CloudCast: Cloud Computing for Short-Term Weather Forecasts

CloudCast provides personalized short-term weather forecasts to clients based on their current location using cloud services, generating accurate forecasts tens of minutes in the future for small areas. Results show that it takes less than two minutes from the start of data sampling to deliver a 15-minute forecast to a client.

Dedicating high-end servers for executing scientific applications that run intermittently, such as severe weather detection or generalized weather forecasting, wastes resources. The infrastructure-as-a-service (IaaS) model used by today's cloud platforms is well suited for the bursty computational demands of these applications. Clouds are emerging as the primary host platform for a variety of applications, such as DropBox, iCloud, and Google Music. These applications let users store data in the cloud and access it from anywhere in the world. Commercial clouds are also well suited for renting high-end servers to execute applications that require computation resources sporadically. Cloud users pay only for the time they actually use resources and the data they transmit to and from the server, which has the potential to be more cost effective than purchasing, hosting, and maintaining dedicated hardware.

With this in mind, we present CloudCast, a new application for short-term, location-based weather prediction using cloud platforms. If severe weather is approaching the user's location, CloudCast automatically sends personalized notifications. To enable this functionality, CloudCast combines two major components. The first component is an algorithm that produces fine-grained, short-term weather forecasts—called *Nowcasting*[1,2]—up to 15 minutes in the future for areas as small as 100 m^2. Nowcasting has the potential to personalize severe weather alerts by programmatically transmitting highly targeted, location-specific warnings to mobile devices based on their GPS coordinates. The second component is a new architecture that allows the execution of Nowcasting on cloud platforms, such as Amazon's Elastic Compute Cloud (EC2; http://aws.amazon.com/ec2). Nowcasting is compute-intensive, requiring high-memory systems for execution. Hence, we use cloud services to execute the Nowcasting algorithm for our CloudCast application. Cloud computing platforms lower the cost of computation by leveraging economies-of-scale. Generating Nowcasts on a dedicated infrastructure is economically infeasible due to its enormous computational demands, which scale quadratically with spatial resolution.

Thus, to understand the feasibility of hosting Nowcasting on cloud platforms, we analyze the network and computation capability of four different cloud services: two commercial cloud services (Amazon EC2 and Rackspace [www.rackspace.com]) and two research cloud testbeds (the Global Environment for Network Innovations' GENICloud[3,4] and ExoGENI [http://wiki.exogeni.net]). Commercial clouds

1521-9615/13/$31.00 © 2013 IEEE
Copublished by the IEEE CS and the AIP

Dilip Kumar Krishnappa, David Irwin, Eric Lyons, and Michael Zink
University of Massachusetts Amherst

use the pay-as-you-use model, charging users for resource usage on an hourly basis. In contrast, research clouds such as GENICloud and ExoGENI cloud provide free resources for the research community. These research platforms offer additional advantages, beyond being free for the research community. First, researchers can use them to develop prototypes of scientific cloud applications. Second, research clouds such as the ExoGENI allow for dynamic configuration of the network topology within the cloud, a feature that isn't provided by commercial clouds. Finally, research clouds are often connected via next-generation research networks—such as the National LambdaRail (NLR) FrameNet (www.nlr.net/framenet.php) or Internet2 ION (https://geni-orca.renci.org/trac/wiki/flukes)—which allow the provisioning of dedicated, isolated network resources. The latter will help researchers better understand how distributed applications that run in the cloud can benefit from new network technologies.

The primary reason weather services cite for not leveraging the cloud is data staging costs. We evaluate these costs for the extreme case of Nowcasting, which requires real-time radar data uploads to predict conditions tens of minutes in the future. Compared to most cloud applications, CloudCast's Nowcasting algorithm has stricter time constraints. Timely execution of the algorithm is critical, because the Nowcast data must be made available to end users before it becomes obsolete. For example, in a severe weather scenario, we can use Nowcast information to warn the public, as well as to guide spotters and other emergency management personnel. Because Nowcasting predicts weather only in the very near-term future, it's important that the algorithm produces results fast. For instance, if it takes 12 minutes to generate and disseminate a 15-minute Nowcast, that leaves just three minutes for users to take action.

Our hypothesis is that the connectivity and diversity of current cloud platforms mitigate the impact of Nowcasting's staging data and computation latency, enabling the platforms to perform the atmospheric modeling and personalized weather forecasting required by CloudCast. In evaluating our hypothesis, we present the CloudCast architecture, which links weather radars to cloud instances, allowing the architecture's components to request computational and storage resources required to generate forecasts based on real-time radar data feeds as requested from clients. We emulate a radar network using PlanetLab sites and conduct extensive bandwidth measurements between each site and cloud instances. We quantify average bandwidth and its variability to determine if the public Internet and today's clouds are sufficient for real-time Nowcasting. We also analyze the computation time and cost of Nowcasting in the cloud for instances offered by cloud services and demonstrate CloudCast live using a deployed prototype radar as a proof-of-concept.

CloudCast Architecture

To get a better sense of how our application works, let's take a look at its architecture. Figure 1 shows the components of the CloudCast architecture: the meteorological command and control (MC&C), which controls the radars' scanning; cloud instances, which are automatically initiated by the MC&C; and the Nowcasting short-term weather forecasting algorithm.

MC&C Architecture

MC&C[5] is the control part of the Collaborative Adaptive Sensing of Atmosphere (CASA) network that determines how the radars will scan the atmosphere in each 60-second heartbeat. The MC&C architecture considers these different factors to determine how to control the radars. For our CloudCast approach, we use the MC&C's detection features. The algorithms detecting the existence and current location of precipitation are ingested into the MC&C's blackboard architecture; it's then able to classify the situation on multiple levels. On a high level, it differentiates between clear air, stratiform rain, and convective regimes, and each regime has a set of tasks associated. In clear air mode, the need for computational resources diminishes, whereas convective mode has strict requirements for data collection and heavily utilizes computers and networks. Rain sensed by the radars will be detected by the MC&C's algorithms; CloudCast then uses the information to determine when to initiate Nowcasts in the cloud. Thus, cloud-based Nowcasts are automatically initiated and halted without user intervention.

Cloud Services

We chose four different cloud services to analyze the feasibility of executing short-term weather forecast applications.

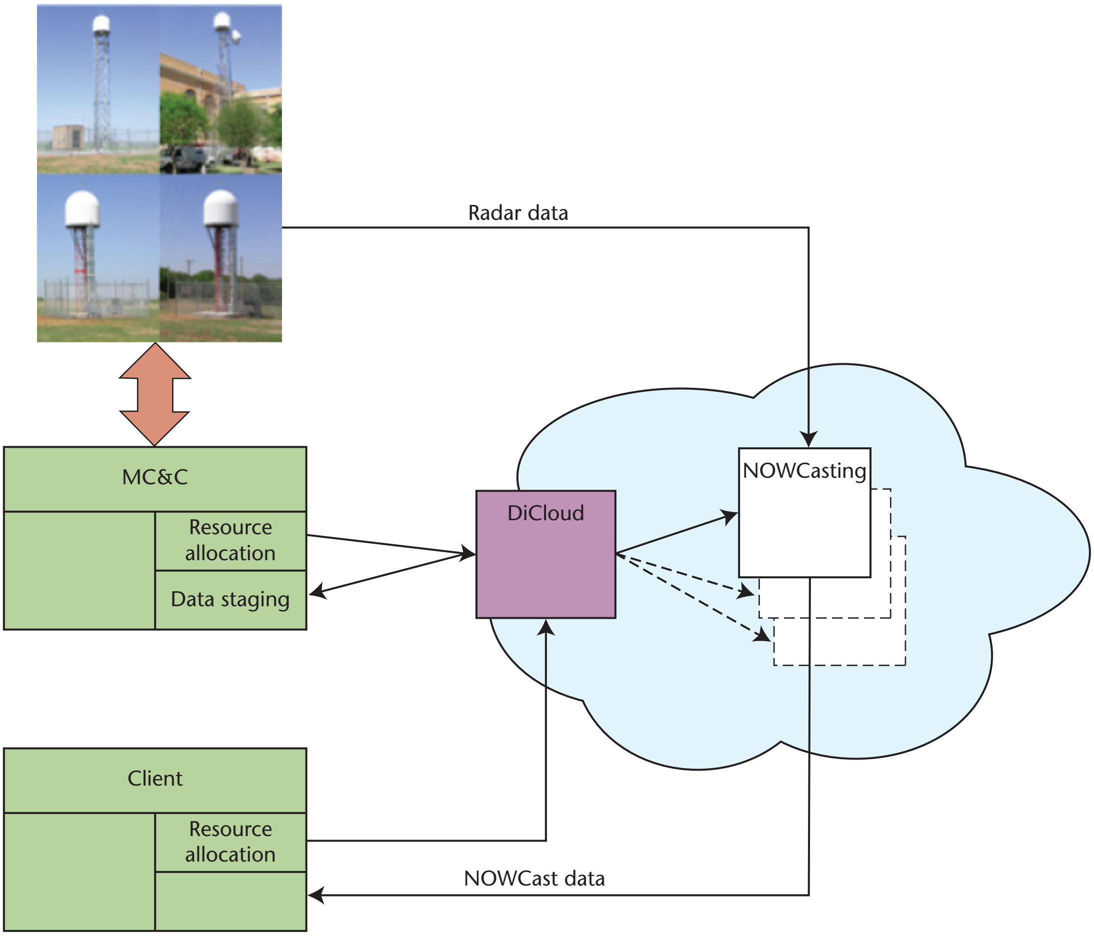

Figure 1. Overview of the CloudCast system architecture. The application is composed of meteorological command and control (MC&C), cloud instances, and the Nowcasting short-term weather forecasting algorithm.

Commercial cloud services. Amazon's EC2 is a cloud service that provides resizable compute capacity to execute applications on demand. EC2 provides on-demand resources with pricing depending on the type of resources used and the duration of 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.

Rackspace Cloud is one of Amazon'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 operating systems.

Research cloud services. GENICloud[3,4] is an open source research cloud testbed that's based on PlanetLab's Slice-Based Facility Architecture (SFA).[6] The testbed supports the management of individual virtual machines (VMs) or VM clusters. GENICloud uses the Eucalyptus[7] open source cloud platform as a base and federates it with SFA to provide a slice-based architecture to acquire cloud instances (VMs) as slivers.

ExoGENI cloud is a software framework and an open source cloud platform that lets users programmatically manage a controllable, shared substrate. The Open Resource Control Architecture (ORCA, which is ExoGENI's control framework) helps provision virtual networked systems via secure and distributed management of heterogeneous resources over federated substrate sites and domains. ORCA lets users create global topologies of nodes connected via layer-2 QoS-provisioned links. ExoGENI gives researchers more flexibility than other cloud services, because it lets them create their own network topology for a compute cluster and choose between several geographically distributed clusters.

Nowcasting

Nowcasting[1,2] refers to short-term (less than 30 minutes) weather forecasting. The Nowcasting algorithm predicts high-impact weather events, such as flood-producing rainfall, severe storms, and hail, in a specific region with sufficient accuracy within a time frame such that appropriate actions can be taken to effectively mitigate the loss of life and property. Because Nowcasting is a short-term weather-prediction system, its applications include warning-decision support for detecting potentially severe weather. Its performance is typically measured in terms of a categorical yes/no (such as rain/no-rain) detection relative to a predetermined measurement threshold representative of a desired threat. This model of measuring performance is well suited for our Nowcasting application, because the Nowcasting algorithm's ability to predict a sufficiently high reflectivity value in a given region is important for end-user emergency decision support.

Measurements

Now that we have a better understanding of CloudCast's architecture, let's investigate the network feasibility of cloud services for CloudCast. As you can see in Figure 1, a potential bottleneck for the real-time operation of Nowcasting in the cloud is the link between the radars and the cloud instances.

To understand the network capability of cloud services for Nowcasting, we perform a series of measurements in a large-scale setting by replicating a distribution system that, at least on the network level, is similar to the Next-Generation Radar (Nexrad) system. Kevin Kelleher and his colleagues[8] give an overview on how data from Nexrads are disseminated by using the National Oceanic and Atmospheric Administration's NOAANet backbone and Internet2. We perform our measurements in two different radar settings: one measurement emulates Nexrad radars and the other uses our four-node radar network that's located in southwestern Oklahoma.[5,9] Both Nexrad and CASA radars use Internet2 as their backbone network and thus have similar network settings. Because we don't have access to Nexrad nodes for our measurement, we use PlanetLab,[6] a global research network that supports large-scale, distributed experiments. Previous work showed that radars generate data at a constant rate of roughly 5 megabits per second (Mbps).[5] For the remainder of our analysis, we 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.

We conduct four different measurement scenarios to depict the network capabilities of cloud services considered for our Nowcasting application. We conduct a serial measurement where data are transmitted to cloud instances from each individual PlanetLab node (replicating Nexrad radars). We conduct this measurement to verify how the cloud instances perform without any competing traffic. We also conduct a parallel measurement, where data from all the PlanetLab nodes transmit data to a cloud instance at the same time. We perform this experiment to verify if the cloud instances can handle the large traffic from all the Nexrad radars at once and still maintain the required minimum threshold throughput of 5 Mbps for our CloudCast application.

Because not all radar nodes around the country would transfer their data to one central instance simultaneously, a more likely scenario is the case in which a particular geographic region's radar nodes will transmit their data to a cloud instance that's close to this subset of radar nodes. To investigate this scenario, we conducted a distributed measurement where only 10 PlanetLab nodes transmit data to a cloud instance in parallel.

The measurements explained so far consider data sent from the radar continuously, but in a real scenario, the data is sent in bursts every minute. From the Nexrad data collected, we've seen that 7.5 Mbytes of data is sent from the radar every minute. Hence, we perform a bursty traffic measurement to understand how the cloud services perform for bursty data sent from the radars. For our measurements, we use Iperf (http://iperf.sourceforge.net) to transit data from radar nodes to cloud instances.

Table 1 gives an overview of the average throughput measurement from Nexrad radar nodes (PlanetLab nodes) and CASA radar nodes to the commercial cloud instances and research cloud instances. 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. The results of serial measurements for both the Nexrad radar nodes and CASA radar nodes in Table 1 show that both the research cloud testbed nodes and the commercial cloud service nodes perform well without competing traffic,

Table 1. Summary of average throughput of all measurements.*

Measurement type	EC2 East Avg. (Mbps)	EC2 West Avg. (Mbps)	Rackspace Avg. (Mbps)	GENICloud Avg. (Mbps)	ExoGENI Avg. (Mbps)	EC2 East Max (Mbps)	EC2 West Max (Mbps)	Rackspace Max (Mbps)	GENICloud Max (Mbps)	ExoGENI Max (Mbps)
Nexrad serial	85.035	36.248	35.335	9.744	110.22	387	80.4	134	52.7	760
Nexrad parallel	3.146	1.249	14.122	7.364	17.2	4.87	10.4	74	32.8	48.1
Nexrad distributed	32.463	9.995	34.159	9.434	112.55	240	44.2	118	52.1	62.6
Nexrad bursty	75.8	89.575	85.75	8.967	79.55	80.2	95.8	92.6	53.5	91.7
CASA serial	165.75	216.75	155.25	8.945	424.25	176	233	216	55.6	444
CASA parallel	46	64.575	102.75	7.965	163.75	47.9	77	144	35.71	179
CASA bursty	128.87	197.66	78.98	9.04	456.10	187.34	213	145	53.21	490

** CASA = Collaborative Adaptive Sensing of Atmosphere; EC2 = Elastic Compute Cloud; GENI = Global Environment for Network Innovations; and Nexrad = Next-Generation Radar.*

with an average throughput above the required threshold of 5 Mbps.

Out of the cloud instances we investigated, the ExoGENI cloud instance performs best without competing traffic, yielding an average throughput of 110.22 Mbps for the Nexrad radar links and 424.25 Mbps for CASA radar links. ExoGENI was followed by Amazon's EC2 East Coast data center cloud instance with 85.03 Mbps; the EC2 West Coast data center cloud instance with 36.24 Mbps; the Rackspace cloud instance with 35.33 Mbps; and, finally, the GENICloud instance with a mere average throughput of 9.71 Mbps for Nexrad radar links. For the CASA radar links, the EC2 West Coast instance performs better than the EC2 East Coast instance, but the average still remains above our required threshold throughput of 5 Mbps for both radar links.

The parallel measurement rows for Nexrad and CASA radars shown in Table 1 provide results that are quite different from the aforementioned serial measurement results. The ExoGENI, Rackspace, and GENICloud cloud instances yield a better average throughput during the parallel measurement than EC2 cloud instances. ExoGENI, Rackspace, and GENICloud cloud instances yield an average throughput of 17.2, 14.12, and 7.68 Mbps, respectively, for Nexrad radar links, which is greater than the threshold throughput of 5 Mbps required for our Nowcasting application. EC2 cloud instance links to Nexrad radar nodes yield an average throughput of 3.14 Mbps and 1.24 Mbps for the East and West Coast data centers, respectively, which is well below the threshold throughput of 5 Mbps. CASA radar links to cloud instances perform better for parallel measurements, as Table 1 shows, yielding an average throughput of more than 5 Mbps for each cloud instance.

These cloud instances perform better when only a subset of nodes is transmitting data in parallel, as shown in the distributed measurement row for Nexrad radars in Table 1. As in the serial measurement scenario, the average throughput results from all the 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 34.15 Mbps. The GENICloud instance provides an average throughput of 9.43 Mbps, and the EC2 cloud service provides an average throughput of 32.46 and 9.99 Mbps in the East and West Coast data centers, respectively.

The results from the bursty traffic measurement experiment, where data are transmitted in bursts of 7.5 Mbytes every minute, yield an average throughput greater than the required threshold throughput of 5 Mbps for each of the cloud instances considered, for both Nexrad radar nodes and CASA radar links. Table 1 summarizes the results from the bursty traffic measurement. From our measurement, we conclude that the networking capabilities of the cloud instances are sufficient for our real-time Nowcasting application. We also infer that the network performance of research cloud testbeds are on par with that of the commercial cloud services, and we can use them as a test instance to execute the Nowcasting application without incurring any additional cost. Additional data from our measurements can be found elsewhere.[10]

Table 2. Nowcast algorithm execution time for live measurement.

Instances	Memory (Gbytes)	Disk (Gbytes)	Cost/hour (US$)	Total cost (US$)	Execution time (s)	Total time (s)
Amazon EC2	7.5	850	0.34	1.13	74.34	95.08
Rackspace	8	320	0.48	1.63	96.53	120.33
GENICloud	8	20	–	–	67.45	78.60
ExoGENI	8	20	–	–	56.83	72.07

Nowcasting Analysis

Having investigated the network feasibility of commercial and research cloud services for our CloudCast application, we can present the cost and computation time analysis of cloud-based Nowcast generation on the cloud instances considered. As a proof-of-concept of our application, we present the results of the live analysis performed with our own radar located on our campus. We also perform an analysis that compares the cost of running Nowcast in the cloud to the cost of using dedicated hardware.

Computation Time

Let's consider the measurement procedure to calculate Nowcasting's cost and computation time for one hour of weather data for the similar instance types offered by Amazon EC2 and Rackspace cloud services. We also calculate Nowcasting's computation time for the same weather data with the instances offered by GENICloud and ExoGENI research cloud services. The weather data used was collected during a severe weather event in May 2011.

For each of the cloud services mentioned in Table 2, we bring up the instances with the Nowcasting image and start the ingest of weather data from the radars. Once the cloud-based Nowcast instance receives the first set of radar scans, it starts to generate 1- to 15-minute Nowcasts, which are kept on the instance's storage. We carry out this operation for one hour of weather data and determine the cost for running this 1-hour Nowcast operation using the cost-tracking services provided by Amazon EC2 and Rackspace in their instance-management console. In addition, the execution time for 15-minute Nowcasting of each weather data from the radar in the cloud instances is measured and the mean is calculated over the whole 1-hour interval.

As Table 2 shows, to generate 15-minute Nowcasts on the commercial cloud services, EC2's average computation time is 74.34 seconds and Rackspace's average is 96.53 seconds. The bootup time of the instances is approximately 7 minutes on average for both EC2 and Rackspace. Hence, generating the first 15-minute Nowcast takes about 8 minutes and 14 seconds for EC2 and 8 minutes and 36 seconds for Rackspace, whereas the subsequent 15-minute Nowcast generation takes only about 74 and 96 seconds for EC2 and Rackspace, respectively. Comparing their cost versus computation time to generate 15-minute Nowcasts, we see that EC2's computation time and cost is less than that of Rackspace.

We also perform the computation time analysis on the research cloud service instances with GENICloud and ExoGENI for the same weather data used to analyze the computation time of commercial cloud services. Table 2 shows the results of our analysis. Both research cloud services offer only one type of instance, which is sufficient for the standard operation of Nowcasting application. As Table 2 shows, both research cloud instances take less time (67.45 and 56.83 seconds, respectively) to compute 15-minute Nowcasts than EC2 and Rackspace. ExoGENI computes the 15-minute Nowcasts the fastest (just 56.83 seconds), and the bootup time for GENICloud and ExoGENI cloud instances are about 2 minutes on average compared to 7 minutes for EC2 and Rackspace.

As a proof-of-concept for our CloudCast application on cloud services, we carry out a live measurement on each of the four cloud instances to calculate the overall time taken for the Nowcasting process—that is, data is generated by the radar, the data is transmitted to the instance executing the algorithm, the 15-minute Nowcast images are generated, and the images are sent to a central webserver to be used by clients. The overall duration of the sum of the individual steps determines how much time a user has between when a severe weather situation is indicated by the Nowcast and when it actually occurs. Obviously, the goal is to maximize that time interval.

For the live measurement analysis, we use the data from our own radar on campus, which is a prototype CASA radar.[9] The last column in Table 2 shows the results from the live measurement carried out on the cloud instances. The average overall time taken for the whole

Nowcasting process was about 95.08 seconds for the EC2 cloud instance, of which 71.98 seconds is consumed in generating 15-minute Nowcasts by the algorithm running on the cloud instance. Thus, it takes about 23.10 seconds for the data to be sent from the radar to the receiving instance, create the predicted images, and transfer the images back to the central server to be accessible by clients. Similarly, the total time taken for the whole Nowcasting process on Rackspace, GENICloud, and ExoGENI cloud instances is 120.33, 78.60, and 72.07 seconds, respectively.

Thus, with our live measurement for the potential and feasibility of performing short-term weather forecasts for mobile devices in the cloud, we found that, for 15-minute Nowcasting, it takes only approximately two minutes to generate the Nowcast images and disseminate them to the client. That gives the clients 13 minutes to take any necessary action based on the 15-minute prediction. Additional information on Nowcasting accuracy on cloud services can be found elsewhere.[11]

Operation Costs

To better understand the operating cost of weather forecast models and the advantages of moving the forecasting to the cloud, here we provide a brief analysis of the forecasting cost.

In the spring of 2011, the CASA Engineering Research Center operated a four-radar network in southwest Oklahoma. From 2 April to 15 June 2011, an intensive operation period (IOP) was defined for a total of 75 days (1,800 hours), representing the climatological peak season for thunderstorms. During this time, the network, covering 10,300 square kilometers, had ongoing convective precipitation for approximately 90 hours, or 5 percent of the IOP. Several of CASA's derived products, including multi-Doppler winds and 15-minute reflectivity Nowcasting, are useful only during these events because X-band Radars aren't able to determine winds in clear air, and Nowcasting algorithms don't predict convective initiation. Our approach was to dedicate individual computers to each product despite the 95 percent idle rate and frequent over-provisioning during smaller scale and weaker events. The machines needed to process the data in a timely manner were purchased in 2011 and cost more than US$4,000 dollars each, not including IT overhead expenses associated with their management.

As a result of this experience, we looked into the IaaS cloud model, a more efficient compute-cloud-based architecture designed to procure computing resources on demand in an automated fashion. A lightweight command-and-control server differentiates between clear-air, stratiform rain and convective regimes, and issues Java-based in-line spot requests to Amazon EC2. Disk images preconfigured with various processing algorithms are uploaded in advance, triggered, and released as weather enters and exits the radar domain. The routines responsible for triggering more resource-intensive algorithms are integrated on-board the radar and require no additional maintenance or overhead. These include reflectivity thresholding (RT) and storm-cell identification and tracking (SCIT) with local radar data, as well as external monitoring routines, such as XML-based RSS feeds, for Weather Forecast Office (WFO) watches and warnings.

Based on current spot prices for machines similar to those used in the 2011 IOP (45 cents/hour), 90 hours of active use would cost about $40 per product, plus $2 per user to stream the resultant data out of the cloud. This represents significant cost savings over the dedicated compute model, assuming a five-year lifecycle. In addition to computing, long-term data storage of the radar data is another substantial cost. The 90 hours of moment data containing storms from four radars—combined with the derived merged products—amounts to roughly 700 Gbytes for the IOP. Current rates of 10 cents per gigabyte per month yield ongoing $70/month costs to keep this data online. Disk arrays are expensive to purchase and maintain, and the cloud storage model appears to be cheaper, although it has fewer advantages than the computing model.

We've shown that commercial and research cloud services are feasible for the execution of our real-time CloudCast application and can provide accurate, short-term weather forecasts to end users. We believe that CloudCast has the potential to support emergency managers and the general public in severe weather events by promptly providing them with potentially life-saving information. In the future, we would like to build a prototype of our architecture in collaboration with cloud services to run Nowcasting on demand for a longer experimentation.

Acknowledgments

We thank Ilia Baldine and his Renaissance Computing Institute (Renci) team for supporting the ExoGENI measurements. This work is supported by US National Science Foundation (NSF) grants CNS-0834243, OCI-1032765, CNS-1238485, and EEC-0313747. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and don't necessarily reflect those of the NSF.

References

1. E. Ruzanski, Y. Wang, and V. Chandrasekar, "Development of a Real-Time Dynamic and Adaptive Nowcasting System," *Proc. Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology*, 2009; https://ams.confex.com/ams/89annual/techprogram/paper_149633.htm.
2. E. Ruzanski, V. Chandrasekar, and Y. Wang, "The CASA Nowcasting System," *J. Atmospheric and Oceanic Technology*, 2011.
3. M. Yuen, "GENI in the Cloud," master's thesis, Dept. of Computer Science, Univ. of Victoria, 2010.
4. A.C. Bavier et al., "Transcloud—Design Considerations for a High-Performance Cloud Architecture Across Multiple Administrative Domains," *Proc. 1st Int'l Conf. Cloud Computing and Services Science*, Springer, 2011; http://christophermatthews.ca/files/closer-final.pdf.
5. M. Zink et al., "Closed-Loop Architecture for Distributed Collaborative Adaptive Sensing: Meteorogolical Command & Control," *Int'l J. Sensor Networks*, vol. 7, nos. 1–2, 2010, pp. 4–18.
6. B. Chun et al., "PlanetLab: An Overlay Testbed for Broadcoverage Services," *SIGCOMM Computer Comm. Rev.*, vol. 33, no. 3, 2003, pp. 3–12.
7. D. Nurmi et al., "The Eucalyptus Open-Source Cloud-Computing System," *Proc. 2009 9th IEEE/ACM Int'l Symp. Cluster Computing and the Grid*, IEEE CS, 2009, pp. 124–131.
8. K.E. Kelleher et al., "Project CRAFT: A Real-Time Delivery System for NEXRAD Level II Data via the Internet," *Bull. Am. Meteorological Soc.*, vol. 88, 2007, pp. 1045–1057; http://dx.doi.org/10.1175/BAMS-88-7-1045.
9. D. McLaughlin et al., "Short-Wavelength Technology and the Potential for Distributed Networks of Small Radar Systems," *Bull. Am. Meteorological Soc.*, vol. 90, 2009, pp. 1797–1817; http://dx.doi.org/10.1175/2009BAMS2507.1.
10. D.K. Krishnappa et al., "Network Capabilities of Cloud Services for a Real Time Scientific Application," *Proc. Local Computer Networks*, IEEE, 2012, pp. 487–495.
11. D.K. Krishnappa et al., "CloudCast: Cloud Computing for Short-Term Mobile Weather Forecasts," *Proc. Int'l Performance, Computing, and Comm. Conf.*, IEEE, 2012, pp. 61–70.

Dilip Kumar Krishnappa *is a PhD student and research assistant in the Electrical and Computer Engineering department at the University of Massachusetts Amherst. His research interests include cloud computing for scientific applications, multimedia, video-on-demand (VoD) systems, and green computing. Krishnappa has an MS in computer engineering from the University of Massachusetts Amherst. Contact him at krishnappa@ecs.umass.edu.*

David Irwin *is an assistant professor in the Electrical and Computer Engineering Department at the University of Massachusetts Amherst. His research interests include cyber-physical systems (CPSs) with an emphasis on sustainability, as well as topics that will influence future CPSs, including green computing, embedded sensor systems, and next-generation clouds and networks. Irwin has a PhD in computer science from Duke University. Contact him at irwin@ecs.umass.edu.*

Eric Lyons *is a research fellow with the Collaborative Adaptive Sensing of the Atmosphere (CASA) organization at the University of Massachusetts Amherst, where he has played a key role in the software design and integration of an intelligent Doppler radar network and has led operations since 2010. His research interests include researching techniques for network bandwidth reduction, dynamic control of heterogeneous systems, creation of meteorological feature detection algorithms, on-demand cloud computing, and creation of interactive displays for meteorologists and emergency managers. Lyons has a PhD in computer science from University of Massachusetts Amherst. Contact him at elyons@cs.umass.edu.*

Michael Zink *is an assistant professor in the Electrical and Computer Engineering Department at the University of Massachusetts Amherst. He also serves as deputy director for technical integration for the NSF engineering research for CASA, where he manages an interdisciplinary team of meteorologists, electrical engineers, social scientists, and computer scientists. He works in the fields of sensor networks and distribution networks for high-bandwidth data. Zink has a PhD in computer science from the Darmstadt University of Technology. Contact him at zink@ecs.umass.edu.*