MOST SIGNIFICANT RESEARCH CONTRIBUTIONS

(I) High Performance Computing:

My most significant contribution to High Performance Computing (HPC), is the ("s-step" conjugate gradient (s-CG) type or Krylov) scalable iterative methods for the solution of large sparse linear systems on HPC systems (HPCS). I have co-authored the first two pioneer papers on the theory and parallel architecture implementation of s-CG, listed below.

The s-step methods combine **s** (e.g. **s**=5) consecutive steps of standard methods into a single big "block" step. Compared to the standard ones, s-step methods have increased concurrency and enhanced data locality. The s-step methods were proved to reduce the required number of global communications (and synchronizations) versus standard methods (for equivalent progress in approximating the solution) by a factor of **1/s**. I have co-authored 10 journal papers that extended the s-step methods to computing spectra of matrices and also demonstrated their scalability in solving engineering problems (e.g. Computational Aerodynamics, Transportation) on HPCS. The s-step methods equivalence to the standard methods has been proved theoretically and also that the increased parallelism is achieved without a substantial increase in computational complexity.

These s-step methods have made a great impact on the scalability of Krylov iterative methods on HPCS. The so-called Chronopoulos-Gear CG (Chron-Gear CG) is the s-step CG (with s=1), which performs 1 global synchronization (gsync) per iteration, versus 2 gsync per iteration in the standard CG. This algorithm was shown to be scalable on modern (petascale) HPCS and GPUs and it has been incorporated in practical software for important engineering simulations. For example, the Parallel Ocean Program (POP) is simulation software for the earth's global ocean circulation model developed and maintained at Los Alamos National Laboratory, USA uses Chron-Gear CG as its core HPCS sparse solver. In the publication: "Early Evaluation of IBM BlueGene/P, Proceedings of the ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis (SC08), Austin, TX, November 15-21, 2008", POP using Chron-Gear CG maintained scalability and gave superior performance to standard iterative methods.

The s-step methods have impacted the design of "communication avoiding" (CA) scalable algorithms for exascale HPCS by leading research groups, e.g. (IEEE Fellows) J. Demmel (Univ. Berkeley, USA), J. Dongarra (Univ. Tenessee, USA). The DOE report ("Top Ten Exascale Report Challenges", DOE ASCAC Subcommitee Report, Feb. 10, 2014 https://science.energy.gov/~/media/ascr/ascac/pdf/meetings/20140210/Top10reportFEB14.pdf) mentions that the "s-step" Krylov iterative methods are suitable for scalable implementation of iterative methods in exascale HPCS.

(II) Load balancing in Distributed Systems:

I co-authored (with my PhD advisee Dr. Grosu) the first paper that models the load balancing problem of jobs mapping to servers in a Distributed System (DS) using Game Theory (GT) from economics. The paper (conference and journal version) that presented the new results are listed below.

Load balancing is indispensable for achieving high performance and scalability of large distributed systems (DS) such as grids or clouds. Effective algorithms for mapping jobs to the DS servers must optimize the DS load, be flexible to adapt to changes in the execution environment and provide some kind of fairness to the users.

The main approaches for jobs allocation in a DS are:

(Centralized): A single centralized decision maker finds the Global Optimal Solution (GOS) to the problem of jobs mapping to resources, which is optimal for the entire DS. This approach has been studied by many authors e.g. work by (IEEE fellow) H. Kameda et al,“ Optimal Load Balancing in Distributed Computer Systems”. This is not suitable for current DS (e.g. Grids, Clouds), where the individual users have requirements for fast execution of their jobs or even deadlines. This is specified in the Service Level Agreement (SLA) in the job submission.

(Distributed): The DS resources allocation problem can be modeled using GT, where users (players, represented by agents) compete for resources. The GT solution is an equilibrium, where no user can increase their payoffs by unilaterally deviating from it. This kind of equilibrium is called the Nash equilibrium. GT offers a viable modeling paradigm for studying this problem. Using this approach, one can obtain solutions in which the decision process is distributed and scalable. Moreover, the system's utilization is improved and the fairness of allocation is guaranteed.

Our paper proposed a distributed GT load balancing approach for jobs allocation in DS. Original distributed algorithms were presented that compute the solution and it was proved that the solution is a Nash equilibrium . These algorithms allocate computational resources to users with Quality of Service guarantees (QoS) (‘fair allocation’) in terms of delivered performance.

These results have impacted the design of techniques for ‘user based’ allocation of system resources in diverse areas such as grids and clouds, peer-to-peer systems, as evidenced by the citations of follow-up research (e.g. from googlescholar: https://scholar.google.gr/citations?user=vpaOim8AAAAJ&hl=en)

Leading research groups (e.g. by IEEE fellows: K. Hwang, S. Das, A. Zomaya) have published GT results similar to (and citing) our GT load balancing techniques.

(III) Loop Scheduling for Clusters (LSC):

My third most significant contribution is on “ Loop Self-Scheduling for Heterogeneous Clusters “. This work extends existing loop self-scheduling schemes from shared memory multiprocessing to distributed computing. I co-authored 10 IEEE conference and 5 journal papers on this topic. Our paper “Chronopoulos, A. T., R. Andonie, M. Benche & D. Grosu. "A class of loop self-scheduling for heterogeneous clusters." *Cluster Computing, 2001. Proceedings. 2001 IEEE International Conference on*, pp. 282-291. IEEE, 2001” is the first paper. This paper extended the loop self-scheduling schemes, which were designed or parallel computers, to schemes suitable for distributed computing. These results have been extended to loops with dependences and were also applied to grid and cloud computing. Many researchers have followed this approach and published new results for heterogeneous clusters, grid and clouds systems citing our work.

(IV) Other significant contributions are in Wireless Networks, Computational Fluid Aerodynamics, Traffic Engineering and Medical Applications.

MOST IMPORTANT PUBLICATIONS

(I) (HPC)

(1) http://www.cs.utsa.edu/~atc/pub/J1.pdf

A.T. Chronopoulos, C.W. Gear, "s-step iterative methods for symmetric linear systems", Journal of Computational and Applied Mathematics, 25(2), 153-168, 1989.

Summary: Author2 was my Ph.D. advisor.

The s-step CG for symmetric sparse linear problems was proposed and analyzed. It is a CA algorithm for sparse problems. Along with other s-step versions, it is useful for scalable implementation of iterative methods on HPCS. Leading researchers have worked on enhancing the s-step algorithms. For example, "A Residual Replacement Strategy for Improving the Maximum Attainable Accuracy of s-Step Krylov Subspace Methods, SIAM. J. Matrix Anal. & Appl., 35(1), 22–43, 2014", by E. Carson and J. Demmel.

Several other papers present implementions of s-CG on modern HPCS and GPUs showing more scalable performance compared to classical CG versions . For example, " Enhancing the Performance of Conjugate Gradient Solvers on Graphic Processing Units, IEEE Transactions on Magnetics, 47(5), pp.1162 – 1165, 2011, by M. Dehnavi et al." reports that the s-CG is 2.5- 3.8 times faster on NVIVIA GPUs than standard CG for a range of sparse linear systems from engineering problems.

(2) http://www.cs.utsa.edu/~atc/pub/J2.pdf

A.T. Chronopoulos and C.W. Gear, "On the efficient implementation of preconditioned s-step conjugate gradient methods on multiprocessors with memory hierarchy", Parallel Computing, 11(1), 37-53, 1989.

Summary: This is the first paper presenting efficient s-CG parallel programming on HPCS. The HPCS synchronizations for each s-CG step are combined into a single communication step. This reduces the algorithm communication complexity and makes s-SG scalable on exascale HPCS and GPUs.

(3) http://www.cs.utsa.edu/~atc/pub/J5.pdf

S.K. Kim, A.T. Chronopoulos,"A class of Lanczos-like algorithms implemented on parallel computers", Parallel Computing, 17(6-7), 763-778, 1991.

Summary: Author1 was my Ph.D. advisee. The paper presents the s-step Lanczos algorithm for computing eigenvalues (eigenvectors) of sparse symmetric matrices and the implementation on parallel computers. The s-step Lanczos is a CA algorithm and it is scalable on HPCS. Leading researchers have worked on enhancing the s-step algorithms. For example: Carson, Erin, and James W. Demmel. "Accuracy of the s-step Lanczos method for the symmetric eigenproblem in finite precision." SIAM Journal on Matrix Analysis and Applications 36, no. 2 (2015): 793-819.

(II) (DS)
(4) http://www.cs.utsa.edu/~atc/pub/J34.pdf

D. Grosu, A.T. Chronopoulos, "Noncooperative load balancing in distributed systems", Journal of Parallel and Distributed Computing, 65(9), 1022-1034, 2005.

The conference version of this paper is: http://www.cs.utsa.edu/~atc/pub/C12.pdf

D. Grosu, A. T. Chronopoulos, "A Game-Theoretic Model and Algorithm for Load Balancing in Distributed Systems", IEEE IPDPS 2002, pp. 146-153, 2002

Summary: Author1 was my Ph.D. advisee. In these publications, the mapping of jobs to resources allocation distributed systems (DS) was modeled as a noncooperative game with QoS for DS users. The game (Nash) equilibrium was shown to be the solution providing QoS guarantees for users. The paper inspired the design of algorithms for QoS allocation of resources in Grids or Clouds

(e.g. (by IEEE fellows) **(a)** K. Hwang, et al "Selfish grids: Game-theoretic modeling and nas/psa benchmark evaluation." IEEE Transactions on Parallel and Distributed Systems, 18, (5) (2007): 621-636. **(b)** Sajal K. Das et al "A game theory-based pricing strategy to support single/multiclass job allocation schemes for bandwidth-constrained distributed computing systems." IEEE Transactions on Parallel & Distributed Systems, 3 (2007): 289-306.)

(5) http://www.cs.utsa.edu/~atc/pub/C11.pdf

D. Grosu, A.T. Chronopoulos, M.Y. Leung, "Load Balancing in Distributed Systems: An Approach Using Cooperative Games", An Approach Using Cooperative Games", IEEE International Parallel and Distributed Processing Symposium, pp. 8-11, Fort Lauderdale, Florida, 15-19 April 2002.

Summary: This is the first paper that proposed cooperative GT to solve the problem of resource allocation in DS. The different decision makers cooperate in making decisions such that each of them will operate at its optimum. Based on the Nash Bargaining Solution (NBS), which provides a Pareto optimal and fair solution, they proposed an algorithm for computing the NBS for this cooperative load balancing game. The main goal of this load balancing scheme is to provide fairness to all the user jobs, i.e. jobs should experience the same expected response time independent of the allocated computer. This research spurred the development of several cooperative resource allocation schemes in grid and cloud computing. (e.g. the journal article A. Y. Zomaya, et al, "A Cooperative Game Framework for QoS Guided Job Allocation Schemes in Grids", IEEE Transactions on Computers, 57, (10), pp. 1413-1422, 2008).

(6) http://www.cs.utsa.edu/~atc/pub/J32.pdf

D. Grosu, A.T. Chronopoulos, "Algorithmic mechanism design for load balancing in distributed systems", IEEE Transactions on Systems, Man and Cybernetics, 34(1), 77-84, 2004.

Summary: In current (Grids, Clouds), resources belong to different self-interested agents or organizations. These agents may manipulate the parameters of the allocation algorithm in their own benefit and their selfish behavior may lead to severe performance degradation and poor efficiency. This paper proposed a mechanism design (MD) GT approach for resource allocation to users in DS. The paper led to the development of several MD approaches for resource allocation in grid and cloud computing.

(III) (LSC)
(7) https://rdcu.be/4qPx

Han, Yiming, and Anthony T. Chronopoulos. "Scalable Loop Self-Scheduling Schemes for Large-Scale Clusters and Cloud Systems." International Journal of Parallel Programming 45, no. 3 (2017): 595-611.

Summary: Author1 was my Ph.D. advisee. In this paper, we extended the previous distributed loop scheduling to a scalable approach. We implemented our scheme on a large scale homogeneous cluster and also on a cloud system heterogeneous environment using the message passing interface protocol.