Using clouds for data-intensive computing in proteomics

Ananth Kalyanaraman, School of EECS, Washington State University, Pullman, WA
Douglas Baxter, Pacific Northwest National Laboratory, Richland, WA
William Cannon, Pacific Northwest National Laboratory, Richland, WA
Email: ananth@eecs.wsu.edu, *douglas.baxter@pnl.gov*, *william.cannon@pnl.gov*

ABSTRACT

Abstract Type: Position

Proteomics is an integral part of systems biology research and one that has become largely a data-intensive field. Due to advances in proteomics technologies, mass spectrometry data archives have been on the rise, as are sequence data generated from next-generation sequencing technologies. However, data mining and applying new methods to existing data sets is currently limited by the need to move and manage large sets of data. For example, mining a proteomics archive of MS/MS spectra for post-translational modifications is currently done on only small subsets of data, not the entire archive which contains over a billion MS spectra occupying over 150 TB as of September 2009. The problem is compounded when trying to co-analyze proteomics archives and nucleotide data from a broad spectrum of sequenced organisms – a capability needed when dealing with environmental samples. Application of the methods is limited by the manual labor required to gather all the data on a single file system. Yet integrated analysis of data from different sources has become far more important for furthering discovery. Cloud computing is expected to be an ideal computing model that can address these limitations and support the massive scale integration and analysis of proteomics and genomics data in a way that is transparent to the application scientist.

In this presentation we will identify the different ways & associated challenges that will be faced while developing cloud computing frameworks for proteomics applications. For ease of exposition, we will organize the presentation from the perspective of two types of core compute operations that are prevalent in proteomics & metaproteomics analysis: i) *Database search* – the primary example of this application class is peptide identification from MS/MS spectral data; and ii) *Large-scale graph analysis* – the primary example of this class is protein family characterization which typically involves computational of large-scale all-against-all sequence-sequence, profile-profile and sequence-profile comparisons. Both these applications are central to the structural and functional characterization of proteins represented from single organism to more complex microbial communities. These applications harbor a high potential to benefit from cloud computing because of large sizes of data and a broad user-base. Yet, their portability into the paradigm throws several challenges, both algorithmic and systemic. Historically, the codebase for these applications have been serial and there are very few implementations that support even traditional parallelization (i.e., on distributed and shared memory machines). Therefore, algorithmic innovations are required to map the underlying problem space to a Map-Reduce model, which is becoming the de facto standard for cloud computing. Furthermore, the evolution and continued upgradation of cloud computing technologies and infrastructures, at both the architectural and system software levels, is poised to impede transition and wider adoption.

To jumpstart discussion, we will present ideas and preliminary findings of our on-going research in the area. The overarching goal is to identify the merits and current limitations that exist along the path toward implementing the grand vision of building cloud computing infrastructures for proteomics research and applying them toward transformative advancement of scientific discovery.

RESPONSE TO CHARGE QUESTIONS

What are the characteristics of applications that would be appropriate for effective utilization of cloud architecture?

An application's fit into the cloud computing paradigm can be argued based on the following aspects (not limited to):

i) Computation should primarily be due to the preponderance of data (i.e., data-intensive);

ii) Application should allow the mapping/porting of the underlying algorithms to cloud computing architectures – and conducive for use of Map-Reduce/Hadoop libraries which have become a de facto standard in cloud platforms;

iii) Presence of a large user-base; and

iv) Functionality to benefit from virtualization.

Proteomics hosts applications that support all of the above.

Database search: Peptide identification, which is the problem of identifying the protein sequence that corresponds to an MS/MS experimental spectrum, heavily relies on database search. Here the characteristics are similar to that of BLAST in that the user specifies a query set (spectra) to be matched against a database (protein sequences). The database indexing, however, is different as candidate matches to spectra are determined based on mass/charge ratio of their parent peptides. The number of candidates to be compared per experimental spectrum drastically increases with the complexity of the underlying organism(s) under study – from x10^5 for a sequenced single microbe to x10^9 for an unsequenced microbial community. This, coupled with the fact that spectral library collections could go into millions, makes the application an ideal candidate to benefit from massive parallelism. State-of-the-art tools that are capable of scaling to hundreds of processors spend several hours even on spectral sets that contain only tens of thousands of spectra. Cloud architectures could significantly help the integration of large amount disk-resident data (both query & database) during analysis, including the storage and management of the large files typically output by these programs.

There are different ways to exploit the Map-Reduce paradigm for peptide identification. Mappers could be responsible for processing chunks of queries, chunks of database, or chunks of both queries and database. Reducers can be used to collect results by query and report top ζ hits, and/or can be used for grouping peptide hits based on the protein sequences they match on the database. This would create a need to use multiple rounds of map-reduce calls. Each approach would generate a different distribution of load among mappers and reducers and would therefore affect the initial allocation of mapper/reducer nodes. Support for virtualization is required due to the way the user would interact with the system. The user should not have to worry about resource or job allocation but is simply required to input a spectral set and a protein sequence database file.

Graph analysis: Applications such as protein family characterization from newly sequenced microbial communities/metagenomic samples primarily rely on comparing the sequence data against already characterized sequence and profile databases. These operations have already consumed millions of CPU hours and tens of terabytes of aggregate memory in the past (as demonstrated by the Sorcerer II GOS project). The underlying all-against-all comparison requires solving problems that are either a) memory-saturating because of graphs that grow quadratically with the number of input sequences need to be processed, or b) computationally intractable (e.g., quasi-clique detection for finding strongly linked groups of protein homologs). Map-Reduce based concepts are yet to be developed to support such advanced graph operations efficiently. However, because graphs can be maintained as adjacency lists and the approximation heuristics used for mining graphs often have built-in sorting, connected component detection and standard traversal operations, efficient implementation may be potentially achieved using Map-Reduce. This area is relatively nascent and needs new development. Traditionally, the implementation of these advanced functionalities has been a sole prerogative of large centers which have access to the required high-end equipment and personnel resources. With the democratization of sequencing, increasing number of researchers world-wide would like these functionalities implemented through a virtual environment without requiring them to invest in code/algorithm development, machines, or personnel. A virtual environment that supports an on-demand implementation of user-defined multi-program pipelines could just prove to be the means to achieve this broader impact goal.

What are the hardware bottlenecks that prohibit cloud architectures from being easily adopted for high throughput biological data analytics?

1. The data transfer rate of user-defined databases to and between cloud machines could become a bottleneck;
2. Within a cluster, the distribution of databases too large to be stored on a single node would require nodes with fast local storage or support for the optional use of fast network interconnect as a feasible alternative for disk access;
3. Another expected bottleneck is the reliability of the parts especially during a global participation of all the cloud components. This leads to requirements for fault resilient applications that avoid fixed processing resource algorithms with algorithms that store explicit sharable state in a reliable global resource.

What are the specific tools that need to be developed or enhanced in order to make cloud architectures easily adopted for biological data and bioinformatics algorithms?

1. The primary tool that needs to be developed or re-implemented is the application code itself. Significant developmental effort is required to re-engineer existing code and their underlying algorithms to take advantage of cloud computing clusters. Most codes are legacy codes, and even traditional parallelization using commodity clusters and/or supercomputers has been a rare commodity so far. For example, in peptide identification, only MSPolygraph offers a reasonable degree of parallelism on commodity clusters. In protein family detection, there is no single piece of serial code; instead what is used are custom-built computational pipelines, which use a mix of existing tools and heuristics (e.g., BLAST, HMMer, etc.) along with newly developed code. The process of porting is further complicated (algorithmically) by the inherent irregularity within data – e.g., irregular graph processing and analytics required for protein family detection, thereby necessitating significant algorithmic innovations alongside code reengineering.
2. New middleware tools for seamless integration of different programs under a virtual environment could be highly resourceful for automating custom-built pipeline where users can *interact* with the system and data under a plug-and-play schema.
3. Visualization tools for representing the results coming from data analytics are necessary. This will be a critical challenge for large-scale data sets, and particularly given the substantial lack of visual tools in the area even under traditional desktop models.
4. Standardized protocols are needed for a processor/compute node to describe its characteristics, and for a process to volunteer/join an existing collection of nodes.
5. Also useful would be a standardized protocol for enabling one sided communication for the loosely connected cloud computing resources (as opposed to the tightly coupled MPI code) as a way to support cloud level Remote Direct Memory Access Protocol.
6. Standardization is also needed for streamlining and integration of the various file and data formats supported in bioinformatics – e.g., fasta files, spectral files, GenBank annotation.