Statement of Research

Scott J. Krieder, 3rd Year Doctoral Student
Department of Computer Science, Illinois Institute of Technology

I. Funding History

As a 3rd year PhD student I have been supported as both a Teaching and Research Assistant in the Department of Computer Science. As a Teaching Assistant, I have mentored students in multiple courses including: Computer Organization, Systems Programming, Operating Systems, Advanced Operating Systems, and Data-Intensive Computing. In addition, I am supported this calendar year by a 2013 Starr/Fieldhouse Fellowship [1] from the Illinois Institute of Technology to pursue a collaboration with Argonne National Laboratory and the Computation Institute at the University of Chicago for my work titled "Towards the Support for Many-Task Computing on Many-Core Computing Platforms". [2][3][4][5]

II. Research Contributions

Current software and hardware limitations prevent Many-Task Computing (MTC) workloads from leveraging hardware accelerators boasting Many Core Computing architectures. Current work aims to address the programmability gap between MTC and accelerators, through the innovative CUDA middleware GeMTC. By working at the warp level, GeMTC enables heterogeneous task scheduling and 10x number of workers compared to CUDA. In order to span multiple accelerators across nodes, this work has adopted the Swift parallel programming system [6], which can both support fine grained millisecond tasks and extreme scale supercomputers at 100K-cores. Currently CUDA developers may only have a maximum of 16 kernels running concurrently, one kernel per streaming multiprocessor (SM). The problem is that all kernels have to start and end at the same time, causing extreme inefficiencies in heterogeneous workloads. By working at the warp level and trading local memory for concurrency the GeMTC framework is able to run up to 84 concurrent kernels. [4] This middleware allows independent kernels (MIMD style) to be launched and managed on many-core architectures that traditionally only support SIMD. [3] As shown in 1 Swift/T [7] makes calls to the GeMTC API and passes tasks into memory on the device. Warp workers pick up tasks from the in memory queue, execute the task with the given parameters and place results on an outgoing result queue. Finally, Swift/T will poll the device and return results from the result queue back to the appropriate task in the swift script. The worker count is dependent on hardware architectures, where current generation GPUs have O(10) Streaming Multiprocessors (SMX), O(100) Warps, and O(1000) cores. [8]

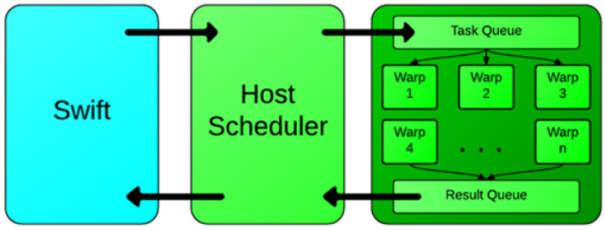

Fig. 1. Flow of a task through Swift/T and GeMTC.

III. Future Direction

Current work designed and implemented the GeMTC framework. A sub-allocator provides improved memory management for dynamic tasks. Integration between GeMTC + Swift/T provides application support, data-flow parallelism, and multi-node scalability. Future work aims to abstract for additional accelerator support (Intel Xeon Phi, AMD GPUs), evaluate real applications such as the Open Protein Simulator (OOPS) [9], and improve current framework performance.

References

[1] I. Research, "Starr research fellowship and fieldhouse research fellowship," http://www.iit.edu/research, 2012.
[2] S. J. Krieder and I. Raicu, "An overview of current and future computing accelerator architectures," 1st Greater Chicago Area System Research Workshop Poster Session, 2012.
[3] S. J. Krieder, B. Grimmer, and I. Raicu, "Early experiences in running many-task computing workloads on gpgpus," XSEDE Poster Session, 2012.
[4] S. J. Krieder and I. Raicu, "Towards the support for many-task computing on many-core computing platforms," Doctoral Showcase, IEEE/ACM Supercomputing/SC, 2012.
[5] B. Grimmer, S. J. Krieder, and I. Raicu, "Enabling dynamic memory management support for mtc on nvidia gpus," Poster Session, EuroSys, 2013.
[6] Y. Zhao, M. Hategan, B. Clifford, I. Foster, G. Von Laszewski, V. Nefedova, I. Raicu, T. Stef-Praun, and M. Wilde, "Swift: Fast, reliable, loosely coupled parallel computation," in *Services, 2007 IEEE Congress on*. IEEE, 2007, pp. 199–206.
[7] J. M. Wozniak, T. G. Armstrong, M. Wilde, D. S. Katz, E. Lusk, and I. T. Foster, "Swift/t: Large-scale application composition via distributedmemory data flow processing," in *Proc. CCGrid*, vol. 13.
[8] NVIDIA, "Nvidia kepler gk110 architecture whitepaper," http://www.nvidia.com/content/PDF/kepler/NVIDIA-Kepler-GK110-Architecture-Whitepaper.pdf, 2012.
[9] A. N. Adhikari, J. Peng, M. Wilde, J. Xu, K. F. Freed, and T. R. Sosnick, "Modeling large regions in proteins: Applications to loops, termini, and folding," *Protein Science*, vol. 21, no. 1, pp. 107–121, 2012.