Annotated Bibliography

Amy Le, Emily Bertelson, Kristina Entzel, Long Tat
CSCI 4576/5576
November 18, 2014

References

[1] K.B. Athreya and Jack Dai. Random logistic maps. I. *Journal of Theoretical Probability*, 13(2):595–608, 2000.

Athreya and Dai explore the concept of a time-varying logistic map. In a sense, they lay out the general groundwork for exploring a spatially-varying logistic map. They find a theoretical explanation for their observations, expressed as probabilities. These findings imply that (and this is beyond the scope of the project) there could be some interesting observations from the spatially-random logistic map, and the observations can be quantified as probabilities. This project's goal is to characterize the map in terms of the average number of period p orbits in any given realization, and also to find the set-valued bifurcation diagram that describes this system.

[2] Center for Computational Research. MPI and parallel computing, 2004–2014. http://www.buffalo.edu/ccr/support/UserGuide/AdvancedTopics/mpi.html.

Explanatory paragraph goes here.

[3] Jeroen S.W. Lamb, Martin Rasmussen, and Christian S. Rodrigues. Topological bifurcations of minimal invariant sets for set-valued dynamical systems. *Proceedings of the American Mathematical Society*, 2013.

Lamb, et. al explore the concept of set-valued bifurcations as an extension of the more common single-valued bifurcation. The kind of problems the authors are interested in are random dynamical systems, such as the Random Logistic Map. This paper serves as a theoretical underpinning for our simulation, and also as a reference for the set-valued bifurcation diagram we plan to generate.

[4] S. Olivier and J. Prins. Scalable dynamic load balancing using upc. In *Parallel Processing, 2008. ICPP '08. 37th International Conference on*, pages 123–131, Sept 2008.

Olivier and Prins implement an asynchronous work-stealing dynamic load balancer with Unified Parallel C (UPC). They evaluate the performance of their balancer with the Unbalanced Tree Search (UTS) benchmark, which is a synthetic tree-structured search space that is highly imbalanced. They observe parallel efficiency of 80% using 1024 processors performing over 85,000 total load balancing operations per second continuously. An additional finding is that the careful use of one sided reads and writes is necessary to minimize the communication overhead. The authors' findings indicate that we should minimize the number of read and write operations as we compute solutions to the fixed point equations in order to keep the communication overhad low.

[5] Various contributers: The Open MPI Project. Open MPI: Open Source High Performance Computing, 2014. http://www.open-mpi.org/faq/.

Explanatory paragraph goes here.

[6] Marc H. Willibeek-LeMair and Anthony P. Reeves. Strategies for dynamic load balancing on highly parallel computers. *IEEE Transactions on Parallel and Distributed Computing*, 4(4), September 1993.

Willibeek-LeMair and Reeves discuss five strategies for dynamic load balancing: sender initiated diffusion, receiver initiated diffusion, hierarchical balancing model, gradient model, and dimension exchange method. The authors consider tasks such as: processor load evaluation, load balancing profitability, task migration, and task selection. Given the trade off between accuracy and increased time for communication, they conclude that the receiver initiated diffusion (RID) is the best method for dynamically load balancing. It is the method that scales the best with the number of processors and requires the least amount of communication overhead. Dynamic load balancing is key for solving systems whose solutions are defined recursively, such as a fixed point iteration. We will use the findings of this paper to guide our program design such that it is the best suited for RID.