Scaling-out CloudBLAST: Combining Technologies to BLAST on the Sky

Andréa Matsunaga, Maurício Tsugawa and José Fortes
Advanced Computing and Information Systems Laboratory
Department of Electrical and Computer Engineering, University of Florida
PO Box 116200, Gainesville, FL, 32611-6200, USA
{ammatsun, tsugawa, fortes}@ufl.edu

Abstract—In a future where multiple cloud providers coexist, end users will have the option to select machines from multiple clouds according to the type and cost of offered resources that best suit their application. The resulting distributed infrastructures, which we refer to as sky-computing systems, will be inherently disconnected and heterogeneous with respect to performance. In order to efficiently scale-out an embarrassingly parallel application in this environment, existing technologies need to be combined and extended. This demonstration illustrates and end-to-end approach to sky computing. In order to achieve scalable management and performance, several technologies and techniques are combined, namely, an IaaS cloud toolkit (Nimbus) to create virtual machines (VMs) on demand, a virtual networking middleware (ViNe) to connect VMs, the MapReduce framework (Hadoop) for parallel fault-tolerant execution of an unmodified application, and a skewed task distribution technique to deal with resource imbalance. A bioinformatics application (BLAST) is used to demonstrate the scalability of this type of sky-computing environment.

1. Introduction

Currently, many cloud providers offer compute cycles as Infrastructure-as-a-Service (IaaS). There are commercial offerings exemplified by the Amazon EC2 and academic research projects such as the Nimbus cloud [1]. Typically, end users are offered a collection of virtual machines (VMs) with full administrative privileges. VMs have performance profiles that differ from one another, even when hosted by a single provider due to the following reasons:

- The servers used by cloud providers are not homogeneous (e.g., EC2 offers instances that differ in the number of compute units, and when servers are added to a farm, new models are used).
- VMs may share a single host and the load of a VM can negatively impact other VMs.

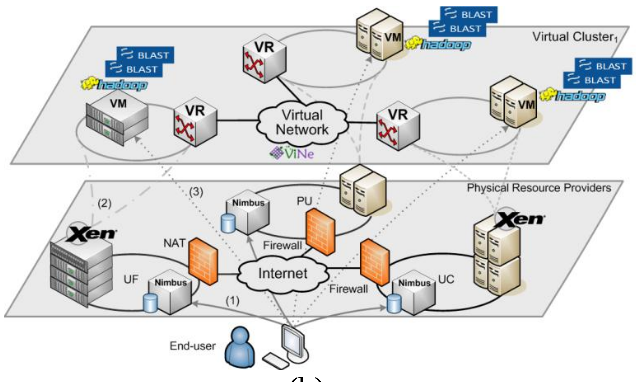
To efficiently use large numbers of heterogeneous cloud resources, several challenges need to the addressed: the parallel execution of applications, inter-cloud communication, inter-cloud management, efficient distribution of tasks, simple integration of application updates, and tolerance to node-failures. The proposed approaches, while generally applicable, are best presented in the context of a representative application.

2. CloudBLAST

CloudBLAST [2] combines virtualization technologies and tools (Xen, Nimbus, ViNe, and Hadoop) to deploy, on-demand, a distributed virtual cluster across multiple clouds. It is demonstrated using resources in 3 sites: at the University of Chicago (UC), at the University of Florida (UF) and at Purdue University (PU) as depicted in Figure 1. In this setup, a well-known bioinformatics application (BLAST) was shown to scale well using the Hadoop MapReduce framework to distribute tasks. In fact, its performance is slightly better than the leading MPI implementation of BLAST.

	UC	UF	PU
Xen VMM	3.1.0	3.1.0	3.0.3
CPU architecture	AMD Opteron	Intel Xeon Prestonia	Intel Xeon Irwindale
CPU clock	2.2 GHz	2.4 GHz	2.8 GHz
CPU cache	1 MB	512 kB	2 MB
VCPUs per node	2	2	2
Memory	3.5 GB	3.5 GB	1.5 GB
Networking	Public	Private	Public
Speed Factor	1.184	1	1.24

(a)



(b)

Figure 1: (a) Characteristics of virtual instances from each cloud provider and (b) overall architecture and operation of the system for preparing (1), deploying (2), and running BLAST (3) on a Xen-based virtual cluster interconnected through ViNe.

3. CloudBLAST: scale-out challenges

There are several interrelated factors that affect the scalability of applications, especially when crossing domains over a wide-area network. Applications might not have been parallelized, VMs in different providers will have distinct performance profiles, inter-site communication may not be present, and potential for failure increases as the number of resources involved grows.

In terms of parallelization, massively parallel applications that can be divided into parts with minimal data dependencies are best suited for sky computing. Many applications from the bioinformatics field are included in this class of applications, Basic Local Alignment Search Tool (BLAST) being a well-known example. A mechanism based on the MapReduce paradigm that enables the parallel execution of these applications by dividing the input without modifying the sequential binary while offering fault-tolerance has been developed and described in [2].

Inter-cloud communication is vital to enable the use of resources across multiple providers. Overlay/virtual network technologies are of key importance to establish communication among VMs guarded by firewalls in different sites. The ViNe project exemplifies such technologies – inter-cloud networking challenges and mechanisms to address them are described in [3].

Inter-cloud management allows the end user to control and configure resources from different providers so that they can be used together to solve a single problem. Typical examples include exchanging network information, establishing master-slave relationships and defining file servers. The Nimbus context broker service [4] used in this demo enables the establishment of a secure channel for exchanging information among appliances to form a virtual cluster.

In the case when homogeneous resources are available, a problem is typically broken into sub-tasks of the same size and distributed to workers. However, when heterogeneous resources are present, the worst performing node dictates the time to solve a problem (the time perceived by the end users), since all sub-tasks must finish. A simple yet powerful solution is to divide work in very small sub-tasks so that each node receives an amount of sub-tasks that is proportional to its performance. The drawback is that, in general, small sub-tasks increase the number of operations needed for scheduling and management, potentially leading to poor overall system performance. Furthermore, in some situations it is not possible to break the work in as many sub-tasks as desired.

Making use of each node's performance profile and dividing the problem accordingly is another alternative. However, this is challenging because (a) very accurate predictions of every node's processing time for a given application configuration are needed and (b) the performance profile can change at run-time (e.g., due to VMs sharing the same resource or due to CPU's dynamic frequency scaling as in Intel Turbo Boost technology). Several machine-learning algorithms and different selections of application-input features have been investigated in [5] for predicting application resource usage, and applying this knowledge to better distribute tasks is the scope of ongoing work. In this demo, a skewed (i.e., bi-modal) distribution of tasks is shown to allow sub-tasks better fit idle slots while avoiding a burst of very small jobs.

4. Scale-out performance

Network virtualization overheads and wide-area network performance (much lower compared to LANs) did not significantly affect the scale-out performance. This is due to the fact that the ViNe technology operates with low overheads, and also because CloudBLAST does not require high-performance networks.

Making use of the setup described previously, BLAST version 2.2.18 was executed with an input consisting of 960 nucleotide sequences to be matched against the 2007 NCBI non-redundant (NR) protein database (3.5GB of data). Results in Figure 2 show the viability of sky computing, as its scalability is comparable to that of a single site.

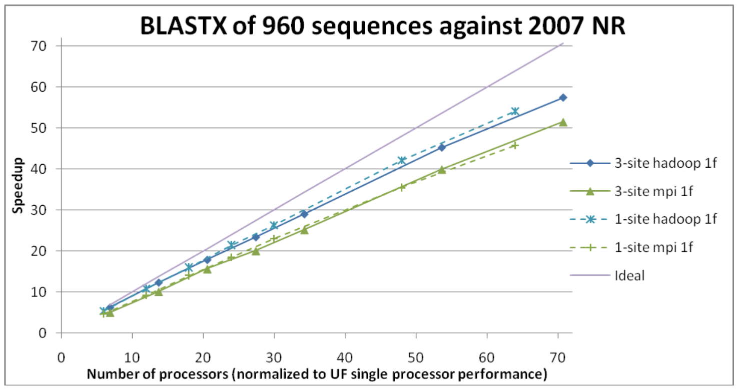


Figure 2. Speedup performance of BLAST when processing 960 sequences against the 2007 NR database. Both Hadoop and MPI implementations scale well when comparing the use of a single cloud provider with the use of three ViNe-connected cloud providers. In addition, Hadoop-based runs were able to complete in the presence of worker failures, while MPI-based runs required full restarts.

To deal with the performance imbalances among resources, skewed task distribution allowed smaller jobs to fill idle slots and slower resources to finish closer to fast resources, leading to better resource utilization. Figure 3 shows that the resources with lower performance (UF) can dictate the time to completion of a job even when each worker receives four tasks on average (top graph), whereas the skewed distribution allows all workers to finish closer to each other (bottom graph).

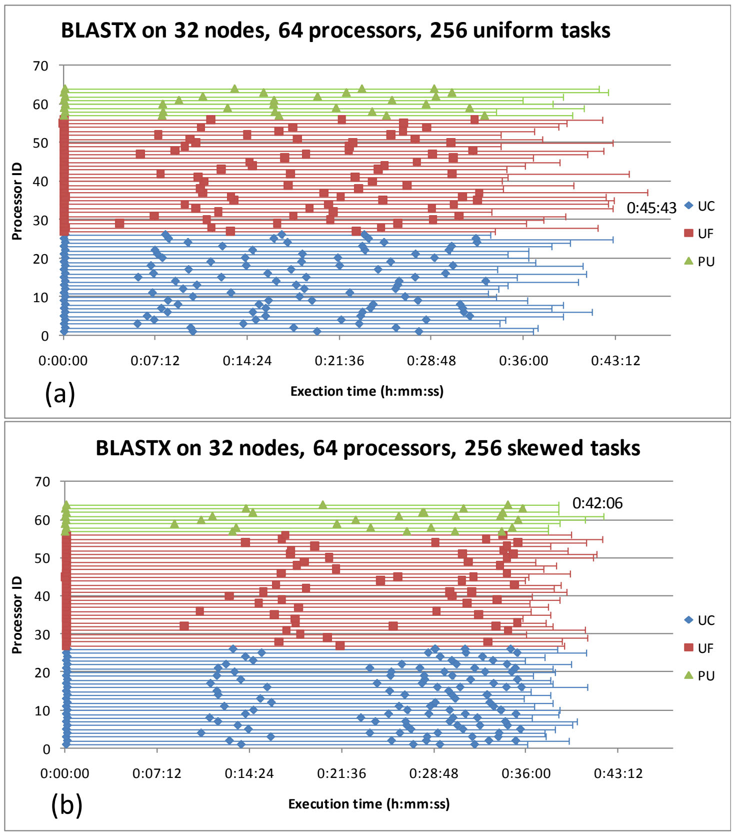
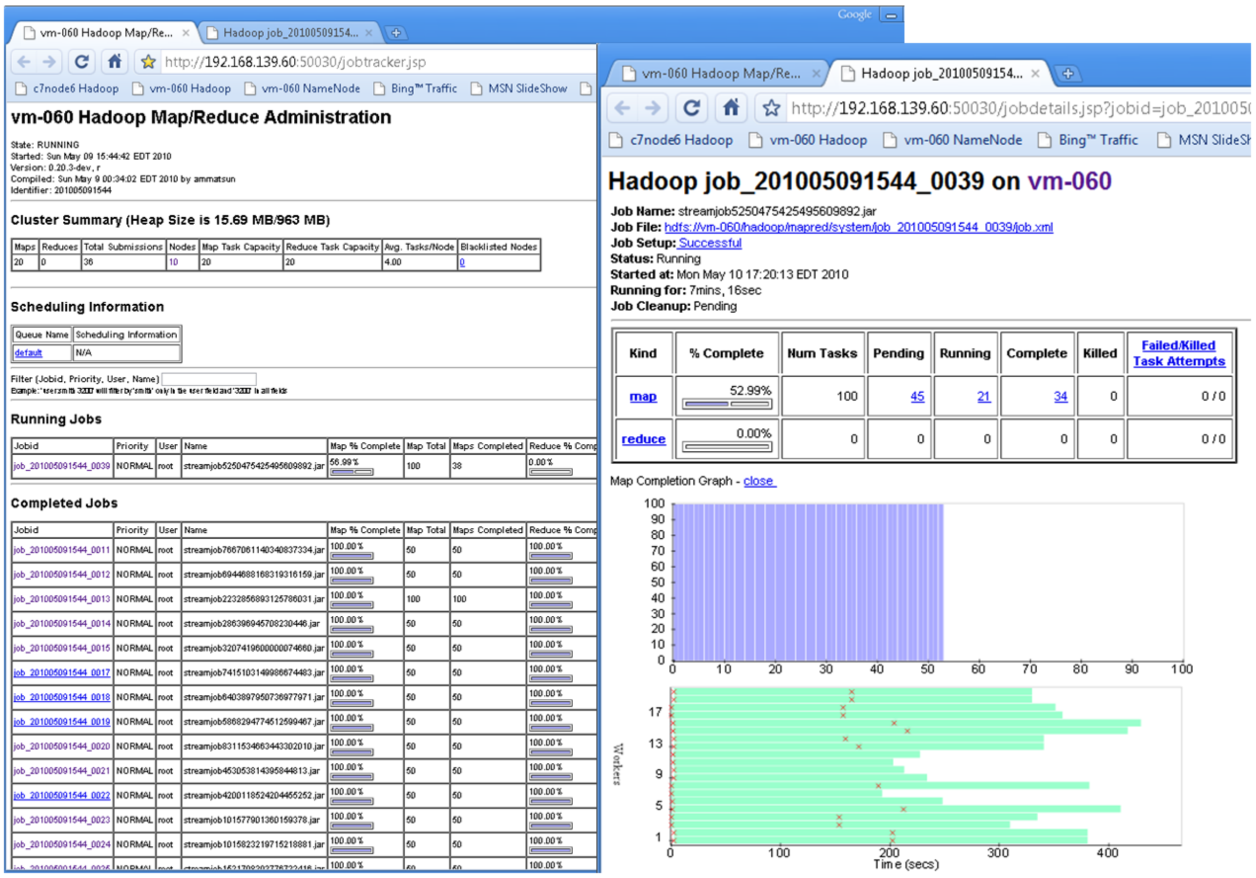


Figure 3: Comparison of executions of a 960-sequence BLAST job divided into 256 tasks with (a) uniform or (b) skewed sizes on 64 processors across 3 different sites (UC, UF and PU). The progress of time is shown in the horizontal axis and the vertical axis represents each of the 64 individual workers. In this particular setup, the overall time perceived by the end user when running with skewed tasks is 8% shorter than when running with uniform tasks. This difference will be more evident in larger scale experiments or when resources demonstrate greater performance heterogeneity – e.g., the EC2 minimum compute unit is approximately half of a single UF VM instance.

5. Sky computing demo

Resources in at least 3 independent clouds (UC, UF and PU) will be used to demonstrate the benefits of sky computing to execute massively parallel applications (in particular, BLAST). All clouds are managed by the Nimbus toolkit, and contextualization service will be used where available to demonstrate the scalability in configuring a virtual cluster. The presence of VMs deployed in non-routable private networks (e.g., all VMs at UF) will require all-to-all communication to be enabled by ViNe overlays. TinyViNe is used in sites where overlay network deployment is challenged by cloud network policies (e.g., UC and PU). All nodes must participate on the overlay, including the end user machine, independently of having public or private addresses. Single-cloud and multi-cloud setups will be used to demonstrate performance of BLAST executions with different sizes of input. The latest version of BLAST (2.2.23+) will also be available to demonstrate the easy integration of application updates. Progress of job execution will be displayed by modified Hadoop web interfaces (Figure 4). Linux applications such as cat and sleep will be used to illustrate the Hadoop overheads and the effects of skewed task distribution. Execution of applications should be successful even in the presence of failures. Runs on FutureGrid (www.futuregrid.org), of which we are partners, are in the works and demonstrations on this new infrastructure will depend on its deployment progress.



vm-060 Hadoop Map/Re... | Hadoop job_20100509154...

http://192.168.139.60:50030/jobtracker.jsp

c7node6 Hadoop | vm-060 Hadoop | vm-060 NameNode | Bing™ Traffic | MSN SlideShow

vm-060 Hadoop Map/Reduce Administration

State: RUNNING
Started: Sun May 09 15:44:42 EDT 2010
Version: 0.20.3-dev, r
Compiled: Sun May 9 00:34:02 EDT 2010 by ammatsun
Identifier: 201005091544

Cluster Summary (Heap Size is 15.69 MB/963 MB)

Maps	Reduces	Total Submissions	Nodes	Map Task Capacity	Reduce Task Capacity	Avg. Tasks/Node	Blacklisted Nodes
20	0	36	10	20	20	4.00	0

Scheduling Information

Queue Name	Scheduling Information
default	N/A

Filter (Jobid, Priority, User, Name)
Example: 'user:smith 3200' will filter by 'smith' only in the user field and '3200' in all fields

Running Jobs

Jobid	Priority	User	Name	Map % Complete	Map Total	Maps Completed	Reduce % Comp
job_201005091544_0039	NORMAL	root	streamjob5250475425495609892.jar	56.99 %	100	38	0.00 %

Completed Jobs

Jobid	Priority	User	Name	Map % Complete	Map Total	Maps Completed	Reduce % Comp
job_201005091544_0011	NORMAL	root	streamjob7667061140340837334.jar	100.00 %	50	50	100.00 %
job_201005091544_0012	NORMAL	root	streamjob6944688168319316159.jar	100.00 %	50	50	100.00 %
job_201005091544_0013	NORMAL	root	streamjob2232856893125786031.jar	100.00 %	100	100	100.00 %
job_201005091544_0014	NORMAL	root	streamjob286396945708230446.jar	100.00 %	50	50	100.00 %
job_201005091544_0015	NORMAL	root	streamjob3207419600000074660.jar	100.00 %	50	50	100.00 %
job_201005091544_0017	NORMAL	root	streamjob7415103149986674483.jar	100.00 %	50	50	100.00 %
job_201005091544_0018	NORMAL	root	streamjob6403897950736977971.jar	100.00 %	50	50	100.00 %
job_201005091544_0019	NORMAL	root	streamjob5868294774512599467.jar	100.00 %	50	50	100.00 %
job_201005091544_0020	NORMAL	root	streamjob8311534663443302010.jar	100.00 %	50	50	100.00 %
job_201005091544_0021	NORMAL	root	streamjob453053814395844813.jar	100.00 %	50	50	100.00 %
job_201005091544_0022	NORMAL	root	streamjob4200118524204455252.jar	100.00 %	50	50	100.00 %
job_201005091544_0023	NORMAL	root	streamjob101577901360159378.jar	100.00 %	50	50	100.00 %
job_201005091544_0024	NORMAL	root	streamjob1015823219715218881.jar	100.00 %	50	50	100.00 %
[illegible]	NORMAL	root	[illegible]	100.00 %	50	50	100.00 %

vm-060 Hadoop Map/Re... | Hadoop job_20100509154...

http://192.168.139.60:50030/jobdetails.jsp?jobid=job_201005

c7node6 Hadoop | vm-060 Hadoop | vm-060 NameNode | Bing™ Traffic | MSN SlideSh

Hadoop job_201005091544_0039 on vm-060

Job Name: streamjob5250475425495609892.jar
Job File: hdfs://vm-060/hadoop/mapred/system/job_201005091544_0039/job.xml
Job Setup: Successful
Status: Running
Started at: Mon May 10 17:20:13 EDT 2010
Running for: 7mins, 16sec
Job Cleanup: Pending

Kind	% Complete	Num Tasks	Pending	Running	Complete	Killed	Failed/Killed Task Attempts
map	52.99%	100	45	21	34	0	0 / 0
reduce	0.00%	0	0	0	0	0	0 / 0

Map Completion Graph - close

Figure 4: Hadoop web interfaces for use in live demo. On the left window, the cluster summary and the history of jobs are listed. On the right, the progress of a Hadoop BLAST job in running state is displayed in graphical mode with each sub-task represented by a separate bar.

References

[1] Globus Alliance. Nimbus Science Clouds. http://workspace.globus.org/clouds.

[2] Matsunaga, A.; Tsugawa, M.; and Fortes, J.A.B. CloudBLAST: Combining MapReduce and Virtualization on Distributed Resources for Bioinformatics Applications. In Proc 4th IEEE Int Conf e-Science, December, 2008.

[3] Tsugawa, M.; Matsunaga, A.; and Fortes, J.A.B. User-level Virtual Network Support for Sky Computing. In Proc 5th IEEE Int Conf e-Science, p.72-79, December, 2009.

[4] K. Keahey and T. Freeman. Contextualization: Providing One-click Virtual Clusters. In Proc 4th IEEE Int Conf eScience, December, 2008.

[5] Matsunaga, A.; and Fortes, J.A.B. On the use of machine learning to predict the time and resources consumed by applications. In Proc 10th IEEE/ACM Int Symp Cluster, Cloud and Grid Computing, 2010.