The Study of Implementing PhyloD application with DryadLINQ

Adrija Sen, Chengming Ge and Ratul Bhawal

Abstract

PhyloD is a statistical tool to identify HIV mutations that defeat the function of HLA proteins in certain patients, thereby allowing the virus to escape elimination by the immune system. [1] Since Dryad is a high-performance, general-purpose distributed computing engine [2], it plays the role of an ideal platform for executing the parallel computation of PhyloD. DryadLINQ guides the development of application in the form of LINQ programming model.

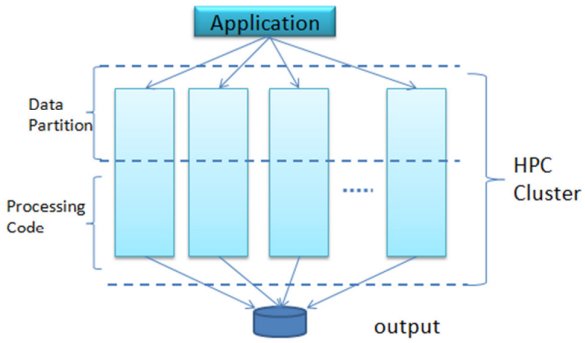
Introduction

HIV virus live and reproduce in human hosts, it attacks human bodies' immune system. To prevent this virus attack, immune system tries to identify and kill the pathogens. The HLV proteins play a role to identify the virus which invades the host cells and alert immune system to attack the HIV virus. The theory of HLV protein is that it provides a fragment of alleles which would fix certain order of codons on the surface of virus. But the virus always replicates fast and in its new reproduction, it is prone to mutation and make out new arrangement of codons for the next generation which escapes HLV protein's identification process. In this case, there is a need to find of alleles which fix well with the virus codons. This is the purpose of PhyloD, to identify and record the mutation of HIV and facilitate the researchers to decoding the complex rules that govern the HIV mutations.

The PhyloD package have three kinds of input data: (i) the phylogenic tree information of the codons, (ii) the information about HLA alleles, and (iii) the information about HIV codons. The workflow of running a single PhyloD application can be divided into three steps: first compute out the cross product of all the alleles and codons to build up various allele-codon pairs. Second, it computes the p-value for each pair, which is the measurement of the association between allele and codon a pair. Third, for each of the p-value, it computes a q-value, which is an indicative measure of the significance of the p-value.

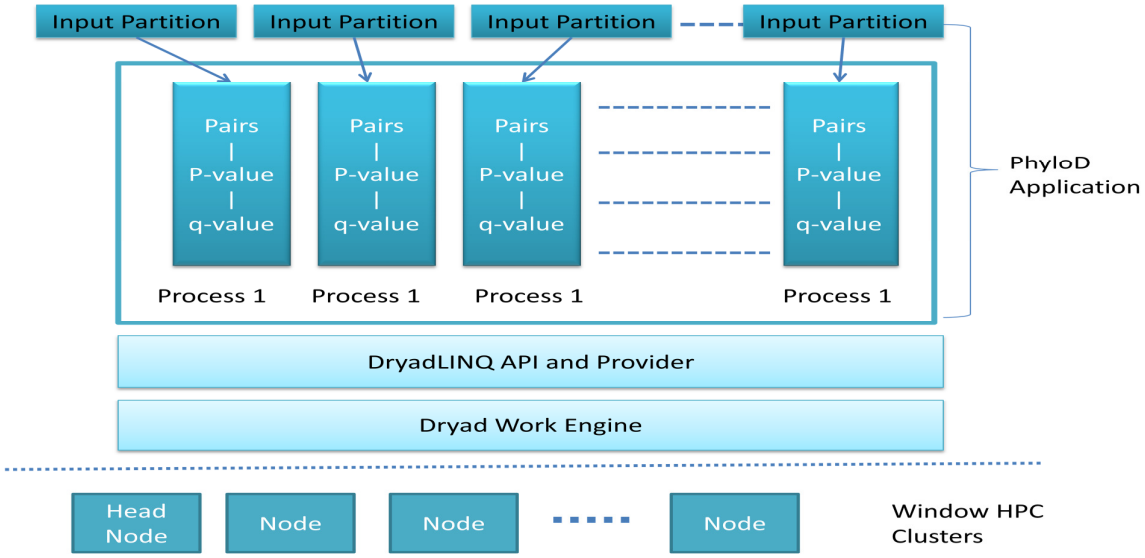
Both allele data file and codon file provide a large amount of data records. Suppose the number of allele is A, the number of codons is C, their cross product would produce A*C pairs, and the application needs to compute A*C times' p-value and q-value. To meet the workload of PhyloD application and reduce the duration cost by computing, a high-performance distributed computing engine is needed. Dryad is a high-performance general-purpose distributed computing engine for running distributed applications on various cluster technologies, including Window HPC Server 2008 [3]. Dryad considers computation tasks as directed acyclic graphs (DAG) where the vertices represent computation tasks and while the edges acting as communication channels over which the data flow from one vertex to another. The channel uses the mechanism of either file or shared memory. DryadLINQ is a compiler which translates LINQ programs to Dryad distributed computations that run on cluster service [4].

Figure 1: Shows how the PhyloD application would be implemented as distributed job.



In our project, we plan to implement parallel computing by executing the PhyloD application with Dryad on the Windows HPC cluster. For the three input files, PhyloD supports file partition by dividing the codon file. In the view of DAG, in the first stage, each node is assigned with one partition and the other two input files with the application instance of PhyloD. A process is created for each of the input partition to build a task. When the processes are invoked by the LINQ lambda expression, each task would be processed on a node in the cluster and computes part of the cross products of allele-codon pairs, p-values on and q-values, these values are all partial output. The program merges all the data sets produced by each process to get the final output file. In this case the PhyloD runs the input files on a cluster (Window HPC Cluster) rather than on client workstation to enhance the effect and short the computing time.

Figure 2: Shows the essential elements of the PhyloD-Dryad/DryadLINQ.



We propose a poster that describes Dryad architecture, its capabilities, and how it can be used to execute PhyloD application on a distributed framework. The poster will include an architecture diagram similar to that shown in Figure 1 and 2. We will include a performance evaluation chart showing the scalability of executing PhyloD on a cluster as against a stand-alone machine. We will also show the final validated result of the PhyloD application. This result will be merged from the partial results created by each Dryad job.

References

[1] The Microsoft Research webpage. http://research.microsoft.com/en-us/um/redmond/projects/MSCompBio/, 2010.
[2] Microsoft Research 2009, DryadLINQ Programming Guide, version 1.0.1.
[3] Microsoft Research 2009, Dryad and DryLINQ: An Introduction, version 1.0.1.
[4] DyradLNQ Tutorial webpage. http://salsahpc.indiana.edu/tutorial/dryadlinq-intro.html