

- MPI
  - Inter-node communication
  - High-performance computing
- Node failure
  - Broken hardware
  - Software bugs
  - Insufficient resources
- Node failure happens commonly for clusters with 1,000+ nodes
  - (1 - p)^1000

We need a system to handle these failures!

Distributed File System

- Decouple data and computing resources
- Replication to take care of node/disk failures
- HDFS
  - Name node
  - Data node

Common Data Analysis Tasks

- Given a large data, find some statistics
- Given a page view log, find the number of users
- Given a page view log, find the number of users group by browser
- Given a page view log, find the number of users from NY state group by browser

Map and Reduce

- PageRank
- PR(x) = \sum_{y links to (x)} (PR(y) / out_degree(y))

- Iterative disk I/O

Spark

- Spark
- Resilient Distributed Dataset (RDD)
  - Immutable
  - Transformations
    - map
    - filter
    - reduceByKey
    - join
    - …
  - Actions
    - count
    - collect
    - …