

Guest Editors' Introduction: Special Section on Many-Task Computing

Ioan Raicu, *Member, IEEE*, Ian T. Foster, *Member, IEEE*, and Yong Zhao, *Member, IEEE*

1 INTRODUCTION

IT is our honor to serve as guest editors of this special section of the *IEEE Transactions on Parallel and Distributed Systems (TPDS)* on many-task computing (MTC). This section focuses on the methods required to manage and execute large multiple program multiple data (MPMD) computations on large clusters, grids, clouds, and supercomputers. We are pleased to present 10 high-quality contributions chosen from 42 submissions, on resource management, data-intensive computing, applications, and MTC on supercomputers, grids, and clouds.

We introduce the term many-task computing (MTC) [2] for computations that bridge the gap between high-performance computing (HPC) and high-throughput computing (HTC) [1]. MTC differs from HTC in its emphasis on using many computing resources over short periods of time to accomplish many computational tasks (both dependent and independent), for which primary metrics are measured in seconds (e.g., FLOPS, tasks/sec., MB/s I/O rates), as opposed to jobs per month. MTC computations comprise multiple distinct activities, coupled via files, shared memory, or message passing. Tasks may be small or large, uniprocessor or multiprocessor, or compute-intensive or data-intensive. The set of tasks may be static or dynamic, homogeneous or heterogeneous, or loosely coupled or tightly coupled. The number of tasks, quantity of computing, and volumes of data may be large.

Today's HPC systems are a viable platform for MTC [3], but large MTC applications can stress HPC hardware and software. Challenges include local resource manager scalability and granularity, efficient utilization of raw hardware, parallel file system contention and scalability, data management, I/O management, reliability at scale, application scalability, and understanding the limitations of HPC systems in order to identify good candidate MTC applications [4]. MTC applications can also be executed on cloud systems, but face other challenges there, for example, relating to internode communication performance.

Three recent MTC workshops (MTAGS, <http://dsl.cs.uchicago.edu/MTAGS10/>) and this special section attracted 142 abstracts and 110 paper submissions, from which 41 papers were accepted. Papers covered resource management, data-intensive computing, applications, and

MTC on supercomputers, grids, and clouds. More than 1,000 people have participated as coauthors, program committee members, reviewers, and attendees in these venues. We are well beyond a critical mass for a new, thriving community, which is quickly expanding.

2 IN THIS SECTION

It was a challenge to select just 10 of the many high-quality submissions for inclusion here. We thank the hundreds of reviewers for their thoughtful work. We briefly introduce the accepted articles in the following.

"A Data Throughput Prediction and Optimization Service for Widely Distributed Many-Task Computing," by Dengpan Yin, Esma Yildirim, Sivakumar Kulasekaran, Brandon Ross, and Tevfik Kosar, presents the design and implementation of an application-layer data throughput prediction and optimization service for MTC in distributed environments. This service uses parallel TCP streams to improve end-to-end data transfer throughput. The authors implement this new service in the Stork Data Scheduler, where the prediction points can be obtained using Iperf and GridFTP.

"ThrifStore: Finessing Reliability Trade-Offs in Replicated Storage Systems," by Abdullah Gharaibeh, Samer Al-Kiswani, and Matei Ripeanu, describes a storage architecture that seeks to provide the reliability and access performance characteristics of a high-end system at reduced cost. ThrifStore uses volatile, aggregated storage to provide a high-throughput frontend, and dedicated low-bandwidth durable storage to enable the restoration of data lost by volatile nodes. The authors evaluate the impact of system characteristics (e.g., bandwidth limitations on the durable and the volatile nodes) and design choices (e.g., replica placement scheme) on data availability and associated system costs (e.g., maintenance traffic).

"Multicloud Deployment of Computing Clusters for Loosely Coupled MTC Applications," by Rafael Moreno-Vozmediano, Ruben S. Montero, and Ignacio M. Llorente, describes the deployment of a computing cluster on top of a multicloud infrastructure, and the use of this cluster for MTC applications. Provisioning cluster nodes with resources from different clouds can improve cost-effectiveness, or enable high availability. The authors evaluate the scalability, performance, and cost of different configurations of a Sun Grid Engine cluster, deployed on a multicloud infrastructure spanning a local data-center and three different cloud sites: Amazon EC2 Europe, Amazon EC2 USA, and ElasticHosts.

"Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing," by Alexandru Iosup, Simon Ostermann, M. Nezh Yigitbasi, Radu Prodan, Thomas Fahringer, and Dick H.J. Epema, analyzes the performance of four different commercial cloud services for scientific

- I. Raicu is with the Illinois Institute of Technology, Chicago, IL 60616, and Argonne National Laboratory, Argonne, IL 60439. E-mail: iraicu@cs.iit.edu.
- I.T. Foster is with the University of Chicago, Chicago, IL 60637, and Argonne National Laboratory, Argonne, IL 60439. E-mail: foster@anl.gov.
- Y. Zhao is with the University of Electronic Science and Technology of China, Chengdu, China. E-mail: yongzh04@gmail.com.

For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org.

computing workloads. The authors compare through trace-based simulation the performance characteristics and cost models of clouds and other scientific computing platforms, for both general and MTC-based scientific computing workloads. Their results indicate that current clouds need an order of magnitude in performance improvement to be useful to the scientific community, and show which improvements should be considered first to address this performance gap.

“Design and Evaluation of Multiple-Level Data Staging for Blue Gene Systems,” by Florin Isaila, Javier Garcia Blas, Jesus Carretero, Robert Latham, and Robert Ross, presents a scalable parallel I/O software system designed to hide the latency of file system accesses to applications on massively parallel platforms. The authors’ solution leverages the hierarchy of networks involved in file accesses to overlap computation, file I/O-related communication, and file system access. They investigate a two-level hierarchy for BlueGene systems, and demonstrate significant performance improvements via a high degree of overlap between computation, communication, and file I/O.

“Parameter Exploration in Science and Engineering Using Many-Task Computing,” by David Abramson, Blair Bethwaite, Colin Enticott, Slavisa Garic, and Tom Peachey, presents Nimrod/K, a set of components and a new runtime engine for the Kepler workflow engine. Nimrod/K provides an execution architecture based on the tagged dataflow concepts developed in the 1980s for highly parallel machines. Nimrod/K provides “Directors” to support task execution orchestration as well as “Actors” that facilitate various modes of parameter exploration. The authors evaluate the power of Nimrod/K to solve real problems in cardiac science.

“Toward Efficient and Simplified Distributed Data Intensive Computing,” by Yunhong Gu and Robert Grossman, describes the design and implementation of a distributed file system called Sector and an associated programming framework called Sphere that processes the data managed by Sector in parallel. The authors describe the directives Sphere supports to improve data locality, and present experimental studies that show that the Sector/Sphere system is about two to four times faster than Hadoop.

“Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud,” by Daniel Warneke and Odej Kao, discusses opportunities and challenges for efficient parallel data processing in clouds and presents Nephele, a data processing framework that explicitly exploit the dynamic resource allocation offered by today’s IaaS clouds for both task scheduling and execution. The authors perform evaluation of MapReduce-inspired processing jobs on an IaaS clouds and compare their results to Hadoop.

“Cloud Technologies for Bioinformatics Applications,” by Jaliya Ekanayake, Thilina Gunarathne, and Judy Qiu, describes experience in applying two cloud technologies—Apache Hadoop and Microsoft DryadLINQ—to two bioinformatics applications, a pairwise Alu sequence alignment application and a sequence assembly program. They use these applications to compare the performance of these cloud technologies and traditional MPI. They also analyze the effect of inhomogeneous data on cloud scheduling, and compare the performance of clouds on virtual and non-virtual platforms.

“Many Task Computing for Real-Time Uncertainty Prediction and Data Assimilation in the Ocean,” by Constantinos Evangelinos, Pierre F.J. Lermusiaux, Jinshan Xu, Patrick J. Haley Jr., and Chris N. Hill, reports on a project that seeks to accelerate ocean uncertainty prediction based on the Error Subspace Statistical Estimation (ESSE) approach. ESSE uses

dynamic data intensive heterogeneous workflows. The authors study a distributed ESSE workflow on a medium-size cluster, examining the I/O patterns exhibited and throughputs achieved by its components as well as the overall ensemble performance seen in practice. They also study the performance/usability challenges of employing Amazon EC2 and TeraGrid to augment ESSE ensembles.

ACKNOWLEDGMENTS

This work was supported in part by the US Department of Energy (DOE) under Contract DE-AC02-06CH11357 and the US National Science Foundation grant NSF-0937060 CIF-72.

REFERENCES

- [1] M. Livny, J. Basney, R. Raman, and T. Tannenbaum, “Mechanisms for High Throughput Computing,” *SPEEDUP J.*, vol. 1, no. 1, 1997.
- [2] I. Raicu, I. Foster, and Y. Zhao, “Many-Task Computing for Grids and Supercomputers,” *Invited Paper, Proc. IEEE Workshop Many-Task Computing on Grids and Supercomputers (MTAGS '08), collocated with Proc. IEEE/ACM Supercomputing '08*, 2008.
- [3] I. Raicu, I. Foster, M. Wilde, Z. Zhang, Y. Zhao, A. Szalay, P. Beckman, K. Iskra, P. Little, C. Moretti, A. Chaudhary, and D. Thain, “Middleware Support for Many-Task Computing,” *Cluster Computing, The J. of Networks, Software Tools and Applications*, 2010.
- [4] M. Wilde, I. Foster, K. Iskra, P. Beckman, Z. Zhang, A. Espinosa, M. Hategan, B. Clifford, and I. Raicu, “Parallel Scripting for Applications at the Petascale and Beyond,” *Computer*, special issue on extreme scale computing, Nov. 2009.



Ioan Raicu is an assistant professor in the Department of Computer Science at Illinois Institute of Technology (IIT), as well as a guest research faculty in the Math and Computer Science Division at Argonne National Laboratory. He is also the founder and director of the Data-Intensive Distributed Systems Laboratory at IIT. His research is in the broad area of distributed systems, with particular interests in resource management at extreme scale focusing on many-task computing, data intensive computing, cloud computing, and grid computing. His work has been funded by the NASA ARC, US DOE ASCR, US NSF/CRA CIFellows, and NSF OCI/CAREER. He is a member of the IEEE.



Ian T. Foster is a Distinguished Fellow at Argonne National Laboratory, the Arthur Holly Compton Professor in the Department of Computer Science at the University of Chicago, and director of the Computation Institute, a joint unit of the University of Chicago and Argonne. His research interests include distributed, parallel, and data-intensive computing, and the application of these methods to scientific problems. His honors include the Lovelace Medal of the British Computer Society and the IEEE Tsutomu Kanai award. He is a fellow of the American Association for the Advancement of Science, ACM, and British Computer Society. He is a member of the IEEE.



Yong Zhao is a professor in the School of Computer Science and Engineering, University of Electronic Science and Technology of China. Before joining the university, he worked at Microsoft on business intelligence projects in cloud computing. He was the key designer of the Swift parallel scripting system, a programming tool for fast, scalable, and reliable loosely-coupled parallel computation. His research areas are in cloud computing, many-task computing, and data intensive computing, with special interests in providing resource management, workflow management, high level language, and scheduling support for large scale computations in clouds and grids. He is a member of the IEEE.