1. Introduction

The importance of simulation is well established with large programs, especially in Europe, USA, Japan and China supporting it in a variety of academic and government initiatives. 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 3 we focus on the possibilities suggested by MapReduce.

The term cloud is being in many ways so let's first define a public data center model that describes the major offerings of Microsoft, Amazon and Google. Their data centers are composed of containers of racks of servers which number between 10,000 and a million. Each server has 8 or more cpu cores and around 64GB of shared memory and one or more terabyte local disk drives. GPUs or other accelerators are not common. There is a network that allows messages to be routed between any two servers, but the bisection bandwidth of the network is very low and the network protocols implement the full TCP/IP stack so that every server can be a full Internet host with optimized traffic between users on the Internet and the servers in the cloud. In contrast supercomputer networks minimize interprocessor latency and maximize bisection bandwidth. Application data communications on a supercomputer generally take place over specialized physical and data link layers of the network and interoperation with the Internet is usually very limited.

2. A Cloud Defined

Each server in the data center is host to one or more virtual machines and the cloud runs a "fabric controller" which manages large sets of VMs fort scheduling and fault tolerance across the servers and acts as the operating system for the data center. An application running on the data center consists of one or more complete VM instances that implement a web service. The basic unit of scheduling involves the deployment of one or more entire operating systems, which is much slower than installing and starting an application on a running OS. Most large scale cloud services are intended to run 24x7, so this long start-up time is negligiblen although running a "batch" application on a large number of servers can be very inefficient because of the long time it may take to deploy all the needed VMs. Data in a data center is stored and distributed over many spinning disks in the cloud servers. This is a very different model than found in a large supercomputer, where data is stored in network attached storage. Local disks on the servers of supercomputers are not frequently used for data storage.

There are more types of clouds than is described by this public data center model. For example, to address a technical computing market, Amazon has introduced a specialized HPC cloud that uses a network with full bisection bandwidth and supports GPGPUs. The major commercial clouds offer higher level capabilities -- commonly termed Platform as a Service PaaS – built on a basic scalable IaaS Infrastructure as a Service. For technical computing, important platform components include tables, queues, database, monitoring, roles (Azure), and the cloud characteristic of elasticity (automatic scaling). MapReduce, which is discussed below, is another major platform service offered by these clouds. Currently the different clouds have different platforms although the Azure and Amazon platforms have many similarities. The Google Platform is targeted at scalable web applications and not as broadly used in technical computing community as Amazon or Azure, but it has been used on some very impressive projects. We expect more academic interest in PaaS as the value of platform capabilities become clearer.

"Private clouds" are small dedicated data centers that have various combinations of the properties above and typically use one of the four major open source (academic) cloud environments Eucalyptus, Nimbus, OpenStack and OpenNebula (Europe) which focus at the IaaS level with interfaces similar to Amazon. FutureGrid is an NSF research testbed for cloud technologies and it operates a grid of cloud deployments running on modest sized server clusters with support for all four academic IaaS. Private clouds do not fully support the interesting platform features of commercial clouds. Open source Hadoop and Twister offer MapReduce features similar to those on commercial cloud platforms and there are open source possibilities for platform features like queues (RabbitMQ, ActiveMQ) and distributed data management system (Apache Cassandra). However, there is no complete packaging of PaaS features available today for academic or private clouds. Thus interoperability between private and commercial clouds is currently only at IaaS level where it is possible to reconfigure images between the different virtualization choices and

there is an active cloud standards activity. The major commercial virtualization products such as VMware and Hyper-V are also important for private clouds but also do not have built-in PaaS capabilities.

3. Mapping Applications to Clouds

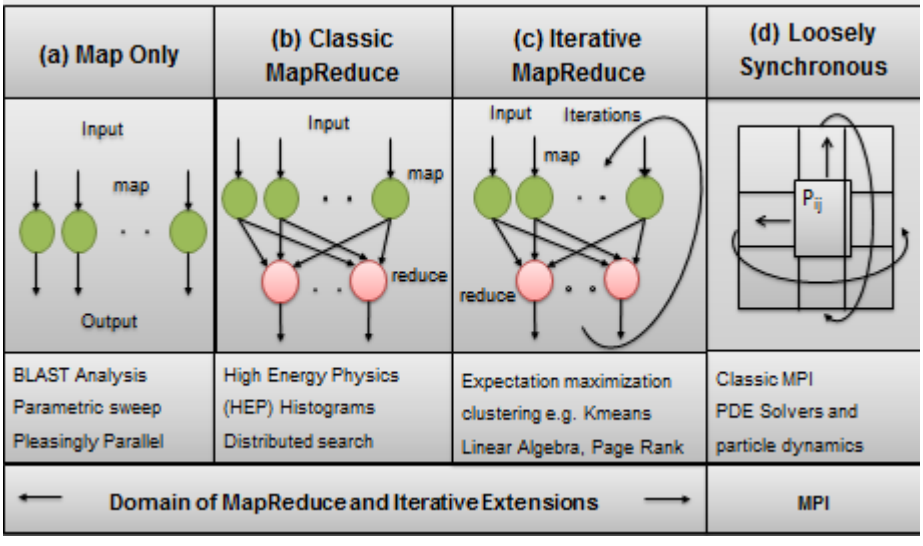

Fig 1: Forms of Parallelism and their application on Clouds and Supercomputers

Previously we 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.

Clouds naturally exploit parallelism from multiple users or usages. The Internet of things will drive many applications of the cloud. It is projected that there will soon be 50 billion devices on the Internet. Most will be small sensors that send streams of information into the cloud where it will be processed and integrated with other streams and turned into knowledge that will help our lives in a million small and big ways. It is not unreasonable for us to believe that we will each have our own cloud-based personal agent that monitors all of the data about our life and anticipates our needs 24x7. The cloud will become increasing important as a controller of and resource provider for the Internet of Things. As well as today's use for smart phone and gaming console support, "smart homes" and "ubiquitous cities" and the current AFRL project build on this vision. We expect a growth in these areas with emergence of cloud supported/controlled robotics.

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 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.

3. CLOUDS AND REPOSITORIES

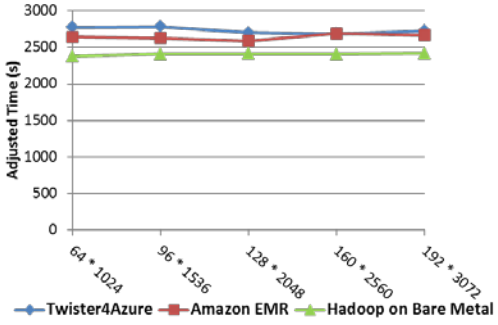

Fig 2: A Map Only example pairs sequence distances

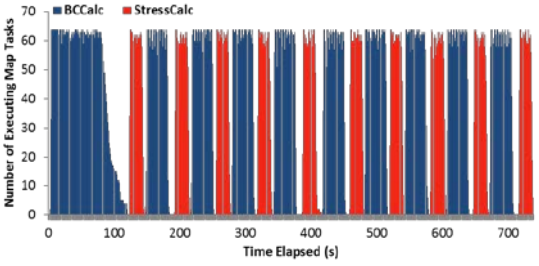

Fig 3: Parallel MDS on Azure4Twister showing communication (white) and two compute map phases

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.

4. Cloud Research Issues

We list areas where is substantial research activity and where we can expect major changes.

- New applications such as Biomedical and bioinformatics applications where cloud architecture brings special challenges in the area of privacy (see later). Furthermore, Clouds have been attractive platforms for these applications as they are emerging big data areas and there is less history in using existing platforms.
- Sensor webs studied in this project are another emerging area where elastic nature of Clouds is well suited for the often bursty nature of sensor data.
- Big data applications based on new MapReduce or Iterative MapReduce environments are attractive on Clouds and result in broad research areas include addressing both programming and storage challenges. Latter include SQL and NOSQL models and the reconciliation of distributed data and centralized cloud computing
- Scheduling models optimized for MapReduce and for other Cloud usage modes such as scalable sensor webs (Sensor Grids or Clouds) where one has Clouds controlling and supporting a distributed Grid of sensors.
- Optimizing the run time features and performance for MapReduce and Iterative MapReduce. This includes new reduction primitives, polymorphic implementation on different systems with for example, exploitation of high performance networks as in classic MPI research.
- Support of federation of clouds and cloud bursting (typically the linkage of private and public Clouds) and on-demand cloud federation.
- New storage models such as data parallel HDFS and Hbase (Bigtable).
- NOSQL table structures such as Cassandra and commercial approaches such as Amazon SimpleDB and Azure Table.
- Economic models for an ecosystem with multiple cloud systems and CI.
- Research on Cloud software stacks. There is research at all levels of the software stack with two rather different emphasis areas. Research on systems that provide basic virtual machine provisioning, deployment and management. This includes Eucalyptus, Nimbus, OpenStack and OpenNebula with virtual networking as a distinct activity. At the other end are integration of capabilities to provide rich Platform-as-a-Service as offered by major commercial systems. Concepts such as appliances provide novel ways of delivering these capabilities.
- Clouds tend to achieve scalability by allowing faults. Research is needed on both, how to expose faults to users as well as services to build fault tolerant applications. Most research in HPC tends to be on forbidding faults; however Clouds highlight a different philosophy with resilient applications running on faulty systems.
- Green IT is naturally synergistic with Clouds and related research includes examining the impact of Cloud features on power use, including the cost of powering idle machines supporting elastic clouds as well as a application aware approaches to power management.

Security policies and mechanisms: Clouds tend to emphasis the need for quality security mechanisms due to the sharing of storage and computing. One research area investigates hybrid architectures with algorithms broken into two; a low cost but non privacy preserving part running on an intrinsically secure private clouds, and a time consuming but privacy preserving part executing on a public cloud. Genomic data (human) and other health records are demanding here. The concept of differential privacy and health data anonymization is an active research topic. As well as basic security for computing and storage there is research on privacy preserving search with the elegant but time consuming concept of Homomorphic Encryption which allows encrypted data to be searched by encrypted queries.

Standards: There are many important standard activities, from those specifying the basic virtual machine structure to higher-level standards defining the PaaS environment, for example, queue and table structures. Although there is some support for these standards – such as OCCI (from OGF) in OpenNebula and OpenStack – this area is still under development. NIST and IEEE are playing leadership roles.

5. 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