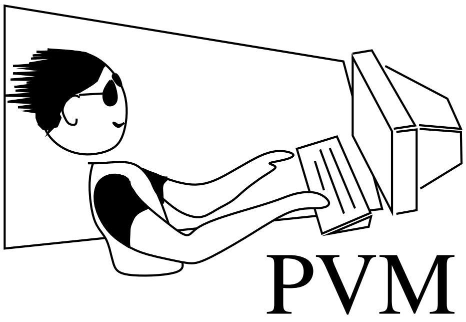
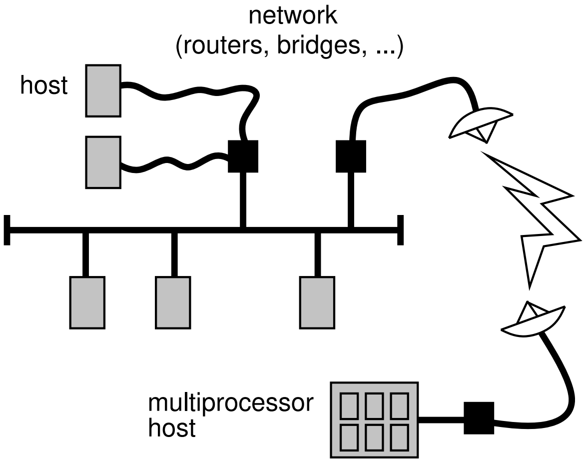
The PVM Project

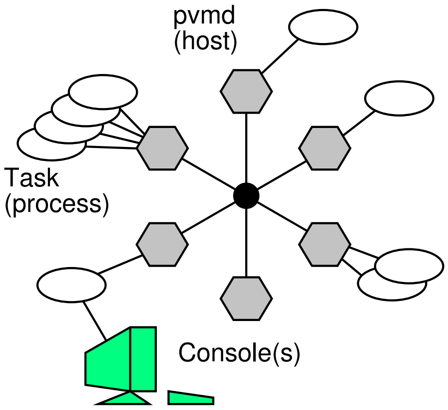


Description
PVM (Parallel Virtual Machine) is a software system that enables a collection of heterogeneous computers to be used as a coherent and flexible concurrent computational resource. The individual computers may be shared- or local-memory multiprocessors, vector supercomputers, specialized graphics engines, or scalar workstations, that may be interconnected by a variety of networks, such as ethernet, FDDI, etc. PVM support software executes on each machine in a user-configurable pool, and presents a unified, general, and powerful computational environment for concurrent applications. User programs written in C, C++ or Fortran, are provided access to PVM through the use of calls to PVM library routines for functions such as process initiation, message transmission and reception, and synchronization via barriers or rendezvous. Users may optionally control the execution location of specific application components; the PVM system transparently handles message routing, data conversion for incompatible architectures, and other tasks that are necessary for operation in a heterogeneous, network environment.

Applications and Environments
PVM is ideally suited for concurrent applications composed of many interrelated parts. PVM is particularly effective for heterogeneous applications that exploit specific strengths of individual machines on a network. As a loosely coupled concurrent supercomputer environment PVM is a viable scientific computing platform. PVM system has been used for a number of applications such as molecular dynamics simulations, superconductivity studies, distributed fractal computations, matrix algorithms, and in the classroom as the basis for teaching concurrent computing.



PVM – Physical View



PVM – Logical View

The PVM Project

Major Features of PVM

- Portable - Runs on nearly every Unix machine, plus many shared- and distributed-memory multiprocessors.
- Heterogeneous - Any types of machines can be combined in a single virtual machine
- Scalable - Virtual machines can include hundreds of host computers, and run thousands of tasks.
- Dynamic configuration - computers can be dynamically added and deleted from the parallel virtual machine by the application or manually.
- Hooks for fault tolerance - Application can be notified via messages of lost processes, processors or addition of new resources.
- Dynamic process groups - user-defined process grouping for functions like broadcast and barrier. Groups can overlap and change dynamically during an application.
- Signals - PVM tasks can send signals to other tasks.
- Multiple message buffers - Allows easier development of PVM math libraries, graphical interfaces, etc.
- Tracing - Call-level tracing built into PVM library.
- Can be customized - User can write manager tasks to implement custom scheduling policies.

PVM Runs On		
Workstations	Operating Systems	Supercomputers
DEC	BSD	Convex C2, Exemplar
IBM	System-5	Cray C90, T-3D
HP	Linux	Intel Paragon, Hypercube
Silicon Graphics		IBM SP-2, 3090
Sun		TMC CM-2, CM-5
NeXT		Kendall Square
Data General		Sequent
386/486/586 Unix boxes		Maspar
		Encore
		Fujitsu
		Stardent
... and others		

Status and Availability
The current release (PVM 3.3, patch level 10) is now available and may be obtained by sending electronic mail to *netlib@ornl.gov* with the message *send index from pvm3.*

Contacts and Support
PVM is an ongoing research project, as such support at levels appropriate for a research project may be reasonably expected. Several machine vendors offer fully supported versions of PVM.

Questions and comments can be sent to *pvm@msr.epm.ornl.gov.* The PVM WWW home page can be found at *http://www.epm.ornl.gov/pvm/pvm_home.html.* There is an active Usenet newsgroup, *comp.parallel.pvm*, for discussion on PVM and help with problems.

This project is supported in part by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy, by the National Science Foundation Science and Technology Center (CRPC), and by the Science Alliance, a state supported program at the University of Tennessee.