Grid-Enabled, High-performance Microscopy Image Analysis

Patrick Widener[1], Wenjin Chen[2], Fusheng Wang[1], Lin Yang[2], Jun Hu[2], Vicky Chu[2], Joel H. Saltz[1], David J. Foran[2], Tahsin Kurc[1]
[1]Center for Comprehensive Informatics,
Emory University, Atlanta, GA
[2]Center for Biomedical Imaging & Informatics,
UMDNJ-Robert Wood Johnson Medical School, New Brunswick, NJ

Abstract. Biomedical images are intrinsically complex with each domain and modality often requiring specialized knowledge to accurately render diagnosis and plan treatment. Having a general software framework, which provides access to high-performance resources, can serve to facilitate high-throughput investigations of micro-scale features as well as algorithm design, development and evaluation. In this paper we describe the requirements and challenges of supporting microscopy analyses of large datasets of high-resolution biomedical images. We present high-performance computing approaches for storage and retrieval of image data, image processing, and management of analysis results for additional explorations. We describe the implementation of a Grid-enabled analysis component for quantitative investigation of tissue microarrays and present empirical performance results.

Keywords: Microscopy imaging, tissue microarrays, large scale analysis, data management

1 Introduction

High-resolution biomedical imaging provides a valuable tool for scientists to investigate the structure and function of biological systems at cellular, sub-cellular, and molecular levels. Information obtained at these scales can provide biomarkers for better prognostic accuracy and lead to new insights into the underlying mechanisms of disease progression. For example, current therapies and treatment regimens for breast cancer are based upon classification strategies, which are limited, in terms of their capacity to identify specific tumor groups exhibiting different clinical and biological profiles. Tumors can be analyzed using tissue microarrays (TMAs) to confirm clinico-pathologic correlations, which have been established with whole tissue sections [1]. The process could involve extracting and assimilating phenotypic and molecular features from images from multiple groups of patients, comparing and correlating this information with the information about the patient under study, and classifying the condition of the patient based on the analysis results.

The field of genomics research has been transformed with advances in high-throughput instruments which can generate large volumes of readings quickly. We are observing a similar trend in biomedical imaging. Advanced microscopy scanners are

capable of rapidly imaging glass slides (usually within several minutes) in their entirety at high-resolution. The continuing increase in speed and resolution of imaging instruments will usher in high-throughput, high-resolution micro-scale feature analysis. Research studies will be able to collect large numbers of high-resolution images from TMAs and whole slides from cohorts of patients. In these studies, multiple images will be captured from the same tissue specimen using different stains and/or imaging modalities; images from the same patient will also be captured at multiple time points in a treatment or observation phase.

As advanced microscope scanners continue to gain favor in research and clinical settings, microscopy imaging holds great potential for highly detailed examination of disease morphology and for enhancing anatomic pathology. In order to realize this potential, researchers need high-performance systems to handle data and computation complexity of the high-throughput micro-scale feature analysis process. In addition, the systems should be able to leverage Grid computing both for taking advantage of distributed resources and for supporting sharing of analytical resources as well as image datasets and analysis results in collaborative studies.

In this paper we describe the requirements and challenges of supporting micro-scale analyses of large datasets of high-resolution biomedical images. We argue that an integrated software framework to address the requirements should provide support for researchers to efficiently store and retrieve large volumes of image data (image storage and management component), execute complex analyses on image datasets (analysis component), and manage, query, and integrate vast amounts of analysis results (results management component). We present high-performance computing approaches for these three components. We describe the implementation of a Grid-enabled analysis component for quantitative investigation of tissue microarrays and present empirical performance results.

2 Requirements and Challenges of Microscopy Image Analysis

The first challenge to enabling high-throughput, high-resolution analyses of micro-scale features is the fact that images obtained from contemporary scanners are very large. For example, scanning a 15 mm x 15 mm section of tissue results in a billion pixels, or 3 GB of RGB data, and image sizes of 10 GB or more are common for images obtained from a larger tissue section or scanned at higher resolutions. Large clinical studies may recruit hundreds of participants; studies on animal models of disease, such as those that make use of models based on mice, may have hundreds of specimens. These studies will generate thousands of images (e.g., studies on morphological changes in mouse placenta can generate up to thousand slides from a single placenta specimen to form a 3-dimensional representation of the placenta) over the course of the study, resulting in multiple terabytes of image data.

The second challenge is the computational complexity of analyzing images. Operations on image data may range from relatively simple intensity/color correction tasks to complex segmentation and feature extraction operations. A researcher may combine these operations into analysis workflows for characterization of micro-scale structures. In addition, multiple workflows composed of a number of interrelated algorithms may be needed to carry out segmentation and classification. One of the reasons for executing multiple workflows is to facilitate algorithm development and

evaluation. The composition of analysis pipelines, the values of input parameters of analysis methods, and the characteristics of input datasets all affect the analysis results and the accuracy of the analysis outcome. Given the large volume of images and the vast number of features, it would not be feasible to manually inspect each image for every feature and fine-tune analysis pipelines. A workable approach is to apply a few hundred variations of analysis pipelines and input parameter values on a few hundred images. Systematic management, comparison, and analysis of the results from these experiments can weed out bad choices (for the intended study), reducing the number of potentially high-quality pipelines to 10-20. These pipelines are then executed on the whole collection of images. Besides increasing accuracy of and confidence in analysis results, an image dataset can be analyzed multiple times with different algorithms and pipelines to detect and extract different types of features.

Another challenge is the management of huge amounts of semantically complex analysis results. Image markups can be either geometric shapes or image masks; annotations can be calculations, observations, disease inferences or external annotations. Many of the analytical imaging results are anatomic objects such as lesions, cells, nuclei, blood vessels, etc. Features such as volume, area, elongation, are extracted from these objects, and the objects are classified (annotated) based on feature characteristics and domain knowledge. Annotations may draw from one or more domain ontologies, resulting in a semantically rich environment. An example query from one of our studies is ”Search for objects with an observation concept (astrocytoma), but also expand to include all its subclass concepts (gliosarcoma and giant cell glioblastoma).” Spatial relationships among the objects are often important to understanding the biomedical characteristics of biology systems. Thus, additional annotations can be derived from existing annotations and spatial relationships among structures and features -- common spatial relationships include containment, intersection or overlap, distance between objects, and adjacency relationships. Large image datasets and complex analyses result in large volumes of metadata about objects, markups, and features computed for each anatomic object, and semantic annotations (about cell types, genomic information associated with cells, etc). For instance, segmentation of whole slide images from brain tumor specimens can lead to 100,000 to 1,000,000 cells in each virtual slide. Classification categories include classes of brain tumor cells, normal brain cell categories, macrophages, endothelial cells, etc. Various markers can be used to identify possible cancer stem cells, mutations, along with markers designed to identify blood vessels. The process of classifying a given cell may involve 10-100 shape, texture, and stain quantification features. As a result, systematic analysis of a large dataset consisting of thousands of images can generate 10^{10} to 10^{13} features.

3 High Performance Computing Approaches

Distributed storage platforms can be leveraged to reduce I/O costs of storing and retrieving very large datasets of high-resolution images. To maximize the efficiency of parallel storage and data accesses for image data, data declustering and indexing techniques can be employed. In an earlier work [2], we evaluated several techniques for data distribution, indexing, and query processing of multi-resolution 3-dimensional image datasets. We implemented Hilbert-curve based, random, and

round-robin distribution strategies for de-clustering of sub-image regions across storage nodes. A two-level R-tree based indexing scheme was employed. The two-level scheme consisted of an R-tree index for each mesh on the local chunks assigned to a node and another R-tree index on the bounding boxes of all the meshes in the dataset. We should note that storage and I/O costs can further be reduced by applying additional optimizations. These optimizations include incremental, adaptive declustering and partial replication. In the incremental, adaptive scheme, a dataset is initially declustered using a simple, but inexpensive, de-clustering algorithm (e.g., a round-robin assignment of image subregions to storage nodes) so that the data can be stored and made available for use quickly. The initial de-clustering can then be incrementally refined using information on data access patterns and a better, but potentially more expensive, de-clustering algorithm. Partial replication can be useful if there are multiple types of queries and/or if it is detected that certain regions of a dataset are accessed more frequently than others. In that case, instead of redistributing the entire dataset, the regions of the dataset can be replicated.

Processing of very large images and image datasets require careful coordination of data retrieval, distribution of data among processing nodes, and mapping of processing tasks to nodes. A combination of multiple parallelism approaches can be employed to quickly render results from a large dataset. Multiple images can be processed concurrently in a bag-of-tasks strategy, in which images are assigned to groups of processors in a demand-driven fashion. High-resolution images, however, may not fit in the main memory of a single processor. In addition, image analysis workflows may consist of operations that can process data in a pipelined, streaming manner. These characteristics of data and operations are suitable for combined use of task- and data-parallelism. We have developed a middleware system, referred to as out-of-core virtual microscope (OCVM)[3,4], based on the DataCutter infrastructure[5] in order to support multiple parallelism approaches. In this system, multiple instances of workflows can be created and executed with each instance processing a subset of images. Within each workflow instance, an image is partitioned into user-defined chunks (rectangular sub-regions) so that I/O operations can be coordinated by the runtime system rather than relying on the virtual memory. The processing operations constituting the workflow can be mapped to processors to reduce I/O and communication overheads. Multiple instances of an operation can be instantiated to allow for data-parallelism. In this setup, the retrieval, communication, and processing of chunks can be pipelined, wherever it is possible, and the chunks can be processed concurrently by multiple instances of an operation.

As we presented in Section 2, high-throughput, high-resolution analyses of micro-scale features will generate vast amounts of results. For example, in one of our projects, an analysis involving 213 whole-slide images segmented and annotated approximately 90 million nuclei. An XML results document for a single image, which included the boundaries of all segmented nuclei in the image along with 23 features computed for each nucleus, was close to 7GB in size. In order to scale to large volumes of data, databases of analysis results can be physically partitioned into multiple physical nodes on cluster based computing infrastructure. The distributed memory on a cluster system can also be leveraged to reduce I/O costs. We investigated the performance of different database configurations for spatial joins and cross-match operations[6]. The configurations included 1) a parallel database

management system with active disk style execution support for some types of database operations, 2) a database system designed for high-availability and high-throughput (MySQL Cluster), and 3) a distributed collection of database management systems with data replication. Our experimental evaluation of cross-match algorithms[7] shows that the choice of a database configuration can significantly impact the performance of the system. The configuration with distributed database management systems with replication (i.e., replication of portions of the database) provides a flexible environment, which can be adjusted to the data access patterns and dataset characteristics.

The other challenge associated with analysis results is the complexity of the results. As we presented in Section 2, semantic metadata is needed to describe analysis results (e.g., nuclear texture, blood vessel characteristics) and the context of the image analyses. An important aspect of semantic information systems is the fact that additional annotations/classifications (also referred to as implicit assertions) can be inferred from initial annotations (also called explicit assertions) based on the ontology and the semantics of the ontology language. Query execution and on-the-fly computation of assertions may take too long on a single processor machine. Pre-computation of inferred assertions, also referred to as the materialization process, can reduce the execution of subsequent queries. Execution strategies leveraging high-performance parallel and distributed machines can reduce execution times and speed up the materialization process[8]. One possible strategy is to employ data parallelism by partitioning the space in which the spatial objects are embedded. Another parallelization strategy is to partition the ontology axioms and rules, distributing the computation of axioms and rules to processors. This partitioning would enable processors to evaluate different axioms and rules in parallel. Inter-processor communication might be necessary to ensure correctness. This parallelization strategy attempts to leverage axiom-level parallelism. A third possible strategy is to combine the first two strategies with task-parallelism. In this strategy, N copies of the semantic store engine and M copies of the rule engine are instantiated on the parallel machine. The system coordinates the exchange of information and the partitioning of workload between the semantic store engine instances and the rule engine instances.

4 A Grid-enabled Implementation for Comparative Tissue Microarray Analysis

In this section we present the implementation of a Grid-enabled system to support analysis of large tissue microarray (TMA) datasets. This system is designed as a modular system and provides: 1) a library of image processing operations, including automated registration, segmentation, feature extraction, and classification; 2) a data model, which represents imaged specimen information, correlated clinical data, and quantitative analysis results; and 3) Grid services for remote access to image analysis algorithms and sharing of algorithms, image data, and analysis results.

The data management component of the system is designed to support multi-modal indexing and querying of imaged tissue discs and correlated clinical data based upon the staining characteristics and expression patterns of the specimen. It provides support to allow one to initiate queries based upon standard, text-based based criteria including the diagnosis of record, histologic type, tumor grade and biomarker used in

the study. The component also supports queries based upon the integrated staining intensity, effective staining area, and effective staining intensity of a given disc. The client interface allows users to interactively select any region or object of interest within a given disc and initiate a query based upon the texton histogram of that particular sub-region, tissue, or cell. Users can refine queries by clicking on any one of the ranked retrievals in which case the selected ranked retrieval serves as the new query input image.

The data analysis component of the system implements a library of operations to perform automatic computer-aided analyses of TMA arrays. The current library includes operations for correcting for artifacts and compensating for mechanical distortions within the specimen and performing automatic segmentation, feature extraction and classification of the tissue samples. These algorithms begin by automatically delineating the outer contour of each tissue disc. After isolating the TMA disc, segmentation of the ROI into regions is automatically executed using texton histograms. Textons are defined as repetitive local features that humans perceive as being discriminative between textures. Our library uses the multiple scale Schmid filter bank composed of 13 rotation invariant filters. The image filtering responses are clustered using K-means to generate a large codebook. A texton library is constructed from the corresponding cluster centers. After normalization, the texton histogram represents the texton channel frequency distribution in a local neighborhood around the centered pixel. In order to compensate for scale changes, the texton histogram is extracted from 5 different window sizes (4, 8, 16, 32, 64 pixels, respectively) and concatenated into one large feature vector. This concatenated texton histogram was used as features to train the classifiers. Given an image, we apply the trained classifiers for each pixel and separate the image into tumor and non-tumor.

We have developed a Grid service, based on the caGrid infrastructure[9], that encapsulates the computation of texton histograms given a set of TMA disc images. caGrid enables remote access to resources. In this way, analysis algorithms do not need to be incorporated to client programs and run on the local machine. Using the caGrid infrastructure, multiple algorithm developers can make their algorithms available through the same service interfaces and using the same object models for method input and output. This enhances the ability share new algorithms and tools among researchers and algorithm developers, because client programs can easily access the new resources without requiring resource specific modifications. Moreover, the algorithms and tools can be hosted on parallel machines to improve performance. In our case, the TMA analytical service support draws from the DataCutter framework[5]. DataCutter is designed and implemented as a stream-filter framework in which a data processing pipeline can be composed as a network of interacting components, referred to as filters. The filters interact with each other by sending and receiving data through communication channels referred to as streams. The DataCutter framework can enable bag-of-tasks type of parallelism as well as allow for combined use of task- and data-parallelism. In our current implementation we support the bag-of-tasks execution model. A single data processing operation or a group of interacting operations is treated as a single task. Multiple instances of these tasks can be instantiated on different computation nodes of a cluster. Images received by the system for analysis are distributed to these instances using a demand-driven strategy (to balance computational load among the task instances) based on a master-

slave implementation. Each slave node sends request to the master node when the slave node is available for taking a new task. The master node schedules and fetches the next task and sends to one of the available slave nodes. After processing of all the image discs is completed, the service returns the results to the client through the Grid.

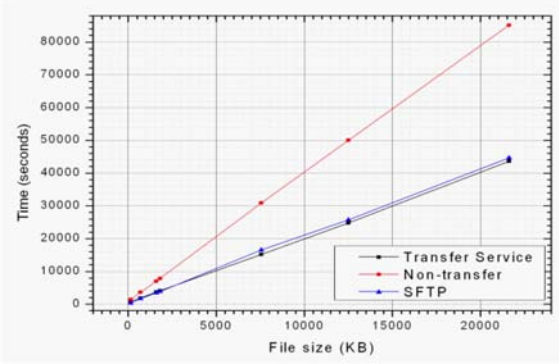

Figure 1. Upload speed of a batch file with 1 to 100 TMA disc images.

We have incorporated the caGrid transfer service in our analytical service to exchange image data and analysis results efficiently between the caGrid service and clients. Compared with the default implementation of using SOAP and XML to transfer data, the caGrid transfer service allows us to upload or download large data set without doing serialization or de-serialization, which is both time and memory demanding. Figure 1 compares the data upload speeds of different data transfer protocols for uploading a batch of files with one to one hundred TMA core images (size from 100KB to 20MB) using a client located at the Cancer Institute of New Jersey to one caGrid analytical service running on a cluster machine at the Ohio State University. Our results demonstrated that the caGrid transfer service performs significantly faster than SOAP messaging protocols, and is comparable with SFTP protocol.

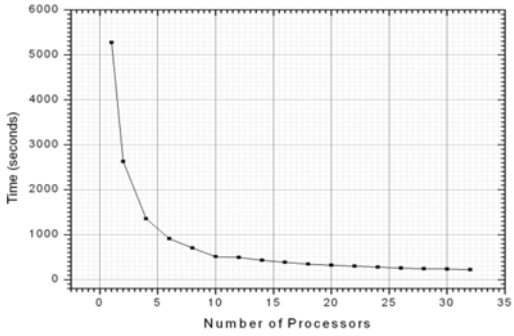

Figure 2. Processing time of TMA disc images versus the number of processors.

Figure 2 shows the processing time of TMA images versus the number of processors, where the cost drops significantly as the number of CPUs increases. The experiment was performed on a cluster machine at Ohio State University, with computing nodes of AMD Dual 250 Opteron CPUs, 8GB DDR400 RAM and 250GB SATA hard drive.

The computing nodes are connected through dual GigE Ethernet. Each node is installed with CentOS 4.0. This work was featured as part of a live demonstration at the July 2009 caBIG® face-to-face meeting held in Washington DC. In the demo, queries were initiated from a client machine at the meeting site and the computation was performed at Ohio State University and/or Emory University, with the image archive located at the Cancer Institute of New Jersey.

5 Related Work

Digital microscopy has become an increasingly important biomedical research tool as hardware instruments for rapid capture of high-resolution images from tissue samples have become more widely available. Several leading institutions have already undertaken ambitious projects directed toward digitally imaging, archiving, and sharing pathology specimens. One related software technology, referred to as Virtual Microscopy (VM), provides access to the resulting imaged specimens [10-14]. The Open Microscopy Environment project [15] develops a database-driven system for analysis of biological images. The system consists of a relational database that stores image data and metadata. Images in the database can be processed using a series of modular programs. These programs are connected to the database; a module in the processing sequence reads its input data from the database and writes its output back to the database so that the next module in the sequence can work on it. OME provides a data model of common specification for storing details of microscope set-up and image acquisition. CCDB/OpenCCDB [16] is a system and data model developed to ensure researchers can trace the provenance of data and understand the specimen preparation and imaging conditions that led to the data. CCDB implements an ontology link to support semantic queries and data sources federation. Several research projects have implemented techniques and tools for efficient management, query, and processing of scientific datasets. Manolakos and Funk[17] describe a Java-based tool for rapid prototyping of image processing operations. This tool uses a component-based framework, called JavaPorts, and implements a master-worker mechanism. Oberhuber[18] presents an infrastructure for remote execution of image processing applications using SGI ImageVision library, which is developed to run on SGI machines. Grid workflow management systems like Kepler[19] and Pegasus[20] seek to minimize the makespan by manipulating workflow-level parameters such as grouping and mapping of a workflow's components. Glatard et. al.[21] describe the combined use of data parallelism, services parallelism and job grouping for data-intensive application service-based workflows on the EGEE Grid. System-S [22] is a stream processing system developed at IBM. The system provides support for declaration and execution of system provided and user-defined operators on continuous streams of data on high-performance machines and in distributed environments. SciDB is a database management system under development, which is being designed to support very large scientific datasets[23]. SciDB is based on multidimensional array storage, rather than traditional relational tables, in order to reduce space and processing costs for scientific data. The MapReduce framework provides a programming model and runtime support for processing and generating large datasets on large cluster systems[24]. In MapReduce, two functions are provided by a user: A map function that processes (key,value) pairs and generates a set of (key,value) pairs; and a reduce function that merges and aggregates all the values with

the same key. The runtime system takes care of scheduling the execution of operations, data retrieval, data partitioning, and inter-processor communication.

6 Conclusions

Microscopy imaging is an underutilized tool in a researcher's arsenal of tools for basic and translational biomedical research. While advanced instruments for imaging tissues have been commercially available, the wider adoption of microscopy imaging in research and clinical settings has been hampered by the paucity of software tools for handling very large image datasets, complex analysis workflows, and managing huge volumes of analysis results. Use of parallel and distributed computing and storage environments can alleviate these challenges. It is possible to achieve good performance, but careful coordination and scheduling of I/O, communication, and computation operations in analysis workflows is necessary. Our work has showed that comprehensive systems for microscopy image analysis need to implement high-performance computing techniques throughout the system, including the storage and management of image data, execution of analysis algorithms, and management and exploration of analysis results. These systems should also leverage Grid computing technologies both for access to distributed computational and storage resources and for efficient sharing of data and tools in collaborative research efforts.

Acknowledgements. This research was funded, in part, by grants from the NIH through contract 5R01EB003587-04 from the National Institute of Biomedical Imaging and Bioengineering and contract 5R01LM009239-03 from the NLM, PHS Grant UL1RR025008 from the CTSA program, by R24HL085343 from the NHLBI, by NCI Contract N01-CO-12400, 79077CBS10, 94995NBS23 and HHSN261200800001E, by NSF CNS-0615155 and CNS-0403342, and P20 EB000591 by the BISTI program. Additional funds were provided by the DoD via grant number W81XWH-06-1-0514 and by the U.S. Dept. of Energy under Contract DE-AC02-06CH11357.

Bibliography

[1] Rimm DL, Camp RL, Charette LA, Costa J, Olsen DA, and Reiss M, "Tissue microarray: A new technology for amplification of tissue resources," Cancer Journal, 7(1), pp. 24–31, 2001.
[2] Zhang X, Pan T, Catalyurek U, Kurc T, and Saltz J. "Serving Queries to Multi-Resolution Datasets on Disk-based Storage Clusters", The Proceedings of 4th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid2004), Chicago, IL, April 2004.
[3] Kumar VS, Rutt B, Kurc TM, Catalyurek UV, Pan TC, Chow S, Lamont S, Martone M, Saltz JH, "Large-scale biomedical image analysis in grid environments", IEEE Transactions on Information Technology in Biomedicine 12(2), 154–161, 2008.
[4] Kumar V, Kurc T, Ratnakar V, Kim J, Mehta G, Vahi K, Nelson Y, Sadayappan P, Deelman E, Gil Y, Hall M, Saltz J, "Parameterized specification, configuration and execution of data-intensive scientific workflows.", Cluster Computing, April 2010.
[5] Beynon M, Chang C, Catalyurek U, Kurc T, Sussman A, Andrade H, Ferreira R, and Saltz J, "Processing large-scale multi-dimensional data in parallel and distributed environments," Parallel Comput., vol. 28, no. 5, pp. 827–859, 2002.
[6] Kumar V, Kurc T, Saltz J, Abdulla G, Kohn S, and Matarazzo C, "Architectural Implications for Spatial Object Association Algorithms", the 23rd IEEE International Parallel and Distributed Processing Symposium (IPDPS 09), Rome, Italy, May, 2009.
[7] Gray J, Nieto-Santisteban MA, and Szalay AS, "The zones algorithm for finding points-near-a point or cross-matching spatial datasets.", CoRR, abs/cs/0701171, 2007.

[8] Kurc T, Hastings S, Kumar S, et al. ”HPC and Grid Computing for Integrative Biomedical Research”, International Journal of High Performance Computing Applications, Special Issue of the Workshop on Clusters and Computational Grids for Scientific Computing, August 2009.
[9] Oster S, Langella S, Hastings S, Ervin D, Madduri R, Phillips J, Kurc T, Siebenlist F, Covitz P, Shanbhag K, Foster I, and Saltz J, “caGrid 1.0: An Enterprise Grid Infrastructure for Biomedical Research,” Journal of the American Medical Informatics Association (JAMIA), vol. 15, pp. 138–149, 2008.
[10] Felten CL, Strauss JS, Okada DH, Marchevsky AM. Virtual microscopy: high resolution digital photomicrography as a tool for light microscopy simulation. Hum Pathol. 30(4):477-83, 1999.
[11] Singson RP, Natarajan S, Greenson JK, Marchevsky AM. Virtual microscopy and the Internet as telepathology consultation tools. A study of gastrointestinal biopsy specimens. Am J Clin Pathol. 111(6):792-5, 1999.
[12] Ramirez NC, Barr TJ, Billiter DM. Utilizing virtual microscopy for quality control review. Dis Markers. 23(5-6):459-66, 2007.
[13] D. H. Okada, S. W. Binder, C. L. Felten, J. S. Strauss, and A. M. Marchevsky, “Virtual microscopy and the internet as telepathology consultation tools: Diagnostic accuracy in evaluating melanocytic skin lesions,” Am J. Dermatopathololology, 21(6), pp. 525–531, 1999.
[14] Afework A, Beynon M, Bustamante F, et al., “Digital dynamic telepathology - the Virtual Microscope.”, In the AMIA Annual Fall Symposium. American Medical Informatics Association, Nov. 1998.
[15] Goldberg I, Allan C, Burel JM, et al., “The open microscopy environment (OME) data model and xml file: Open tools for informatics and quantitative analysis in biological imaging.”, Genome Biol. 6(R47), 2005.
[16] Martone M, Tran J, Wong W, Sargis J, Fong L, Larson S, Lamont S, Gupta A, Ellisman M, “The cell centered database project: An update on building community resources for managing and sharing 3d imaging data.”, Journal of Structural Biology 161(3), 220-231, 2008.
[17] Manolakos E and Funk A, “Rapid prototyping of component-based distributed image processing applications using JavaPorts.”, In Workshop on Computer-Aided Medical Image Analysis, CenSSIS Research and Industrial Collaboration Conference, 2002.
[18] M. Oberhuber M, “Distributed high-performance image processing on the internet.”, MS Thesis, Technische Universitat Graz, 2002.
[19] Ludascher B, Altintas I, et al., “Scientific workflow management and the kepler system”, Research articles. Concurr. Comput.: Pract. Exper., 18(10):1039–1065, 2006.
[20] Deelman E, Blythe J, Gil Y, et al., “Pegasus: Mapping scientific workflows onto the grid.”, Lecture Notes in Computer Science: Grid Computing, pages 11–20, 2004.
[21] Glatard T, Montagnat J, and Pennec X, “Efficient services composition for grid-enabled data-intensive applications”, In Proceedings of the IEEE International Symposium on High Performance Distributed Computing (HPDC’06), Paris, France, June 19, 2006.
[22] Andrade H, Gedik B, Wu K, Yu P, “Scale-Up Strategies for Processing High-Rate Data Streams in System S.”, in The 25th International Conference on Data Engineering (ICDE 2009), Shangai, China. 1375-1378, 2009.
[23] Cudre-Mauroux P, Lim H, Simakov J, et al., “A Demonstration of SciDB: A Science-Oriented DBMS”, in 35th International Conference on Very Large Data Bases (VLDB'09). Lyon, France, 2009.
[24] Dean J, and Ghemawat S, “MapReduce: Simplified Data Processing on Large Clusters”, in 6th Symposium on Operating Systems Design and Implementation (OSDI'04). 2004: San Francisco, CA.