# A Data Diffusion Approach to Large Scale Scientific Exploration

**Ioan Raicu[*], Yong Zhao[*], Ian Foster[#*+], Alex Szalay[-]**

*{iraicu,yongzh }@cs.uchicago.edu, foster@mcs.anl.gov, szalay@jhu.edu*

*[*]Department of Computer Science, University of Chicago, IL, USA*
*[+]Computation Institute, University of Chicago & Argonne National Laboratory, USA*
*[#]Math & Computer Science Division, Argonne National Laboratory, Argonne IL, USA*
*[-] Department of Physics and Astronomy, The Johns Hopkins University, Baltimore MD, USA*

## Abstract

Scientific and data-intensive applications often require exploratory analysis on large datasets. Such analysis is often carried out on large scale distributed resources where data locality is crucial in achieving high system throughput and performance. [1] We propose a "data diffusion" approach that acquires resources for data analysis dynamically, schedules computations as close to data as possible, and replicates data in response to workloads. As demand increases, more resources are acquired and "cached" to allow faster response to subsequent requests; resources are released when demand drops. This approach can provide the benefits of dedicated hardware without the associated high costs, depending crucially on the nature of application workloads and the performance characteristics of the underlying infrastructure.

This data diffusion concept is reminiscent of cooperative Web-caching [2] and peer-to-peer storage systems [3]. Other data-aware scheduling approaches assume static or dedicated resources [4, 5, 6, 7, 8, 9, 10], which can be expensive and inefficient (in terms of resource utilization) if load varies significantly, where our dynamic resource allocation alleviates the problem. The challenges to our approach are that we need to co-allocate storage resources with computation resources in order to enable the efficient analysis of possibly terabytes of data without prior knowledge of the characteristics of application workloads.

To explore the proposed data diffusion, we have developed Falkon [11, 12], which provides dynamic acquisition and release of resources ("executors") and the dispatch of analysis tasks to those executors. We have extended Falkon to allow executors to cache data to local disks, and perform task dispatch via a data-aware scheduler. The integration of Falkon and the Swift parallel programming system [13] provides us with access to a large number of applications from astronomy [14, 15, 16, 13], astro-physics, medicine [13], and other domains, with varying datasets, workloads, and analysis codes.

## References

[1] A. Szalay, J. Bunn, J. Gray, I. Foster, I. Raicu. "The Importance of Data Locality in Distributed Computing Applications", NSF Workflow Workshop 2006.
[2] R. Lancellotti, M. Colajanni, B. Ciciani, "A Scalable Architecture for Cooperative Web Caching", Proceedings of Workshop in Web Engineering, Networking, 2002.
[3] R. Hasan, Z. Anwar, W. Yurcik, L. Brumbaugh, R. Campbell. "A Survey of Peer-to-Peer Storage Techniques for Distributed File Systems", International Conference on Information Technology: Coding and Computing (ITCC'05), 2005.
[4] O. Tatebe, N. Soda, Y. Morita, S. Matsuoka, S. Sekiguchi. "Gfarm v2: A Grid file system that supports high-performance distributed and parallel data computing", Conference on Computing in High Energy and Nuclear Physics (CHEP04), 2004.
[5] W. Xiaohui, W.W. Li, O. Tatebe, X. Gaochao, H. Liang, J. Jiubin. "Implementing data aware scheduling in Gfarm using LSF scheduler plugin mechanism", International Conference on Grid Computing and Applications (GCA'05), 2005.
[6] M. Ernst, P. Fuhrmann, M. Gasthuber, T. Mkrtchyan, C. Waldman. "dCache, a distributed data storage caching system," Conference on Computing in High Energy and Nuclear Physics (CHEP), 2001.
[7] P. Fuhrmann. "dCache, the commodity cache," 21st IEEE Conference on Mass Storage Systems and Technologies, 2004.
[8] F. Schmuck and R. Haskin, "GPFS: A Shared-Disk File System for Large Computing Clusters," In Proceedings of the First Conference on File and Storage Technologies (FAST), 2002.
[9] S. Ghemawat, H. Gobioff, S.T. Leung. "The Google file system," 19th ACM SOSP, 2003.
[10] J. Dean, S. Ghemawat. "MapReduce: Simplified Data Processing on Large Clusters", OSDI'04: Sixth Symposium on Operating System Design and Implementation, 2004.
[11] I. Raicu, Y. Zhao, C. Dumitrescu, I. Foster, M. Wilde. "Falkon: a Fast and Light-weight tasK executiON framework", to appear at IEEE/ACM International Conference for High Performance Computing, Networking, Storage, and Analysis (SC07), 2007.
[12] I. Raicu, C. Dumitrescu, I. Foster. "Dynamic Resource Provisioning in Grid Environments", TeraGrid Conference 2007.
[13] Y. Zhao, M. Hategan, B. Clifford, I. Foster, G. von Laszewski, I. Raicu, T. Stef-Praun, M. Wilde. "Swift: Fast, Reliable, Loosely Coupled Parallel Computation," IEEE Workshop on Scientific Workflows, 2007.
[14] I. Raicu, I. Foster, A. Szalay. "Harnessing Grid Resources to Enable the Dynamic Analysis of Large Astronomy Datasets", IEEE/ACM International Conference for High Performance Computing, Networking, Storage, and Analysis (SC06), 2006.
[15] I. Raicu, I. Foster, A. Szalay, G. Turcu. "AstroPortal: A Science Gateway for Large-scale Astronomy Data Analysis", TeraGrid Conference 2006.
[16] J.C. Jacob, et al. "The Montage Architecture for Grid-Enabled Science Processing of Large, Distributed Datasets," Earth Science Technology Conference 2004