Implementation of Parallel Processing Environment using PVM

Presented by

- Manabendra Das (012850G)
- Soumik Sikdar (012850D)
- Asim Kumar Roy (012850F)

Under the supervision of

 Mr. Utpal Roy

Objective of Project

- Setting up a parallel processing environment on a LAN
- Use of Message Passing Model (implementing Parallel Virtual Machine)
- Testing the environment by executing a simple parallel program

What is Parallel Processing ?

- A form of computing in which a number of activities are carried out concurrently, generally on multiple processors, so that the effective time required to solve a problem is reduced.
- Communication and synchronization of the processes forms the core of parallel programming issues.

Why Parallel Processing ?

- Save time
- Solve larger problems
- Compute resources available on LAN, WAN, even Internet
- Cost saving
- Overcoming memory constraints

Different Parallel Programming Models

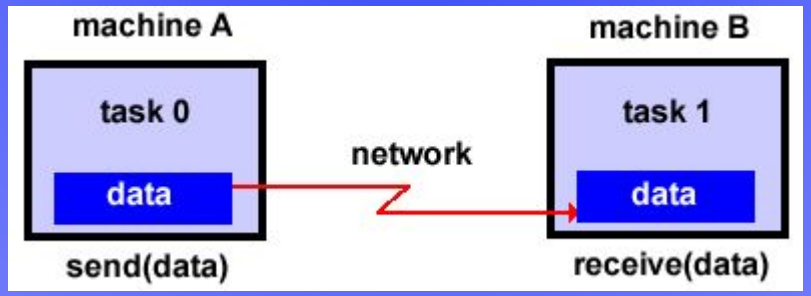
- Shared Memory Model
- Message Passing Model
- Hybrid Model

Shared Memory Model

- Tasks share a common address space, which they read and write asynchronously
- Locks and semaphores may be used to control access to the shared memory
- There is no need to specify explicitly the communication of data between tasks
- It becomes more difficult to understand and manage data locality

Message Passing Model

- A set of tasks that use their own local memory during computation
- Tasks exchange data through communications by sending and receiving messages



Hybrid Model

- Any two or more parallel programming models are combined
- Currently, a common example of a hybrid model is the combination of the message passing model (MPI) with the shared memory model (OpenMP)

What is PVM ?

- A software application that enables you to turn TCP/IP networked computers into a single virtual machine in order to run parallel programming
- PVM system is the message-passing model
- It makes a collection of computers appear as one large *virtual* machine

Why PVM ?

- Ability to establish a parallel machine using ordinary (low cost) workstations
- Ability to connect heterogeneous Machines
- Use of simple C/Fortran functions
- Takes care of data conversion and low-level communication issues
- Easily installable and configurable

But...

- Programmer has to explicitly implement all parallelization details
- Difficult debugging

PVM principles

- Virtual Machine
- Tasks
- Message Passing
- Heterogeneity
- Process Management
- Dynamic Process Groups
- Multiprocessor Support

How to start PVM ?

- **From the PVM directory:**

  $ pvm <*hostfile*>

  <*hostfile*>: a file that contains host names

- **In PVM console:**

  pvm> conf

  shows configuration (active host names)

  pvm> quit

  quits PVM console but PVM still runs

  pvm> halt

  terminates PVM