LA-UR-19-27106

Investigating the use of BlueField with OpenSHMEM and MPI over OpenUCX distributed applications

Liliana Aguirre Esparza, New Mexico State University
lili91@nmsu.edu

Mentor: Wendy K. Poole, HPC-ENV
wkpoole@lanl.gov

Co-mentor: Steven Poole, ALDSC/HPC
swpoole@lanl.gov

Introduction

We are moving into highly heterogeneous computing systems and distributed data centers. With this evolution comes issues of continued exploitable performance. Today's data center servers must deal with very large storage and computational workloads, which currently spend very valuable CPU cores on ancillary processing, instead of focusing on the applications requirements. This in turn results in a decrease in focused performance. It is crucial that we address these issues and find new areas and methods to relieve the pressure off CPU cores that should be used for applications performance.

Goal

The objective of this summer's research was to begin the development of a methodology that will allow us to profile, gather and analyze any prospective computations from kernel from both the Department of Defense (DoD) and LANL, to be offloaded onto the BlueField Multicore System on Chip to improve application performance. The goals were to build, run and understand the provided benchmarks. These benchmarks are written in the MPI and OpenSHMEM communication libraries. This would allow us to gain an understanding of the applications and take steps towards creating a library for Smart NICs.

The BlueField Multicore System On A Chip (SoC) and Controller Card [1]

BlueField SoC Features

Bluefield is a highly integrated system on chip (SoC), which is optimized for NVMe storage systems, Network Functions Virtualization (NFV), security systems, and embedded applications. The BlueField SoC was developed by Mellanox Technologies as a way to address performance, network and cybersecurity concerns. The chip includes:

- a set of 64-bit Armv8 A72 core CPUs,
- a PCIe switch,
- and a network controller.

The BlueField SoC was developed as a solution for building Just-A-Bunch-Of-Flash (JBOF) systems, All-Flash-Array and storage applications for NVMe over Fabrics. The PCIe switch on the BlueField supports up to 32 lanes of both Gen3 and Gen4, which allows transfers of more than 200Gb/sec of data to and from the SSDs.

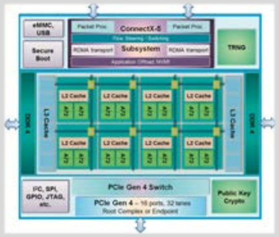


Figure 1. BlueField Architecture

Controller Card

The Mellanox BlueField Controller Card is designed to control and take advantage of all the features of the BlueField SoC. The card has a PCIe standard form factor, and can be used as a high performance networking card.

Ethernet/ InfiniBand
NVMe Over Fabrics
BlueField
NVMe

Figure 2. BlueField SoC

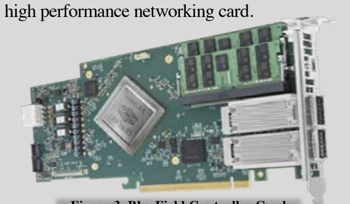


Figure 3. BlueField Controller Card

Benchmarking

Several profiling tests were performed on a DoD application using a sampling tool, in order to understand which part of the application was consuming the most time. The application was written to support both MPI and OpenSHMEM, therefore each test was duplicated to account for both libraries. The tests were performed on both Trinitite (Intel) and Capulin (Arm) in order to observe the effect of memory and number of cores on the overall application performance.

Methodology

The first set of tests were performed on Trinitite and Capulin using 2 nodes. Each node on Trinitite has 128GB of memory and 32 Haswell cores, and each node on Capulin has 256GB of memory and 56 Arm cores. The tests were done using 25%, 50%, and 75% memory. For each memory percentage three different tests were implemented using diverse program arguments. A second set of tests was performed using 4 nodes, and a third using 8 nodes.

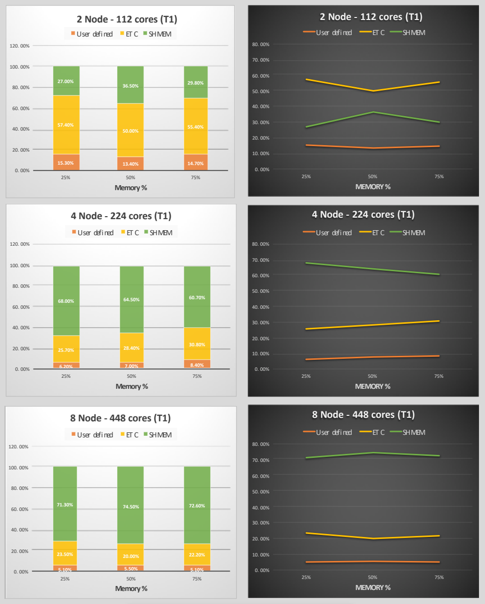


Figure 4. Capulin Test 1 Results

Profiling Results

The tests on Trinitite revealed no clear pattern. The results varied based on the memory percentage and the program arguments. However, the results on Capulin yielded a clearer pattern. Based on these results it was concluded that time spent on communication between PEs using both OpenSHMEM and MPI increased in correlation to the number of cores. The profiling results for test one with OpenSHMEM at 25%, 50%, and 75% memory, using 2, 4 and 8 Capulin nodes are displayed in Figure 4. A similar pattern was shown for tests 2 and 3.

BlueField Testing

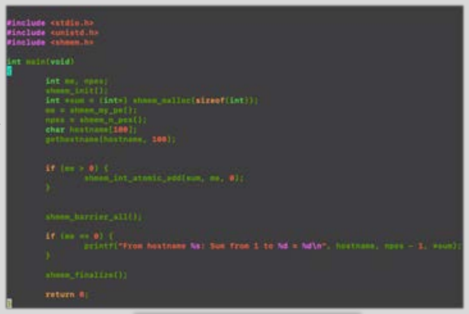
A simple program was used to test the use of atomic operations across PEs on the host nodes and the BF cards. The code was adapted from a program found on the official OpenSHMEM website.[2]

Testing Environment

The system used to perform these tests was made up of several types of nodes. For our purposes we used the Jupiter nodes with BlueField cards attached. The Jupiter nodes consist of 10 cores, each with 2 threads, resulting on a total of 20 PEs per node. Each Jupiter node also has one BlueField controller card and SoC, which consists of 16 cores.

Test Program

For the test program, shown in Figure 5 we used 20 cores on both Jupiter nodes 8 and 9, and all 16 in each of the Bluefield cards, for a total of 72 PEs. It was noted that in order to run the program in parallel, the code must be on both the host and BF the cards. The program was ran using an appfile specifying how many cores each node and BF card should use to run the program. The test program atomically adds all PE rank values and stores the result into the symmetric object "SUM" on PE 0. Once all rank values have been added, the result is printed.



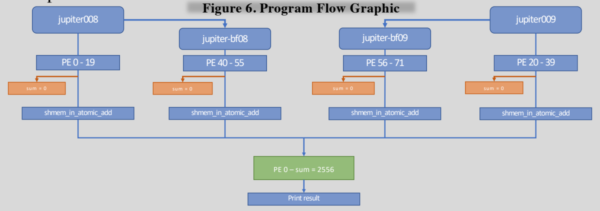
```
#include <stdio.h>
#include <unistd.h>
#include <shmem.h>

int main(void)
{

        int me, npes;
        shmem_init();
        int *sum = (int*) shmem_malloc(sizeof(int));
        me = shmem_my_pe();
        npes = shmem_n_pes();
        char hostname[100];
        gethostname(hostname, 100);


        if (me > 0) {
                shmem_int_atomic_add(sum, me, 0);
        }


        shmem_barrier_all();

        if (me == 0) {
                printf("From hostname %s: Sum from 1 to %d = %d\n", hostname, npes - 1, *sum);
        }

        shmem_finalize();

        return 0;
}
```

Figure 5. Test Program Code

Results and Observations

After the running the program it was noted that the order of the entries in the appfile, determines the PE ranks. Based on the output, it was concluded that atomic operations are correctly executed across all PEs. Figure 6 demonstrates the program's flow with regard to individual processes.

Figure 6. Program Flow Graphic



Future Work

The profiling data gathered gives us a greater understanding of the provided application and allows us to hypothesize about what computations may be offloaded to improve performance. Based on the results, a good place to start offloading is communication computations. Furthermore, the information gathered about how the BlueField relates to the host node will allow us to find the best way to break down the application so the BlueField can handle specific computations in a way that actually improves performance. The next step would be to perform the same profiling tests from Trinitite and Capulin (or larger systems) on a system equipped with BF cards and compare those results to the previously gathered information.

Acknowledgements

I would like to acknowledge the mentoring I have received from Wendy Poole, Steve Poole, and Brody Williams. I also would like to acknowledge the Department of Defense as well as Mellanox Technologies for the support to this project.

References

[1] Mellanox Technologies, http://www.mellanox.com/products/bluefield-overview/
[2] OpenSHMEM.org, https://github.com/openshmem-org/openshmem-examples/blob/master/C/sum2n.c