The Naval Research Laboratory Cray XD1

Wendell Anderson, *Naval Research Laboratory (Code 5593)*, **Dr. Marco Lanzagorta** *ITT Corp,* **Dr. Robert Rosenberg** *NRL (Code 5593)* and **Dr. Jeanie Osburn** *NRL (Code 5592)*

ABSTRACT: *In June 2006, the Naval Research Laboratory expanded its Cray XD1 to 432 dual core Opteron 275 dual core CPUS and 150 FPGA (144 Vertex 2 and 6 Vertex 4) making it the largest XD1 in the world. This paper will examine our experiences with porting NRL codes to the XD1 and our efforts with bringing the FPGA technology into the High Performance Computing Environment.*

KEYWORDS: XD1, FPGA, Dual Cores, HPC

1. Introduction

Since its inception the Center for Computational Sciences (CCS) of the Naval Research Laboratory (NRL) has functioned at the leading edge of High Performance Computing. In that position the center has two different and somewhat conflicting roles at NRL. On the one hand, it is tasked with providing HPC resources to scientists at the Naval Research Laboratory and the Office of Naval Research who require access to the fastest computers in the world. On the other hand the center performs research to push forward the computing capabilities of HPC computers into the realms of the increasingly complex models required by the Department of Defense. When it was time to replace the Cray Multi-Threaded architecture (MTA) at NRL, the Cray XD1 was chosen. For the scientific applications supported by the center the new machine provided an increase in the raw number of floating point operations that could be done each second by a factor of 144. It also supplied the resources needed to research the contributions of multi-core processors and Field Programmable Gate Arrays (FPGA) to HPC.

2. XD1 System

The NRL Cray XD1 consists of thirty-six chassis with six nodes in each chassis. Each node of the NRL system consists of two Opteron 275 2.2 GHz dual core processors with 8GBs of shared memory, and a 73GB 10K rpm 3.5 in. SATA drive. The full XD1 system thus contains 864 cores with a cumulative raw speed of 3.5 teraflops per second. The XD1 system has 144 Xilinx Vertex-II and six Vertex-4 FPGA,

Each node runs a Cray modified version of the SUSE Linux (kernel 2.65.5). The XD1 has both the gnu and Portland Group Fortran and C/C++ compilers available. For performance reasons most codes are compiled and linked with the Portland Group compiler. MPI support is provided through mpich 2.6. Users may improve the performance of their applications by using the tuned AMD Core Math Library (ACML) or the Cray Scientific Library.

In addition to the local disks on each node, the XD1 also has available 30 TBs of fibre channel external disk storage. This space is available from any of the processors via the Lustre disk system. The Lustre disk system provides a high bandwidth option for saving large amounts of data from a program.

User access to the XD1 is available through an XD1 node dedicated to logins. One additional node is devoted to monitoring the XD1 and four more nodes are dedicated to supporting the Lustre disk system. The remaining 210 nodes are compute nodes that are available to users only via the PBSPro batch queuing system. Currently we are running two queues on the XD1, a small queue of six nodes that contain the nodes with attached Vertex 4's and a second large queue that contains the other compute nodes.

Programming support for the FPGA is provided through the standard Xilinx software and through three higher order languages: Mitrion's Mitrion-C, Celoxica Handel-C and DSPlogic. The first two packages run natively on the XD1, while the last package only runs on a Windows PC (the code generated for the must be transferred from the PC to the XD1)

3. Scientific Applications

The XD1 has proven to be a popular resource among the researchers that use the NRL facility with over 3.3 million CPU hours of computational time used since the initial installation of the system. The top six codes in terms of CPU hours are given in Table 1.

Code	CPU Hrs
ARMS	1,350,000
NOZZLE	800,000
NRLMOL	600,000
ADF	120,000
CHARMM	90,000
STARS3D	80,000

Table 1: Code Usage by Time

By far the largest usage was by the Adaptive Refined Magneto-hydrodynamic Solver (ARMS), a massively parallel, flux-corrected transport based code built upon Message Passing Interface communications and NASA Goddard's PARAMESH parallel adaptive meshing toolkit. ARMS performs three-dimensional, time-dependent simulations of solar magnetic storms. Dr. C. R. DeVore and Dr. S. K. Antiochos are using the code to simulate solar storms resulting from a solar eruption. Dr. Judy Karpen is using ARMS to test the hypothesis that concentration of explosive solar activity in filament channels is a result of flux cancellation driven by convective motions beneath the solar surface concentrating the shear in such sites.

The second largest usage is NOZZLE, an MPI code that solves the compressible Navier-Stokes equation on structured multi-block domains using a domain decomposition model for parallel processing. Dr. Andreas Gross is using the NOZZLE program to support Air Force Programs that require the numerical simulation (with or without turbulence) of Coanda wall jet experiments.

The NRLMOL[2] code is an NRL developed code that implements the Density-Functional formalism for clusters and molecules using MPI to parallelize the problem with a master–slave architecture. Dr. Tunna Baruah and Dr. Mark Pederson have been using this code on the XD1 to study molecular vibrational effects on the simulation of a light-harvesting molecule [3].

The ADF code (http://www.scm.com/) is a commercial universal density code for chemists. Dr. Stefan Badescu and Dr. Victor Bermudez of NRL are using the code to perform a quantum-chemical analysis of the interaction of chemical warfare agents with materials.

The CHARMM code (Chemistry at HARvard Macromolecular Mechanics http://www.charmm.org/) is a program for macromolecular simulations, including energy minimization, molecular dynamics and Monte Carlo simulations. Dr. Alex MacKerell and Deva Priyakumar use CHARMM to study the interaction of urea with P5GA RNA.

The STARS3D [4] code is a frequency-domain parallel code for three-dimensional structural/acoustic/seismic simulations. It is based on a high-order finite element method and incorporates several advanced numerical features like hierarchic basis functions, infinite elements, perfectly matched layer approximations and domain-decomposition (FETI) solver. Dr. Seeker Day is using the code to study wideband acoustic radiation and scattering from submerged elastic structures.

4. FPGA Scientific Applications

Users of the XD1 are in the beginning stages of incorporating the FPGA into their codes. The initial applications have come from those users who have developed VHDL codes on a local system with an FPGA and now wish to run their codes on a system with multiple FPGAs. Porting their codes to the XD1 has involved connecting the top-level component of their VHDL program to either registers or memory that can then be read/written by the Opteron processor.

The initial user of the FPGA was Ken Rice, a graduate student of Tarek M. Taha at Clemson University. They are working on accelerating large-scale models of the neocortex, implementing George and Hawkins' recognition algorithm in hardware. This model utilizes a network of nodes -- where each node implements Pearl's Bayesian belief propagation. At present the application has modeled up to 321 nodes using a total of 64 of the XD1's Xilinx-II FPGA.

Commander Charles Cameron of the United States Naval Academy has been using the XD1 to evaluate ray tracing software for lens design and general optical systems processing He has used an MPI C-program to study a system with 22 planar surfaces, two paraboloid reflectors, and one hyperboloid refractor. In going from one core to 839 he achieved an efficiency of 97.9%. His next step is to perform the ray tracing on the XD1 FPGA.

NRL is currently working with Mitrion to port their SGI RASC FPGA BLASTN implementation to the Cray XD1. This work consists in changing the Mitrion code that uses 128 bit data paths from a paired set of QDRAM banks on the SGI to 64-bit data paths from a single QDRAM. Dr, Anthony Malanoski of NRL is planning on using this BLAST to determine organism identification from relatively short sequence reads from an experimental tool.

Other scientists are in the preliminary stages of investigating the use of the XD1 FPGA for adaptive beam forming, cryptography. hyper-spectral image processing, line of sight calculations, and molecular dynamics.

5. FPGA Programming

Users have had success porting their VHDL codes to the FPGA. The main issue has been learning the Cray API between the Opteron and the FPGA. Most have attacked the problem by using the HELLO WORLD program provided by Cray and the documentation provided with it to write the codes necessary to load the net lists to the FPGA and to move data between the Opteron memory and the FPGA RAMS. The other action that the users had to perform was to add their VHDL to the Xilinx software top.prj

NRL has obtained three software packages: Mitrion's Mitrion-C, Celoxica's Handel-C and DSPlogic that provide simpler methods of programming the FPGA rather than the cumbersome task of writing VHDL or Verilog codes. The Mitrion package runs on the XD1 and new versions are closely tied to a specific (and maybe not yet released version) of the Xilinx compiler, so that maintaining the Mitrion software package and incorporating bug releases has presented a challenge. The Celoxica package that runs under Linux, and thus could be run on the XD1, still has not been released. Celoxica has been providing us with temporary licenses for the Windows PC version of their software. This has led to additional work to install and support the software on PCs, make it available to our scattered user, and update the licenses every time the software license expires. Finally, DPSlogic only runs on a Windows PC, so its availability to users is on an as needed basis. The learning curve on these packages has been steeper than we expected and we have discovered that there is no substitute for experience in writing programs for the FPGA.

6. Performance measurements

As part of our evaluation of the XD1, we have looked at the efficiency of the Opteron dual cores and the I/O performance of the disk systems.

The XD1 has the earliest of the AMD dual cores – the Opteron 275. While the 275 has the same L1 and L2 cache for each core as the single core version of the chip, the dual cores share the same DDR memory controller as the single chip processor. This sharing of memory bandwidth can lead to a degradation of the performance of codes running on the dual core chips.

In order to further investigate and quantify the effect of sharing the memory bandwidth between cores, ten applications were run with two different scenarios: n nodes using all 4 cores on the node and 2*n nodes using only one core of each dual core processor.

Since the one core and two core cases were run over the same number of processors, the running times assuming no memory contention should have been the same. In order to determine how close we came to obtaining this ideal, we define the efficiency as

$$100*(1-(T_4-T_2)/T_2)$$

where T_2 is the wall clock time for running the code on N nodes using one core per processor and T_4 is the wall time running it on N/2 nodes using all the cores on the nodes. Since the same number of processors are used in both cases, if the times T_2 and T_4 are the same, we have perfect efficiency (100%). If T_4 is twice T_2 then there is no advantage as running two copies of the application, each on N/2 nodes would finish at the same time as running the application consecutively on N nodes using only one-half of the cores and the efficiency is 0%.

The timing results comparing using one of the cores and using both of the cores are given in Table 2. In two of the cases (NOZZLE and AVUS) we saw efficiencies greater than 100% probably the result of the cores being more closely coupled when using four per node. Six of the remaining eight showed efficiencies of greater than 50% indicating that while the sharing of memory bandwidth degraded the processing efficiency, significant improvements were achieved. Only two of the

applications (LANCZOS at 22% and RFCTH2 at 39%) showed only minor improvements. The poor performance of LANCZOS was expected as this is a sparse matrix code and the ratio of main memory accesses to computations is high due to a high percentage of cache misses.

Application	One Core	Both Cores	Efficiency %
STATIC	313	450	56
CAUSAL	275	293	93
LANCZOS	771	1371	22
NRLMOL	14283	16260	90
ARMS	2090	2524	79
NOZZLE	27498	27286	101
AVUS	1197	963	120
HYCOM	823	849	97
OOCORE	5274	7716	54
RFCTH2	279	448	39

Table 2 Dual Core Efficiency

The running time of applications on high performance computers can be significantly influenced by the time it takes to move data between memory and disk. Users may need to read input data, write out calculated results, store and retrieve data on temporary scratch space, and save restart files. A program running on the XD1 has two disk systems available – a high-speed parallel file system Lustre available to all the nodes on the XD1 and the local low speed SATA disk drives that are available only to the local node. Generally input, output, and restart files will be stored only on the Lustre disk system, as they need to be available to the user from any node including the login node. Temporary scratch files however may be written to the local disk and deleted when the run is completed.

Last year we presented results on the I/O rates to the Lustre and local disk systems. Since then we have upgraded the Lustre disk system by adding an additional controller and devoting 4 nodes to the running of Lustre instead of 2. Table 3 presents the Lustre rates with the first entry indicating the old rate and the second the rate for the upgraded system. Local disk given in Table 4 did not change.

I/O speed improved significantly for up to 4 nodes for writes and 8 nodes for reads. After this, the performance levels off. Although the data rate to the local disk is slower, the application does avoid competing with other nodes in the system that may also be writing to the Lustre system.

NODES	Read (MB/sec)	Write (MB/sec)
1	206/156	165/417
2	325/326	324/7821
4	629/630	646/1298
8	794/1224	709/1393
16	892/1460	862/1250
32	859/1420	893/1280

Table 3 Lustre I/O rates

NODES	Read (MB/sec)	Write (MB/sec)
1	46	58
2	98	105
4	114	209
8	273	406
16	374	804
32	720	1565

Table 4 Local Disk I/O rates

7. XD1 issues

Since the XD1 is a new product of Cray and the NRL system was the largest XD1 ever fielded by Cray we expected that we would see a substantial number of problems with running programs on the system. We did in fact see significant problems, most of them related to the interaction between the XD1 and the Lustre file system.

One of the first problems noted was the slowness of queries that required disk accesses when programs running on the XD1 were using most of the network bandwidth to the Lustre nodes. A standard “ls” request to list the files in a directory on the disk could take as much as five minutes as the operating system struggled to get access to the required information from the disk. A second “ls” request to the same directory resulted in a near instantaneous response as the required data had been cached by the system in memory and no disk accesses were requested.

This slowness also manifests itself when trying to rebuild a disk that has failed. Since we are running a RAID disk system we can reconstruct a single disk if it fails. The rebuild takes one to two hours when done with an empty system, but takes up to three days when the procedure is run on a busy system.

The NRL XD1 uses AFS for user’s home directories. If a user issues a simple AFS request like “fs lq” to check the file system quota and the current directory is on Lustre the process hangs. NRL is currently investigating the problem to determine an appropriate remedy.

Some of our users have reported that their programs crash when writing large files to the disk. This often occurs when writing files that can be used to restart the program at some intermediate point. This has been a tough problem to track down as the problem is difficult to reproduce and a job may run for days before the problem appears. Test jobs run by Cray support personnel have identified GART (used by the hardware to map RDMA accesses to physical addresses) problems on a node allocated to the job. Cray is currently running tests to see if they can more reliably reproduce the problem.

The most bothersome problem from a user's prospective has been the error messages printed out when their MPI jobs die. Typically the message is of the form

```
mpiexec:Error: read_rai_startup_ports:
Failed to read barrier entry token
from rank 3 process on node#"
```

Not only does this message provide almost no useful information about the reason for the program dying, but leads the user to believe that the problem is associated with the Lustre disk system even if the cause is a divide by zero. Cray is currently working on the MPI mpiexec to provide more useful error messages.

8. Conclusions

The XD1 has proven to be a popular resource for the scientists using HPC resources at NRL. The system is constantly loaded and the queues full most of the time. As seen by the examples in this report a wide range of research has been conducted using the XD1.

Codes using the FPGA have developed slower than expected. Programming in VHDL/Verilog is too time consuming for most of our HPC users. Those users who have used the XD1 with these codes had already developed them for use on local systems and are porting their codes to the XD1 tot take advantage of the large number of FPGA. The higher order languages are still in their infancy. Mitrionics has had several releases and the installation process is still not mature. The Celoxica Linux version is still in beta release and not yet ready for the general public. DSPlogic presents a paradigm different from what most of our users are familiar. The learning curve for all the products is moderate.

Shortly after our final purchase of the XD1, Cray announced that they were discontinuing the XD1 product line. While Cray continues to support the product, development of improvements has stopped. Bug fixes are limited to those that do not require a major rework of the software. The latest release of the operating system has been in test for 4 months and will be the final one. Even so, we expect that the XD1 will be a viable part of our center for at least the next three years

Acknowledgments

In addition to the scientists mentioned in the report, we would also acknowledge Ray Yee the onsite XD1 Cray engineer for providing answers to our many XD1 questions, Jace Mogill of Mitrionics who ported the Mitrion BLASTN FPGA implementation from the SGI RASC implementation to the XD1.

About the Authors

Wendell Anderson is a mathematician and the head of the Research Computers Section of the CCS. Jeanie Osburn is head of the Operational Computer Section of the CCS. Marco Lanzagorta is a physicist for ITT Corporation. and Robert Rosenberg is a Computer Scientist at NRL Their interests includes High Performance Computing, data visualization, and computer performance.

References

[1] **W. Anderson, M. Lanzagorta, R. Rosenberg, and J, Osburn,** "Early Experiences with XD1 at the Naval Research Laboratory", *Cray Users Group Meeting Users Group Meeting*, Lugano, SW. May, 2006

[2] **M. Pederson, D. Porezag. J. Kortus and D. Patten,** "Strategies for massively parallel local-orbital-based electronic calculations", *Physica status Solidi,* KnB217 197 (2000)

[3] **T. Baruah, M. Pederson. and W. Anderson,** "Massively Parallel Simulation of Light Harvesting in an Organic Molecular Triad", *DOD HPCMP Users Group Meeting*, Nashville, TN. June, 2005

[4] **S. Dey and Dibyendu K. Datta,** "A parallel hp-FEM infrastructure for three-dimensional structural acoustics", *International Journal for Numerical Methods in Engineering, Vol 68 Issue 5, pp 583-603.*