RESEARCH STATEMENT

Henry R. Moncada (hrmoncada@gmail.com)

My research interests span the areas of parallel programming, mathematical modeling, and machine learning. A common thread in my research is in understanding the theory and design of mathematical modeling framework to be used on scalable parallel architectures. Broadly speaking my Ph.D. research belongs to the area of *parallel computing, electromagnetics mathematical modeling and 3D structure arrays*. Most of my work has greatly benefited from interaction with a number of colleagues and my advisor. I describe my work below.

Background and Current Work

My research consists of design a faster implementation of an algorithm to generate 3D spatially variant lattices (SVL) structures and improve its performance when it is running on a parallel computer system. The algorithm is used to synthesize a SVL for a periodic structure. The algorithm has the ability to spatially vary the unit cell, the orientation of the unit cells, lattice spacing, fill fraction, material composition, and lattice symmetry. The algorithm produces a lattice that is smooth, continuous and free of defects. The lattice spacing remains strikingly uniform even when the lattice is spatially varied which it is important for maintain the consistency of lattice's properties throughout its structure. Periodic structures like photonic crystals or metamaterial devices can be enhanced using the spatially variant algorithm, thus unlocking new physical mechanisms [7, 8].

A portable computer program for parallel architectures was developed for this computer simulation. To write the code, we pick a general-purpose programming language that supports structured programming and two computer science tools. The Fastest Fourier Transform in the West (FFTW) was used to handle the Fourier transform of the unit cell device and the Portable, Extensible Toolkit for Scientific Computation (PETSc) for handle the numerical linear algebra operations and Message Passing Interface (MPI) for distributed memory to improve the performance of the code that generates 3D SVL when it is executed on a parallel system.

A study of the performance, efficiency and scalability of the SVL code on the two architectures on Stampede2 (KNL and SKX) was develop on this research. In a parallel computer system each individual component is identical in nature and can perform the entire tasks of the larger system, althougth usually at a slower rate. Of course it will be naive to expect that we could just bunch together a number of parallel components and get high performance. The performance of any parallel system is governed by

- The architecture used to connect the many components.
- The resource management (load balancing) algorithm used to allocate tasks amongst the parallel components.

On the other site, a scalable code can handle proportionally very small to large tasks of computational operations. To scale the SVL code, we use the isoefficiency analysis to proportionally increase the problem size and resources to study the scalability of the SVL code. This type of analysis is important to understand the performance of the SVL code when it is executed on a supercomputer and find the best way to maximize its performance. For my Ph.D. research, two main goals were stablished to be achieved with the code design of SVL parallel application.

- Performance: The capacity to reduce the execution time to solve the problem when the computing resources increase. We metric the performance of a parallel application normally by comparing the execution time with multiple processors and the execution time with just one processor.
- Scalability: Measure of a parallel systems capacity to increase performance when the complexity, or size of the problem, increases. An application is said to be scalable when its efficiency is maintained by increasing the number of processes and the size of the problem, proportionally, reflecting its capacity in making use of available resources effectively. The efficiency is defined as the ratio of speedup to the number of processors and measures the fraction of time for which a processor is usefully utilized.

  $$E = \frac{S}{p} = \frac{T_{seq}}{pT_{par}}$$

  where p is the numbert of processes, T_{seq} serial runtime of the sequential algorithm, and T_{par} parallel runtime of the algorithm to be solve on p processors

- Increase number of processors, decrease efficiency
- Increase problem size, increase efficiency

A scalable parallel system can keep efficiency (made the cost-optimal) by adjusting the number of processors and the problem size simultaneously. Note that an algorithm may have different performance, efficiency and scalabilioty on different parallel architecture.

However, there are architectural and algorithm limitations factors that limited the performance and scalability of the 3D SVL. To made sense of the effect of these limitation we use the isoefficiency analysis [11]. The isoefficiency analysis is a way to measure scalability and allows us to calculate at what growth rate we should increase the problem size while increasing the number of processors p in order to keep the efficiency fixed. The reason we cannot simply double the problem size when we double the number of processors is because of parallel overhead (for example, communication overhead), which typically depends on both the problem size and the number of processors[4].

The Isoefficiency function, parallel execution time can be expressed as a function of problem size, overhead function, and the number of processing elements. We can write parallel runtime as

$$T_{par}(N,p) = \frac{W - T_{over}(W,p)}{p}$$

The overhead function $T_{over}(W,p)$ is the part of the parallel system cost (processortime product) that is not incurred by the fastest known serial algorithm on a serial computer. The parallel efficiency can be expressed as a function of the total overhead T_{over} and the sequential execution time T_{seq} as follows:

$$E = \frac{S}{p} = \frac{T_{seq}(N)}{p\,T_{par}(N,p)} = \frac{T_{seq}(N)}{W - T_{over}(W,p)}$$

where E is the desired efficiency to be maintained. We assume that it takes unit time to perform one basic computation step of an algorithm. This assumption does not impact the analysis of any parallel system because the other hardware-related constants, such as message startup time, per-word transfer time, and per-hop time, can be normalized with respect to the time taken by a basic computation step. With this assumption, the problem size W is equal to the serial runtime $T_{seq}(N)$ of the fastest known algorithm to solve the problem on a sequential computer.

$$E = \frac{S}{p} = \frac{W}{W - T_{over}(W,p)}$$

We can solve for W in terms of p to give a function that tells us how we should scale up W as we increase p.

$$W = \frac{E}{1-E} T_{over}(W,p) = K\,T_{over}(W,p)$$

Highly scalable systems have small isoefficiency function.

Our research results show that the SVL isoefficiency scaling cannot maintain the efficiency as we scale up the problem size and the computer system resources simultaneously. On the other hand, the SVL code increases its efficiency as the problem size increases. Considering a more significant problem could show improving scalability. The lower values we get in the efficiency are due to the parallel overhead. We can not account for all the overhead as interprocessor communication. We suspect that memory pressure and contention are causing some of the overhead. We can try to reduce that effect by improving the way the linear system is solved [2, 5]. Our future work consists of improving the efficiency and scalability by reducing the overload performance of communication overhead. Moving data limited the performance achievement of the SVL code. Improving the scalability as the problem size and the number of processors grow is an essential factor in getting a better performance of the SVL code.

A Research Agenda

In the course of my research, I have noticed a number of hardware advances in parallel computing architecture, insights into algorithms as well as analysis techniques aid in the design of elegant and simple solutions. I envisage

the field of mathematical modeling created from the ground up, building upon the foundations of a number of fields. One in particular field in computer science get my attention, machine learning. Even though, I did not work directly with machine learning. Many of the mathematical armories I am awarded. It has been used in machine learning.

In the near future, I am interested in the principles involved in the design of necessary application of machine learning in combination with frameworks of mathematical modeling and parallel computing. Parallelism can play a crucial role in improving the performance of machine learning techniques. Since machine learning uses methods of data analysis that automates analytical models technique that teaches computers to do what comes naturally to humans and animals; learn from experience and improve their learning over time in autonomous fashion, by feeding them data and information in the form of observations and real-world interactions. Machine learning requires the use of intensive data analysis to improve the quality of their learning over time, and parallel computing can play a fundamental key in speed up the computational process of improving the performance of machine learning models. The flexibility of machine learning techniques models allows researchers, data scientists, engineers, and analysts to produce reliable, repeatable decisions and results and uncover hidden insights from complexity relationships and trends in the data and mathematical model techniques[10].

In the future, my research will involve a right mix of futuristic and present-day research where performance and scalability will remain essential, in improving the performance of any application. One part of my work will focus on fundamentally different proposals and radical solutions that involve mathematical modeling and machine learning in combination with parallel tools libraries to perform complex calculations so that we can achieve performance level without too much cost in calculation time and overhead will still be required as a way to improve quality code performance.

As an example, many areas in which parallel computing in combination with numerical techniques can be used. Remain an open research field for machime learning. Below, I am pointing out few ones that I am particularly interested in developing.

- Biology: The relations between biology and the field of machine learning is long and complex. Nowadays, modern biology can benefit from the advancements made in the area of machine learning techniques. The success or failure of machine learning techniques on a given problem is sometimes a matter of the expertise of the user and the quality of the indices used to evaluate the results. It should be clear that choice, tuning, and diagnosis of machine learning applications are far from mechanical, although caution should be taken when judging the superiority of some machine learning approaches over other categories of methods involved in the classifier design should be cross-validated to obtain an unbiased estimate for classifier accuracy. In addition, using mathematical models to understand the biological processes that shape population and community dynamics, with a particular interest in the evolution of infectious diseases and modeling problems in systems biology[10].
- Physics: Laboratoiers like CERN handles so much data in a single run of the Large Hadron Collider. Machine learning algorithms are used to check for anomalous data and to present a summarised report for humans to work with. Therefore, instead of hard-coding programs for a specific singular task, we can now teach machines to learn and adapt themselves for different tasks at hand and intuitive models of physical systems can be replaced by abstract models of data and mechanical patterns of cause and effect. In addition, the use numerical analyses to study the dynamics of material mixing, molecular dynamics, elasticity, material transport, radiation transport and multi-scale material modeling.
- Geoscience: An improving in better sensing technologies, in better computational resources for running large-scale simulations models, and internet-based democratization of data that has enabled the collection, storage, and processing made geosciences witnessed a major revolution from being a data-poor field to a data-rich field. These acculation of data and technology advance made application of machine learning techniques very useful in geosciences and remote sensing areas[6].
- Electrodynamics Applications : Machine learning prediction is becoming increasingly popular. It provides great advantages for problems that are difficult for human but easy to solve for machine. For example, machine learning prediction of electromagnetic frequencies had been tried to be estimated using machine learning. Also, the use parallel computing as well as mathematical tools to impletement and speed up computational simulation eletrodynamics applications.

- Network/Cyber Security : With the increased popularity of machine and deep learning tools. A lot of security practitioners believe that these approaches are the magic silver bullet we have been waiting to solve all of our cybersecurity challenges. To make a broad statement, we are trying to find anomalies, identify malicious behavior or malicious entities; call them hackers, attackers, malware, unwanted behavior, etc. In abstract terms, these events are not statistical. For example, an increase in network traffic might be statistically unusual, but from a security point of view, that rarely ever represents an attack. To address these issues properly, we need to mix experience and knowledge. We have to work with experts to capture their knowledge and use that on the algorithms to reveal actual security insights or issues, just like you work with a teacher (the expert) to have him guide you on your journey [1, 9].
- Computer Vision: Machine learning methods have added an enormous boost to the rapidly developing field of computer vision. Today a lot of new applications of computer vision techniques have been introduced and are now becoming parts of our everyday lives. Data collection has become increasingly exponentially putting on a rigorous challenge to the scalability and robustness of the computer vision and machine learning algorithms. The manipulation of images viewpoint variation, illumination, occlusion, scale, deformation, background clutter, and intra-class variation variables are used to learn what distinguishes them rather than specify the difference in objects recognition, face recognition and indexing, photo stylization or machine vision in self-driving cars. Making harder to understand the internal mechanics of this technology[3].

Machine learning has been accelerating the upward trend over the last few years. It provides great advantages for problems that are difficult for human but easy to solve for machine. As a result, machine prediction is becoming increasingly popular. Practically every commercial and scientific domain has witnessed a major revolution in data acculation, from being a data-poor field to a data-rich field. This acculation of big data offers immense potential for machine learning techiques that can be widely applied to the science and engineering problems.

In contrast, I intend to devote the other part of my work on practical systems, which have immediate relevance and impact in the industry. I intend to work closely with a number of researchers in related fields. Similarly, I intend to collaborate with industry in understanding and developing solutions for practical problems. I believe my past experience of research work done jointly with a number of colleagues will help to be problem solver created and help me achieve my goals. I am excited at the prospect of learning, contributing, giving shape and making an impact in this upcoming and challenging fields.

References

[1] Marco Barreno, Blaine Nelson, Russell Sears, Anthony D. Joseph, and J. D. Tygar. Can machine learning be secure? In *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security*, ASIACCS '06, pages 16–25, New York, NY, USA, 2006. ACM.

[2] Edoardo Coronado-Barrientos, Guillermo Indalecio, and Antonio García-Loureiro. Improving performance of iterative solvers with the axc format using the intel xeon phi. *The Journal of Supercomputing*, Mar 2018.

[3] Floriana Esposito and Donato Malerba. Machine learning in computer vision. *Applied Artificial Intelligence*, 15(8):693–705, 2001.

[4] Ananth Y. Grama, Anshul Gupta, and Vipin Kumar. Isoefficiency: Measuring the scalability of parallel algorithms and architectures. *IEEE Parallel Distrib. Technol.*, 1(3):12–21, Aug 1993.

[5] Raphael Hunger. *Floating point operations in matrix-vector calculus.* Technische Universitt Mnchen - Associate Institute for Signal Processing, 2007.

[6] David J. Lary, Amir H. Alavi, Amir H. Gandomi, and Annette L. Walker. Machine learning in geosciences and remote sensing. *Geoscience Frontiers*, 7(1):3 – 10, 2016. Special Issue: Progress of Machine Learning in Geosciences.

[7] Raymond C. Rumpf and Javier Pazos. Synthesis of spatially variant lattices. *Opt. Express*, 20(14):15263–15274, Jul 2012.

[8] Raymond C. Rumpf, Javier J. Pazos, Jennefir L. Digaum, and Stephen M. Kuebler. Spatially variant periodic structures in electromagnetics. *Phil. Trans. R. Soc. A*, 373(2049):20140359, aug 2015. 00001.

[9] C. Sinclair, L. Pierce, and S. Matzner. An application of machine learning to network intrusion detection. In *Computer Security Applications Conference, 1999. (ACSAC '99) Proceedings. 15th Annual*, pages 371–377, 1999.

[10] Adi L Tarca, Vincent J Carey, Xue-wen Chen, Roberto Romero, and Sorin Drăghici. Machine Learning and its Applications to Biology. *PLoS computational biology*, 3(6):e116, 2007.

[11] Hai-Xiang Yang, Tian-Ruoand Lin. Isoefficiency analysis of cgls algorithm for parallel least squares problems. In Bob Hertzberger and Peter Sloot, editors, *High-Performance Computing and Networking*, pages 452–461, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.