FC with RG – Near Memory

Andrej Vrtanoski, Darryl Bailey, Yu Pan

Project Goals:

- Learn about the underlying architecture of the Lucata Pathfinder and frameworks needed to program it.
- Implement & run programs on the Lucata Pathfinder.

Background & Motivation:

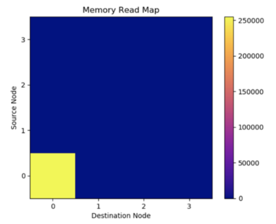
- Lucata Pathfinder: near-memory HW architecture with the concept of migrating threads
- Migrating threads: Move computation (thread context) to memory
- The Lucata Pathfinder tackles the problem of dominating communication overhead of data-intensive sparse applications.

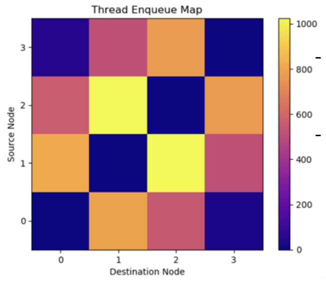
Project Overview and Methods:

- C/C++: Primary language worked in.
- Cilk: Programming Language used to implement parallelism in the Lucata Pathfinder
- Simulator: Environment used to test and run Cilk code before running on the Pathfinder.
- Data allocation and Distribution: Pathfinder memory_web library used to implement methods for distribution amongst the multiple Pathfinder nodes.

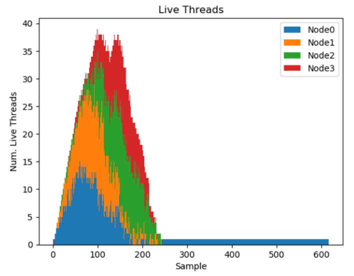
Experiments and Results:

- Implemented a matrix multiplication algorithm in order to learn how to use the Lucata systems, the tools it provides and the Lucata c libraries that it provides.





- Experimented with cilk_for vs cilk_spawn_at to optimize for local thread spawns.
- Converted our code from using malloc (figure 1) to malloc_2d (figure 2) to stripe the data across the node.



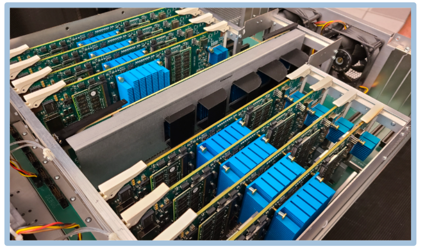
- Learned how to use the Lucata emulator to generate the following debug graphs.
- Scheduled code locally on the Lucata system using ssh in order to benchmark the performance of the dense matrix multiple.
- Implemented a 2d block distribution algorithm.

Serial

{"N":16,"time_ms":75.19,"ticks":13157848}

Parallel

{"N":16,"time_ms":2.50,"ticks":437077}



Lessons Learned:

- Programming with knowledge of the hardware is difficult.
- Dense matrix multiplication was not meant for the Pathfinder. It is definitely more suited to sparse data structures.

Next Steps and Future Challenges:

- Implement PageRank with GraphBLAS.
- Run programs on the Lucata HW rather than just simulate.
- Look into algorithms the Pathfinder is suitable to run.