2:00 – 3:00	Invited Talk "*Parallel Programming in the Age of Ubiquitous Parallelism*" *Keshav Pingali,* University of Texas, Austin **Abstract**: Multicore and manycore processors are now ubiquitous, but parallel programming remains as difficult as it was 30-40 years ago. During this time, our community has explored many promising approaches including functional and dataflow languages, logic programming, and automatic parallelization using program analysis and restructuring, but none of these approaches has succeeded except in a few niche application areas. In this talk, I will argue that these problems arise largely from the computation-centric foundations and abstractions that we currently use to think about parallelism. In their place, I will propose a novel data-centric foundation for parallel programming called the operator formulation in which algorithms are described in terms of actions on data. The operator formulation shows that a generalized form of data-parallelism called amorphous data-parallelism is ubiquitous even in complex, irregular graph applications such as mesh generation/refinement/partitioning and SAT solvers. Regular algorithms emerge as a special case of irregular ones, and many application-specific optimization techniques can be generalized to a broader context. The operator formulation also leads to a structural analysis of algorithms called TAO-analysis that provides implementation guidelines for exploiting parallelism efficiently. Finally, I will describe a system called Galois based on these ideas for exploiting amorphous data-parallelism on multicores and GPUs.
3:00 – 3:30	Invited Talk "*Graph Mining Using the Ex-MATE System*" Gagan Agrawal, The Ohio State University **Abstract**: Scalable and convenient programming models for developing graph analysis applications continue to be an open question. Though more specialized APIs have been proposed recently, the Map-reduce framework has been widely used as the infrastructure for processing large-scale datasets in various domains, including graph analysis. Recent work has shown that an alternate API MATE (Mapreduce with an AlTErnate API), where a reduction object is explicitly maintained and updated, reduces memory requirements and can significantly improve performance for many applications. This approach is promising for graph mining, though it also requires that large (disk-based) reduction objects can be supported. This talk will describe a system, Extended MATE or Ex-MATE, which supports this alternate API with reduction objects of arbitrary sizes. We develop support for managing disk-resident reduction objects and updating them efficiently. We evaluate our system using three graph mining applications and compare the performance to that of PEGASUS, a graph mining system implemented based on the original map-reduce API and its Hadoop implementation. Our results on a cluster with 128 cores show that for all three applications, our system outperforms PEGASUS, by factors ranging between 9 and 35.
3:30 – 4:00	Coffee Break
4:00 – 4:30	"*Towards Scalable Optimal Sequence Homology Detection*" Jeff Daily, Sriram Krishnamoorthy and Ananth Kalyanaraman Pacific Northwest National Laboratory, and Washington State University

	Abstract: The field of bioinformatics and computational biology is experiencing a data revolution — experimental techniques to procure data have increased in throughput, improved in accuracy and reduced in costs. This has spurred an array of high profile sequencing and data generation projects. While the data repositories represent untapped reservoirs of rich information critical for scientific breakthroughs, the analytical software tools that are needed to analyze large volumes of such sequence data have significantly lagged behind in their capacity to scale. In this paper, we address homology detection, which is a fundamental problem in large-scale sequence analysis with numerous applications. We present a scalable framework to conduct largescale optimal homology detection on massively parallel supercomputing platforms. Our approach employs distributed memory work stealing to effectively parallelize optimal pairwise alignment computation tasks. Results on 120,000 cores of the Hopper Cray XE6 supercomputer demonstrate strong scaling and up to 2:42 x 10^7 optimal pairwise sequence alignments computed per second (PSAPS), the highest reported in the literature.
4:30 – 5:00	"*Towards Highly Scalable X10 Based Spectral Clustering*" Hidefumi Ogata, Miyuru Darayathna and Toyotaro Suzumura Tokyo Institute of Technology, and IBM Research, Tokyo **Abstract**: Large graph analysis has become a widely studied area in recent years. Clustering is one of the most important types of analysis that has versatile applications such as community detection in social networks, image segmentation, graph partitioning, etc. However, existing clustering algorithms do not intend for large scale graphs. To solve this problem, we implemented spectral clustering in X10 that is a programming language aimed for developing highly scalable applications on Post-Petascale supercomputers. Our spectral clustering is based on the algorithm proposed by Shi and Malik. After evaluating scalability and precision, we found that our implementations are scalable in terms of execution time and precise for analyzing real data.
5:00 – 5:30	"*External Memory based Distributed Generation of Massive Scale Social Networks on Small Clusters*" Sandeep Gupta **Abstract**: Small distributed systems are limited by their main memory to generate massively large graphs. Trivial extensions to current graph generators to utilize external memory leads to large amount of random I/O hence do not scale with size. In this work we offer a technique to generate massive scale graphs on small cluster of compute nodes with limited main memory. We develop several distributed and external memory algorithms, primarily, shuffle, relabel, redistribute, and, compressed-sparse-row (CSR) convert. The algorithms are implemented in MPI/pthread model to help parallelize the operations across multicores within each core. Using our scheme it is feasible to generate a graph of size 238 nodes (scale 38) using only 64 compute nodes. This can be compared with the current scheme would require at least 8192 compute node, assuming 64GB of main memory. Our work has broader implications for external memory graph libraries such as STXXL and graph processing on SSD-based supercomputers such as Dash and Gordon