# Managing and Executing Loosely-Coupled Large-Scale Applications on Clusters, Grids, and Supercomputers

[1]Ioan Raicu, [2]Yong Zhao, [1,3,4]Ian Foster, [3,4]Mike Wilde,
[3]Zhao Zhang, [3]Ben Clifford, [3]Mihael Hategan, [3]Sarah Kenny
iraicu@cs.uchicago.edu, yozha@microsoft.com, foster@mcs.anl.gov, wilde@mcs.anl.gov,
zhaozhang@uchicago.edu, benc@ci.uchicago.edu, hategan@mcs.anl.gov, skenny@uchicago.edu

[1]Department of Computer Science, University of Chicago
[2]Microsoft Corporation
[3]Computation Institute, Argonne National Laboratory & University of Chicago
[4]Mathematics and Computer Science Division, Argonne National Laboratory

## Overview

With the advances in e-Sciences and the growing complexity of scientific analyses, more and more scientists and researchers are relying on workflow systems for process coordination, derivation automation, provenance tracking, and bookkeeping. Workflows are not a new concept and have been around for decades [1][3][6][7][2]. In the scientific community, several systems for scientific programming and computation have emerged [4][5][12][14].

We present Swift [14], a system that bridges scientific workflows with parallel computing. It is a parallel programming tool for rapid and reliable specification, execution, and management of large-scale science workflows. Swift takes a structured approach to workflow specification, scheduling and execution. It consists of a simple scripting language, SwiftScript, for concise specifications of complex parallel computations based on dataset typing and iterations, and dynamic dataset mappings for accessing large-scale datasets represented in diverse data formats. The runtime system relies on the CoG Karajan [9] workflow engine for efficient scheduling and load balancing, and it integrates the Falkon [10] light-weight task execution service for optimized task throughput and resource efficiency when executing many independent jobs on large compute clusters. It combines three techniques to achieve this goal: (1) multi-level scheduling techniques to enable separate treatments of resource provisioning and the dispatch of user tasks to those resources [10][15]; (2) a streamlined task dispatcher able to achieve order-of-magnitude higher task dispatch rates than conventional schedulers [10]; and (3) performs data caching and uses a data-aware scheduler to leverage the co-located computational and storage resources to minimize the use of shared storage [11][13][16].

We believe the synergy found between Swift and Falkon can address issues such as scalability, reliability, scheduling and monitoring, data management, collaboration, workflow provenance, and workflow evolution. The science community is demanding both specialized, domain-specific languages to improve productivity and efficiency in writing concurrent programs and coordination tools, and generic platforms and infrastructures for the execution and management of large-scale scientific applications, where scalability and performance are critical.

Both Swift [17] and Falkon [18] have been Globus Incubator [19] projects since 2007. They have both been used in a variety of environments from clusters (i.e. TeraPort [22]), to multi-site Grids (i.e. Open Science Grid [24], TeraGrid [21]), to specialized large machines (SiCortex [22]), and supercomputers (i.e. IBM BlueGene/P [23]). Large-scale applications from many domains (i.e. astronomy [10][26], medicine [8][10][28], chemistry [25], molecular dynamics [29], and economic modeling [27][30]) have been run at scales of tens of thousands of jobs on thousands of processors, with an order of magnitude larger scale on the horizon. Both projects are being actively developed primarily at University of Chicago, in the CS Department and Computational Institute with funding from NSF, DOE, and NASA.

## References

[1] Ahuja, S., Carriero, N., and Gelernter, D. “Linda and Friends”, IEEE Computer 19 (8), 1986, pp. 26-34.

[2] Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., Trickovic, I., Weerawarana, S. “Business Process Execution Language for Web Services, Version 1.1. Specification”, BEA Systems, IBM Corp., Microsoft Corp., SAP AG, Siebel Systems (2003).

[3] Barbacci, M. R., Weinstock, C. B., Doubleday, D. L., Gardner, M. J., Lichota, R. W. “Durra: A Structure Description Language for Developing Distributed Applications”, Software Engineering Journal, IEEE, pp. 83-94, 1996.

[4] Dean, J. and Ghemawat, S. "MapReduce: Simplified data processing on large clusters." OSDI, 2004.
[5] The Fortress Programming Language, http://fortress.sunsource.net/, 2008.
[6] Foster, I. "Compositional Parallel Programming Languages", ACM Transactions on Programming Languages and Systems 18 (4), 1996, pp. 454-476.
[7] Foster, I. and Taylor, S. Strand. "New Concepts in Parallel Programming. Prentice Hall, Englewood Cliffs, N. J. 1990.
[8] Functional MRI Data Center, http://www.fMRIDC.org, 2008.
[9] Laszewski, G.v., Hategan, M. and Kodeboyina, D. "Java CoG Kit Workflow", in Taylor, I.J., Deelman, E., Gannon, D.B. and Shields, M. eds. Workflows for eScience, 2007, 340-356.
[10] Raicu, I., Zhao, Y., Dumitrescu, C., Foster, I., Wilde, M. "Falkon: a Fast and Lightweight Task Execution Framework", IEEE/ACM Supercomputing (SC'07), November 2007.
[11] Raicu, I., Zhao, Y., Foster, I., Szalay, A. "A Data Diffusion Approach to Large Scale Scientific Exploration," Microsoft eScience Workshop at RENCI 2007.
[12] Star-P, http://www.interactivesupercomputing.com, 2008.
[13] Szalay, A., Bunn, A.., Gray, J., Foster, I., Raicu, I. "The Importance of Data Locality in Distributed Computing Applications", NSF Workflow Workshop 2006.
[14] Zhao, Y., Hategan, M., Clifford, B., Foster, I., Laszewski, G.v., Raicu, I., Stef-Praun, T., Wilde, M. "Swift: Fast, Reliable, Loosely Coupled Parallel Computation", IEEE International Workshop on Scientific Workflow (SWF07), Collocated with SCC 2007.
[15] Raicu, I., Dumitrescu, C., Foster, I. "Dynamic Resource Provisioning in Grid Environments", TeraGrid Conference 2007.
[16] Raicu, I., Zhao, Y., Foster, I., Szalay, A. "Accelerating Large-scale Data Exploration through Data Diffusion", to appear at IEEE/ACM International Workshop on Data-Aware Distributed Computing 2008.
[17] Swift Globus Incubator, http://dev.globus.org/wiki/Incubator/Swift
[18] Falkon Globus Incubator, http://dev.globus.org/wiki/Incubator/Falkon
[19] Globus Incubator Management, http://dev.globus.org/wiki/Incubator/Incubator_Management
[20] TeraPort Cluster, http://teraport.uchicago.edu/, 2007.
[21] Catlett, C., et al. "TeraGrid: Analysis of Organization, System Architecture, and Middleware Enabling New Types of Applications," HPC 2006.
[22] SiCortex, http://www.sicortex.com/, 2008.
[23] IBM BlueGene/P (BG/P), http://www.research.ibm.com/bluegene/, 2008.
[24] Open Science Grid (OSG), http://www.opensciencegrid.org/, 2008.
[25] Zhao, Y., Raicu, I., Foster, I., Hategan, M., Nefedova, V., Wilde, M. "Realizing Fast, Scalable and Reliable Scientific Computations in Grid Environments", Grid Computing Research Progress, Nova Pub. 2008.
[26] Jacob, J.C., et al. "The Montage Architecture for Grid-Enabled Science Processing of Large, Distributed Datasets." Earth Science Technology Conference 2004.
[27] Stef-Praun, T., Madeira, G., Foster, I., and Townsend, R. "Accelerating solution of a moral hazard problem with Swift", e-Social Science, 2007.
[28] Stef-Praun, T., Clifford, B., Foster, I., Hasson, U., Hategan, M., Small, S., Wilde, M and Zhao,Y. "Accelerating Medical Research using the Swift Workflow System", Health Grid , 2007.
[29] Moustakas, D.T., Therese Lang, P.T., Pegg, S., Pettersen, E., Kuntz, I.D., Broojimans, N., Rizzo, R.C. "Development and Validation of a Modular, Extensible Docking Program: DOCK 5", J. Comput. Aided Mol. Des., 2006, 20, 601-619.
[30] Hanson, D. "Enhancing Technology Representations within the Stanford Energy Modeling Forum (EMF) Climate Economic Models", ACEEE Workshop on Energy and Economic Policy Models: A Reexamination of Fundamentals, 2006.

## Presenting Author Bio

Ioan Raicu is a Ph.D. Candidate in the Distributed Systems Laboratory in the Computer Science Department at University of Chicago, under the guidance of Dr. Ian Foster. His main area of research is in Grid Computing with an emphasis on resource management in large scale distributed systems and grid computing. Within resource management, he is particularly interested in efficient task dispatch and execution systems, resource provisioning, data management, scheduling, and performance evaluations. His work has been funded by the NASA Ames Research Center Graduate Student Research Program, as well as the DOE Office of Advanced Scientific Computing Research.