Optimizing IRS Benchmark for IBM Cell Processors

Balasubramanya Bhat

Suraj Kasi Satyanarayana

Project Goals

The goal of this project is to run the IRS (*Implicit Radiation Solver*) parallel benchmark [1][2][3] on multiple IBM cell processor [5] nodes using MPI and accelerate the runtime of the benchmark on each node using the multiprocessing capability available within the IBM cell processor (PPE and SPEs).

Purpose of IRS Benchmark

The name IRS stands for *Implicit Radiation Solver* [1][2][3]. IRS solves a diffusion equation on a three-dimensional, block structured mesh. The single CPU instruction mix for this benchmark is roughly 40%load/store, 18% floating point, 31% fixed-point, and 11% branch.

Runtime Analysis of this benchmark on Henry 2

In order to analyze the current behavior of this benchmark, we ran it on *Henry 2* cluster using MPI for communication and *gprof* for profiling the execution time. Following are the top 10 functions in terms of the execution times.

```
Each sample counts as 0.01 seconds.
```

% time	cumulative seconds	self seconds	calls	self s/call	total s/call	name
44.70	3.50	3.50	826	0.00	0.00	rmatmult3
10.22	4.30	0.80	6	0.13	0.86	MatrixSolveDriver
9.45	5.04	0.74				sock_msg_avail_on_fd
4.73	5.41	0.37	820	0.00	0.00	icdot
3.19	5.66	0.25	19	0.01	0.01	volcal3d
2.55	5.86	0.20				socket_msgs_available
2.55	6.06	0.20	1227	0.00	0.00	norml2
2.17	6.23	0.17	6	0.03	0.03	divgradpert3d
2.04	6.39	0.16	2071	0.00	0.00	setpz1
2.04	6.55	0.16	6	0.03	0.03	DiffCoef

Planned Solution

We can see that above 10 functions make up for 83.64% of the total execution times at each node. We plan to accelerate the above 10 functions using the multi-processing capability of the IBM cell processors [5]. Each cell processor has 1 PPE and 8 SPEs (6 SPEs on Sony PS cell processor [4]). The main processing happens on the PPE. We plan to do functional and domain partitioning of the above functions and run them on multiple SPEs. From the above table, we can see that rmatmult3 and MatrixSolveDriver are responsible for more than 55% of the total computation, forming the biggest hotspots within the benchmark. These functions deal with matrices and can be easily parallelized using the techniques studied in the course.

Timeline for the project

Dates	
Oct25 – Nov2	Understanding the benchmark code
Nov3 – Nov9	Porting the benchmark onto the IBM cell processors without any modifications to the code.
Nov10-Nov15	Converting the hotspot functions into kernels and running it on the cell processor.
Nov16- Nov24	Testing and debugging the code.
Nov 24 -	Document the steps and write report.

Deliverables

The deliverables for the project includes the modified source code for the IRS bench mark, project report, original and the improved timing metrics.

References

[1] https://asc.llnl.gov/computing_resources/purple/archive/benchmarks/irs/
[2] http://moss.csc.ncsu.edu/~mueller/cluster/cluster03/g1/irs.pdf
[3] https://asc.llnl.gov/computing_resources/purple/archive/benchmarks/irs/irs.readme.html
[4] http://moss.csc.ncsu.edu/~mueller/cluster/ps3/
[5] http://www-03.ibm.com/technology/cell/index.html