Northwest Trajectory Analysis Capability: A Platform for Enhancing Computational Biophysics Analysis

Elena S. Peterson[1], Eric G. Stephan[1], Abbie Corrigan[1], Roberto Lins[1], Thereza A. Soares[1], Randy Scarberry[2], Stuart Rose[2], Leigh Williams[2], Canhai Lai[1], Terence Critchlow[1], T.P. Straatsma[1]
[1]Computational Sciences & Mathematics Division, PNNL, Richland, WA, USA
[2]Computational & Statistical Analytics Division, PNNL, Richland, WA, USA

Abstract - *As computational resources continue to increase, the ability of computational simulations to effectively complement, and in some cases replace, experimentation in scientific exploration also increases. Today, large-scale simulations are recognized as an effective tool for scientific exploration in many disciplines including chemistry and biology. A natural side effect of this trend has been the need for an increasingly complex analytical environment. In this paper, we describe Northwest Trajectory Analysis Capability (NTRAC), an analytical software suite developed to enhance the efficiency of computational biophysics analyses. Our strategy is to layer higher-level services and introduce improved tools within the user's familiar environment without preventing researchers from using traditional tools and methods. Our desire is to share these experiences to serve as an example for effectively analyzing data intensive large scale simulation data.*

Keywords: Data Management, Computation Biology Software, Molecular Dynamics, Data Intensive Computing.

1 Introduction

The Data Intensive Computing for Complex Biological Systems (DICCBS) project, at Pacific Northwest National Laboratory (PNNL) and Oak Ridge National Laboratory (ORNL), performs leading-edge computational biology and computational chemistry with the goal of improving the understanding of complex protein interactions [1]. Scientists on this project make extensive use of several high-performance computational tools including: NWChem [2], a complex, parallel computational chemistry/biophysics simulation code; and DIAna [3], a parallel data analysis tool. For example, some of the scientific research that has already benefited from this infrastructure resulted in the design of a protein-based scaffold with potential application as biosensors for *in vitro* diagnostics of diseases and environmental pollutants. As scaffold, scientists at PNNL have used the computationally designed protein Top7 [4] [5]. Because the Top7-based scaffold exhibits an unusual stability at extreme temperature and chemical conditions, these biosensors will have applications outside the highly controlled laboratory setting. The successful usage of this technology relies on the molecular level understanding of the factors behind the stability of Top7. This can be, in principle, achieved by performing a wide array of simulations of several protein variants under a variety of physico-chemical conditions (pH, temperature, ionic strength). However, to manage and analyze the several TB of generated data would be rather challenging by traditional means.

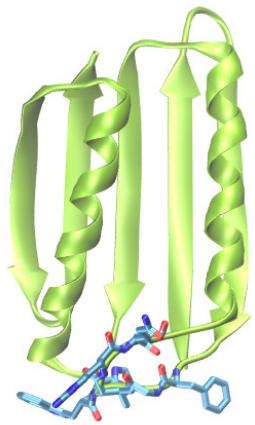

Figure 1. The three-dimensional structure of the Top7-based scaffold. Residues in stick correspond to the new sequence inserted in the original Top7 protein.

While the data files produced in this experiment are not particularly large by today's standards, the overall collection of simulation and analysis results is significant. The problem deepens if comparative analysis involving different simulations are necessary to answer a specific scientific question. Also the analytic tools require a large set of specific inputs and produce a complicated set of results data. An efficient method for handling the data and results involved in this paradigm is required [6]. This includes the ability to search for input data and results data based on metadata about the simulation as well as general provenance information about the setup and running of each simulation (i.e. who ran it, when it was ran, general input parameters, etc). The ability to tie all this together in a tool suite allows the researchers to work in one environment that meets their exact needs.

2 Background and Approach

Our goal was to develop an analytical environment to increase the efficiency of large scale data analyses without being intrusive to the scientist's analytical approach. Our first task was to discover how the scientists performed their analysis. In our initial requirements investigation we discovered researchers relied upon many command line tools and visualization tools that they had grown accustomed to, but turned out to be inefficient and cumbersome. The inefficiencies of their process and tools didn't create problems until they started on larger scale data analysis which required dealing with larger data and other collaborators. This generated some new requirements:

- Search capabilities: Scientists could no longer store simulations on local disk. The scientists required a more sophisticated means to store the simulations without needing to track the simulations by hand or continually download hundreds of gigabytes of files to search through their results.
- Data conversion/visualization tools: Various visualization tools were used by scientists but each had their own data formatting requirements. The scientists would have to remember which tool could graph which type of analysis file and then convert their data to that tool's particular format.
- Data Management/Archive: As stated earlier, there was too much data to manage on local disks so a long term storage system was needed. Along with storing the data they wanted to be able to share it with their collaborators.

Based on the requirements analysis it was abundantly clear that in order for NTRAC to be useful it had to dovetail high level capabilities into the existing infrastructure and capabilities. Our motivation was in part focused on seeking early adoption to address user needs immediately because we knew scientists were ultimately more interested in accomplishing their science than waiting for an end-to-end solution.

With this rationale we designed a solution that provided users multiple access levels. On one level is a suite of advanced client side analytical capabilities and data management tools. On another level, however, the scientists use traditional methods to directly store and retrieve data. We also gave users the ability to mix and match the advanced capabilities with traditional methods. An example of this was allowing users to have full access to the analytical tools without the GUI.

For scientists wanting to use traditional methods NTRAC imposed little impact. These users would be requested to share their data in an archive shared area, rather than store it locally. For scientists having more demanding needs such as searches and visualization they could take advantage of the new tools. We made our approach generic enough to allow our new tools to be plugged into any underlying archive infrastructure.

Based on these requirements we developed a preliminary design that included user interface mockups, application interfaces, and example visualizations. The requirements analysis, preliminary design, and extensive user interviews provided us with a basis to develop a system that was applicable to the users needs and unobtrusive in its implementation. Because many of the software components to be developed were not directly dependent on each we were able to design and implement them in parallel. This allowed us to get prototypes into the users hands quickly.

3 Architecture

There are three main parts to the overall architecture of our system all three of which are both integrated and can be used alone. First is the user interface tool suite that allows the users to search for simulation files, launch the analysis tool(s), and visualize the outputs of that analysis. Then there is the data storage and movement architecture, and finally the metadata [7] services architecture.

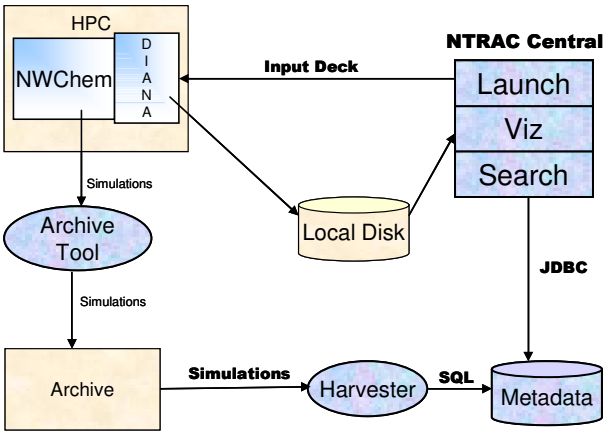

Figure 2. NTRAC Architecture depicting data flow from simulation to storage and to user analysis.

3.1 NTRAC Central – User Environment

For integrated analysis capabilities we designed a client that provided an intuitive user interface for the scientists to work in. Scientists could use this tool exclusively to search for all or part of archived NWChem simulations, to launch DIAna to analyze the simulation results, and to visualize the analysis results.

The NTRAC workflow typically begins with the search capability. Queries to select simulations for DIAna analysis are performed using a simple database call to the metadata repository (described below). Search criteria can include selections about the simulation run itself (e.g. scientist name,

simulation name, system type, method, and status) or properties of the simulation. The query results are presented to the user through a table/excel like interface and they can sort and subset from there.

Once a set of simulations is chosen, those files are downloaded to the local machine through the same protocols described below as the archive tool. The user can then use the DIAna input tool to select their input deck to the analysis tool. DIAna is a complex analysis tool written in Fortran 90. Analyses are carried out using an input deck describing the required analyses and proper inputs. The format of the file requires extensive knowledge of DIAna and the kind of analyses a user wants to carry out. As a result formatting an input file correctly was quite difficult even to the expert users. To make DIAna more usable and robust, the user interface guarantees the scientist that the input file created is well formed and will provide the results requested. Once the initial input is created users have the option of fine tuning their input prior to analysis by editing the file directly from the user interface. This accomplished two objectives, creating a sense of trust with the scientists – they could see exactly what the user interface was producing, and because the DIAna tool is changing rapidly they could make edits/additions without waiting for an update to the user interface.

Once the DIAna analysis is successfully completed we provide a rich set of graphing tools to visualize the results. As a foundation for our visualization environment we incorporated the JGraph [8] open source software to support the ability to provide various types of graphs given one set of inputs. To handle the numerous output formats we created a generic parser structure that requires only a few minor changes to produce a new type of visualization. As well, the various visualization screens all have similar behavior and look and feel.

To support persistence in user sessions we created a "context" file that tracks the state of the tools so that a user isn't constrained to performing all of their analysis in one execution of the client.

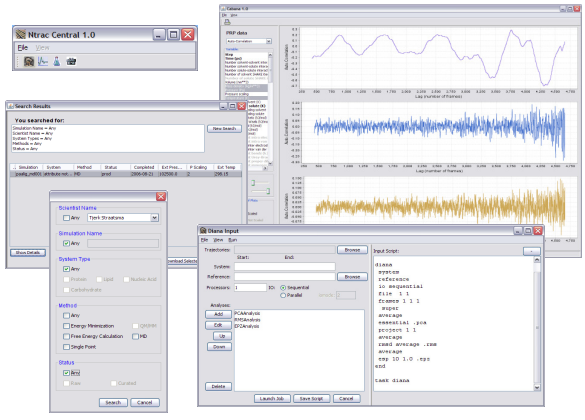

Figure 3. Examples of NTRAC Central user interface

3.2 Data Management Tools

In order to move beyond the initial ad hoc data management approach used by our scientists, we needed to address the major underlying issues that gave rise to it: the system administrators for each computational and storage resources were balancing different and conflicting requirements for these resources. As a result, each cluster and archive had its own management policy and configuration, making it extremely difficult to provide consistent capabilities across the computation environment utilized by scientists. The original policies were so diverse it wasn't always easy to know that the best data transfer method was being chosen. Lacking a ubiquitous data transfer capability, scientists ended up using only a subset of the available resources and thus were unable to take full advantage of the available capabilities.

To create that ubiquitous data transfer capability meant having a single tool that would pick the most efficient transfer method. We wrote a light-weight copy script in Python that works the same on all compute nodes. This script relies upon a protocol registry database that describes how available computational resources talk to each other. Using a data driven approach for determining the best method of data transfer rather than hardwiring a solution has given us the flexibility to dynamically alter data transfer strategies. Based on the selected protocol the script automatically builds the transfer commands and transfers the files. By masking the complexity of the environment through this script, our scientists no longer need to worry about the nuances of the underlying capabilities

3.2.1 Archive

For archiving we relied upon the PNNL's Environmental Molecular Science's Laboratory (EMSL) [9] 300 terabyte archive, NWfs. NWfs was created as a long term storage system but not necessarily as a data repository and definitely not for data sharing. As such it is large and regularly backed up but has only minimal data management support and no on-box computation capability. We worked with the NWfs team to create a solution by creating a "user-group", setting permissions appropriately and adding our researchers and their collaborators to that group. This gave the researchers one place for all of their data to be shared. While this is obviously not the entire data repository solution it did help us to lay a foundation with the scientists for creating a better long term solution.

3.3 Metadata Services

Because of the limited access to the archive, harvesting metadata [10] similar to the way web crawlers search the web to refresh search engines was a reasonable approach. The metadata services consist of three components: an intelligent harvester, a results parser, and a metadata store.

While archiving and sharing data sets is an important data management capability, being able to efficiently identify the subset of information currently of interest is critical in effectively managing the repository over time. We believe a metadata service approach, which supports complex user queries over metadata extracted from the simulation and analysis data sets, is the best way to deliver this capability.

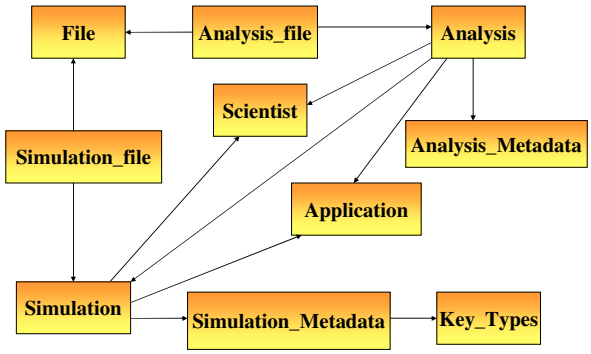

Figure 4. Metadata service schema

We have two services that obtain metadata. The first is an automatic extraction of metadata from known file types. We use the Defuddle [11] [12] parser to extract approximately 40 metadata attributes from NWChem and DIAna. This service automatically crawls the archive to identify new files, determines if the file is of a known type, and if it is, extracts the metadata. The second service is built on top of the copy capability, and allows scientists to manually annotate their data through a simple GUI when they archive it. This allows us to obtain high-level information, which is outside the scope of the automatically generated metadata, about the results being archived. These annotations are extremely important since they allow the simulation to be placed in the appropriate context long after the experiment was performed.

The extracted metadata is stored in the archive as a collection of self-describing XML files, one for each archive file. We also load this information into a PostgresQL™ database for ease of search later on.

While the use of an intermediate, self-describing, XML format may seem inefficient at first glance, we believe it is important to the long-term success of NTRAC. Both NWChem and DIAna are actively evolving, constantly adding new capabilities and improving existing features. XML is an extremely flexible data description format. This flexibility is enhanced by Defuddle code, which generates an XML representation of a file based on a declarative description of the original file format. As a result, we believe that this architecture will allow us to easily adapt to changes in the underlying simulation and analysis codes.

4 Discussion

Our development approach was to target early adopters and put discrete software capabilities into the researchers' hands as quickly as possible. This rapid prototyping approach gave the developers early evidence about the usefulness of the tools and guided new capability development or changed the requirements to better fit the scientists' needs.

The data management developers teamed with our internal computing infrastructure to define and implement faster methods of managing the large amounts of simulation results data files that often had to be moved from the compute machines to the storage areas. We were able to replace the long and painful process with a simpler and faster one and gain the trust of our users immediately.

During a three month evaluation period the value of this approach became apparent. During the early part of the evaluation period the secure copy protocol (scp) out performed the secure file transfer protocol (SFTP) by a factor of 100%. At the end of the evaluation period sftp became inefficient and scp outperformed sftp by almost 100%. The many aspects (i.e. network configuration, firewalls, etc) that account for these results are out of our control and beyond the capability of the general scientist to understand and manipulate. This simple litmus test helps give our users a best practices approach to data transfer.

To develop the metadata services we worked with users to develop a schema to support metadata collection and defined a raw data translator using Defuddle team to make configuring the metadata services completely data driven.

Without a metadata service identifying data of interest requires scientists to manually identify the appropriate simulation result files, download them, and inspect their metadata to determine which data should participate in the analysis. This is an extremely time consuming and painful process. Obviously, the complexity of identifying the appropriate simulation, or set of simulations in the case of a comparative analysis, increases as the data archive grows. As a result, it is easy to overlook relevant simulations. Furthermore, the data and analysis files are intended for software processing, not manual examination, and thus it is challenging to determine which datasets should be analyzed. Without using a metadata service setting up an analysis could take from days to weeks. Now with the help of the NTRAC Central search tool the metadata repository can perform queries in milliseconds.

The client tools were created to ease the process and provide support for the current workflow again without being intrusive or creating something unknown to the scientists. We were able to increase efficiency and accuracy to the overall process and gained user acceptance by providing small pieces in a timely manner.

As mentioned earlier this resource proved invaluable because now visualization and analysis were stream-lined and avoided the hours of mundane steps it took to reformat results during analysis. This capability now enabled the scientists to see their data in a matter of minutes.

5 Conclusion

NTRAC proved to be a robust, cost effective solution to support computational biophysics analysis at PNNL. As the analysis tools built around NWChem evolve and become available to the scientific community at large we will also provide them with our NTRAC Central solutions.

Through our experiences we found NTRAC serves as an example for ways research teams can enhance the analytical experience that data intensive applications require even while the application is evolving. Our long term vision is to apply the lessons of NTRAC to other large scale bioinformatics and computational biology problems and to incorporate leading edge research such as incorporation of scientific workflows and provenance tracking to capture data lineage, and interconnect with other biological grids.

6 Acknowledgements

This work was funded by the U. S. Department of Energy Office of Advanced Scientific Computing Research. PNNL is operated by Battelle for the U. S. Dept. of Energy

7 References

[1] Data-Intensive Computing for Complex Biological Science http://www.biopilot.org

[2] R.A. Kendall; E. Apra; D.E. Bernholdt; E.J. Bylaska; M. Dupuis; G.I. Fann; R.J. Harrison; J. Ju; J.A. Nichols; J. Nieplocha; T.P. Straatsma; T.L. Windus; A.T. Wong. "High Performance Computational Chemistry: An Overview of NWChem a Distributed Parallel Application"; Computer Shys. Comm, 128, (260-283), 2000.

[3] T.P. Straatsma. "Data Intesive Analysis of Biomelecular Simulations"; International Conference of Computational Methods in Sciences and Engineering, 963(2), (1379-1382), 2007.

[4] B. Kuhlman; G Dantas; G.C. Ireton; G. Varani; B.L. Stoddard; D. Baker. "Design of a novel globular protein fold with atomic-level accuracy"; Science, 302, (1364-1368), 2003.

[5] T.A. Soares; T.P. Straatsma. "Design of the Top7 protein as a scaffold for antigen-binding epitopes"; Presented by Thereza Soares (Invited Speaker) at the American Chimical Society NORM, June, 2007.

[6] T.P. Straatsma. "Data-intensive computing laying the foundation for biological breakthroughs"; Breakthroughs, 10, Spring 2007.

[7] K. Jeffery. "Metadata: An Overview and Some Issues"; ERCIM News, 35, October 1998.

[8] James S. Plank. "Jgraph – A Filter for Plotting Graphs in PostScript"; USENIX Technical Conference Proceedings, (61-66), Winter 1993.

[9] Environmental Molecular Sciences Laboratory http://www.emsl.pnl.gov

[10] http://lftp.yar.ru

[11] B. Wu; T.D. Talbott; K.L. Schuchardt; E.G. Stephan; J.D. Myers. "Mapping Phyisical Formats to Logical Models to Extract Data and Metadata: The Defuddle Parsing Engine"; IPAW'06 Internation Provenance and Annotation Workshop, May 2006.

[12] Martin Westhead, Ted Wen, Reobert Carroll. "Describing Data on the Grid"; fourth International Workshop on Grid Computing (134), 2003.