Data Intensive Applications on Clouds

Geoffrey Fox
School of Informatics and Computing
Indiana University
Bloomington IN 47408, USA
gcf@.indiana.edu

ABSTRACT

The cyberinfrastructure supporting science will include large-scale simulation systems headed to exascale combined with cloud like systems supporting data intensive and high throughput computing, pleasingly parallel jobs and the long tail of science. Clouds offer economies of scale, elasticity supporting real time and interactive use and powerful new programming models such as MapReduce. We stress that iterative extensions of MapReduce will be necessary to get good performance on for several data mining (analytics) applications. We give several illustrations mainly from bioinformatics. We suggest that the data deluge implies a corresponding increase in the computational resources needed to support analysis and this suggests new architectures for large scale data repositories.

Categories and Subject Descriptors

D.1.3 [Software Programming Techniques]: Concurrent Programming

General Terms

Algorithms, Design, Experimentation, Performance

Keywords

mapreduce, iterative mapreduce, data intensive science, data analytics, exascale, cloud, performance, pleasingly parallel, high throughput computing

1. INTRODUCTION

The importance of simulation in science is well established with large programs, especially in Europe, USA, Japan and China supporting it. The requirements and consequent architecture of large scale supercomputers is well understood although there are important challenges in meeting performance goals seen by international drives to reach first petascale (starting 15 years ago) and now exascale performance. Performance on closely coupled parallel simulations drives both hardware (low latency high bandwidth networks, high flop CPU's) and software that can exploit it. Grids covered both the linkage of such computers and broader computing facilities. This has spurred rise in high throughput computing, workflow and service oriented architectures (Software as a service); concepts of lasting value. Major data intensive applications like LHC data analysis highlighted the many important pleasingly parallel applications that these were a major driver of Grid and many task systems. Now the strong commercial interest is driving clouds and we can ask how they fit in? Clouds offer on-demand service (elasticity), economies of scale from sharing, a plethora of new jobs making clouds attractive for students & curricula and several challenges including security. Clouds lie in between grids and HPC supercomputers in their synchronization costs so all the high throughput jobs run on grids should perform well on clouds. In this paper, we suggest that there is a class of explicitly parallel jobs that do not need the highest performance interconnect and will have good performance and good user experience on clouds. We describe this in an application analysis in section2. Of course, HPC supercomputers can do "all applications" subject to reservations about limited I/O (disk) capabilities. However, they are overkill for many problems and it seems better to reserve such machines for the high-end applications that require them and use commodity cloud environments when appropriate. We stress that clouds offer not just a new humongous data center architecture but striking new software models spurred by the competitive Platform as a Service PaaS market. In section 2 we focus on the possibilities suggested by MapReduce.

2. MAPPING APPLICATIONS TO CLOUDS

Previously I discussed mapping applications to different hardware and software in terms of 5 "Application Architectures"[1] mainly aimed at simulations and extended it to data intensive computing [2, 3]. One category, synchronous, was popular 20 years ago but is no longer significant. It describes applications that can be parallelized with each decomposed unit running the identical machine instruction at each time. Another category, asynchronous is typically not important in practical computational science and engineering. There was also a category of metaproblems, which describe the domain supported by workflow with coarse grain interlinked components. The other categories were pleasingly parallel (essentially independent) and loosely (bulk) synchronous which are critical application classes that possibly combined in metaproblems describe the bulk of eScience. As mentioned above, pleasingly parallel problems whether parameter searches for simulations or analysis of independent data chunks (as in LHC events) are very suitable for clouds. Loosely synchronous problems include partial differential equation solution and particle dynamics and after parallelization, consist of a succession of compute-communication phases.

Looking at data intensive applications we can re-examine the pleasingly parallel and loosely synchronous category as shown in figure 1 above. This introduces map-only (identical to pleasing parallel), and separates off MapReduce and Iterative MapReduce classes from the large loosely synchronous class whose remaining members are the last sub category d) on the right of figure 1. This area requires HPC architectures with low latency high bandwidth interconnect. The MapReduce class b) consists of a single map (compute) phase followed by a reduction phase such as gathering together the results of queries following an Internet search or LHC data analysis (histogram) of different datasets. As implemented in Hadoop, one would normally communicate between Map and Reduce phases by writing and reading files. This leads to excellent fault tolerance and dynamic scheduling features. At SC11, there was some buzz in favor of data analytics and Hadoop but that this is not clearly reasonable as many data analysis (mining) applications involve kernels that do not fit Map only or MapReduce categories. Many algorithms including those with linear algebra (needing to be parallelized) fall into the

DataCloud-SC'11, November 14, 2011, Seattle, Washington, USA.
ACM 978-1-4503-1144-1/11/11.

category c) Iterative MapReduce in figure 1. Problems in this category consist of multiple (iterated) Map phases followed by reduction or collective operation communication phases. They do not have the many local communication messages typically needed in parallel simulations shown in fig 1d) but rather larger collective operations mixing compute and communication. We do not expect traditional MapReduce to be broadly useful but the Iterative extension is much more promising but the breadth of its applicability needs much more study. Iterative MapReduce is a programming model that can have the performance of MPI and the fault tolerance and dynamic flexibility of the original MapReduce. Open source Java Twister[4, 5] and Twister4Azure[6, 7] have been released as an Iterative MapReduce framework. Figure 2 compares Twister4Azure with Amazon and a classic HPC configuration on a map-only case while figure 3 shows Azure4Twister having a smooth execution structure and modest communication overhead (the uncolored gaps) on a parallel data analytics algorithm. We expect the commonly used expectation maximization (EM) approach used for example in Multidimensional Scaling MDS application of fig 3, to be particularly attractive for iterative MapReduce as EM can have large compute/communication ratios. Category c) extends the clear value of clouds in the categories a) and b) of figure 1.

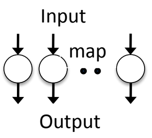
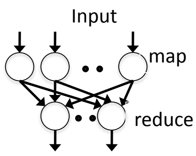
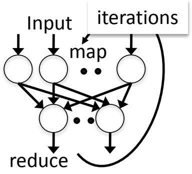
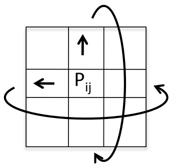
a) Map Only	**b) Classic MapReduce**	**c) Iterative MapReduce**	**d) Loosely Synchronous**
			
CAP3 Analysis Smith Waterman Distces Parametric sweeps PolarGrid Matlab data analysis	High Energy Physics (HEP) Histograms Distributed search Distributed sorting Information retrieval	Expectation maximization Clustering e.g. Kmeans Linear Algebra Multidimensional Scaling Page Rank	Many MPI scientific applications such as solving differential equations and particle dynamics
⟵ **Domain of MapReduce and Iterative Extensions** ⟶			**MPI**

Fig 1: Four applications classes and their mapping to run time/ programming models

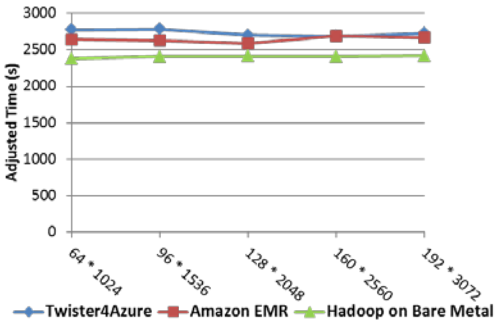


Fig 2: A Map Only example pairs sequence distances

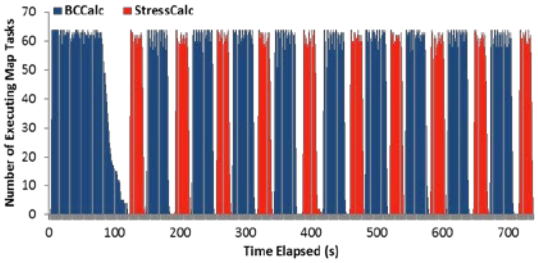


Fig 3: Parallel MDS on Azure4Twister showing communication (white) and two compute map phases

3. CLOUDS AND REPOSITORIES

It is traditional to set up data repositories for large observational projects. Examples are EOSDIS (Earth Observation), GenBank (Genomics), NSIDC (Polar science), and IPAC (Infrared astronomy). The fourth paradigm implies an increase in data mining (analytics) based on such data and this implies repositories need computing as well as data. We also expect that one should bring the computing to the data and not vice versa. Thus we do not expect researchers to download large petabyte data samples to their local cluster; rather we expect repositories to be associated with cloud resources (as cheapest and elastic) that allow data analytics on demand. Again further work is needed here. Some questions include the data storage architecture (database or NOSQL) and how one supports mining of multidisciplinary science involving data from different fields stored in different clouds.

5. ACKNOWLEDGMENTS

Our work is partially supported by Microsoft and by the National Science Foundation under the FutureGrid Grant No. 0910812.

6. REFERENCES

1) Fox, G.C., R.D. Williams, and P.C. Messina, *Parallel computing works!* 1994: Morgan Kaufmann Publishers,

2) calculating all Jaliya Ekanayake, Thilina Gunarathne, Judy Qiu, Geoffrey Fox, Scott Beason, Jong Youl Choi, Yang Ruan, Seung-Hee Bae, and Hui Li, *Applicability of DryadLINQ to Scientific Applications*. January 30, 2010, Community Grids Laboratory, Indiana University.

3) Judy Qiu, Jaliya Ekanayake, Thilina Gunarathne, Jong Youl Choi, Seung-Hee Bae, Yang Ruan, Saliya Ekanayake, Stephen Wu, Scott Beason, Geoffrey Fox, Mina Rho, and H. Tang, *Data Intensive Computing for Bioinformatics*. December 29, 2009.

4) SALSA Group. *Iterative MapReduce*. 2010 [accessed 2010 November 7]; Twister Home Page Available from: http://www.iterativemapreduce.org/.

5) J.Ekanayake, H.Li, B.Zhang, T.Gunarathne, S.Bae, J.Qiu, and G.Fox, *Twister: A Runtime for iterative MapReduce*, in *Proceedings of the First International Workshop on MapReduce and its Applications of ACM HPDC 2010 conference June 20-25, 2010*. 2010, ACM. Chicago, Illinois.

6) *Twister for Azure*. [accessed 2011 May 21]; Available from: http://salsahpc.indiana.edu/twister4azure/.

7) Thilina Gunarathne, Bingjing Zhang, Tak-Lon Wu, and Judy Qiu, *Portable Parallel Programming on Cloud and HPC: Scientific Applications of Twister4Azure*, in *IEEE/ACM International Conference on Utility and Cloud Computing UCC 2011*. December 5-7, 2011. Melbourne Australia. http://www.cs.indiana.edu/~xqiu/scientific_applications_of_twister4azure_ucc_17_4.pd