**Web Services for Building and Executing Biosequence Analysis Pipelines**

Adam Hughes\*, Saliya Ekanayake\*†, Judy Qiu\*†, and Geoffrey Fox\*†

\*Pervasive Technology Institute

†School of Informatics and Computing

Indiana University, Bloomington IN, USA

{adalhugh, sekanaya, xqiu, gcf@indiana.edu}

*Abstract*— The advent and continued refinement of modern high-throughput sequencing techniques have led to a proliferation of raw biosequence data, as labs routinely generate millions of sequence reads in a matter of days. Analyzing these results is beyond the computational capacity of single-lab resources, necessitating the use of high-performance computing resources, which presents significant overhead in terms of manual data manipulation and job monitoring. To overcome this bottleneck, application pipelines are being developed in conjunction with a generalized service-based portal system to abstract the details of job creation and management. This infrastructure will eventually provide service interfaces to map-reduce technologies, leveraging the computing resources offered by traditional grids and clusters, as well as emerging Cloud platforms.

This paper and the accompanying live demonstration first briefly detail the design and implementation of a software package to automate the setup steps for a specific sequence analysis workflow and a generalized service-oriented architecture (SOA) to expose this pipeline and other analysis tools as web services. Then, a simple web interface for submitting and managing jobs on high-performance computing platforms is presented as an example of the flexibility in building applications afforded by web service composition.

Keywords-biosequence; bioinformatics; SOA; service-oriented architecture

# Introduction

Genome research projects generate large quantities of data in the form of biosequence reads, each consisting of several hundred base pairs. In order to derive useful information from these data, scientists employ many software tools to search generated biosequences against existing databases, to visualize these search results, and to create and manage biosequence annotations. The sheer size of the experimental data, which may consist of hundreds of millions of sequence reads generated from a single run (as discussed in [1] and [2]), renders the computational resources of single labs inadequate for completing such analyses in a timely fashion. Consequently, the use high-performance computing resources is generally required to make progress against the avalanche of sequencing results facing genomic researchers. Creating and managing the computational jobs necessary to carry out these analyses can be a tedious manual task, potentially costing the researcher valuable time and effort. In order to overcome this productivity bottleneck, robust application workflows for automating most of the steps involved in sequence analysis and a service-based portal system to abstract the intricacies of managing interactions with underlying high-performance computing machines are being developed. This infrastructure will be expanded to provide service interfaces to map-reduce technologies and make the computing resources offered by traditional grids and clusters, as well as emerging Cloud platforms, more accessible to typical sequencing projects.

This paper and the accompanying live demonstration first briefly detail the design and implementation of a software package to automate the setup steps for a specific sequence analysis workflow and a generalized service-oriented architecture (SOA) to expose this pipeline and other analysis tools as web services. Then, a simple web interface for submitting and managing live jobs on high-performance computing platforms is presented as an example of the flexibility in building applications afforded by web service composition.

.

# Related Work

A common early step in typical biosequence analysis workflows is sequence alignment, wherein a set of input sequences are compared to a databases of well-known sequences, such as the human genome, or against each other. Many of these pipelines employ the Smith Waterman-Gotoh (SW-G) algorithm, a local sequence-alignment tool which calculates similarities between regions of protein or DNA sequences and scales like O(N2) [4, 5].  Recent work has focused on multiple parallel implementations of this algorithm, including MPI, DryadLINQ, Apache Hadoop, and Twister (Iterative Map-Reduce) to examine Alu clustering[6,7,8].    
  
The pipeline model discussed here presents a flexible architecture which allows for the use of any of these Smith Waterman-Gotoh implementations, as well as other alignment and analysis tools as they become available.  By exposing this entire pipeline framework through standard web service interfaces, the described portal system builds on earlier gateway ideas [3] and enables the researcher to employ the appropriate computational strategy based on the specific problem at hand and available computational platforms, serving as a simple gateway to grid, cluster, and Cloud resources.

# Biosequence Classification Workflow

One important aspect of biosequence analysis involves classifying biological sequences of either DNA or proteins into clusters of similar nature, enabling biologists to draw conclusions about a set of sequence reads relative to each other. Such classification begins with the calculation of a value to indicate the similarity between each pair of sequences. This value is usually known as the distance between two sequences. Thus, a matrix of distances for each pair is generated and is fed to a clustering algorithm implementation. The clustering algorithm will assign a group number to each sequence such that sequences with similar distances to other sequences fall into the same group.

The previous steps are the essence of the classification process. A visualization step, however, dramatically increases the usefulness of such results for analytical purposes. Therefore, a point visualization based on the same distance matrix is attached to the chain of algorithms. The goal of the visualization algorithm is to find 3-dimensional (3D) coordinates for each sequence such that each pair of points has the same distance as the corresponding two sequences in the distance matrix. A visualization tool is then able to produce a 3D picture depicting the clustering of sequences. The classification scheme described here is shown in Fig. 1.

Several pieces of software work in a flow (pipeline) to complete one classification task. Because of the large input sets typical of such a sequence classification study, the use of HPC resources is vital to the timely completion of these analyses. In order to simplify the creation of HPC jobs to execute analysis pipelines, a job configuration and submission tool has been developed as a simple Graphical User Interface (GUI) to select the suite of algorithms and the computer cluster nodes, and to allow configuring the algorithm parameters. The job is saved to the local machine, which makes it easy to perform a re-run or to use some of the algorithm configurations for a new job. shows a screen shot of this job-building tool, which will be demonstrated during the live exhibition.

# Service-Oriented Architecture

Exposing computational tools, such as the analysis pipeline described above, through simple web service interfaces further simplifies the task of managing HPC jobs. To this end, the SALSA Portal is being developed as a suite of generic web services for creating and managing HPC jobs on a wide variety of computing platforms. Because these utilities are presented as services, they are readily consumable by many types of software clients [9], including web sites and desktop applications, allowing end users to compose the services into utilities that satisfy their individual needs. Furthermore, the service architecture is fully extensible, enabling access to new analysis tools and computing platforms, as they become available, with minimal new development effort.

To accomplish these goals, certain foundational functionalities are required for managing computational jobs under most conditions, and these *use cases* form the basis for the Portal’s design. The major areas of functionality provided by Portal services are as follows: *Security*, *Resource Discovery*, *Data Transfer*, *Job Management*, and *Portal Administration*.

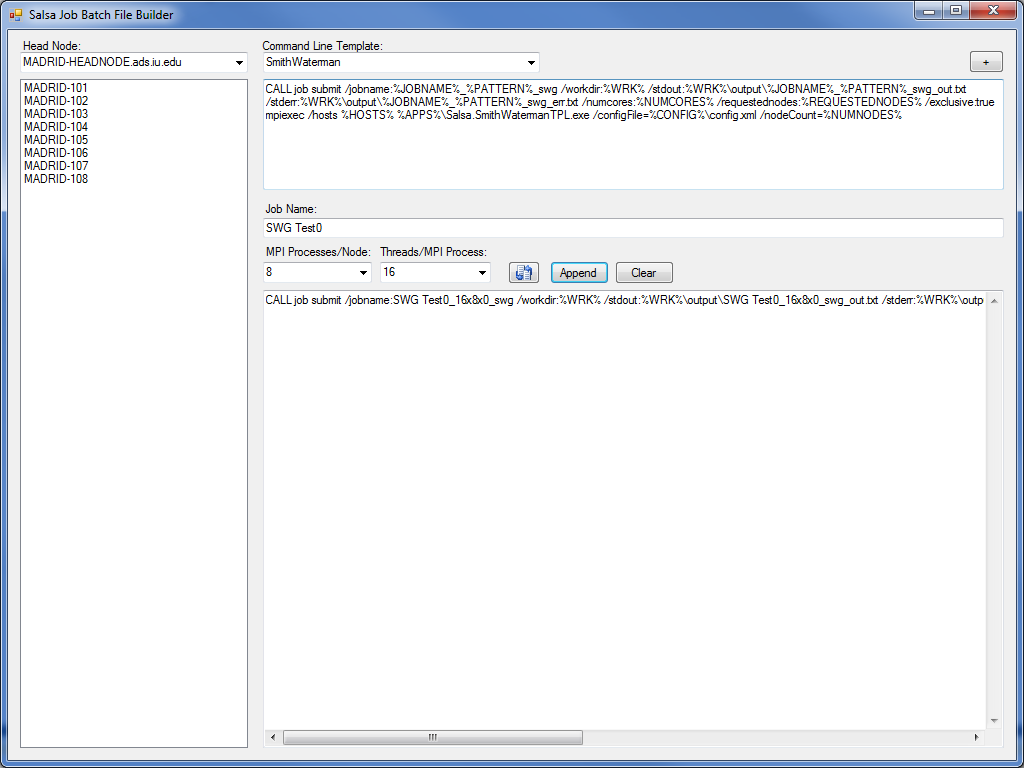
To address these use case requirements, the SALSA Portal was designed as a Service-Oriented Architecture (SOA) arranged in logical layers [10], as shown in Fig. 3, allowing for software reuse and the composition of available functionalities. Each layer consists of a set of software components that may be used to carry out the various actions required for job and resource management.

Although multiple web-service technology frameworks are available for the implementation of portal-type systems, the Windows Communication Foundation (WCF) provides developers with a comprehensive set of features that make WCF very attractive for enterprise-quality software production in a variety of settings, as in [9]. Consequently, the SALSA Portal is implemented through WCF services.

By exposing the basic functions involved in managing computational jobs through simple web service interfaces, the SALSA Portal enables developers to rapidly develop applications in a variety of technologies in order to best serve specific scientific domains and user communities. As shown in Fig. 4, a simple web interface has been developed which accesses a basic set of the SALSA web services in

****

1. General Bioseqeunce Classification Scheme



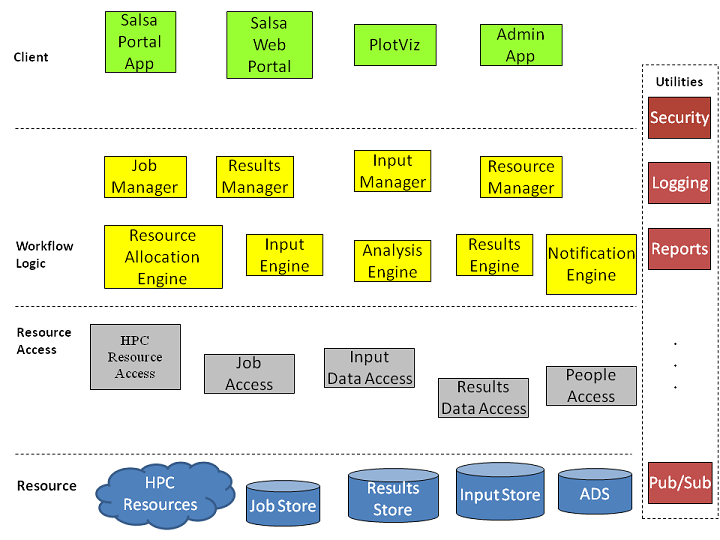
1. A simple user interface for creating biosequence analysis pipeline jobs to run on HPC resources.

order to carry out jobs based on the biosequence analysis

pipeline discussed above. This illustrative example will also be shown live during the accompanying demonstration session.

# Conclusion

The deluge of data produced by modern experimental techniques in the physical sciences is making it increasingly obvious that analysis and visualization no longer belongs solely in the realm of local laboratory computing resources.  Instead, such problems must be approached utilizing state-of-the-art software tools running on distributed high-performance computing resources, both traditional clusters and various Cloud platforms.  The substantial usage barrier for making full utilization of these resources often leaves researchers frustrated and with a mounting cache of raw data.

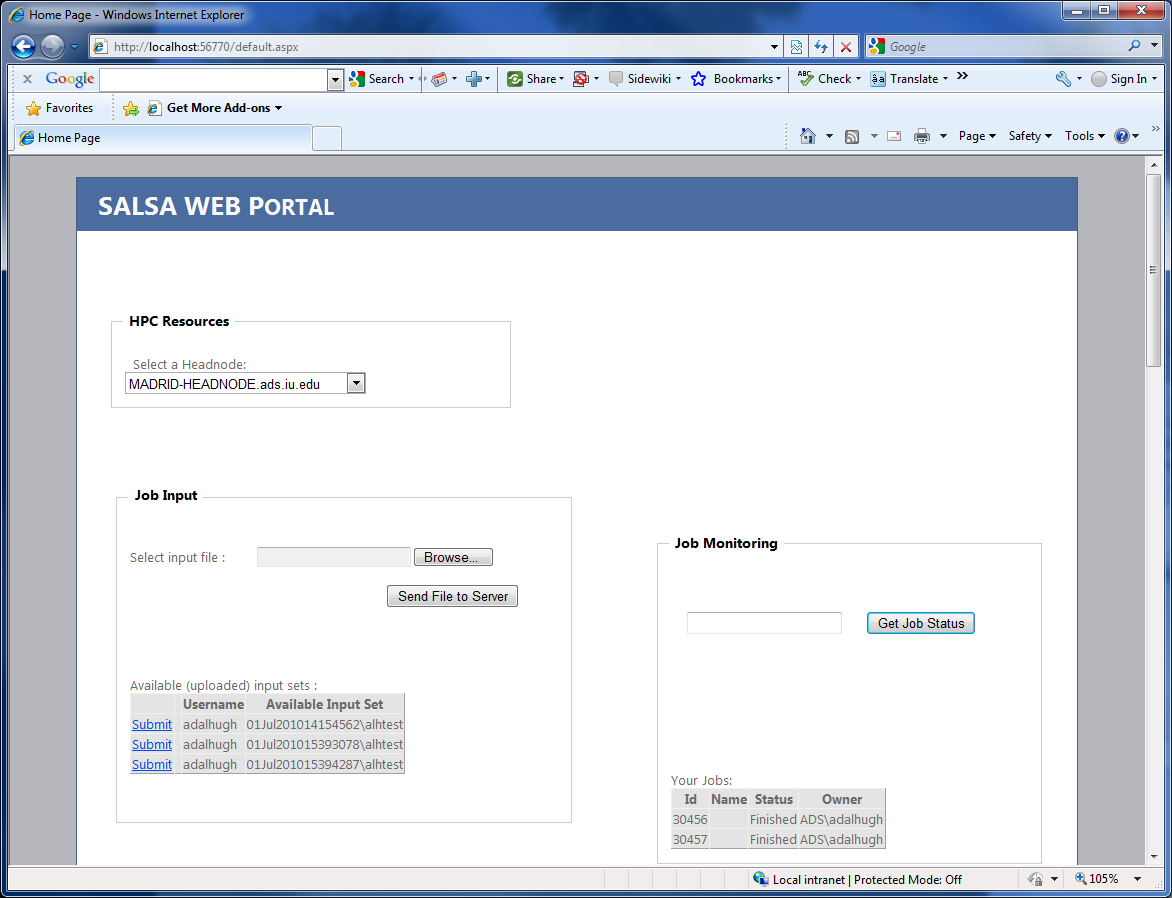


1. The multi-tiered, service-oriented architecture of the SALSA Portal services. All *Manager* components are exposed as web services and provide a loosely-coupled set of HPC functionalities that can be used to compose many different types of client applications.

The combination of algorithm “chaining” to form analysis pipelines and web services to allow easier access to such computational tools holds much promise for easing the current tedious data manipulation burden facing many researchers in the biology and health science fields. The extensible architecture of the SALSA Portal allows for relatively simple expansion to support new software tools and computing platforms as they are developed in the coming years.  This flexibility will be exhibited during the live exhibition through a demonstration of a simple web application orchestrated through composition of SALSA web services as described here.

##### Acknowledgment

This work is supported by the National Institutes of Health under grant 5 RC2 HG005806-02.



1. A simple web client that utilizes SALSA Portal web services to provide HPC resource discovery, file transfer, and computational job control functionalities to end users.

##### References

1. M. Marguilies et al., “Genome sequencing in microfabricated high-density picolitre reactors,” *Nature*, vol. 441, no. 7089, . pp. 376-380, Sep. 2005.
2. B. Richter and D. Sexton, “Managing and Analyzing Next-Generation Sequence Data,” *PLoS* *Computational Biology*, vol. 5 no. 6, Jun. 2009.
3. M. Pierce, C. Youn, and G. Fox, “The gateway computational web portal,” *Journal of Concurrency and Computation: Practice and Experience*, vol. 14, no. 13-15, pp. 1411-1426, Nov.-Dec. 2002.
4. T. Smith and M. Waterman, “Identification of common molecular substances,” *Journal of Molecular Biology*, vol. 147, pp. 195-197, 1981.
5. O. Gotoh, “An improved algorithm for matching biological sequences,” *Journal of Molecular Biology*, vol. 162, pp. 705-708, 1982.
6. J. Ekanayake, T. Gunarathne, X. Qiu, “Cloud Technologies for Bioinformatics Applications,” accepted to the *Journal of IEEE Transactions on Parallel and Distributed Systems,* Special Issues on Many-Task Computing, accepted for publication.
7. J. Qiu, et al., “Data Intensive Computing for Bioinformatics,” in *Data Intensive Distributed Computing*. Hershey, PA:IGI Global: 2011.
8. G. Fox, et al., “Biomedical Case Studies in Data Intensive Computing,” keynote talk*, 1st International Conference on Cloud Computing (CloudCom 2009)*, Beijing Jiaotong University, China Dec. 2009.
9. J. Lowy, *Programming WCF Services*, 1st ed. Sebastopol, CA: O’Reilly Media, 2007.
10. T. Erl, *Service-Oriented Architecture (SOA): Concepts, Technology, and Design*, 9th ed. Crawfordsville, IN: R.R. Donnelley, 2009.