

IT-308 Operating Systems

Message Passing Interface

Project Report Final

Team #50

Instructor : Prof. Sanjay Chaudhary

Title:

Message Passing Interface:

We are planning to implement a basic search engine (keywords will be searched in already downloaded documents residing on local machine).

Team Members:

- Jayesh Hathila (201001045)
- Ayush Jain (201001036)
- Prateek Agrawal (201001046)
- Vidhan Agarwal (201001034)

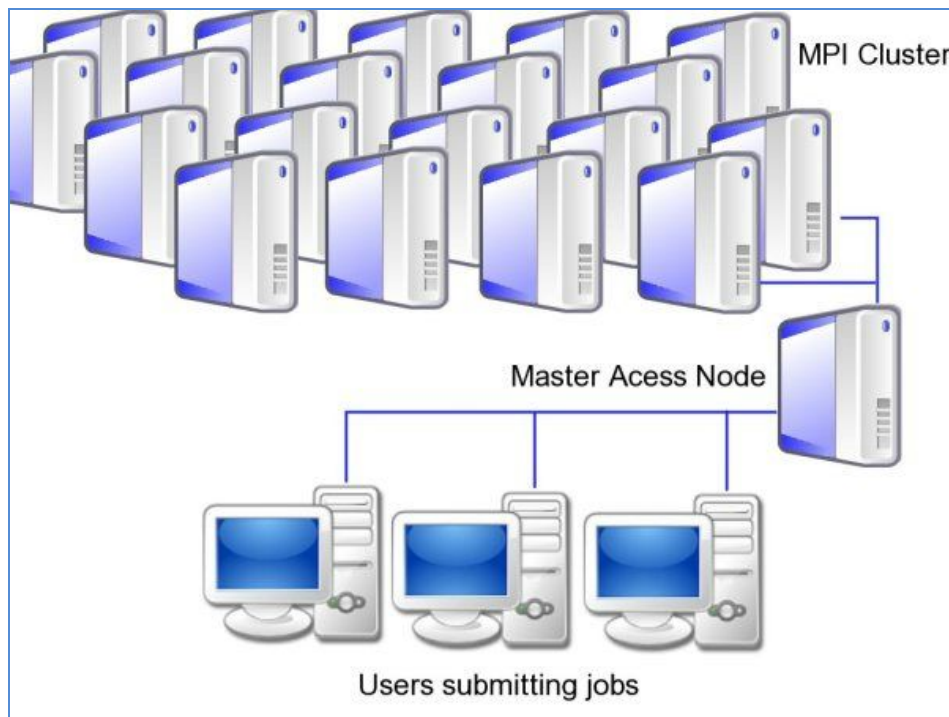
Brief Description:

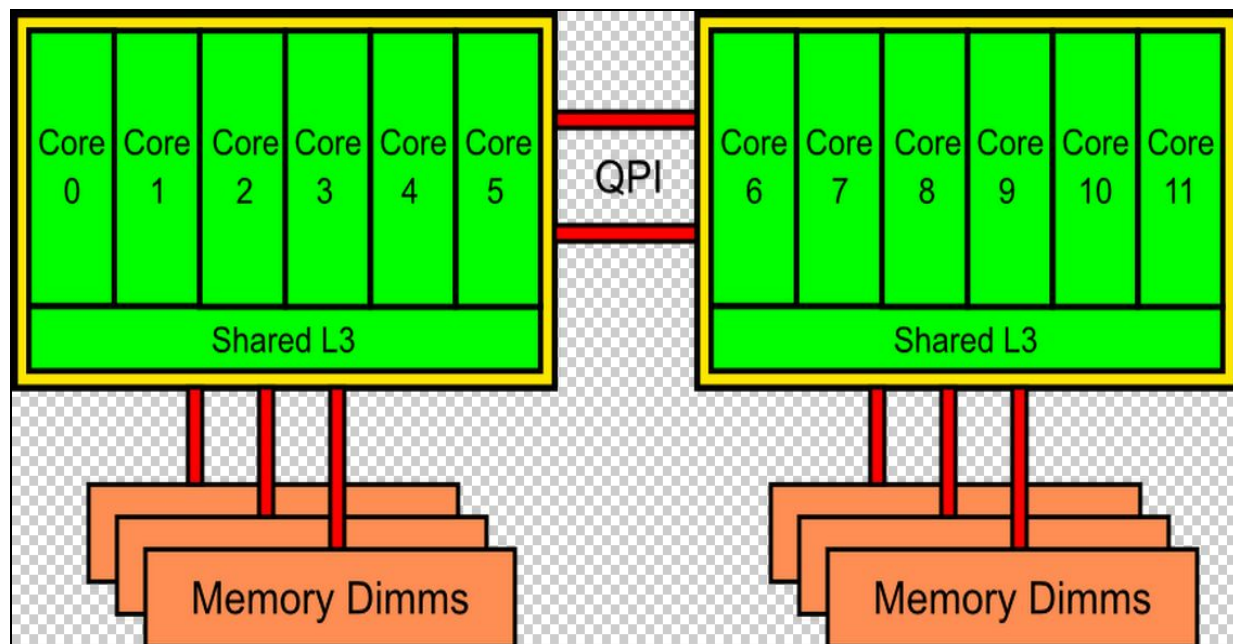
Message Passing Interface (MPI) is a standardized and portable message passing system designed by a group of researchers from academia and industry to function on a wide variety of parallel computers. As we are new to the topic, so we don't have in-depth knowledge pertaining to the application that our project could be based upon. Based on some readings and search by the team members, we are planning to implement parallel processing using shared memory(tentative).We will run program parallel on two cores (as a demo) to decrease the overall execution time which is now a day's implemented in most of the distributed computing systems. Possibly we will try to migrate

running process to save the current state and recover from data corruption on occurrence of some interrupt.

Finally, we have developed a module which can count the frequency of word or words based on inputs. Files are present in different areas and different processes are working in every area. So each process is allocated a specific number of files. Now the process works in every file count the frequency and return a value in buffer.

Architectural or model diagram for MPI





Modules to be developed:

As we will follow evolutionary incremental model of software development life cycle, and as per its definition we will make modules in below listed phase:

- 1) Simple code to pass message to ensure system configurations and working environment.
- 2) We will run a simple calculation using two processor cores utilizing concepts of parallel programming
- 3) We will implement a real time application which uses parallel programming in its true sense.

Technical Specifications:

We have used our code in following OS: Ubuntu 12.1.0 and Fedora17
Languages used are C and MPI libraries.

Following tools were used while development of the project:

MPICH1, MPICH2, GCC(for sub module compilation),
MPICC(for mpi implemented programs),
G++(as pre-requisite for mpicc)

Algorithms Used:

For sorting we have used Insertion Sorting.
Few parallel programming algorithms like MPI_GATHER, MPI_BCAST,
and MPI_SCATTER have been implemented.

Data Structures:

We have used linked list and double pointers.
Our code is working on a User level mode.

Implementation:

Task1: We plan to form clusters using Beowulf. We successfully connected the laptops locally but were fixed at making clustering through MPI, which was not running.

Task2: We are planning to implement a basic level of search engine in which input will be a string and output will be the relevant document with highest priority according to keyword matching.

We will first tokenize the string then we will assign a document and a keyword to search.

(This will happen in circular fashion.).Next increment will shed more light on implementation of the application

Task3: We finally have implemented the algorithm and it correctly counts the frequency of the letters and words. We have used MPI_BCAST and MPI_GATHER to successfully implement it.

We used following commands and technique to connect and transfer data:

ssh, scp, sftp.

Below written steps were followed to form MPI cluster

ON MACHINE 1:

log in as user_name

ssh-keygen

<enter key twice when prompted for a password>

cd .ssh

cp id_rsa.pub fubar

ON MACHINE 2:

log in as user name

ssh-keygen

<enter key twice when prompted for a passphrase>

cd .ssh

cp id_rsa.pub snafu

scp fubar:/home/user_name/.ssh/fubar .

cat fubar >> authorized_keys

ON MACHINE 1:

log in as user_name

cd .ssh

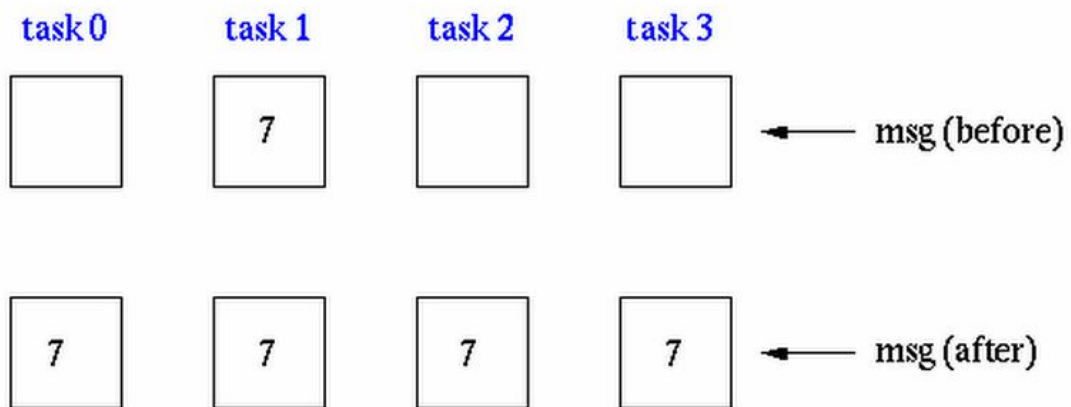
scp snafu:/home/user_name.ssh/snafu .

cat snafu >> authorized_keys

MPI_Bcast

Broadcasts a message to all other processes of that group

```
count = 1;  
source = 1;          broadcast originates in task 1  
MPI_Bcast(&msg, count, MPI_INT, source, MPI_COMM_WORLD);
```



`MPI_Gather (void *sendbuf,`
 `int sendcnt,`
 `MPI_Datatype sendtype,`
 `void *recvbuf,`
 `int recvcnt,`
 `MPI_Datatype recvtype,`
 `int root,`
 `MPI_Comm comm)`

Annotations:

- Address of variable to send
- Number of elements send to root
- Variable Data Type
- Significant only at root
- Address of variable to receive
- Significant only at root
- Number of elements from each
- Significant only at root
- Variable Data Type
- root, where the data is gathered to
- MPI_COMM_WORLD

Test Results:

We will be testing our program by various input output cases and by running it in different Linux versions.

Test Case 1: Brief: Input/ Output

Input Set: We give different file names and keep changing the string whose frequency is to be found.

Observations: The total frequency of the words in each file is calculated.

Test Case 2: Brief: running in different versions of Linux

Input Set: Ran the program in different versions of Linux: Ubuntu 12.1, Ubuntu 10.04 and fedora 17.

Observations: Successful correct outputs were given in each. It implies that various modules are compatible in different versions of OS.

Snapshots:

```
File Edit View Terminal Help
jayesh@jayesh:~/mpi_stuff$ mpicc new.c
jayesh@jayesh:~/mpi_stuff$ mpirun -np 4 ./a.out
filename a corresponding frequency 0
filename test1.txt corresponding frequency 0
filename tst.c corresponding frequency 1
filename b corresponding frequency 0
filename test.c corresponding frequency 1
filename wap.txt corresponding frequency 3
filename test corresponding frequency 0
filename test.txt corresponding frequency 0
filename wordlist.txt corresponding frequency 5
jayesh@jayesh:~/mpi_stuff$
```

```

jayesh@jayesh: ~/mpi_stuff
File Edit View Terminal Help
jayesh@jayesh:~/mpi_stuff$ ssh-keygen
Bad escape character 'ygen'.
jayesh@jayesh:~/mpi_stuff$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/jayesh/.ssh/id_rsa):
Created directory '/home/jayesh/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/jayesh/.ssh/id_rsa.
Your public key has been saved in /home/jayesh/.ssh/id_rsa.pub.
The key fingerprint is:
26:b1:f8:5b:45:ac:48:52:46:7b:4a:3f:03:cc:15:53 jayesh@jayesh
The key's randomart image is:
+--[ RSA 2048 ]-----+
|      .+ +oE      |
|      = o o      |
|      . 0 . o      |
|      = 0 o      |
|      . = S .      |
|      . o +      |
|      . .      |
|      o      |
|      .      |
+-----+
jayesh@jayesh:~/mpi_stuff$

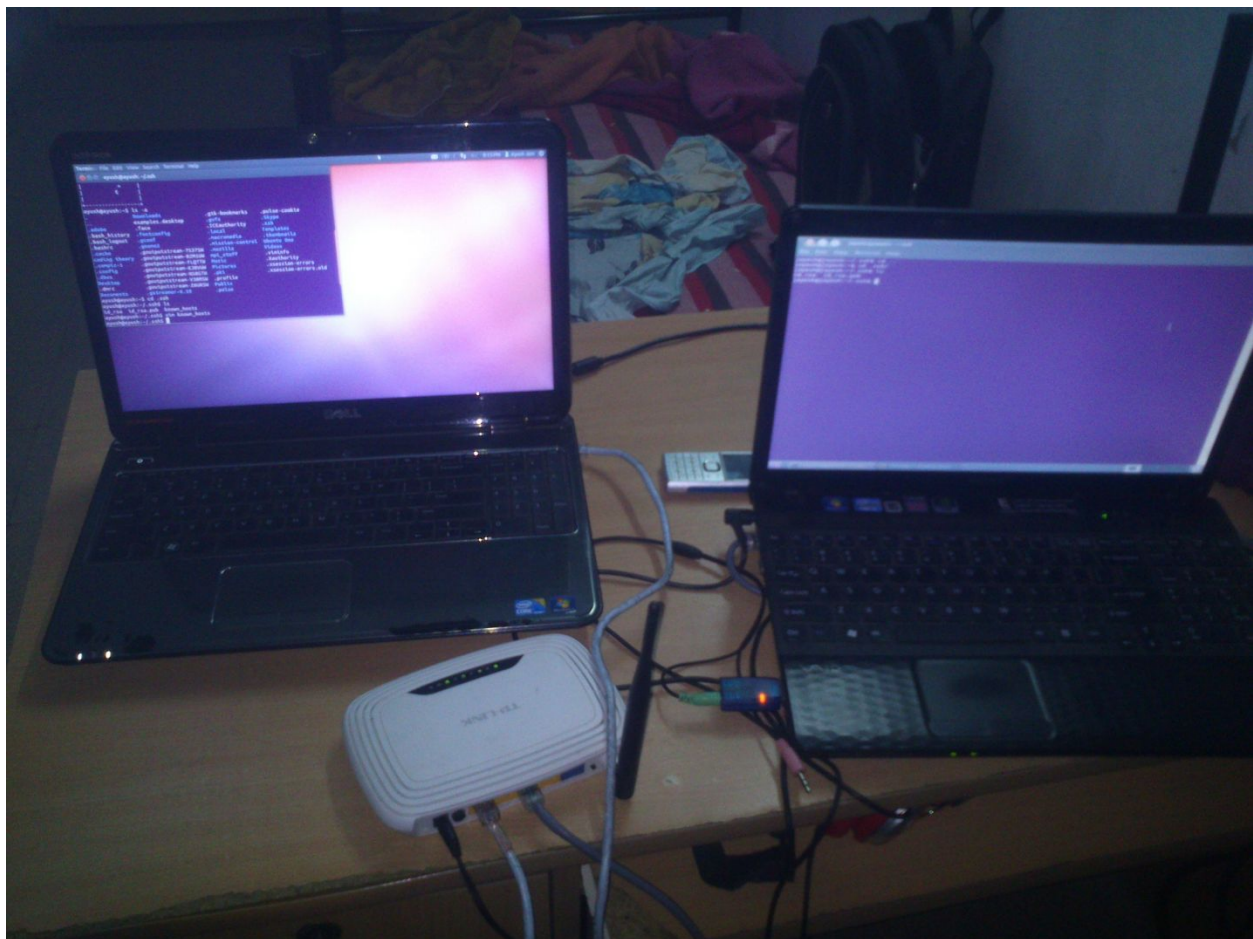
```

```

drwx----- 2 jayesh jayesh 4096 2013-04-14 11:40 .ssh
-rw-r--r-- 1 jayesh jayesh 0 2013-03-11 04:15 .sudo_as_admin_successful
drwx----- 2 jayesh jayesh 4096 2013-03-11 04:55 .synaptic

```

```
File Edit View Terminal Help
jayesh@jayesh:~/mpi_stuff$ ls
a      b      hello.c  os_project  pq.c  q      test1.txt  test.txt  wap.txt
a.out  first.c  new.c    p.c        pqw.c  test  test.c    tst.c     wordlist.txt
```



Member Wise Work Distribution:

Everyone was involved in coding of cluster formation and reading various literature of MPI provided by Sir.

Implementation of frequency Searching and message passing feature has been done by Jayesh Hathila and Ayush Jain.

Report and proposal prepared by Prateek Agrawal and Vidhan Agarwal.

Testing of various functions in different OS and of various I/O cases has been done by Prateek Agrawal.

References:

Using MPI – by William Gropp, Ewing Lusk

<http://www.tc.cornell.edu/Services/Education/Topics/MPI/Basics.bak/5.2+mpirun+mpiexec.htm>

<http://www-unix.mcs.anl.gov/mpi/mpich1/mpich-nt/>

<http://www.lam-mpi.org/tutorials/one-step/lam.php>

<http://www.linuxquestions.org/questions/linux-newbie-8/problem-getting-started-with-kernel-thread-4175447493/>

<http://www.linuxquestions.org/questions/linux-virtualization-90/problem-forming-clustur-of-laptops-via-router-mpi-4175451386/>

<http://www.linuxquestions.org/questions/linux-networking-3/ssh-hostname-mpi-clustering-4175451608/>

<http://stackoverflow.com/questions/13209686/openmpwhy-am-i-not-getting-different-thread-ids-when-i-uses-pragma-omp-parallel>

<http://stackoverflow.com/questions/15009855/connecting-2-laptops-for-mpi>

<http://www.lam-mpi.org/MailArchives/lam/2013/03/14339.php>