Scalable Programming and Algorithms for Data Intensive Life Science Applications

Judy Qiu
School of Informatics and Computing
Indiana University, Bloomington.
xqiu@indiana.edu

Cloud computing [1] offers new approaches for scientific computing that leverage the major commercial hardware and software investment in this area. Closely coupled applications are still unclear in clouds as synchronization costs are still higher than on optimized MPI machines. However loosely coupled problems are very important in many fields and can achieve good cloud performance even when pleasingly parallel steps are followed by reduction operations as supported by MapReduce. It appears that many data analysis problems fit the MapReduce paradigm but there is no definitive analysis here. For example analysis of LHC (Large Hadron Collider) data corresponds to a data selection step followed by forming histograms; this naturally corresponds "perfectly" to the MapReduce paradigm. In Life Science, "all-pairs" applications like BLAST can run well with MapReduce but are particularly simple corresponding to "pleasingly parallel" or "map only" structure. Finally there are applications involving steps like the dimension reduction or clustering algorithms illustrated below where pleasing parallel operations (such as alignment and sequence distance computation) and followed by data mining steps involving iterative operations – such as those present in matrix algebra. Such iterative algorithms are the mainstay of large scale scientific computing and are linked directly to data with data assimilation in weather and climate area [2]. Even in the "birthplace" of MapReduce – Information Retrieval – the Page Rank algorithm needs iterative MapReduce. Thus we pose the following questions.

1) What data analysis problems in science can use clouds and/or MapReduce
2) What data analysis problems need iterative algorithms poorly supported by basic MapReduce
3) What are tradeoffs in performance, usability, flexibility and fault tolerance between MPI and Iterative MapReduce
4) What are requirements for workflow systems needed to support complicated science data processing.

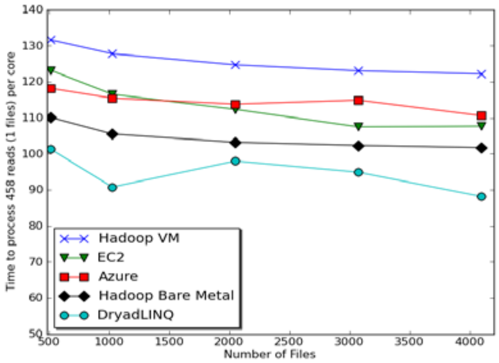

Figure 1 Time to process a single biology sequence file (458 reads) per core with different frameworks

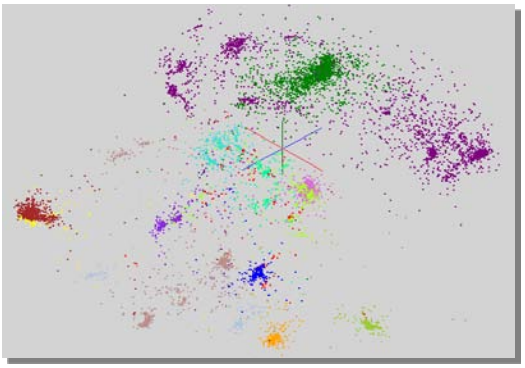

Figure 2 Results of 17 clusters for full sample using Sammon's version of MDS for visualization.

Here we examine different ways for using clouds for pleasingly parallel applications where we have compared five different approaches using two biomedical applications. We look at the cloud infrastructure service based virtual machine utility computing models of Amazon AWS and Microsoft Windows Azure; MapReduce based computing frameworks Apache Hadoop (deployed on raw hardware as well as on virtual machines) and Microsoft DryadLINQ. We compare performance showing strong variations in cost between different EC2 machine choices and comparable performance between the utility computing (spawn off a set of jobs) and managed parallelism

(MapReduce). The MapReduce approach offered the most user friendly approach. Typical results [3] are shown in Fig. 1.

A typical bioinformatics pipeline of Smith-Waterman distance Computation, Deterministic Annealing Clustering and MDS visualization is shown below in Fig. 3, which can give results such as Fig. 2 where the results of 30,000 Metagenomics sequences in 3D are shown. The visualization uses dimension reduction where we have implemented two powerful methods GTM (Generative Topographic Mapping) and MDS (Multidimensional Scaling) [4] [5].

Only MDS can be used for DNA sequence visualization as GTM requires a vector representation of original high dimensional data whereas MDS only requires the N by N matrix of dissimilarity scores between sequences. Multiple Sequence Alignment needed to obtain a uniform vector representation of sequences is typically infeasible. The distance matrix calculation needed by MDS is very suitable for cloud implementation as the computations are independent. However both clustering and MDS require parallel implementation as they are expensive $O(N^2)$ computations; the run time of these on a 768 core cluster is about 3 hours for 30,000 sequences with a speed up of 500. These parallel implementation run poorly on clouds or MapReduce as their iterative algorithms require the long running processes and low latency of MPI. Thus we see hybrid cluster-cloud architectures as needed for this class of problem where a complete workflow is gotten by linking separate services in clouds and closely coupled clusters.

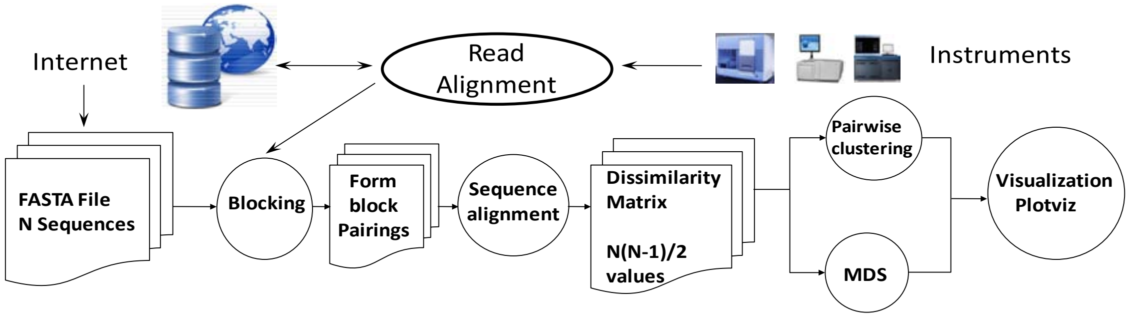

Figure 3 Pipeline for analysis of metagenomics Data

We have developed new interpolation algorithms for both MDS and GTM which can exploit clouds and MapReduce for the dominant part of the computation for large problems. These perform a basic dimension reduction for a sample of the data (20,000-100,000 points) which runs using MPI on a cluster; the remaining points are interpolated which is a pleasingly parallel cloud application. We will present performance results for run time and quality of dimension reduction.

Alternatively we have extended MapReduce in an open source system, Twister [6] [7], that supports iterative computations of the type needed in clustering, MDS and GTM. This programming paradigm is attractive as it supports all phases of the pipeline in Fig. 1. We present performance comparisons between MPI, MapReduce and Twister on kernel applications such as matrix multiplication as well as the core services of Fig. 1 in [6]. Other approaches to Iterative MapReduce are Pregel [8] HaLoop [9] and Spark [10]

References

[1] Jaliya Ekanayake, Xiaohong Qiu, Thilina Gunarathne, Scott Beason, Geoffrey Fox High Performance Parallel Computing with Clouds and Cloud Technologies to appear as a book chapter to Cloud Computing and Software Services: Theory and Techniques, CRC Press (Taylor and Francis), ISBN-10: 1439803153.

[2] How supercomputers and hybrid workflows helped beat tornadoes to the chase, iSGTW International Science Grid This Week September 1 2010.

[3] Thilina Gunarathne, Tak-Lon Wu, Judy Qiu, and Geoffrey Fox, Cloud Computing Paradigms for Pleasingly Parallel Biomedical Applications, Proceedings of Emerging Computational Methods for the Life Sciences Workshop of ACM HPDC 2010 conference, Chicago, Illinois, June 20-25, 2010.

[4] Seung-Hee Bae, Jong Youl Choi, Judy Qiu, Geoffrey Fox Dimension Reduction and Visualization of Large High-dimensional Data via Interpolation, Proceedings of ACM HPDC 2010 conference, Chicago, Illinois, June 20-25, 2010.

[5] Jong Youl Choi, Seung-Hee Bae, Judy Qiu, Geoffrey Fox, Bin Chen, and David Wild, Browsing Large Scale Cheminformatics Data with Dimension Reduction, Proceedings of Emerging Computational Methods for the Life Sciences Workshop of ACM HPDC 2010 conference, Chicago, Illinois, June 20-25, 2010.

[6] Jaliya Ekanayake, Hui Li, Bingjing Zhang, Thilina Gunarathne, Seung-Hee Bae, Judy Qiu, Geoffrey Fox Twister: A Runtime for Iterative MapReduce, Proceedings of the First International Workshop on MapReduce and its Applications of ACM HPDC 2010 conference, Chicago, Illinois, June 20-25, 2010.

[7] Twister software at http://www.iterativemapreduce.org/

[8] Grzegorz Malewicz, Matthew H. Austern, Aart J. C. Bik, James C. Dehnert, Ilan Horn, Naty Leiser, and Grzegorz Czajkowski Pregel: A System for Large-Scale Graph Processing, Proceedings of the 2010 international conference on Management of data, Indianapolis, Indiana, USA Pages: 135-146 2010

[9] Yingyi Bu, Bill Howe, Magdalena Balazinska, Michael D. Ernst HaLoop: Efficient Iterative Data Processing on Large Clusters, Proceedings of the VLDB Endowment, Vol. 3, No. 1, The 36th International Conference on Very Large Data Bases, September 1317, 2010, Singapore.

[10] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, Ion Stoica Spark: Cluster Computing with Working Sets poster at http://radlab.cs.berkeley.edu/w/upload/9/9c/Spark-retreat-poster-s10.pdf