Table of Contents Author Index Home

Get Acrobat Reader Get Postscript Version Get PDF Version

# Performance and Portability of the Message Passing Toolkit on CRAY J90 and CRAY T3E Systems

Andrea Hudson and Alex Wang
NASA Goddard Space Flight Center and Hughes STX

## 1. Background

For 7 years, the computing center at the NASA Goddard Space Flight Center which supports the Earth and space scientists has been primarily a CRAY parallel vector environment (PVP). In 1989, the center procured a CRAY Y-MP with 4 processors and 64 MW of memory. This system was upgraded to a CRAY C98 with 6 processors and 256 MW of memory in 1993. By this time, most of the large production codes were well vectorized and in a few cases, multitasked. Multitasking was beneficial for those jobs that were highly parallel (greater than 95 percent).

By 1995 there was a need to provide greater total throughput with fewer dollars. Consequently, Goddard and Cray Research, Inc. agreed to replace the CRAY C98 with three CRAY J90 systems. Since individual J90 processors are about 4-5 times slower than a CRAY C98 processor, the change in hardware highlighted a need to have more of the production applications multitasked. It was imperative that some programs be multitasked to the greatest extent possible, because some projects had specific turn-around requirements.

To address this need, a team of staff members from the computing center was formed to work with the different members of the science community. Representatives from each of the laboratories defined the applications that they believed to be most critical to their work and those were the applications that the teams worked to multitask and/or optimize for the parallel vector environment.

As emphasis on multitasking grew, it was clear that scaling beyond 12-16 processors on the CRAY J90 system was unlikely, even on those applications with high levels of parallelism. Though the ability to multitask was a short-term solution to providing sufficient job turn-around, it was clear that over the long term the only way to be able to guarantee adequate throughput for higher resolution and more complex applications was an explicit form of parallelism. This leads us to message passing development.

Currently, there are two possible choices for developing portable, message passing code: Parallel Virtual Machine (PVM) and the Message Passing Interface (MPI). Most groups are leaning towards MPI development since it appears to be the defacto standard for parallel programming via message passing. The CRAY specific message passing, or data passing, library is SHMEM. For applications running on a CRAY platform (T3E) performance will typically be best if SHMEM is used.

All of these elements, and the fact that CRAY provides the MPT (Message Passing Toolkit), present a terrific opportunity to explore the development and portability of one of the message passing paradigms on the CRAY J90s and/or T3E. In April of this year, a CRAY T3E was made available at Goddard through a cooperative agreement between CRAY Research and NASA. Earth and space science grand challenge problems, along with other NASA-funded earth and space science projects, are allocated time on this system to explore parallel development.

## 2. Hardware

As previously mentioned, the current hardware configuration for Earth and space science production computing at Goddard consists of three CRAY J90 systems. At this time, one system has 32 scalar enhanced (SE) processors. Another of the CRAY J90 systems has 32 (classic) processors and the other system has 16 (classic) processors. The two 32-processor systems are NFS mounted with one of the systems being delineated for interactive work. Both systems accomodate batch work with most of the large multitasking work delegated to the non-interactive system. The 16-processor system is dedicated to time critical work for one of the larger user community groups.

Collectively, the three CRAY J90s have a total of 2 GW of memory. The system running most of the large memory, multitasking work has 1 GW and the other 2 systems have 512 MW each.

For data storage, the science community has access to the UniTree mass data storage system which runs on a CONVEX platform. This system is connected via HiPPI, FDDI and Ethernet to the CRAY platforms. The two 32 processor CRAY systems run the Cray Data Migration Facility (DMF) software for medium-term (less than 35 days) storage. There is approximately a 50-terabyte capacity in the mass data storage system. The computing center operates one of the most active data archival facilities in the country with occasionally over 100 gigabytes stored and over 100 gigabytes retrieved in a day.

To accomodate research and development into new computational techniques, a T3E system is available to grand challenge investigators to the Earth and space science high-performance computing project, now known as the National Coordination Office for Computing, Information and Communications. In addition NASA-funded Earth and space scientists can submit proposals for time on the system. Approximately 20 percent of the time on the T3E, scalable testbed, is to be available to this broader community of NASA funded earth and space scientists.

## 3. Methodology for Testing MPT

To possess detailed knowledge of both the portability and the performance issues of each of the components of the MPT, a worthwhile effort has been to instrument example programs with some basic communication primitives. We implemented point-to-point, half-to-half and broadcast tests in each of the MPT protocols: MPI, PVM and SHMEM.

It is critical to note that the testing of the MPT on the CRAY J90 (PVP) system was done in a "shared memory" mode. Essentially, this means that the MPT libraries emulate shared memory (memory to memory) communication. There is an option to do communication over TCP sockets. Since the testing described in this paper was designed to determine the feasibility of using MPT for message passing development, a decision was made to use the most optimal form of communication available.

Specifically, for PVM code, the shared memory implementation was run in stand-alone mode, which means that no PVM daemon was required. This is typically the best mode of operation for a single PVM code executing within a single PVP system. For PVM and SHMEM programs to run in shared memory mode it is necessary to convert global and static data to TASKCOMMON. CRAY has provided a feature which allows you to do this conversion at compile time.

For MPI under MPT on the CRAY J90 (PVP) systems it is also necessary to convert global and static data to TASKCOMMON. To run, it is necessary to use the mpirun command with option -nt instead of -np. The -nt option will specify the number of tasks for the shared memory implementation. This will provide multiple tasks within a single UNIX process. If -np will run on a specified number of processors and communication will be conducted with TCP sockets.

In each case, the size of the messages were allowed to vary. In addition, data on the transfer rates was collected so that each of the message passing protocols could be compared.

Implementation of the PVM and MPI example programs contain many similar elements since successful communication requires the sending and receiving processor to "handshake." In other words, each processor involved in the communication makes a call to a send or receive routine.

There are three primitive examples which were explored: point-to-point, half-to-half and broadcast. The point-to-point example program demonstrates basic communication between two processors. One processor sends a message to the other processor and this processor receives the message. The half-to-half example program has half of the processors in the group send messages to the other half, which in turn receive the messages and the receiving processors send back the messages to the original processors. The broadcast example has one processor send out a message to all of the other processors in the group, and they receive the message.

Of note is the UNICOS environment in which we developed and tested the primitive communication routines. There were no changes to default environment variables, outside of varying the number of processors for different runs. For example, in each compilation, the default version of the f90 compiler was invoked. By switching to a different/newer version under modules, it may be true that a slightly different executable would be produced. The mode in which the executable codes were produced and run emulated the mode typically used by scientific computing technical teams when developing an application. The version of the Message Passing Toolkit used on each system was MPT 1.1.0.0.

## 4. Analysis of Portability

A few problems were encountered with respect to portability across the systems. For the MPI examples, point-to-point and half-to-half, the MPI_ANY_TAG value that is set in the include file mpif.h on the T3E had to be hard-coded to another value. When this was completed, the programs compiled and ran without problem.

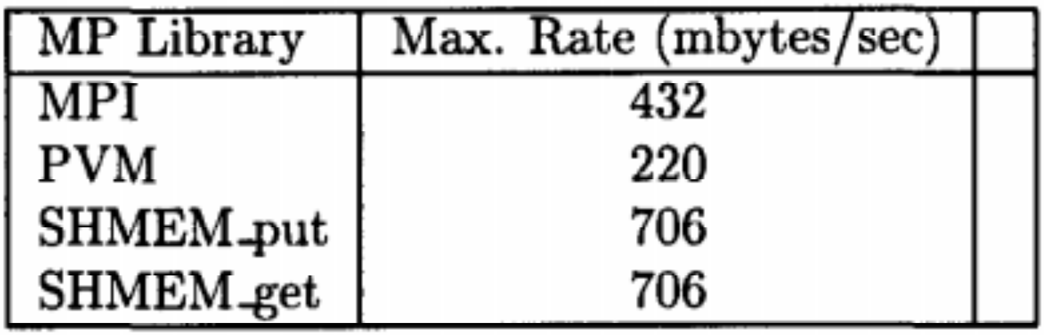
One outstanding issue has been that some of the MPI runs on the J90s fail in a non-deterministic fashion. The same applications have shown no problem on the T3E. Cray Research, Inc. states in their *MPI Programmer's Manual* that it is not completely efficient to preserve MPT behavior in a shared memory/multitasked environment. The MPT implementation on PVP systems may occasionally have unexpected behavior. These cases need to be documented as much as possible and, if feasible, corrected.

## 5. Analysis of Performance

In this section, performance of the individual MPT libraries on each system, J90 and T3E, will be presented. Different modes of communication were used on each of the systems. Therefore, the performance of the J90 and T3E examples should not be compared because it is not an "apples to apples" comparison. The performance numbers presented here are provided for analysis of individual message passing libraries on the respective systems.

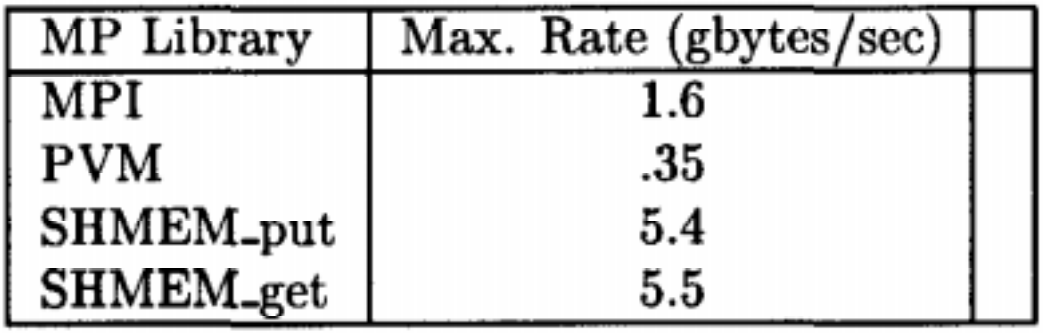
All of the performance numbers presented were gathered in dedicated runs.

Table 1. Comparison of point-to-point transfers using the MPT libraries in a shared memory environment



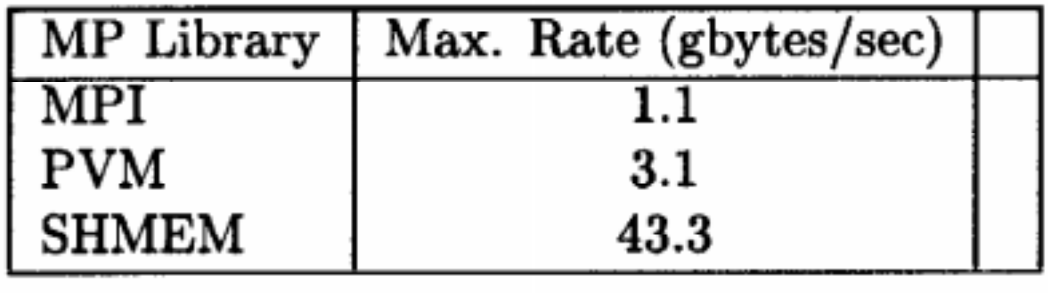
| MP Library | Max. Rate (mbytes/sec) | |
|---|---|---|
| MPI | 432 | |
| PVM | 220 | |
| SHMEM_put | 706 | |
| SHMEM_get | 706 | |

Table 2. Comparison of half-to-half transfers between 8 processes using the MPT libraries in a shared memory environment



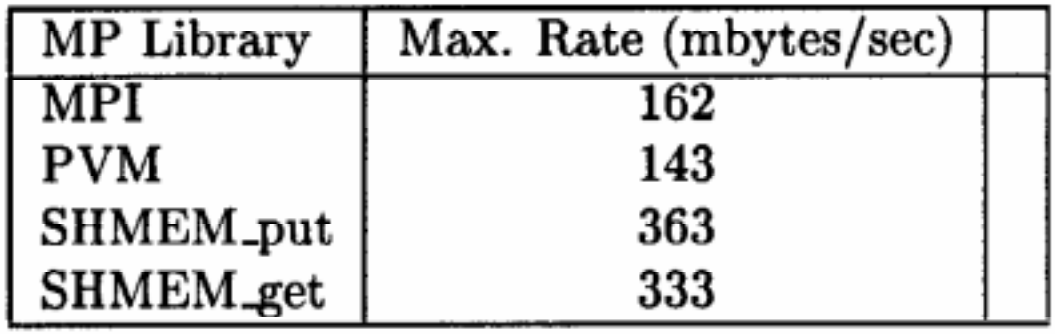
| MP Library | Max. Rate (gbytes/sec) | |
|---|---|---|
| MPI | 1.6 | |
| PVM | .35 | |
| SHMEM_put | 5.4 | |
| SHMEM_get | 5.5 | |

Table 3. Comparison of broadcast transfers to 8 processes using the MPT libraries in a shared memory environment



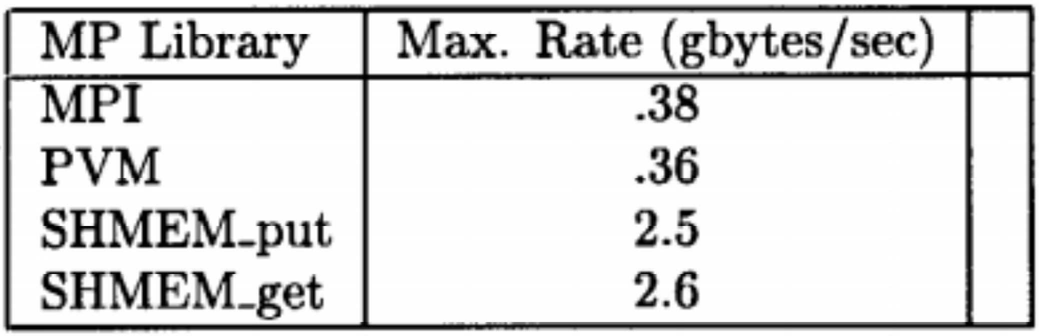
| MP Library | Max. Rate (gbytes/sec) | |
|---|---|---|
| MPI | 1.1 | |
| PVM | 3.1 | |
| SHMEM | 43.3 | |

Table 4. Comparison of point-to-point transfers using the MPT libraries in a T3E environment



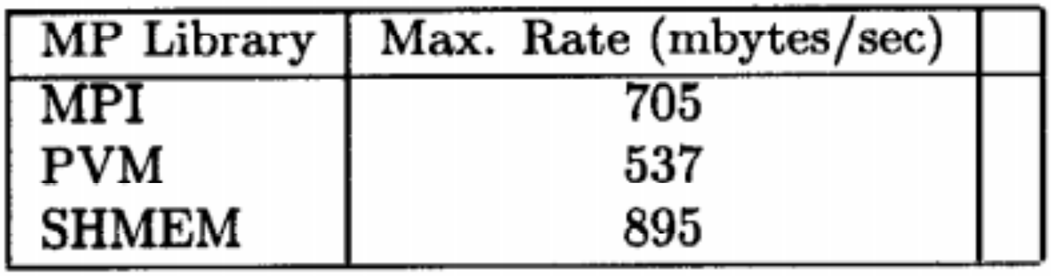
| MP Library | Max. Rate (mbytes/sec) | |
|---|---|---|
| MPI | 162 | |
| PVM | 143 | |
| SHMEM_put | 363 | |
| SHMEM_get | 333 | |

Table 5. Comparison of half-to-half transfers between 8 processes using the MPT libraries in a T3E environment



| MP Library | Max. Rate (gbytes/sec) | |
|---|---|---|
| MPI | .38 | |
| PVM | .36 | |
| SHMEM_put | 2.5 | |
| SHMEM_get | 2.6 | |

Table 6. Comparison of broadcast transfers to 8 processes using the MPT libraries in a T3E environment



| MP Library | Max. Rate (mbytes/sec) | |
|---|---|---|
| MPI | 705 | |
| PVM | 537 | |
| SHMEM | 895 | |

## 6.Conclusion

In summary, having the capability of designing message passing code on a CRAY J90 (PVP) system provides an additional platform for that development. This can be useful to offset the work of one or the other system, or when either system may be temporarily unavailable for system maintenance or system crashes. However, this is only of real value if the software issues are well understood by the developer. If message passing development on a CRAY PVP system via MPI or PVM is pursued, it is recommended that each mode of development, shared memory and socket communication, is understood.

If non-deterministic run-time problems occur with MPI or PVM programs on the CRAY PVP system, and that is not the platform which will ultimately be used to run the executable, then it may not be worthwhile to spend time debugging in this environment. A recommendation would be to follow up the issue with CRAY and verify the development on the CRAY T3E.

If portability between J90 and T3E systems is paramount, then it may be essential to run using TCP communications on the CRAY J90 systems if PVM or MPI is being used. This will likely provide much greater portability across systems, but communications performance will be sacrificed on the CRAY J90 systems.

Table of Contents | Author Index | CUG Home Page | Home