The PVM Project

A. Beguelin, J.J. Dongarra, G.A. Geist, W. Jiang, R. Manchek, K. Moore, and V.S. Sunderam

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 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.

Status and Availability

The beta release of PVM 3.0 is now available and may be obtained by sending electronic mail to *netlib@ornl.gov* with the message *send index from pvm3.* There are a number of improvements planned for the beta version. A notable one is performance; we expect to improve the performance over the course of the next few months as we tune and adjust the internals. The beta release contains a users guide and installation notes; PVM has been tested on a wide variety of systems.

Some of the new features of PVM include:

- Runs on nodes of multiprocessors - initially Intel iPSC/860 other machines like CM-5, Paragon to be added later.
- Dynamic configuration - computers can be dynamically added and deleted from the parallel virtual machine by the application or manually.
- Multiple message buffers - allowing easier development of PVM math libraries, graphical interfaces, etc.
- Signals - PVM tasks can send signals or notify sets of tasks about events.
- Dynamic process groups - user-defined process grouping for functions like broadcast and barrier. groups can overlap and change dynamically during an application.
- Cleaner naming convention for PVM routines - C routines start with pvm_ Fortran routines start with pvmf. For example pvm_send.
- Improved routines - receive by source or type (or user defined context) spawn multiple tasks with options for argv and debug, pack and unpack messages using stride, and more query functions.
- New more comprehensive User's Guide and PVM 3.0 Reference Manual.

PVM is an ongoing research project, as such support at levels appropriate for a research project may be reasonably expected. Questions and comments can be sent to *pvm@msr.epm.ornl.gov.*

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.