

# Network Capabilities of Cloud Services for a Real Time Scientific Application

Dilip Kumar Krishnappa, Eric Lyons, David Irwin, and Michael Zink  
University of Massachusetts Amherst  
{krishnappa, zink}@ecs.umass.edu {elyons, irwin}@cs.umass.edu

**Abstract**—Dedicating high-end servers for executing scientific applications that run intermittently, such as severe weather detection or generalized weather forecasting, wastes resources. While the Infrastructure-as-a-Service (IaaS) model used by today’s cloud platforms is well-suited for the bursty computational demands of these applications, it is unclear if the network capabilities of today’s cloud platforms are sufficient. In this paper, we analyze the networking capabilities of multiple commercial (Amazon’s EC2 and Rackspace) and research (GENICloud and ExoGENI cloud) platforms in the context of a Nowcasting application, a forecasting algorithm for highly accurate, near-term, e.g., 5-20 minutes, weather predictions. The application has both computational and network requirements. While it executes rarely, whenever severe weather approaches, it benefits from an IaaS model; However, since its results are time-critical, enough bandwidth must be available to transmit radar data to cloud platforms before it becomes stale. We conduct network capacity measurements between radar sites and cloud platforms throughout the country. Our results indicate that ExoGENI cloud performs the best for both serial and parallel data transfer with an average throughput of 110.22 Mbps and 17.2 Mbps, respectively. We also found that the cloud services perform better in the distributed data transfer case, where a subset of nodes transmit data in parallel to a cloud instance. Ultimately, we conclude that commercial and research clouds are capable of providing sufficient bandwidth for our real-time Nowcasting application.

## I. INTRODUCTION

Cloud platforms are emerging as the primary data warehouse for a variety of applications, such as DropBox, iCloud, Google Music, etc. These applications allow users to store data in the cloud and access it from anywhere in the world. Commercial clouds are also well suited for providing high-end servers for rent to execute applications that require computation resources sporadically. Cloud users only pay for the time they actually use the hardware and the amount of data that is transmitted to and from the server, which has the potential to be more cost effective than purchasing, hosting, and maintaining dedicated hardware. Amazon’s Elastic Compute Cloud (EC2) [1] and Rackspace [8] are two of many commercial cloud services where users can *rent* compute and storage resources based on their needs and are charged based on the usage time of compute resources, the amount of data that is transferred, and the amount of data that is stored.

Commercial clouds use the pay-as-you-use model where the users are charged for resource usage on an hourly basis. In contrast, research clouds like GENICloud [26], [10] and the ExoGENI cloud [2] provide free resources for the research

community. Apart from the fact that the usage of these cloud platforms is free for the research community, they bear the following additional advantages. First of all, researchers can use them to develop prototypes of scientific cloud applications. Large-scale implementation of these applications will still have to happen in commercial clouds since the research clouds provide only a limited number of resources (e.g., available compute nodes or overall storage space). Second, research clouds such as the ExoGENI (with its NEuca extensions [5]) allow for dynamic configuration of the network topology within the cloud, a feature that is not provided by commercial clouds. Third, specific research clouds are connected via next-generation research networks (NLR FrameNet [6] or Internet2 ION [3]) that allow the provisioning of dedicated, isolated network resources. The latter will help researchers to better understand how distributed applications that run in the cloud can benefit from new network technologies. This will, for example, allow us to investigate how a dedicated layer 2 connection between the source and the receiving instance in the cloud will impact the overall performance of the application.

Recent work has shown the benefits of using cloud computing for data-intensive scientific applications [17], [13]. While these operate on massive data sets uploaded to the cloud *a priori*, their computation typically takes place offline with flexible deadlines, if any. Additionally, scientific applications often reuse the same data set for repeated executions, e.g., searches over a large parameter space, which mitigates the upfront (time and monetary) cost of uploading data to the cloud. However, many scientific applications exist that require hard deadlines for data processing. One important application is short-term, fine-grained weather forecasting, called *Nowcasting*, which produces highly accurate forecasts 10s of minutes in the future for areas as small as  $100m^2$  [24], [23].

Compared to most applications that are hosted in the cloud at present, Nowcasting has stricter real-time constraints. Timely execution of the algorithm is critical since the Nowcast data has to be made available to end users as soon as possible. For example, in a severe weather scenario Nowcast information can be used to warn the public, or guide spotters and other emergency management personnel. Since Nowcasting predicts weather only in the very near-term future (on the order of minutes in some cases), it is important that the algorithm produces results very fast. Assuming that it would take 12 minutes to generate a 15 minute Nowcast, would leave just 3 minutes for the users of these data to take action.

In our specific scenario, we are investigating whether there are several factors that impact the real-time performance of the Nowcasting algorithm. One contributing factor is the throughput on the path between the radar nodes to the location where the Nowcasting algorithm is executed. The second factor is the available compute resources that determine how fast data from different radars can be merged and how fast the Nowcasting algorithm can be executed. In this paper, we focus on the network characteristics of cloud services and how it might impact the performance of our Nowcasting algorithms and any other real time scientific applications.

To analyze the network capabilities of cloud services for real-time scientific applications, we perform a series of measurements with EC2 instances on the East and West coast, instances from Rackspace’s cloud, as well as instances from GENICloud and ExoGENI cloud. The measurements involve data transfer from National Weather Services NEXRAD radars to the cloud instances. Since, we do not have access to the NEXRAD radars, we have mimicked the NEXRAD radar location with the nearby PlanetLab nodes for measurements.

The remainder of the paper is outlined as follows. In Section II, we provide a brief description of the cloud services we use, while Section III presents the measurement setup and the results of the measurement. Section IV provides a network comparison of the commercial cloud service and research cloud testbeds for real-time scientific applications. Related work in this area is presented in Section V and Section VI concludes the paper.

## II. CLOUD SERVICES

In this paper we have chosen four cloud services for the network capability analysis for our real-time application of short-term weather forecasting. We have considered two commercial cloud services—Amazon’s EC2 and Rackspace Cloud Hosting—as well as two research cloud testbeds—GENICloud and ExoGENI cloud. In this section, we give a brief description of these cloud services before explaining our measurement methodology in Section III.

### A. Elastic Compute Cloud (EC2)

Amazon’s Elastic Compute Cloud (EC2) [1] is a cloud service which provides resizable compute capacity to execute applications on demand. Amazon EC2 provides a variety of services including cloud servers, storage, Virtual Private Cloud, and CloudWatch. Amazon provides an easy-to-use web service interface which allows users to obtain and configure cloud resources at any of Amazon’s AWS data centers. It provides users with complete control of their computing resources and lets users run applications on Amazon’s computing environment. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing users to quickly scale capacity, both up and down, as their computing requirements change.

EC2 provides on-demand resources with pricing depending on the type of resources used and the duration of the usage. The cost of using commercial cloud services also depends on

additional factors such as the amount of I/O performed and the amount of storage used, both of which can incur significant costs for researchers using cloud resources. Wang et. al [25] provide a list of example applications that can be executed on Amazon’s Elastic Compute Cloud (EC2).

### B. Rackspace Cloud

Rackspace Cloud [8] is one of Amzon’s competitors in the area of commercial cloud hosting. Rackspace offers services including cloud servers, cloud storage and cloud-based website hosting. Cloud servers are available in eight different sizes (with respect to available RAM and disk space) and support a variety of Linux and Windows operating systems. In [16] and [19], the authors provide a brief description of Rackspace and compare its services with other cloud providers.

### C. GENICloud

GENICloud [26], [10] is an open source research cloud testbed which is based on the Slice-Based Facility Architecture (SFA) used by PlanetLab [11]. It supports the management of individual VMs or clusters of VMs. GENICloud uses the Eucalyptus [20] open source cloud platform as a base and federates it with SFA to provide a slice-based architecture to acquire cloud instances (virtual machines) as slivers, similar to acquiring virtual machines on PlanetLab. GENICloud as a platform consists of a small set of nodes at various sites connected internally to provide a cloud testbed for trusted researchers. The GENICloud resources can be acquired using the Sface [9] GUI providing valid credentials, or a Web based GUI similar to the one for PlanetLab.

### D. ExoGENI Cloud

ExoGENI cloud [2] is a software framework and an open-source cloud platform, which allows users to programmatically manage a controllable, shared substrate. Based on this substrate, researchers can create their own cluster infrastructure by combining servers, storage, and network links in an arbitrary fashion. An ExoGENI deployment is a dynamic collection of interacting control servers that collaborate to provision and configure resources for each experimenter according to the policies of the participants. ORCA (ExoGENI’s control framework) helps to provision virtual networked systems via secure and distributed management of heterogeneous resources over federated substrate sites and domains. ORCA allows users to create global topologies of nodes connected via layer 2 QoS-provisioned links. Based on these features ExoGENI cloud offers a variety of opportunities for experimentation and research and also for developing new resource control and management policies via plugins. The ExoGENI cloud, similar to GENICloud, uses a slice-based architecture on top of OpenStack [7] or Eucalyptus [20]. ExoGENI gives researchers more flexibility than other research clouds, as well as commercial clouds, since it allows them to *i*) create their own network topology for a compute cluster, and *ii*) choose between several geographically distributed clusters.

### III. MEASUREMENTS AND RESULTS

The previous section provides an overview of the cloud services we have considered for our analysis. In this section, we investigate the network performance of the cloud services for our real-time scientific application of weather forecasting by performing a series of measurements. The weather prediction algorithm (Nowcasting) uses radar data as input. NEXRAD radars are the current source of data for weather predictions in the U.S. Since we do not have access to NEXRAD radar data feeds, to perform measurements in a large-scale setting we *replicate* a distribution system that, on the network level mimics the NEXRAD system. For the measurements we make use of PlanetLab [11] a global research network that supports large-scale, distributed experiments. Thus, the PlanetLab nodes take on the role of a radar data source for the Nowcasting application in our measurement.

To make our experiment as realistic as possible we went through the exercise of choosing PlanetLab nodes that are close in physical location to the NEXRAD radars. Unfortunately, close proximity between NEXRADs and PlanetLab nodes is not always given. This is due to the fact that locations for weather radars are chosen based on parameters like coverage and beam blocking, which often places them in remote areas. Although there are 159 NEXRAD radar sites in US, we could find only 103 PlanetLab nodes close to those locations, out of which there were around 60 PlanetLab nodes active at any given time. Hence, in our measurements we use results from around 60 PlanetLab nodes compared to 159 NEXRAD radars. The measurement results provided in the paper were performed during the last week of March 2012. According to [27], radars generate data at a constant rate of roughly 5 Mbps. For the remainder of the paper, we will use 5 Mbps as the minimum required throughput between a radar node and the cloud instances to allow real-time data transmission for Nowcasting operation. This threshold can be varied based on the application's need.

#### A. Serial Data Transfer

In this section, we present the results from a series of measurements we performed in which PlanetLab nodes transmit data to a cloud instance in a serial manner. With these measurements we intend to evaluate the average throughput of each individual path between a radar node and cloud instances in the absence of competing traffic to a specific cloud instance. To investigate if the location of the EC2 instance has an impact on throughput we performed the measurement twice, once with an EC2 instance in a West Coast data center and another in the EC2 East Coast data center. Also, to investigate if the time of the day has any impact on our measurement results we perform our measurements twice for each cloud instance, once during the day and once at night time. Approximate time for the day measurement was around noon and for the night measurement was around midnight (PST). For these measurements we used Iperf [4] to transmit data via TCP.

Figure 1 shows the results of the measurements from PlanetLab nodes to the EC2 East Coast data center, EC2 West Coast

data center and Rackspace cloud instances, while Figure 2 shows the results of the serial measurement from PlanetLab nodes to GENICloud and ExoGENI cloud instances during the day. During this measurement we transmit data from each individual PlanetLab node to cloud instances for 15 minutes. We would have transmitted for a longer time period but due to high data transmission rate, the overall data volume that can be transmitted by PlanetLab nodes is limited to low bandwidth burst after 10.8 GB of total data transfer. PlanetLab uses this policy to avoid any DDoS attacks. Tables I and II show a summary of the measurements performed on the commercial and research clouds. As shown in the tables, the average throughput over all transmissions is 36.248 Mbps for the case of a West Coast EC2 instance, 85.035 Mbps for an East Coast EC2 instance, 35.335 Mbps for a Rackspace instance, 9.744 Mbps for a GENICloud instance and 110.22 Mbps for an ExoGENI instance for the measurements performed during the day. The average throughput on all the cloud instances is greater than the threshold throughput of 5 Mbps required for our Nowcasting application. This implies that when only one radar node is used for our Nowcasting application, the network links between the radar node and cloud instances offer sufficient capacity to execute the application in real time.

Tables I and II show the results for the measurement performed at night and it can be seen from the average throughput for the serial measurements row that there is minimal improvement compared to the day time measurements. This improvement shows that there is slightly less network traffic at night but the improvement in average throughput is not drastic. This is good news for our Nowcasting application, since its real time constraints do not allow for a delayed execution. Otherwise, the results show that it makes sense to perform data upload for non time-critical applications to night time to benefit from the slightly increased network performance.

Though the average data throughput for the serial measurement on all the cloud instances is well above the 5 Mbps requirement, it can be seen from Figures 1 and 2 that for some of the paths from the nodes to the cloud instances the average throughput is well below the threshold. About 13% of the nodes have average throughput of less than 5 Mbps to the EC2 machines for both East and West Coast, while about 18% of the nodes have an average throughput of less than 5 Mbps to the Rackspace cloud instance and GENICloud cloud instances and 10% of the nodes to ExoGENI cloud instance have less than 5 Mbps average throughput. An overview for all measurements is given in Table III.

In the specific case of the serial measurement these cases are most likely below the 5 Mbps threshold due to issues with the specific PlanetLab node or the access link of that node. To affirm our conjecture we take a look at the throughput measured (as shown in Figures 1 and 2) for node 3. One can see that the throughput is consistently low for all five measurements which leads us to conclude that this is a PlanetLab node or access link related issue. By inspecting nodes 4, 5, and 6, one can also observe nodes that consistently produce good throughput results for all five measurements.

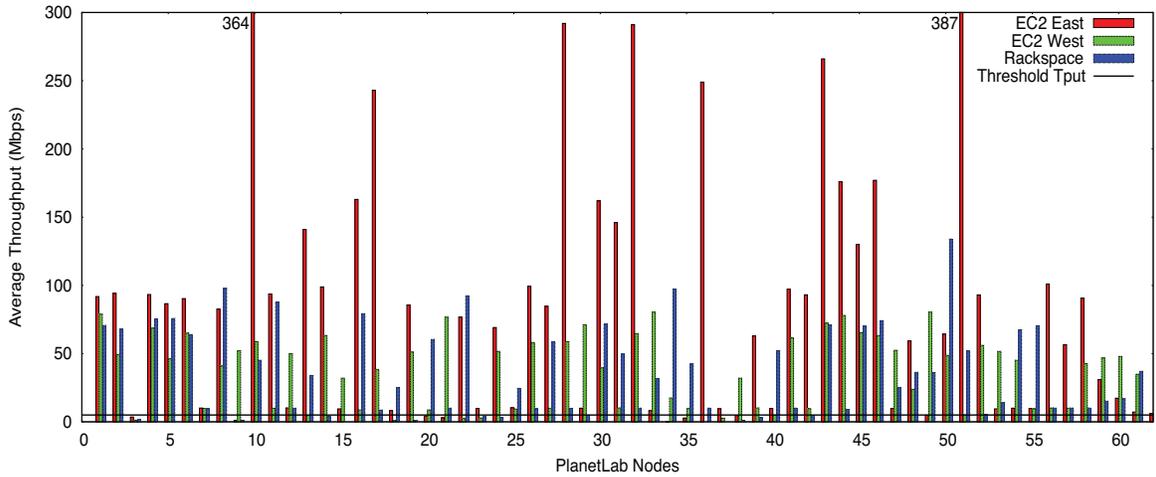
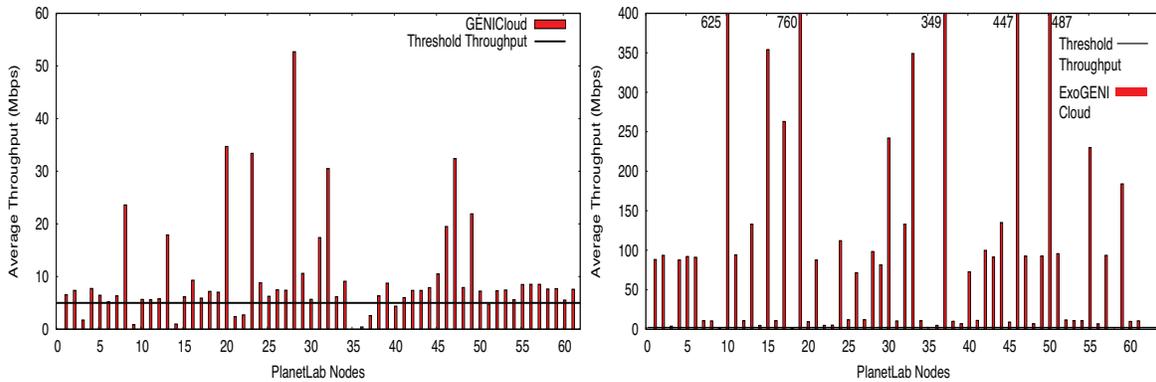


Fig. 1. Serial Measurement between PlanetLab Nodes and Commercial Cloud Instances. Y-axis is cut off at 300 Mbps to better depict low-throughput results. Throughput values for PlanetLab nodes 10 and 50 are 364 Mbps and 387 Mbps.



(a) Serial Measurement between PlanetLab Nodes and GENICloud Instance (b) Serial Measurement between PlanetLab Nodes and ExoGENI Cloud Instance. Y-axis is cut off at 400 Mbps to better depict low-throughput results.

Fig. 2. Serial Measurement between PlanetLab Nodes and Research Cloud Instances

### B. Parallel Data Transfer

The previous section (Section III-A) gives an insight into the data link capacity of radar nodes to cloud instances without any significant, impeding traffic at a cloud instance. In this section, we perform a series of measurements where data were transmitted from the PlanetLab nodes mimicking the NEXRAD radars to a cloud instance in parallel. The intention is to verify if cloud services can handle the traffic generated by a large set of senders and still maintain the threshold throughput of 5 Mbps needed for Nowcasting. As in Section III-A we perform our measurements twice, once during the day and once at night and we use Iperf [4] to transmit data to receiving cloud instances in parallel.

Figure 3 shows the average throughput values of the parallel data measurements from PlanetLab nodes to the commercial cloud services and Figure 4 shows the average throughput values of the parallel data measurements between PlanetLab nodes and the research cloud instances for the measurement performed during the day. The average throughput over all

transmissions is 3.146 Mbps, 1.249 Mbps and 14.122 Mbps for EC2 East Coast, EC2 West Coast and Rackspace cloud instances, respectively. The average throughput over all transmissions is 7.364 Mbps for the GENICloud instance and 17.2 Mbps for the ExoGENI cloud instance. Tables I and II show the average throughput results of the parallel measurements on all the cloud instances for both day and night. As it can be seen from the results of the night measurement the improvement on the average throughput is very small compared to the day measurement, which implies that there is not a significant reduction in competing traffic from other sources to the cloud data centers during the night.

The results from the parallel measurements show that the throughput of each individual PlanetLab node to an EC2 instance is lower than the threshold throughput of 5 Mbps. Thus, in this scenario real-time operation of the Nowcasting application cannot be guaranteed. GENICloud, Rackspace and ExoGENI instances perform comparatively well during the parallel measurement producing an average throughput of

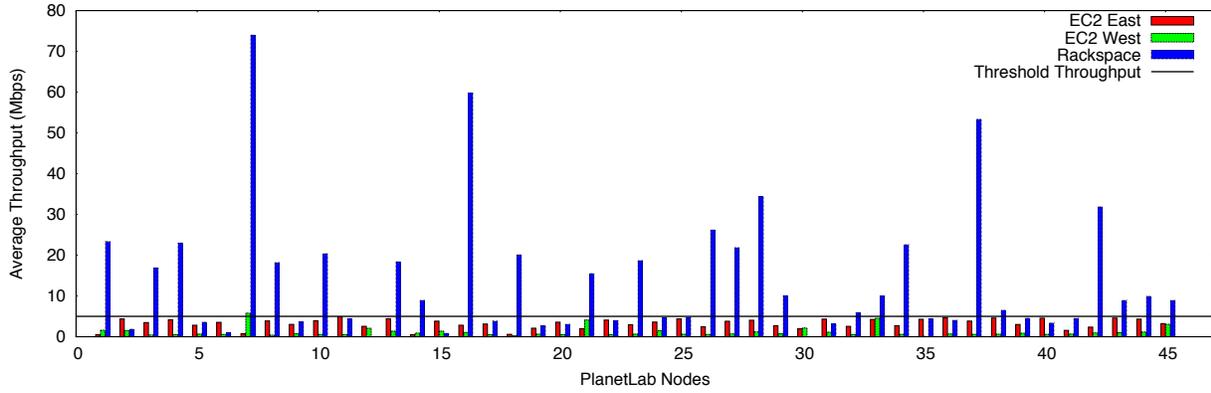


Fig. 3. Parallel Measurement between PlanetLab Nodes and Commercial Cloud Instances

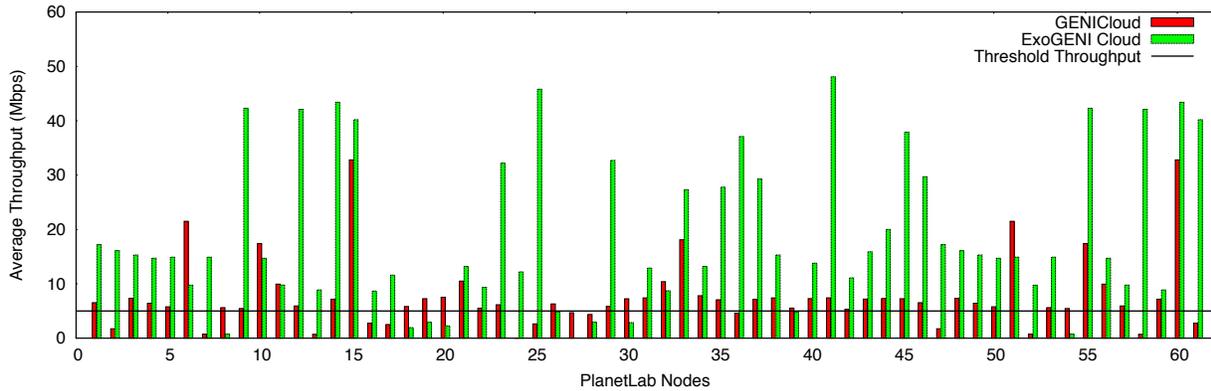


Fig. 4. Parallel Measurement between PlanetLab Nodes and Research Cloud Instances

greater than 5 Mbps. But, we note that there is significant reduction in the average throughput for the parallel measurement from that of serial measurement for Rackspace and ExoGENI instances, which might reduce the efficiency of Nowcasting algorithm in the case of many sources transmitting in parallel. Even though the average throughput of GENICloud, Rackspace and ExoGENI cloud instances are above 5 Mbps threshold throughput, there are about 48%, 22% and 17% of the nodes with throughput of less than 5 Mbps to Rackspace, GENICloud and ExoGENI cloud instances, respectively.

The results from this measurement show that in the case of parallel transmissions and with a large number of senders the average throughput is low. Note, that due to PlanetLab specifics only approximately 60 nodes transmitted in parallel. This is significantly less than 159 radar nodes in which case the average throughput should be significantly worse. These measurement results encouraged us to investigate an alternative approach in which data from a subset of radar nodes are transmitted to a cloud instance instead of the case where *all* radars transmit their data to a single instance. The results from this approach are presented in the next section.

### C. Distributed Data Ingest

In Section III-B, we present the results of a parallel measurement from PlanetLab nodes to the cloud instances where all the nodes transfer data to the instance simultaneously. In

reality, not all radar nodes around the country would transfer their data to one central instance simultaneously. A more likely scenario is the case where a group of radar nodes that belong to a geographic region will transmit their data to a cloud instance that is close to this subset of radar nodes<sup>1</sup>. Assuming we divide the radars into a subset of 10 nodes, we intend to determine the average throughput between the PlanetLab nodes and cloud instances when only a subset of radar nodes (10 nodes) transmit data in parallel. Since some of the cloud instances provide a very low average throughput of less than 5 Mbps (threshold throughput) for the parallel transmission, we intend to verify if cloud instances perform better with our distributed approach. As in Sections III-A and III-B, we perform our measurements twice, once during the day and once at night and we use Iperf to transmit data to receiving instances in different clouds.

Figure 5 shows the average throughput values of the distributed data measurements from PlanetLab nodes to the commercial cloud instances, while Figure 6 shows the average throughput values of the distributed data measurements between PlanetLab nodes and the research cloud instances. The average throughput of the measurement during the day

<sup>1</sup>Since a merging process for data from adjacent radars with overlapping scanning areas is required as a pre-processing step for Nowcasting sub-grouping of radars is a reasonable approach.

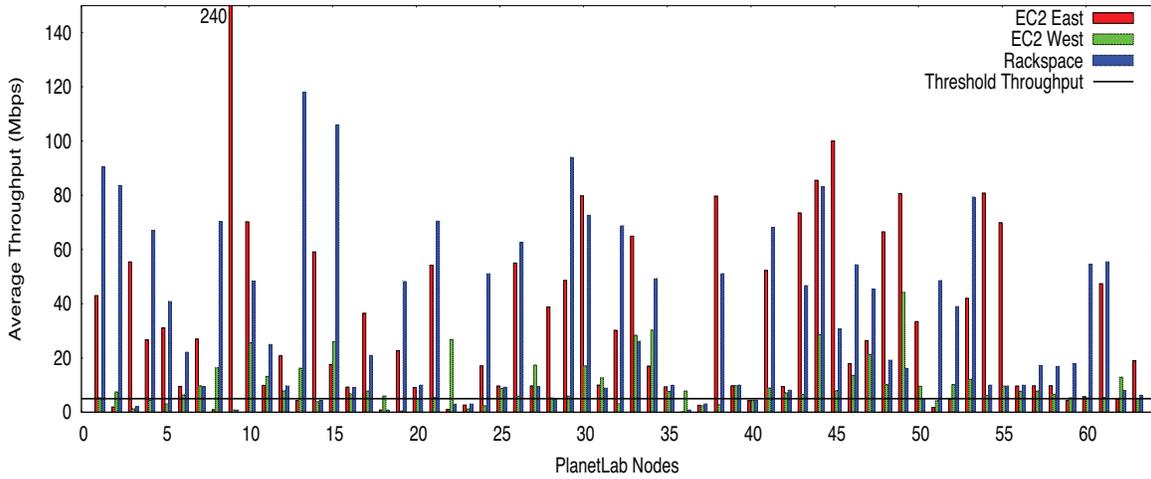


Fig. 5. Distributed Measurement between PlanetLab Nodes and Commercial Cloud Instances. Y-axis is cut off at 150 Mbps to better depict low-throughput results.

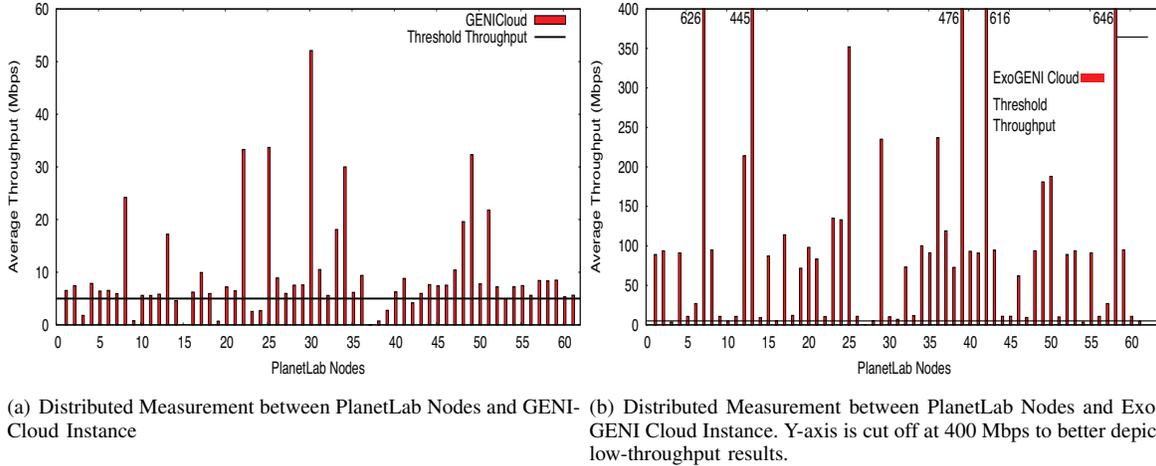


Fig. 6. Distributed Measurement between PlanetLab Nodes and Research Cloud Instances

is 32.463 Mbps, 9.995 Mbps and 34.159 Mbps for EC2 East Coast, EC2 West Coast and Rackspace instances, respectively, while the average throughput for the distributed measurement on GENICloud and ExoGENI instances is 9.434 Mbps and 112.55 Mbps, respectively. As shown in Figures 5 and 6, when only 10 nodes are transmitting in parallel to a cloud instance the average throughput is above the threshold throughput of 5 Mbps. The results from our distributed measurements show that the average throughput of the GENICloud instance does not change much from the parallel measurement explained in the previous section while there is a huge difference in the average throughput of the other cloud instances compared to the parallel measurement throughput (Section III-B). From the results obtained during this measurement we can infer that when only a subset of radar nodes transmit data in parallel to cloud instances, the average throughput is greater than the threshold throughput and sufficient to execute the Nowcasting application in real time. Tables I and II provide the results of the average throughput obtained from the distributed

measurement performed at night. As in Sections III-A and III-B, there is not a huge difference in average throughput from the measurement performed during the day.

As in Section III-A, though the average throughput of the distributed measurement from the PlanetLab nodes to the cloud instances is above the threshold throughput of 5 Mbps, there are about 22% of the nodes have a throughput of less than 5 Mbps to EC2 East and West Coast instances, 15%, 17% and 6% of the nodes with less than 5 Mbps throughput to Rackspace, GENICloud and ExoGENI cloud instances, respectively.

#### D. Dedicated Network Resources

To investigate how dedicated network resources have an impact on the data throughput for the Nowcasting application we performed a measurement that includes a layer 2 connection between a mimicked radar node and a cloud instance. In this measurement we transmitted data in parallel from PlanetLab nodes to an ExoGENI cloud instance over regular IP while, at the same instant, a node from a different ExoGENI cluster also

TABLE I  
SUMMARY OF AVERAGE THROUGHPUT OF ALL MEASUREMENTS IN COMMERCIAL CLOUD SERVICES

Measurement type	Average (Mbps)						Maximum (Mbps)					
	EC2 East		EC2 West		Rackspace		EC2 East		EC2 West		Rackspace	
	Day	Night	Day	Night	Day	Night	Day	Night	Day	Night	Day	Night
Serial	85.035	86.80	36.248	37.058	35.335	50.657	387	360	80.4	84.8	134	145
Parallel	3.146	4.278	1.249	1.064	14.122	12.458	4.87	10.8	10.4	11.2	74	43.9
Distributed	32.463	35.26	9.995	9.98	34.159	32.812	240	245	44.2	64.2	118	105

TABLE II  
SUMMARY OF AVERAGE THROUGHPUT OF ALL MEASUREMENTS IN RESEARCH CLOUD TESTBEDS

Measurement type	Average (Mbps)				Maximum (Mbps)			
	GENICloud		ExoGENI		GENICloud		ExoGENI	
	Day	Night	Day	Night	Day	Night	Day	Night
Serial	9.744	9.922	110.22	115.40	52.7	53	760	764
Parallel	7.364	8.462	17.2	41.19	32.8	51.1	48.1	417
Distributed	9.434	9.889	112.55	98.527	52.1	52	626	631

transmits data over a dedicated layer 2 link to the receiving instance. The average throughput of the parallel measurement from PlanetLab nodes to the ExoGENI cloud instance is 34.68 Mbps where as the layer 2 throughput between the ExoGENI cloud instances is 572 Mbps. This result shows that dedicated network resources can provide guaranteed throughput to cloud instances for applications that require a minimum throughput.

#### IV. COMPARISON OF CLOUD SERVICES

In Section III, we present the network performance measurement results of cloud instances for our real-time short term weather prediction application. In this section, we compare the network performance of research cloud testbeds for our application with that of commercial cloud services (Amazon’s Elastic Compute Cloud and Rackspace).

Table I and Table II give an overview of the average throughput measured between PlanetLab nodes and the cloud instances. Table III shows the percentage of PlanetLab nodes that have average throughput below the 5 Mbps threshold for all measurements we performed. The results of serial the measurements row in Tables I and II show that both, the research cloud nodes and the commercial cloud service nodes perform well without competing traffic with an average throughput above the required threshold of 5 Mbps. ExoGENI instances perform best without competing traffic yielding an average throughput of 110.22 Mbps followed by EC2 East Coast data center instance with 85.035 Mbps, EC2 West Coast data center instance with 36.248 Mbps, Rackspace instance with 35.335 Mbps and then GENICloud instance with an average throughput of 9.719 Mbps. We also note that there is not much improvement in average throughput results for the measurement performed at night.

The parallel measurement row shown in Tables I and II provides results that are very different to the serial measurements presented above. ExoGENI, Rackspace and GENICloud instances yield an average throughput of 17.2 Mbps, 14.122 Mbps and 7.681 Mbps, which is greater than the threshold throughput of 5 Mbps required for Nowcasting. EC2 cloud instances yield an average throughput of 3.146 Mbps and 1.249

Mbps for East Coast and West Coast data center respectively, which is well below the threshold throughput of 5 Mbps required for our real-time application of Nowcasting.

The distributed measurement row in Tables I and II shows that each of the cloud instances considered perform better when only a subset of nodes are transmitting data in parallel. As in the serial measurement scenario the average throughput results from all cloud instances are greater than the threshold throughput of 5 Mbps. The ExoGENI cloud instance performs better than the other three cloud instances with an average throughput of 112.55 Mbps while the Rackspace cloud instance provides an average throughput of 34.159 Mbps. GENICloud instance provides an average throughput of 9.434 Mbps and EC2 cloud service provides an average throughput of 32.463 Mbps and 9.995 Mbps in the East and West Coast data centers, respectively. The measurement results show that GENICloud instance is the most consistent of cloud instances providing almost identical average throughput irrespective of the number of nodes used in data transmission.

From the measurement results presented in this paper, we can conclude that the networking capabilities of the cloud instances are sufficient for the real-time operation of our weather forecasting application called Nowcasting. We can also infer that, the network performance of research cloud testbeds are in par with that of the commercial cloud services and can be used as a test instance to execute our Nowcasting application without incurring any additional cost.

We would also like to mention that the measurement results presented in this paper can also be used to verify which cloud services offer sufficient network capacity for other applications that require a certain throughput. One such example is a camera sensor network for security or monitoring for which data are transmitted from a set of distributed cameras to a central processing node. Assuming the processing would be performed on a cloud instance and the minimum throughput requirement for a single camera stream is known one can simply use the results presented in this paper to determine which cloud instances can support such an application.

Since our measurements show that clouds offer sufficient

TABLE IV  
COMPUTATION TIME ANALYSIS OF CLOUD SERVICES

Instances	Memory	ECUs	Disk	Cost	Exec. Time
EC2	7.5 GB	4	850 GB	\$.34/hr	74.341s
Rackspace	8 GB	4	320 GB	\$.48/hr	96.53s
GENICloud	8 GB	4	20 GB	-	67.45s
ExoGENI	8 GB	4	20 GB	-	68.84s

capacity to transmit radar data for Nowcasting in real time we were also interested in investigating the compute performance of different cloud instances. To measure the performance of cloud instances we executed a 15 minute Nowcast and measured the time until the final Nowcast results is calculated. For these measurements we have chosen cloud instances that offer similar resources. The results (see Table IV) show that it takes a little bit over one minute to complete one Nowcast run for a 15-minute forecast besides for the case of the Rackspace instance where the execution takes over 1.5 minutes. In general, the results show that commercial and research clouds offer sufficient networking and compute resources to allow a timely execution of the Nowcasting algorithm.

## V. RELATED WORK

A substantial amount of research has been carried out to investigate the feasibility of running scientific applications in commercial clouds such as Amazon's AWS. Hazelhurst examines the performance of the bioinformatics application WCD [14]. Deelman et.al. provide details of performance and storage costs of running the Montage workflow on EC2 [12]. The High-Energy and Nuclear Physics (HENP) STAR experiment has examined the costs and challenges associated with running their analysis application in the EC2 cloud [18]. Ramakrishnan et.al. have examined the usefulness of cloud computing for e-Science applications [21]. In addition, standard benchmarks have also been evaluated on Amazon EC2. Rehr et al. show that Amazon EC2 is a feasible platform for applications that do not need advanced network performance [22]. Ramakrishnan et al. perform a comprehensive comparison of the performance of Amazon EC2 with HPC platforms, using real applications representative of the workload at a typical supercomputing center [15].

To the best of our knowledge, this work is the first to look into the feasibility of Amazon EC2 and Rackspace cloud services for a real-time weather forecasting application.

## VI. CONCLUSION

In this paper, we investigated the feasibility of cloud services for both commercial cloud services (EC2 and Rackspace) and open source research cloud testbeds (GENICloud and ExoGENI cloud) for short term weather predicting real time application called Nowcasting. We have investigated the network characteristics of the cloud services by measuring the throughput between radar nodes and the cloud instances. We have mimicked the NEXRAD radars by using PlanetLab nodes close the NEXRAD radar locations for the measurements.

Our results indicate that ExoGENI cloud performs the best for both serial and parallel data transfer. We also found that the cloud services perform better in the distributed data transfer case, where a subset of nodes transmit data in parallel to a cloud instance.

The measurements performed in this paper demonstrate that cloud services are capable of handling traffic from radar nodes without adding significant delay since in most cases the throughput between the PlanetLab node and cloud instances is above the required throughput. We conclude that today's cloud connectivity allows the execution of Nowcasts in real-time to deliver accurate weather prediction to the end user. We can also conclude that research cloud testbeds provide good network connectivity for real-time scientific applications and can be used for real-time research purposes on the cloud.

## VII. FUTURE WORK

In future work, our goal is to identify the bottleneck (e.g., with packet pair measurements) between the sender and the receiver in the cloud. This might help us to identify if the low network performance for the commercial clouds is caused by saturation of their data center network that host the cloud resources.

## VIII. ACKNOWLEDGEMENTS

This work is supported by U.S. NSF grants CNS-0834243, OCI-1032765, and EEC-0313747. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the National Science Foundation. We would like to thank Ilia Baldine and his team at RENCi for supporting the ExoGENI measurements.

## REFERENCES

- [1] Amazon's Elastic Compute Cloud. <http://aws.amazon.com/ec2/>.
- [2] ExoGENI. <http://wiki.exogeni.net>.
- [3] Internet2 ION. <https://geni-orca.renci.org/trac/wiki/flukes>.
- [4] Iperf. <http://sourceforge.net/projects/iperf/>.
- [5] NEuca. <https://geni-orca.renci.org/trac/wiki/NEuca-overview>.
- [6] NLR Framenet. <http://www.nlr.net/framenet.php>.
- [7] OpenStack. <http://www.openstack.org>.
- [8] Rackspace Cloud. <http://www.rackspace.com>.
- [9] Sface. <http://svn.planet-lab.org/wiki/SfaceGuide>.
- [10] A. C. Bavier, M. Yuen, J. Blaine, R. McGeer, A. A. Young, Y. Coady, C. Matthews, C. Pearson, A. Snoeren, and J. Mambretti. Transcloud - Design Considerations for a High-performance Cloud Architecture Across Multiple Administrative Domains. In *CLOSER*, May 2011.
- [11] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman. PlanetLab: An Overlay Testbed for Broad-coverage Services. *SIGCOMM CCR*, July 2003.
- [12] E. Deelman, G. Singh, M. Livny, B. Berriman, and J. Good. The Cost of Doing Science on the Cloud: The Montage Example. In *SC*, November 2008.
- [13] C. Evangelinos and C. N. Hill. Cloud Computing for Parallel Scientific HPC Applications: Feasibility of Running Coupled Atmosphere-Ocean Climate Models on Amazon's EC2. In *CCA*, October 2008.
- [14] S. Hazelhurst. Scientific Computing Using Virtual High-performance Computing: A Case Study Using the Amazon Elastic Computing Cloud. In *SAICSIT*, October 2008.
- [15] K. Jackson, L. Ramakrishnan, K. Muriki, S. Canon, S. Cholia, J. Shalf, H. Wasserman, and N. Wright. Performance Analysis of High Performance Computing Applications on the Amazon Web Services Cloud. In *CloudCom*, December 2010.

TABLE III  
PERCENTAGE OF PLANETLAB NODES WITH THROUGHPUT BELOW 5MBPS THRESHOLD.

Measurement type	EC2 East		EC2 West		Rackspace		GENICloud		ExoGENI	
	Day	Night	Day	Night	Day	Night	Day	Night	Day	Night
Serial	12.9%	14.5%	12.5%	18.7%	18.7%	15.6%	15.6%	14.6%	9.8%	8.2%
Parallel	100%	68.8%	100%	97.3%	48.8%	17.7%	22.2%	15.5%	17.7%	24.4%
Distributed	12.6%	16.6%	21.8%	35.9%	15.6%	15.6%	17.1%	15.6%	5.7%	13.4%

- [16] H. Jin, S. Ibrahim, T. Bell, W. Gao, D. Huang, and S. Wu. Cloud Types and Services. In *Handbook of Cloud Computing*, pages 335–355. Springer US, 2010.
- [17] G. Juve, E. Deelman, K. Vahi, G. Mehta, B. Berriman, B. Berman, and P. Maechling. Scientific Workflow Applications on Amazon EC2. In *E-Science*, December 2009.
- [18] K. Keahey, R. Figueiredo, J. Fortes, T. Freeman, and M. Tsugawa. Science Clouds: Early Experiences in Cloud Computing for Scientific Applications. In *CCA*, October 2008.
- [19] A. Li, X. Yang, S. Kandula, and M. Zhang. CloudCmp: Shopping for a Cloud Made Easy. In *HotCloud*, June 2010.
- [20] D. Nurmi, R. Wolski, C. Grzegorzczak, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov. The Eucalyptus Open-Source Cloud-Computing System. In *CCGRID*, May 2009.
- [21] L. Ramakrishnan, K. R. Jackson, S. Canon, S. Cholia, and J. Shalf. Defining Future Platform Requirements for e-Science Clouds. In *SoCC*, June 2010.
- [22] J. J. Rehr, J. P. Gardner, M. Prange, L. Svec, and F. Vila. Scientific Computing in the Cloud. *ArXiv e-prints*, December 2008.
- [23] E. Ruzanski, V. Chandrasekar, and Y. Wang. The CASA Nowcasting System. *Journal of Atmospheric and Oceanic Technology*, January 2010.
- [24] E. Ruzanski, Y. Wang, and V. Chandrasekar. Development of a Real-time Dynamic and Adaptive Nowcasting System. In *Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology*, January 2009.
- [25] L. Wang, J. Tao, M. Kunze, A. Castellanos, D. Kramer, and W. Karl. Scientific Cloud Computing: Early Definition and Experience. In *HPCC*, September 2008.
- [26] M. Yuen. GENI in the Cloud. Master’s thesis, University of Victoria, 2010.
- [27] M. Zink, E. Lyons, D. Westbrook, J. Kurose, and D. Pepyne. Closed-loop Architecture for Distributed Collaborative Adaptive Sensing: Meteorological Command & Control. *IJSNET*, February 2010.