PROCEEDINGS OF SCIENCE

# Building a Chemical-Protein Interactome on the Open Science Grid

**Rob Quick**[*a], **Soichi Hayashi**,[a] **Samy Meroueh**,[b] **Mats Rynge**[c] **Scott Teige**,[a] **Bo Wang**,[b] **David Xu**[b]

[a] *High Throughput Computing Group - Research Technologies Indiana University*
[b] *Department of Biochemistry and Molecular Biology - Indiana University School of Medicine*
[c] *Information Sciences Institute - University of Southern California*
*E-mail:* rquick@iu.ed, hayashis@iu.edu, smeroueh@iu.edu, rynge@isi.edu, steige@iu.edu, dadiaobo@gmail.com, yx5@umail.iu.edu

The Structural Protein-Ligand Interactome (SPLINTER) project predicts the interaction of thousands of small molecules with thousands of proteins. These interactions are predicted using the three-dimensional structure of the bound complex between each pair of protein and compound that is predicted by molecular docking. These docking runs consist of millions of individual short jobs each lasting only minutes. However, computing resources to execute these jobs (which cumulatively take tens of millions of CPU hours) are not readily or easily available in a cost effective manner. By looking to National Cyberinfrastructure resources, and specifically the Open Science Grid (OSG), we have been able to harness CPU power for researchers at the Indiana University School of Medicine to provide a quick and efficient solution to their unmet computing needs. Using the job submission infrastructure provided by the OSG, the docking data and simulation executable was sent to more than 100 universities and research centers worldwide. These opportunistic resources provided millions of CPU hours in a matter of days, greatly reducing time docking simulation time for the research group. The overall impact of this approach allows researchers to identify small molecule candidates for individual proteins, or new protein targets for existing FDA-approved drugs and biologically active compounds.

*International Symposium on Grids and Clouds (ISGC) 2015,*
*15 -20 March 2015*
*Academia Sinica, Taipei, Taiwan*

[*] Speaker.

 

PoS(ISGC2015)024

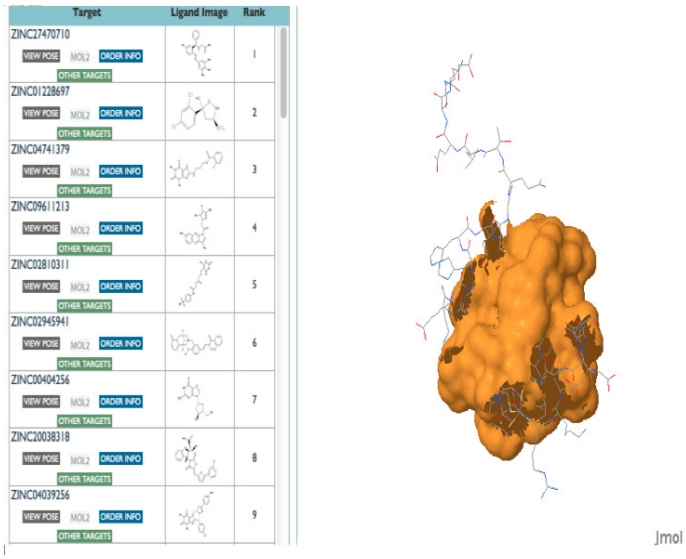


**Figure 1:** Sample visualization of a protein (CBFA2T1 Byclin-D-related protein) along with ranking table

## 1. INTRODUCTION

The Structural Protein-Ligand Interactome or SPLINTER [3] project led by the Indiana University School of Medicine predicts the interaction of molecules with proteins through structure-based molecular docking and scoring. The docking is calculated using AutoDock Vina [6], a program in which the interaction of a small molecule with a single protein is simulated (Figure 1). While a single docking requires trivial computing resources, the SPLINTER project attempts to build a comprehensive interactome of thousands of potential treatment molecules with thousands of proteins, a task requiring tens of millions of CPU hours.

The serial nature of each individual job makes the high throughput computing environment provided by the Open Science Grid [4] a perfect fit for this workflow. By using the HTCondor scheduling system [2] and the Pegasus [1] workflow management system, the High Throughput Computing (HTC) group at Indiana University was able to provide the SPLINTER researchers with more than 35 million CPU hours between January 2013 and March of 2015.

By utilizing existing OSG provided infrastructure, minimal development effort (less than 1 FTE Week) by the Indiana University HTC team was required to implement the workflow. At production levels some unanticipated issues were discovered and addressed. These issues included data storage for input and output and optimizing resource discovery, matchmaking and usage. Analysis of many individual ligand-protein pairs per individual flocking job was implemented to minimize batch system overhead.

## 2. BACKGROUND

The Open Science Grid provides shared resources, services and software for the scientific community. It is used as a high throughput computing environment in which problems are addressed by

breaking them down into many individual, independent jobs. To enable SPLINTER to use OSG, it was necessary to develop a workflow in which the docking jobs can be isolated, executed, and managed. Pegasus was selected for workflow management because it enables an application to be executed in different settings, as well as describing the workflow in abstract terms. In addition to Pegasus, DAGMan [5] was used to execute the tasks, and HTCondor was used to manage the individual workflow tasks. HTCondor handles the job submission part by placing them on queues and identifying available resources. Finally AutoDock Vina was used for the simulation of the ligand-protein interaction. It uses the same molecular structure input file format as AutoDock, is designed to be simple and easy to run, and is faster [6] than AutoDock.

## 3. METHODS

Due to the large amount of input and output files in a SPLITER run, the workflow consists of two parts: the main workflow and sub-workflows. They are defined as follows:

1. The user provides the protein and ligand structure input files.

2. Pegasus generates the main workflow configuration file with information such as location of the input and output files, configuration of the HTCondor job script and environment variables.

3. Sub workflow configuration files are created. Each sub-workflow manages up to 20,000 individual ligand-protein interaction simulations.

4. Each individual job contains a small cluster of of AutoDock Vina calls. Task clustering in this case makes the jobs long enough for grid overheads to be negliable.

5. Pegasus manages job submission of the grouped ligand-protein sub-workflows to OSG resources.

## 4. RESULTS

A initial run was submitted during January and February of 2013 and consisted of approximately 3900 proteins and 5000 ligands constituting over 19 million docking simulations. This run accounted for 1.42 million wall clock hours and completed in 27 days. The daily average total wall clock time delivered was 52,593 hours. Peak values exceeded 100,000 hours per day (Figure 2).

Subsequent SPLINTER analyses between March 2013 and December 2013 consumed an additional 2.6 million CPU hours. From January 2014 to this writing, SPLINTER consumed more than 34M CPU hours. Table 1 shows the distribution of wall time hours provided per OSG resource provider for this subset of tasks in the workflow.

| Institution | Millions of hours |
|---|---|
| Fermi National Accelerator Lab, FermiGrid | 4.758 |
| Fermi National Accelerator Lab, CMS Tier 1 | 4.535 |
| University of California, San Diego CMS Tier 2 | 3.686 |
| California Institute of Technology, CMS Tier 2 | 3.528 |
| University of Nebraska, Lincoln | 3.161 |
| Massachusetts Institute of Technology, CMS Tier 2 | 2.318 |
| New York Syracuse | 1.504 |
| Midwest Atlas Tier 2 | 1.494 |
| University of Nebraska, Omaha | 1.192 |
| Brookhaven National Lab, Atlas Tier 1 | 0.772 |
| Other | 7.090 |
| Total | 34.038 |

**Table 1:** Computation hours by institution for the period January 2014 to March 2015. Midwest Tier 2 consists of resources from Indiana University, University of Illinoins, Urbans, University of Michigan and the University of Chicago.

## 5. OTHER CONSIDERATIONS

Each individual ligand-protein simulation generated a single output file. While small (approximately 1 kilobyte), the very large number of these files would have caused access time difficulties if they had been directed to a single directory. Since the output filename was constructed from the ligand and protein input filenames no special consideration for its destination was required. Each sub-workflow created a destination directory with a name derived from the unix time of creation where the output was stored. Once a sub-workflow was completed the files were concatenated into a single archive file, compressed and delivered to the end user. Ordering was later added to simplify administration upon unpacking output at the data's final destination.

## 6. CONCLUSION

The Open Science Grid and the Pegasus workflow management system were used to execute tens of millions of small tasks. Problems associated with the segmentation of the total task, minimization of batch queue overhead, management of submission of sub-tasks and mechanisms for handling the resulting output files were addressed. The methods described here are directly applicable to the large class of problems defined by exhaustive analysis of pairs of input items.

## 7. ACKNOWLEDGEMENTS

This research was done using resources provided by the Open Science Grid, which is supported by the National Science Foundation and the U.S. Department of Energy's Office of Science.

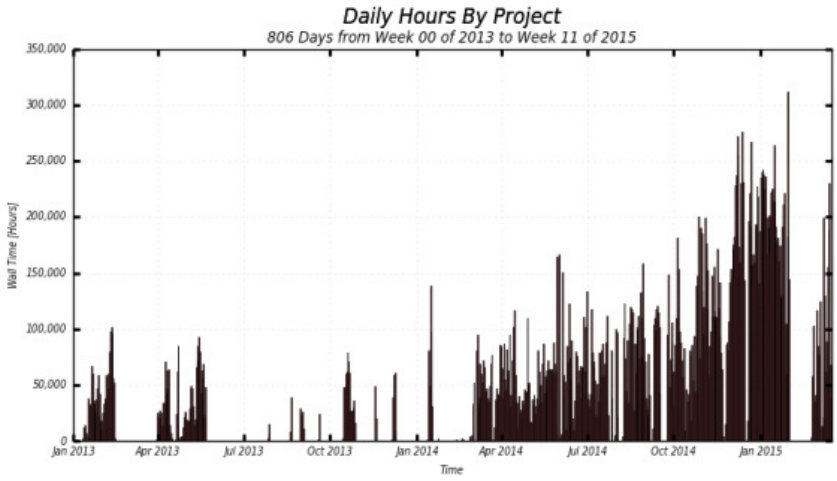


**Figure 2:** Over the lifetime of the SPLINTER project on the Open Science Grid access to computing resources has consistently grown. Single day production has reached over 300k wall hours.

## References

[1] E. Deelman, G. Singh, M. hui Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good, A. Laity, J. C. Jacob, and D. S. Katz. *Pegasus: a framework for mapping complex scientific workflows onto distributed systems. SCIENTIFIC PROGRAMMING JOURNAL*, **13**:219-237, 2005.

[2] M. J. Litzkow, M. Livny, and M. W. Mutka. *Condor - a hunter of idle workstations*. In ICDCS, pages 104-111. IEEE Computer Society, 1988.

[3] X. Peng, F. Wang, L. Li, K. Bum-Erdene, D. Xu, B. Wang, A. A. Sinn, K. E. Pollok, G. E. Sandusky, L. Li, J. J. Turchi, S. I. Jalal, and S. O. Meroueh. *Exploring a structural protein-drug interactome for new therapeutics in lung cancer.* Mol. BioSyst., DOI: [10.1039/C3MB70503J] 2014.

[4] R. Pordes, D. Petravick, B. Kramer, D. Olson, M. Livny, A. Roy, P. Avery, K. Blackburn, T. Wenaus, F. Wurthwein, I. Foster, R. Gardner, M. Wilde, A. Blatecky, J. McGee, and R. Quick. *The open science grid*. Journal of Physics: Conference Series, **78(1)**:012057, 2007.

[5] D. Team *Dagman - directed acyclic graph manager*. http://research.cs.wisc.edu/htcondor/dagman/?dagman.html/

[6] O. Trott and A. J. Olson. *Autodock vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading*. Journal of Computational Chemistry, **31(2)**:455-461, 2010.