TaxoGrid: Molecular Phylogeny on Garuda Grid

Amit Saxena, Sonal Dahale, Sankalp Jain, E. Ramakrishnan, Vivek Gavane, Renu Gadhari, Pankaj Vats, Sunitha Manjari K, Rashmi Mahajan and Rajendra Joshi
Bioinformatics Group, Centre for Development of Advanced Computing
Pune - 411007, Maharashtra, INDIA

Abstract

Molecular phylogeny is a fundamental aspect of understanding the evolution and is one of the downstream genome analysis pipelines. This pipeline has been incorporated in TaxoGrid portal and deployed on the Indian grid computing initiative GARUDA. TaxoGrid is a grid-based portal which implements the phylogenetic analysis, viz. ortholog detection using database similarity searching, multiple sequence alignment and phylogenetic reconstruction. The analysis is executed in an automated manner as a bioinformatics workflow and is capable of efficiently handling voluminous data owing to the availability of parallel implementation of tools over the vast compute resources of grid.

Keywords: *Embarrassingly parallel, scientific computing, molecular phylogeny, bioinformatics workflow, grid.*

1. Introduction

TaxoGrid is the implementation of phylogeny which finds evolutionary relationship between genes, genomes and organisms. It is a fundamental tool to understand the origin of biochemical pathways, regulatory mechanisms in cells as well as the development of complex systems [1]. One of the bottleneck for such studies is compute-intensive nature of the phylogenetic reconstruction as the search space depends on the number of characters. Character based methods like maximum parsimony and maximum likelihood scan each column of the alignment to arrive at the best tree topology while accommodating all the information [2],[3].

TaxoGrid exploits the need of compute intensive and high-throughput nature of Phylogeny analysis using a grid environment. It is based on embarrassingly parallelism principle which can distribute the data across the clusters available in a grid. Each data chunk is associated with a compute pipeline which executes in an independent manner over the available grid nodes. Each compute pipeline performs phylogenetic analysis using bioinformatics tools viz. ssearch (Smith Waterman), ClustalW and RaxML[1]. All such compute pipelines are spawned across the available clusters in the grid and are executed independently. The input is divided into subset and associated with a compute pipeline at the client end. Gridway[2] is used as a meta-scheduler for grid job submission. It is an open source meta-scheduling technology that enables large-scale, secure, reliable and efficient sharing of computing resources. The actual execution location is transparent to the user, as Gridway manages the jobs across the cluster based on scheduling policies and availability of resources in grid. Service oriented architecture (SOA) is the key technology that keeps client interface clean from the intricacies of the server side coding comprising of job submission to heterogeneous hardware platforms.

The phylogeny service can be called from the client using standard service protocols. The TaxoGrid client is a web based portal developed using J2EE technology. Phylogeny service is deployed using OPAL[3]; an open source framework, developed by National Biomedical Computation Resource. It is a wrapper around compute intensive scientific applications. It provides features such as scheduling, standards-based Grid security and data management in an easy-to-use and configurable manner. OPAL [4] supports DRMAA[4] libraries provided by Gridway to call APIs from Java code. Thus all the technologies work in synchronization to provide a simple user-friendly interface powered by a very powerful Grid based job execution at back-end. TaxoGrid has been deployed on Garuda [5] which is a collaborative grid infrastructure of Indian research community. The infrastructure is based on a nationwide grid of computational nodes, mass storage and scientific instruments. It aims to provide technology required to enable analysis of data and solving compute intensive problems. The backend network infrastructure is National knowledge network NKN which provides a high speed network backbone. The Grid has adopted a pragmatic approach for using the existing grid infrastructure and SOA-based technologies.

[1] http://sco.h-its.org/exelixis/software.html

[2] http://www.gridway.org

[3] http://www.nbcr.net/software/opal

[4] http://www.drmaa.org

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

2. Embarrassingly Parallel Approach

One of the important aspect of parallelism involves division of a bigger task into many smaller tasks where there is no dependency between two parallel tasks. In a scenario where the same kinds of operations are to be performed on a big dataset, it can be distributed to different processors which perform similar operations on these divided data in a parallel and independent manner. As the input for phylogeny is a set of genes or protein sequences from various organisms of interest along with a database for retrieving orthologs, there exists an inherent scope of parallelism which helps in developing the distributed parallel workflow for phylogenetic tree reconstruction. To tap on the availability of multiple resources available on the grid, the compute pipeline for the phylogeny workflow can be executed in parallel on various clusters available on the grid.

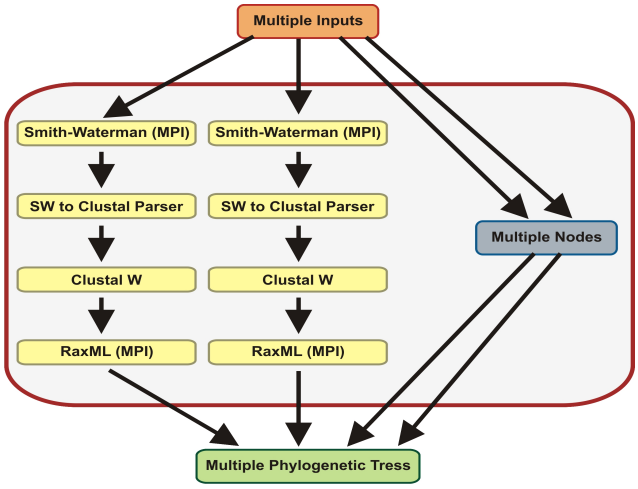

Fig. 1 TaxoGrid Process Pipeline over Grid.

3. Tools used in the Phylogeny Workflow

The tools used for molecular phylogeny are depicted as a bioinformatics workflow in Fig. 1. It consists of Smith Waterman (SW), ClustalW and RaxML pipeline. Every pipeline is independent of each other with its own set of inputs and database. Thus each pipeline is executed parallel on different nodes of Garuda grid infrastructure at different geographical locations transparent to the user executing the job. A customized parser is written to convert the output of SW into ClustalW readable format. The individual tools available in open domain are used very actively by the bioinformatics community for various research and analysis activities.

Orthologous genes for a given query gene set are obtained using Message Passing Interface (MPI) version of Smith-Waterman comparison included in the Fasta package version 35 [6] which implements the highly accurate and sensitive approach of dynamic programming for database similarity searching against the UniProt knowledge base Release 2011 04 [7]. The default cutoff criteria used to parse the pair wise alignments are E-value: 0.00001; % identity: 50; % overlap: 80.

The custom parser automates the parsing and also provides the user a handle to change these parameters. The orthologs thus obtained are subjected to multiple sequence alignment using the MPI-version 0.13 of ClustalW [8]. The multiple alignments are converted to interleaved format for compatibility with RAxML MPI based version 7.0.4 [9] for the tree regeneration.

4. TaxoGrid Architecture

The architecture of TaxoGrid is shown in Fig. 2. The technologies include Apache Tomcat, Apache Axis, Opal tool-kit and Gridway. Tomcat is a powerful web container used for deployment of Opal tool-kit and Axis Engine. Axis is a ready-made framework for constructing SOAP processes such as clients, servers and gateway. Axis adds more functionality than a SOAP engine; it also includes a simple stand alone server that plugs into servlet engines such as Tomcat. It has an extensive support for the Web Service Definition Language (WSDL) and emitter tool that generates Java classes from WSDL. Opal uses Axis Engine for implementing a ready-made general purpose scientific application service wrapper. Opal has a flexibility of choosing the underneath platform for the execution of scientific applications which can be a cluster or Globus [10] based grid or any resource that can be accessed using DRMAA APIs.

Changes were made in Opal source code to support Gridway based job submission using DRMAA libraries. This enhanced Opal layer with Gridway support hides the complexity of Grid-based job submission and takes onus of selecting the best available resources based on the policy configurations. TaxoGrid exploits the advantages of Opal and Gridway to dynamically allocate concurrent molecular phylogeny jobs to the available grid nodes.

Service oriented architecture (SOA) is the key technology that keeps the client interface clean from intricacies of the server side coding consisting of job submission to heterogeneous hardware platform.

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

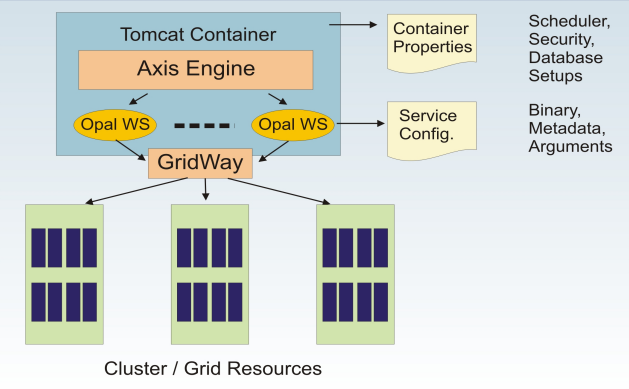

Fig. 2 TaxoGrid Architecture.

5. Implementation Details

TaxoGrid is implemented as a Grid service accessible by a Flex based client interface. The implementation details of TaxoGrid are mentioned in the points below:

5.1 Installation of phylogeny application pipeline on Garuda Grid

The phylogeny application pipeline consisting of Smith Waterman, ClustalW and RaxML has been compiled and installed on Garuda. Garuda resources need not be homogeneous and don't require separate compilation of the applications based on the processor, OS and libraries at different locations. Parsers are also written in bioinformatics friendly language like Perl and deployed with the pipeline code. The individual tools have been installed and configured on the various grid nodes of Garuda located at Bengaluru, Hyderabad, Chennai and other places.

5.2 Phylogeny service implementation and deployment

The phylogeny service is implemented as a single Java Class using Apache Axis. Opal toolkit is used as a wrapper to the TaxoGrid pipeline. For every application deployment one Opal configuration file is created. The application behavior parameters are stored in the configuration file and is passed as a parameter inside the deployment descriptor. Opal provides a configuration file with all the default values. By default, the job manager provided by Opal is Fork which can be changed to any of the supported job managers from the list. We have used DRMAA job manager to support Gridway based jobs. The Fig. 3 shows a snapshot of the configuration file used in Opal tool-kit configuration.

```
# full qualified class name (FQCN) of the job manager being used
# pal.jobmanager=edu.sdsc.nbcr.opal.manager.ForkJobManager
  opal.jobmanager=edu.sdsc.nbcr.opal.manager.DRMAAJobManager
# opal.jobmanager=edu.sdsc.nbcr.opal.manager.GlobusJobManager
# opal.jobmanager=edu.sdsc.nbcr.opal.manager.RemoteGlobusJobManager

## BEGIN: information for the DRMAA job manager
## ------------------------------------------------
# the parallel environment (PE) being used by DRMAA
drmaa.pe=mpich
## ------------------------------------------------
## END: information for the DRMAA job manager
```

Fig. 3 Opal Jobmanager Configuration Snapshot.

5.3 Client Interface development

The client is a web-based portal developed using J2EE framework. Fig. 4 shows a snapshot of the interface. It is very user friendly and graphically rich. Latest technologies such as Flex and AJAX are incorporated. It also supports browser based visualization of the generated phylogenetic trees.

Fig. 4 TaxoGrid Client Interface.

6. Benchmarking

The Mycobacterium case study was taken up to perform TaxoGrid benchmarking. TaxoGrid was used to reconstruct the phylogenetic trees of the entire proteome of Mycobacterium tuberculosis H37Rv [11], the causative agent of tuberculosis. Of the 3988 proteins of M. tuberculosis, orthologs were detected for 985 proteins using stringent filter criteria as defined earlier. As depicted in Fig. 5, the function of majority (~ 80%) of the conserved proteins is related to housekeeping activities like information storage, cellular processing, signaling and metabolism. The remaining 20% proteins are either conserved hypothetical proteins typically found only in genus Mycobacterium or are the antigenic proteins that are candidates for diagnostics/vaccine development.

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

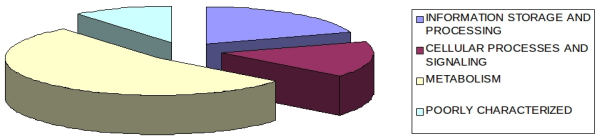

Fig. 5 Distribution of the M. Tuberculosis H37Rv genes according to COG database.

The reconstruction of phylogenetic trees for 985 proteins was carried out in triplicate i.e. , three runs each on 20, 15, 10 and 5 nodes (1 node has 8 computing cores) and average run time for each node was 346 minutes, 421 minutes, 651 minutes and 1144 minutes i.e., ~5 hours on 20 cores of GARUDA Grid as compared to ~99 hours using a regular desktop PC. The benchmark result are shown in Fig. 6 as a representation of decrease in execution time as the no. of computing cores increases.
Benchmarking was carried over nodes of Garuda Grid, which are geographically distributed across various Indian cities like Bengaluru, Hyderabad, and Chennai. The ~20 times speed-up achieved has enormous implications in reducing the time taken to process the genomics data.
The speed-up helps especially during time-crunch situations like epidemic scares of the avian-flu or H1NI outbreaks, wherein there is a lot of pressure to select the strains capable of offering protection in terms of vaccine candidates or to short-list genes for drug development.

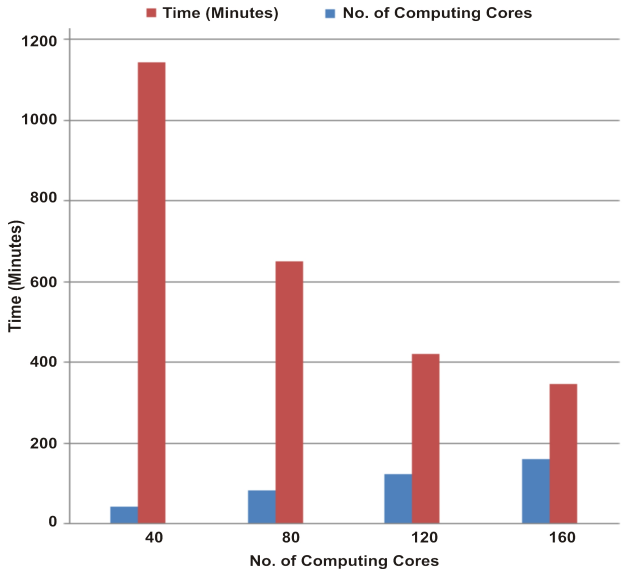

Fig. 6 TaxoGrid benchmark (time scale) on Garuda Grid.

7. Discussion

TaxoGrid is a unique 'first-of-a-kind' phylogeny pipeline over a grid. The phylogeny pipeline is implemented as a web service over grid thereby providing ease of availability and re-usability. Phylogeny service is deployed using OPAL open source framework which is a wrapper around scientific applications specifically compute intensive scientific applications. The phylogeny service can be called from client using the standard service protocols. A web-portal is developed using Flex technology, which provides an easy-to-use interface and hides the complexities associated with grid systems from the users. Optimum utilization of grid hardware is done by using parallel run of the computational pipelines using embarrassingly parallel approach. It enables rapid execution for given data along with the user friendliness of the interface. TaxoGrid's modular design enables enhancements as per the advances in technology. It is capable of replacing the underlying architecture with an all-together new infrastructure like cloud [12] without much change in the code.
The benchmarking data also supports the initiative of deployment of such a computational intensive application workflow. The benchmark result shows that the application is capable of consuming the enormous computation power of grid without early saturation, provided input data does not exhaust.

8. Conclusions

Molecular phylogeny a fundamental part for genome analysis is one of the most computationally-intensive applications in life sciences. This is due to an exhaustive search space, which can be searched using parallel computation. Phylogenetic studies have a direct impact in understanding the spread of disease during epidemics and evolution of pathogenic strains responsible for disease manifestation. TaxoGrid ensures the optimal use of the computational resources available at the disposal of a computational grid thereby providing a great opportunity for fast execution of the molecular phylogenetic reconstruction of the multiple genes.

Acknowledgments

The authors acknowledge the Department of Electronics & Information Technology, Ministry of Communications & Information Technology (Government of India) and Bioinformatics Resources & Applications Facility (BRAF) of C-DAC.

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

References

[1] M. Medina, "Genomes, phylogeny, and evolutionary systems biology," Proceedings of the National Academy of Sciences of the United States of America, vol. 102 Suppl, no. Suppl 1, pp. 6630–6635, 2005.
[2] M. Ott, J. Zola, A. Stamatakis, and S. Aluru, "Large-scale maximum likelihood-based phylogenetic analysis on the IBM BlueGene/L," Proceedings of the 2007 ACMIEEE conference on Supercomputing SC 07, vol. 8, no. 1, p. 1, 2007.
[3] S. Roch, "Toward extracting all phylogenetic information from matrices of evolutionary distances," Science, vol. 327, no. 5971, pp. 1376–1379, 2010.
[4] S. Krishnan, B. Stearn, K. Bhatia, K. Baldridge, W. Li, and P. Arzberger, "Opal: SimpleWeb Services Wrappers for Scientific Applications," 2006 IEEE International Conference on Web Services ICWS06, vol. 18, no. 22, pp. 823–832, 2006.
[5] B. B. P. Rao, S. Ramakrishnan, M. R. R. Gopalan, C. Subrata, N. Mangala, and R. Sridharan, "e-Infrastructures in IT: A case study on Indian national grid computing initiative GARUDA," Computer Science Research and Development, vol. 23, no. 34, pp. 283–290, 2009.
[6] W. R. Pearson and D. J. Lipman, "Improved tools for biological sequence comparison," Proceedings of the National Academy of Sciences of the United States of America., vol. 85, no. 8, pp. 2444–2448, 1988.
[7] T. U. Consortium, "Ongoing and future developments at the Universal Protein Resource." Nucleic Acids Research, vol. 39,no. Database issue, pp. D214–D219, 2011.
[8] K.-B. Li, "ClustalW-MPI: ClustalW analysis using distributed and parallel computing." Bioinformatics, vol. 19, no. 12, pp. 1585–1586, 2003.
[9] A. Stamatakis, "RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models." Bioinformatics, vol. 22, no. 21, pp. 2688–2690, 2006.
[10] I. Foster and C. Kesselman, "The Globus Toolkit," The Grid Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers Inc., pp. 259–278, 1999.
[11] J.-C. Camus, M. J. Pryor, C. Mdigue, and S. T. Cole, "Reannotation of the genome sequence of Mycobacterium tuberculosis H37Rv." Microbiology, vol. 148, no. Pt 10, pp. 2967–2973, 2002.
[12] B. Coombe, "Cloud Computing Overview," IT Architect, no. December, pp. 1–12, 2011.

Amit Saxena received MCA degree from Mohan Lal Sukhadia University, Udaipur, India, in 2003. His research interests include grid computing and Java related technologies.

Sonal Dahale received M. Sc. (Biochemistry) degree from DAVV University, Indore India, in 2006. Her research interests include molecular phylogeny and metabolic pathway reconstruction. She has published in PlosOne journal.

Sankalp Jain received M.Tech (Computer Science) degree from DAVV University, Indore India, in 2008. His research interests grid computing and perl programming.

E. P. Ramakrishnan received ME (Computer Science) degree from Anna University, Chennai, India, in 2008. His research interests include user interface design and implementation in Java.

Vivek Gavane received M.Tech (AI) degree from RJPV University, Bhopal India, in 2005. His research interests artificial intelligence and parallel programming.

Renu Gadhari received BE (Computer Science) degree from BAMU University, Aurangabad India, in 2006. Her research interests development in J2EE.

Pankaj Vats received M.Tech (Bioinformatics) degree from SASTRA University, Tamil Nadu, India, in 2007. His research interests include Genome Analysis and Machine learning.

Sunitha Manjari Kasibhatla received Advanced Diploma in Bioinformatics from University of Pune in 2000. Her research interests include comparative genomics. She has published 6 research articles in journals like PlosOne, Nucleic Acids Research, BMC Bioinformatics, Proteins Peptide Letters, Journal of Virology and Journal of Bacteriology.

Rashmi Mahajan received M. S. (Computer Science) degree from UNCC University, North Carolina USA, in 1998. Her research interests software design and development using OOPS and Java technology.

Rajendra Joshi received his PhD in Biochemistry from National Chemical Laboratory, Pune in 1994. His research interests include use of high performance parallel computers for biological research. He has ~25 papers published in international peer reviewed journals.

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.