Migrating Users to the CRAY T3E from the CRAY T3D and Intel Paragon

Jay Boisseau, Ken Steube, and Max Pazirandeh, San Diego Supercomputer Center
San Diego, CA USA

ABSTRACT: SDSC's CRAY T3D and Intel Paragon users were granted access to SDSC's CRAY T3E in January '97. In this paper we describe the issues involved in migrating all users from the T3D and Paragon to the T3E in just a few months. In particular we discuss techniques for porting T3D-specific (e.g. utilizing T3D features still not implemented or Cray PVP front-end features) and Paragon-specific (e.g. utilizing the NX message passing library) user applications and optimizing applications for the T3E. The discussion is sufficiently general to also be of value for T3E sites not migrating users from other platforms.

Introduction

Many high performance computing centers purchasing CRAY T3E systems will have to address the issues involved in migrating users from other platforms. SDSC recently faced these issues when we migrated users from our Intel Paragon and CRAY T3D systems. The issues in porting from the T3D to the T3E are usually subtle ones due to the change from **cf77** to **f90** and from a hosted system to a self-hosted system. On the other hand, porting from an Intel Paragon to the T3E can be a much larger task. There are many potential problems that arise in porting codes in either case; our goal is to provide some assistance to those who will help others migrate to T3E systems.

SDSC's MPP Systems

SDSC received its T3E in November 1996. After system testing and acceptance, we opened it to friendly-users for a brief period to prepare the system for production-style use. The system was available to all users on March 3, 1997. The T3D was decommissioned two weeks later when users had some time to port their codes to the T3E. Our Intel Paragon was removed from service to the general public two weeks after that. Thus, in the span of 4 weeks, all MPP users at SDSC had to migrate to the T3E.

SDSC's CRAY T3E has 256 processing elements (PEs). Each PE includes a 300 MHz DEC Alpha 21164 processor with 128 megabytes core memory. 240 of these processors are available to parallel jobs, with the remaining processors devoted to interactive serial jobs and system functions. We have approximately 135 Gbytes of disk storage, the bulk of which is available to users in the **/work** file system. The **/work** file system is managed by an automated purge which deletes files as needed if they are not accessed for over 80 hours.

Small jobs (32 processors for less than 60 minutes) may be run interactively, but jobs requiring more resources are forced to run in batch. Similar limits on interactive jobs were imposed on Paragon and T3D users. The purpose of this restriction is to force large jobs to be run in NQS (now called NQE) so that better resource management can be performed.

Porting from the CRAY T3D

SDSC's T3D had only 128 processors and T3D jobs had the restriction of having to use 2^n processors. Except for a few dedicated runs, the largest jobs typically run on the SDSC's T3D were on 64 processors. SDSC's T3E has twice as many processors and there is no power-of-2 restriction on the number of processors per job on the T3E. SDSC users commonly use up to 128 processors on parallel jobs and can get 192 with relative ease. Dedicated runs on up to 240 processors are also available.

The list of major issues we addressed in porting user codes from the CRAY T3D to the CRAY T3E includes:

Compiling, Debugging, and Running Jobs

- **cf77** changed to **f90**
- **mppldr** became **cld** (we recommend linking with **f90** or **cc**)
- **-Wf'-dp'** simplified to **-dp**

Monitoring and Controlling Jobs

- **ps -MPel** lists parallel jobs instead of **ps -ALM**
- **grmview** lists parallel jobs and describes state of each processor
- SDSC provides **mpp.pl** to list information about torus usage and jobs

MPI

- MPI is part of the Message Passing Toolkit (CRAY-supplied version of MPI)
- **mpirun** has been replaced by **mpprun**

PVM

- The PVM libraries are supported and automatically linked as on the T3D
- Process spawn calls may be used on the T3E
- The current implementation of MPI has a bug passing string arguments. You must use **fcdtocp** and **cptofcd** to convert Fortran character descriptors to C pointers and back.

SHMEM

- Most SHMEM calls implemented
- **shmem_put** and **shmem_get** similar performance
- **shmem_put** calls to same destination no longer guaranteed to arrive in order
- cache coherence on **shmem_put** now automatic
- **shmem_udcflush** and **shmem_udcflush_line** are no-ops on the T3E

CRAFT

- not available on T3E
- Portland Group expected to incorporate some features of CRAFT into **pghpf**

The change of **shmem_udcflush** and **shmem_udcflush_line** to no-ops might impact certain benchmark programs since they commonly flush the data cache to normalize timing results. Without **shmem_udcflush** timings will be somewhat less reproducible.

Porting from the Intel Paragon

One important difference between the Paragon and the T3E is the difference in data sizes. Be default, floating-point values are 64 bits of precision on the T3E instead of the 32 bits on the Paragon. The T3E does not have double precision and only allows access to 32-bit variables through the compiler's **-s default32**

option. Also, integer values are 64-bits on the T3E, while they are only 32-bits on the Paragon (see the **-i 32** option for help with this). The T3E does not provide 16-bit integers.

Compiling, Debugging, and Running Jobs

- Must use module command to configure your environment to gain access to compilers and libraries
- **f77** changed to **f90**
- Double-precision is not available on the T3E, compile with **-dp**
- T3E batch jobs are released when the number of nodes they request are free, and the nodes are not reserved for the job on the T3E as they are on the Paragon
- The debugger is **totalview** instead of **ipd**. CRAY's **totalview** has a nice graphical user interface as well as a command-line interface, but lacks the ability to query the list of message ready to be received and the list of unsatisfied receives
- On the T3E, integer variables are not initialized to zero, and an arithmetic exception results from using an uninitialized integer variable (the compiler option **-e0** can help with this)

Monitoring and Controlling Jobs

- **ps -MPel** lists parallel jobs instead of **pspart** or **ps**.
- **grmview** lists parallel jobs and describes state of each node
- SDSC provides **mpp.pl** to summarize information about torus usage and running jobs

MPI

- MPI is part of the Message Passing Toolkit (CRAY-supplied version of MPI)
- **mpirun** has been replaced by **mpprun**
- **f90** replaces **mpif77** and **CC** or **cc** replaces **mpicc** (libraries automatically linked, include files located with no special compiler options)
- The current implementation of MPI has a bug passing string arguments. You must use **fcdtocp** and **cptofcd** to convert Fortran character descriptors to C pointers and back.

PVM

- The PVM libraries are available and automatically linked with your code
- You do not need to run the PVM daemon since its functionality is incorporated elsewhere
- Channels feature not implemented, **pvm_channel_src_setup** call not available

Porting a code from a 32-bit system such as the Paragon to a 64-bit system like the T3E can be easy, or it can lead to lots of difficulties. For example, in some C programs pointer values are stored in integer variables. This causes a serious problem on the T3E since integers are only half the precision of a pointer, and the result it s segmentation violation.

Porting from Fortran 77 to Fortran 90

This section is far from complete, but it offers a few common problems that may arise in this part of the conversion. On both earlier systems, the Paragon and the T3D, the compiler was a Fortran 77 compiler with some extensions on each platform. The only compiler available on the T3E is Fortran 90-compliant. Some extensions to Fortran 77 had to be removed from users codes or converted to standard Fortran 90 syntax/usage. For example, now functions defined in external files *must* be declared EXTERNAL. Another difference is that the Paragon bitwise operations had to be converted to the Fortran 90-standard calls (**iibits** replaces **ibits** for example). Also on the Paragon, some parameters of the **open** statement had to be changed to conform to Fortran 90 usage in user codes.

Performance Enhancements

As expected, users reported significant speedups after porting their codes to the CRAY T3E. We have received only preliminary numbers so far, but a sample of these speedups is listed below for codes ported from the T3D to the T3E:

- Splatter (rendering software): shading 5x, rendering 3x, compositing 3x
- GAMESS (ab-initio chemistry): 6x
- MHD code using Roe solver with adaptive mesh refinement: 3.5x
- Pseudo-spectral direct numerical solver: 2x
- SPH hydrodynamics code: 3.5x

These numbers show a dispersion but are not far fro the average speedup factor of 4 that we expected. We received very little Paragon/T3E performance data for comparison, but one user reported an improvement of a factor of 10 in porting GAMESS from the Paragon to the T3E, which was also expected. (We converted Paragon service units to T3E service units at a 10:1 ratio and T3D service units to T3E service units at a 4:1 ratio; these subsequent reports from users verified the validity of those ratios.)

Sources of performance improvement include the increase in clock speed from 150 to 300 MHz and the implementation of a 96 kbyte level 2 on-chip cache. This cache is three-way set associative and feeds data into the 8 kbyte data and instruction caches you might be familiar with from experience with the CPU used in the T3D. The new CPU also has a Missed Address File, which allows the CPU to avoid some stalls for data loads where the data shares a cache line with another outstanding load. The six stream buffers would provide significant memory speedup if we were able to use them at SDSC. They help manage the latency of an off-page memory reference.

Remarks

Overall, we were generally pleased with the ease of migrating users from the Intel Paragon and CRAY T3D to the CRAY T3E. The CRAY T3E proved to be relatively stable and offered a fairly complete set of tools to enable users to port their codes. Our users reported only the problems mentioned above, with many users experiencing no problems at all. Users observed significant performance increases immediately, even before optimizing their codes for the different architecture of the T3E processors.