Assessing benefits of hardware acceleration using CUDA in an MPI environment.

URL:- http://www4.ncsu.edu/~axavier/ps-project.html

Single Author Info:
UserName:- Anitta Jose(000935568)

Group Info:-
UserName:- Allen Pradeep Xavier(000935568)
UserName:- Anitta Jose(000920671)
UserName :- Sreekanth Mavila (000916577)

Problem Description and Solution approach

We have selected three NAS PB benchmarks.

MG → (MultiGrid)Approximate the solution to a three-dimensional discrete Poisson equation using the V-cycle multigrid method[6]

FT →(Fast Fourier Transform) Solve a three-dimensional partial differential equation (PDE) using the fast Fourier transform (FFT)[6]

IS → (Integer Sort) Sort small integers using the bucket sort.[6]

We would like to optimize the performance using cudaMPI.

We have profiled these benchmarks using gprof[5]. Below are the bottlenecks we could find for each of these benchmarks

1) MG :- Subroutine **resid** which computes the residual execute for 44.07% of time though the number of calls being made to that call is only 170. This function calculates the residual in a loop. So making this loop calculation to execute in parallel we hope to achieve a better performance[7].
2) FT:- Subroutine **fftz2** shows the maximum % execution time. But there are 230912 calls made to this function. So we can conclude this high execution time may not be because this subroutine computation intensive. The subroutine which shows next highest execution time is **evolve .** This also has a loop calculation which can be parallelized[7].

3) IS:- Two subroutines **rank** and **randlc** shows maximum execution time. Here randlc is a random number generator which is invoked 8388631 times. But at the same time rank is invoked only 11 times and it gives a similar exection time as randlc. We plan to optimize performance of this function by parallelizing the loop calculation in this function[7].

Timeline

Milestones	**Deadline**
1. Identifying the exact location of bottleneck in the function identified using gprof.	11th November
2. Writing CUDA kernel . Decide on Fortran/C/Using Fortran to Cuda compiler / Using PGI Fortran Cuda compiler. [1][2][3][4]	11th November
3. Optimization of individual benchmarks.	Final day of project submission

Task Assignment

Task Owner	**Task**
Group Task	Decide on Fortran/C/Using Fortran to Cuda compiler / Using PGI Fortran Cuda compiler. [1][2][3][4]
Allen Pradeep Xavier	Optimization of MG
Anitta Jose	Optimization of IS
Sreekanth Mavila	Optimization of FT

References

1. **http://www-ad.fsl.noaa.gov/ac/Accelerators.html**
2. **http://www.pgroup.com/resources/cudafortran.htm**
3. **http://www.cs.uaf.edu/sw/cudaMPI/**
4. Message Passing for GPGPU Clusters: cudaMPI
   Orion Sky Lawlor. Department of Computer Science, University of Alaska Fairbanks
   202 apman Building, Fairbanks, Alaska 99712, USA. lawlor@alaska.edu
5. http://www.cs.duke.edu/~ola/courses/programming/gprof.html
6. http://en.wikipedia.org/wiki/NAS_Parallel_Benchmarks
7. http://www.nas.nasa.gov/News/Techreports/1994/PDF/RNR-94-007.pdf