Resource Management in Data-Intensive Clouds: Opportunities and Challenges

(Invited Paper)

David Irwin, Prashant Shenoy, Emmanuel Cecchet, and Michael Zink
University of Massachusetts, Amherst
140 Governors Drive
Amherst, MA 01267
{irwin,shenoy,cecchet}@cs.umass.edu, zink@ece.umass.edu

***Abstract*—Today's cloud computing platforms have seen much success in running compute-bound applications with time-varying or one-time needs. In this position paper, we will argue that the cloud paradigm is also well suited for handling data-intensive applications, characterized by the processing and storage of data produced by high-bandwidth sensors or streaming applications. The data rates and the processing demands vary over time for many such applications, making the on-demand cloud paradigm a good match for their needs. However, today's cloud platforms need to evolve to meet the storage, communication, and processing demands of data-intensive applications. We present an ongoing GENI project to connect high-bandwidth radar sensor networks with computational and storage resources in the cloud and use this example to highlight the opportunities and challenges in designing end-to-end data-intensive cloud systems.**

I. Introduction

After initial proposals over 40 years ago [19], cloud computing platforms have finally popularized the on-demand model of computation originally inspired by public utilities, where consumers provision and pay for computing resources only when they use them. The pay-for-use model is generally more cost-effective and efficient for both consumers and providers. While consumers tend to under-utilize their on-site IT facilities, providers are able to sustain high aggregate utilization by taking advantage of statistical multiplexing to simultaneously satisfy many consumers. Cloud platforms, such as Amazon's family of web services, using an Infrastructure-as-a-Service (IaaS) model provide abstractions that are general enough to support a wide range of existing distributed computing platforms tailored to specific application scenarios. Amazon allows consumers to rent virtual machines (EC2), storage volumes (EBS), and storage space (S3) on-demand and pay only for resources they use. While pricing models vary for each resource, consumers typically pay a fixed rate for both their length of use and their aggregate network and disk I/O bandwidth.

Many consider IaaS platforms a natural evolution of ongoing work on high-performance scientific and grid computing, which focuses predominantly on supporting large-scale execution of computationally-intensive scientific tasks [11]. Due to the generality of IaaS platforms, applications with other models of computation have also become increasingly popular. In particular, Google's family of services, including GFS [9], MapReduce [6], BigTable [4], and others [7], tailor abstractions for sub-tasks that are useful for efficiently storing and searching unstructured, and largely static, customer and web data. These projects have become increasingly popular beyond Google with the emergence of the open-source Apache Hadoop project. For instance, MapReduce exposes a programming paradigm that allows users to efficiently manipulate—by filtering, sorting, and aggregating—multi-terabyte, and even petabyte, datasets.

Unlike general scientific computations, which may encompass highly-synchronized distributed compute-bound tasks, MapReduce instead focuses narrowly on supporting batched processing of a special class of embarrassingly-parallel, data-intensive tasks. In addition to MapReduce, numerous other "big data" cloud platforms supporting a variety of data layout and consistency models, e.g., Dynamo [7], Cassandra, PNUTS [5], are also emerging. Thus, with the advent of a wide-range of these cloud-inspired data-intensive paradigms, cloud usage has grown increasingly data-intensive. However, interestingly, cloud platforms and their programming paradigms largely remain separate from, and agnostic to, both the data sources they operate on and the networks that transmit that data. As a result, there is an opportunity to improve coordinated provisioning of the cloud computing and storage resources that process and archive data with the data sensing and network resources that produce and transmit data.

In parallel with these developments in the cloud for supporting data-intensive applications, environmental sensors that collect data to understand and address both immediate and long-term environmental problems have become increasingly important pieces of societal infrastructure. These sensors are evolving from being largely disconnected and producing low-bandwidth data streams to being directly integrated into the network fabric and producing streaming high-bandwidth data streams. For example, weather radars, such as those used in the NEXRAD system in the United States, are capable of producing data rates near 200 megabits/second. Recently, scientists [23] have prototyped denser networks of smaller, steerable radars that produce similar data rates but provide more accurate higher resolution images of smaller regions than NEXRAD. Networks of pan-tilt-zoom video cameras being deployed across both the southern and northern border by the U.S. border patrol, as well as large astronomical radio telescopes, also represent real-world examples of high-bandwidth sensors that produce streaming data.

Unlike the largely static data sources in use on existing cloud platforms, high-bandwidth sensors have both time-varying resource needs and real-time performance demands. Since sensors collect data in the real world, their needs are driven primarily by unpredictable real world events. For instance, weather radars may produce more data and require higher-bandwidth during intense thunderstorms than during periods of calm. Likewise, pan-tilt-zoom cameras may require low latency network connections during times of intense border activity, but may not require network resources at all when performing conventional monitoring functions. In addition to time-varying needs, closed-loop adaptive sensor networks exhibit both latency and bandwidth requirements to transmit data to back-end processors, process the data, and use the result to influence sensor steering decisions. To experiment with these high-bandwidth sensor networks, we have deployed a small testbed of high-bandwidth sensors [17], [10], including both radars and cameras, in Western Massachusetts as a platform for experimentation as part the NSF GENI initiative [15].

In this position paper, we argue that the time-varying, data-intensive requirements of these high-bandwidth sensor network platforms can benefit from tight integration with both the compute and storage resources offered by cloud computing platforms, and the emerging "big data" cloud software platforms. We also discuss additional cloud requirements, notably layer 2 network integration, to facilitate high-bandwidth sensor network applications. We first lay out, in Section II, a motivating application scenario that highlights the benefits of coordinated provisioning of high-bandwidth sensors and cloud resources. We then use our application scenario to derive a set of architectural requirements in Section III for a general control plane that is able to accommodate end-to-end high-bandwidth sensing applications that also incorporates both networking and cloud resources. Finally, before concluding, in Section IV, we provide a brief overview of our ongoing work, focusing on our ViSE testbed for high-bandwidth sensors, as well as the GENI project. We discuss the benefits of integrating ViSE with GENI, and how GENI's goals are well-matched to integrating sensors with cloud substrates.

II. Motivating Sensor → Cloud Application

Our prior experience [10], [13], [17] has shown that a key characteristic of high-bandwidth sensor systems is the need for servers to process the data sensors produce in real-time to drive subsequent actuation. For example, our steerable weather radars are capable of producing data at a rate of nearly 200 megabits per second. In a network of these steerable radars [13], [23], data centers aggregate data feeds from multiple radars, process the data in real-time, and use the results to steer ("actuate") each radar in subsequent observations. The observe-process-actuate feedback loop gives these networks the potential to closely track fast moving weather phenomena, such as tornadoes. Weather radars are not the only real-world examples of high-bandwidth sensors capable of actuation: the U.S. Border Patrol is deploying networks of pan-tilt-zoom (PTZ) video cameras to continuously monitor the northern border for smugglers [14], and as part of a "virtual fence" on the southern border [8]. For instance, in related work [17], we multiplex PTZ cameras between concurrent adaptive sensing tasks, including continuous monitoring, object tracking, and fixed-point sensing.

A. Sensor Feedback Loop

High-bandwidth sensors capable of actuation have a number of characteristics that distinguish them from more typical low-power embedded sensors. Rather than being deployed "off-the-grid" in remote settings and communicating using wireless radios, the energy and bandwidth demands of these sensors necessitates connections to both the power grid and wired network links. The system is inherently distributed since multiple sensors may need to coordinate their actuations to achieve specific network-wide tasks, such as sensing the same region from multiple vantage points. In some cases, as with long-range radars that track mesoscale weather systems across entire regions, the coordination may cover large geographically-disparate areas. Further, high-bandwidth sensor systems are data-driven since they use sensor data to drive subsequent actuation and vice-versa. Thus, the resulting control loop must meet timeliness constraints on sensing, data processing, and actuation. For instance, if a severe weather system is approaching an area, there may be tighter timeliness requirements, requiring more resources, than during a calm period. Finally, applications with different goals may choose to steer these sensors in different directions. As a result, the system must be capable of service multiple concurrent applications with differing requirements [17], such as wind estimation to track tornadoes or rainfall estimation to predict floods.

To accurately sense changing phenomena, the network's feedback loop must be quick and responsive or else sensors may not be able to keep up. The key to a fast turnaround from initial observation to informed actuation is the availability of the network resources—to transmit the data—and the computing resources—to process it. Of course, a high-bandwidth sensor network's need for computing and network resources fluctuates over time based on its real-world observations. While high-bandwidth network links and significant computing power may be necessary to track a powerful storm, these resources may not be necessary during periods of relative calm. Due to these natural fluctuations, dedicating multiple high-bandwidth, and in some cases nationwide, network links along with entire data centers is not desirable, since these expensive and useful resources would otherwise sit idle for significant periods of time. Instead, the system's ability to *reserve*—during periods of intense activity—and *release*—during periods of inactivity—both computing and network resources is crucial. This is exactly the type of elastic behavior cloud computing platforms target. However, existing for-profit cloud computing platforms, such as Amazon's EC2, are available only using the public Internet, and are incapable of reserving backbone network resources and linking them to edge servers and storage.

As we show in Section IV, the ability to reserve high-bandwidth is important for the real-time data-intensive sensing applications our GENI/ViSE testbed supports. We have found that providing a Layer 2 fabric is also important in developing and managing holistic applications that include resources from multiple substrates, from sensors to network links to the cloud. We provide a high-level outline of an example workflow from a radar testbed to a back-end cloud computing platform, to highlight the benefits of linking together both sensing and cloud resources using high-bandwidth Layer 2 connections.

B. Radar Application

In radar sensing, a collection of daemons typically work together to gather, process, and transmit data to multiple destinations. An initial time-series daemon operates close to the radar to take streaming data from its analog-to-digital interface card and batch it together into one or more radials, where each radial contains sensor readings from a specific angle of the radar's antenna. This daemon also communicates with the radar over a separate control line to determine the angle of the data stream, *i.e.*, the position of the radar's antenna in space for each radial. Multiple daemons may communicate with the time series daemon to fetch the data for storage or manipulation. For example, one daemon may fetch the batched data and store it in files. For the radars we study in [13], the radar produces files every 30 seconds, where each file is roughly 1 gigabyte in size, at a rate of nearly 200 megabits per second. Alternatively, a moment data daemon may fetch the data to produce various moment data, including both reflectivity and velocity data. This daemon transforms the raw time-series data by reducing its resolution and normalizing its scale to make it comparable with other different types of radars. The moment data daemon may also write the data to a file for archival (at roughly 8 megabytes per file), generate graphics from the data to use on a local radar map, or post the data to an LDM (Local Data Manager) [12] queue that operates like a simple publish/subscribe system to distribute data products.

There may be multiple LDM receivers running on remote servers listening for new data to post to the queue. A server may also be listening for multiple queues to post data from multiple radars. Once the queues post data, software triggers various detection algorithms to execute on the data. The output of these detection algorithms may then inform subsequent actuation along a reverse path. If we consider applying this application in a severe weather scenario without the capability for reserving both network and cloud resources there are multiple potential bottlenecks. For instance, while there may be ample compute resources available for processing, the LDM queues must wait until enough data posts from multiple radars before triggering detection algorithms. Thus, without sufficient bandwidth the detection algorithms may stall. Since these algorithms not only provide immediate forecasting, but also inform the direction of future actuations, any bottlenecks are magnified because they actually delay subsequent sensing. In effect, the system behaves less like multiple independent workflows, and more like a closed loop pipeline, where any stall in the pipeline causes the entire loop to stall. Of course, if ample bandwidth is available, but ample data center resources are not, there may also be a stall in the data pipeline.

Since our system is setup as a collection of daemons that communicate over standard pipes and sockets, their placement in a network is flexible. For example, it is not feasible to stream raw time-series data over the public Internet. As a result, the moment data daemon typically operates locally to reduce the data's resolution, allowing the detection algorithms to run only over coarse moment data. However, higher bandwidth links will change the current trade-offs that dictate today's daemon placement. Another advantage of the workflow is that the input to each daemon may be either live data or archived files. Storing archived time-series and moment data in the cloud lowers the barrier to processing that data: for instance, scientists may generate the moment data above from raw time series data using a simple MapReduce job [6]. Further, the ability to package up such a sensing-to-processing workflow into one or more virtual appliances to enable turn-key deployment on different cloud computing substrates, such as Amazon's EC2 [1] or the open-source Eucalyptus project [2].

III. Requirements

Below we list a few requirements for next-generation cloud platforms to enable support for high-bandwidth sensors.

Dedicated Virtual Networks. Since current cloud platforms are only accessible over the public Internet, they do not offer latency or bandwidth guarantees to applications. Next-generation cloud platforms, including GENI, should allow consumers to reserve dedicated virtual network links from source to destination. Reserving both network and computing resources for elastic sensing applications requires policies at both the data center and network level to decide how to adapt co-located applications.

We are leveraging our current work on empirical measurements of virtual machine and cloud resource isolation [3], [20], as well as our recently developed algorithms to optimize the cost associated with reconfiguring an application's capacity in the cloud [18] based on these measurements, to make informed decisions. Additionally, we are leveraging our prior work on VM migration [22], [21] to inform these policies as well. We are currently expanding ViSE to include resources from EC2, connected via Layer 2 using OpenVPN, and connected to NLR's dynamic VLAN service (Sherpa) to provision high-bandwidth network links between ViSE and other GENI participants. In particular, ViSE connects to Eucalyptus clusters at Duke University and the Renaissance Computing Institute through provisioned and reserved NLR paths as part of GENI, which will soon be available as a research and deployment platform.

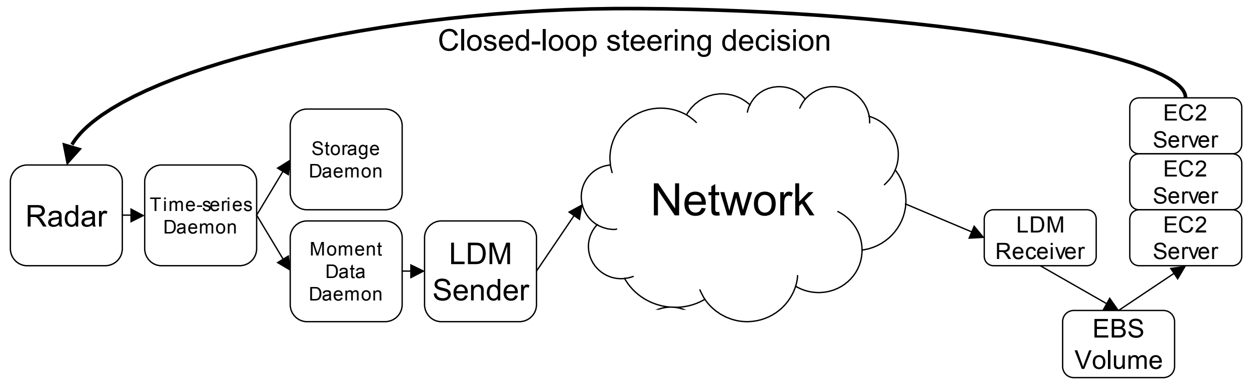

Fig. 1. A simple illustration of a radar workflow and closed-loop adaptive control that spans radar sensor, local processing nodes, the network, cloud storage, and cloud computing platforms. Data flows through multiple daemons and network links before reaching cloud processing platforms, which process the data to inform subsequent sensing tasks. Any of these steps, including the network as well as available cloud processors can serve as a bottleneck.

By including provisioned network links as part of a single platform for sensing applications, our vision is complementary to recent efforts, e.g., CASA [23] and LEAD [16], to make both sensors and IT resources more flexible in response to real-time weather phenomena. Our example radar workflows are inspired by our current joint work with CASA at the UMass-Amherst. While LEAD recognized the need for flexible and adaptive control of both sensing and IT resources, our goal is to link steerable sensors to cloud resources via provisioned network paths that quicken the observe-process-actuate feedback loop present in adaptive steerable sensor networks.

High-performance Storage Systems. In addition to on-demand access, commercial cloud storage platforms present characteristics that make them attractive for hosting high-bandwidth sensor network applications. For instance, clouds effectively offer infinite storage at a linear price that is decoupled from processing. Typically, storage costs follow a step function where each additional step may incur steep costs, especially for large disk arrays. Further, since storage platforms like EBS and S3 are typically replicated, consumers get data redundancy in addition to storage space. However, despite these characteristics, clouds will need to evolve to effectively support high-bandwidth sensor networks. Below we list a few requirements for next-generation clouds. Current cloud platforms leverage both virtual machine technology to isolate and multiplex resources and commodity disks. The performance of these technologies may not be suitable for high-bandwidth applications. Integrating high-performance storage systems will prove beneficial in these cases.

Large-scale Processing Platforms for Time-series Data. While there are many "big data" platforms available today, they do not focus specifically on the type of time-series data and processing required by sensors. Platforms, such as MapReduce, may be applicable to some types of time-series data processing. For instance, the simple data manipulations performed by the moment data daemon are easily transformed into MapReduce-style computations. However, computations of forecasts from time-series data typically involve numerical solutions to partial differential equations that do not conform to MapReduce's embarrassingly parallel style of computation. Other big data platforms largely focus on providing structure to efficiently query and retrieve unstructured data. For these style of computations, new platforms may be required to fully utilize the scale of cloud resources that are available.

IV. Ongoing Work

GENI consists of a federated collection of research testbeds donated by universities, industry research labs, and nation-wide networks. GENI's goal is to provide a shared research platform to support a wide range of realistic and repeatable network science and engineering experiments at scale. In many ways, GENI resembles a cloud computing platform: at its core, it must multiplex collections of heterogeneous physical and virtual hardware components among multiple concurrent experiments. Thus, basic commercial cloud computing platforms represent a partial realization of a core GENI function. As with GENI, cloud platforms expose open web services APIs for third-parties, *i.e.*, GENI researchers, to request isolated collections of virtualized hardware components, *i.e.*, GENI slices, to deploy their applications, *i.e.*, GENI experiments.

However, despite the similarities, there are key architectural differences between GENI and commercial cloud platforms. For instance, no single entity will operate the GENI prototype, since it will consist of a federation of autonomous testbeds hosted and controlled by a variety of institutions. Further, since GENI will initially operate as a non-profit sponsored by the NSF, resolving scheduling conflicts on highly utilized portions of the testbed may not be as easy as allocating resources to the highest bidder. Perhaps the primary architectural difference, though, results from GENI's goal to support a wide rage of hardware components, potentially including, not only machines, but also storage volumes, network links, mobile devices, and sensors. Exposing APIs that allow researchers to reserve entire networks of heterogeneous devices promotes the development of holistic end-to-end systems, including the application scenario we discuss in the next section, that combines real-world tasks, such as sensing, with back-end processing and storage tasks.

A. ViSE Integration

As part of GENI, we have integrated our ViSE [1] high-bandwidth sensor testbed with one of GENI's candidate control frameworks. ViSE currently consists of an Internet-accessible gateway node along with three geographically-distributed sensor nodes. Each node is roughly 10 kilometers from the others, and they communicate using 802.11b over long-distance antennas. Each node includes three distinct sensors, a Davis VantagePro2 Weather Station, a Sony SNC-RZ50N Pan-Tilt-Zoom Camera, and a Raymarine RD424 Radome Radar Scanner. We primarily use ViSE as a platform for experimenting with closed-loop control of adaptive sensor networks using steerable sensors. Applications actuate sensors to capture data at a specific time, location, spatial region, etc., stream that data over both wireless and wired networks to compute clusters for analysis, and use the new results to actuate and refocus sensors on important regions as conditions change. For example, recent work [13] explores how shared high-bandwidth sensor systems can intelligently prioritize and compress data when not enough bandwidth exists to transmit all of the sensor data.

B. Indirect Benefits of Integration

Beyond enabling new types of cross-hardware applications and experiments, integrating both high-bandwidth sensor testbeds and cloud computing platforms has a number of indirect benefits for GENI's prototype development. For instance, GENI's goal of developing a platform that supports the broadest possible range of network experiments covering the broadest possible range of substrates necessitates an extensible design, where core GENI entities, *e.g.*, Component/Aggregate Manager, Clearinghouse, Slice Controller, define interfaces for mapping their functions onto a range of different substrates and experiments [2]. For example, component/aggregate managers must support both high-level GENI functions, such as interactions with Clearinghouses, Slice Controllers, etc., and interfaces to interact with specific substrate technologies. While Clearinghouses need not interface with specific substrate technologies, they need substrate-specific knowledge to facilitate experiment resource discovery and allocation.

However, extensible platforms too often "can do anything, but are good for nothing." As such, their design and implementation must interleave both general platform development with specific platform use-cases. For GENI, a use-case requires stitching different architectural components together—from control frameworks, to substrates, to experiment workflow tools—and augmenting them to support a specific class of experiment. Integrating sensors with cloud computing resources represents an example of this type of "vertical" integration for a general class of data-intensive experiments. While our initial focus is on data produced from our own ViSE sensornet testbed, a data-intensive experiment may organize itself around a collection of data sources, intermediaries, and data sinks. For instance, the data sources could be high-bandwidth reflectivity data from a ViSE radar sensor, continuous web crawls, or streams of measurement data, the intermediaries could be processing nodes or network elements, and the data sinks could be storage volumes or object stores.

Integrating cloud computing platforms also offers the same benefits to GENI as it does to medium- to small-sized businesses: a cost-effective means for scaling the size of an infrastructure while holding human administrative burdens constant. Further, GENI has yet to incorporate virtualized storage allocated independently of processing nodes to archive sensor/measurement data, which is a pre-requisite for effective integration of high-bandwidth sensors. Thus, GENI can benefit from the addition of the storage paradigms already offered by cloud computing platforms, including Amazon's EBS and S3. Finally, sensors and storage volumes will likely have time-varying demands that are not correlated with the demand for compute servers. Integrating sensors and storage require GENI to support applications that bind different resources with different lifetimes together, such as computation and sensors (short lifetime) and storage (long lifetime).

[1]ViSE is an acronym for **Vir**tualized **S**ensing **E**nvironment

[2]See [15] for details on GENI nomenclature.

V. Conclusion

This position paper argues for tightly integrating high-bandwidth environmental sensors, such as weather radars and video cameras, with cloud resources using reserved network links. We describe a motivating application scenario derived from a heterogeneous high-bandwidth sensing testbed we are building as part of the GENI initiative. Since GENI's goal is to build a platform that supports research on a broad range of heterogeneous devices, it provides an opportunity to overcome current challenges in providing this type of coordinated provisioning between sensor networks, network providers, and cloud computing providers.

References

[1] Amazon elastic compute cloud. http://wwww.amazon.com/ec2, March 2010.

[2] Eucalyptus. http://www.eucalyptus.com, March 2010.

[3] Sean Barker and Prashant Shenoy. Empirical evaluation of latency-sensitive application performance in the cloud. In *Proceedings of the 1st ACM Multimedia Systems Conference*, Scottsdale, AZ, February 2010.

[4] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber. Bigtable: A distributed storage system for structured data. *Transactions on Computing Systems*, 26(2), June 2008.

[5] Brian F. Cooper, Raghu Ramakrishnan, Utkarsh Srivastava, Adam Silberstein, Philip Bohannon, Hans-Arno Jacobsen, Nick Puz, Daniel Weaver, and Ramana Yerneni. Pnuts: Yahoo!'s hosted data serving platform. *Proceedings of the VLDB Endowment*, 1(1), August 2008.

[6] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. In *Proceedings of the 6th Symposium on Operating Systems Design and Implementation*, December 2004.

[7] Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall, and Werner Vogels. Dynamo: Amazon's highly available key-value store. In *Proceeings of the Symposium on Operating Systems Principles*, pages 205–220, December 2007.

[8] A. Francoeur. Border patrol goes high tech. In *photonics.com*, August 2009.

[9] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. The google file system. In *Proceedings of the Symposium on Operating Systems Principles*, pages 29–43, December 2003.

[10] David Irwin, Navin Sharma, Prashant Shenoy, and Michael Zink. Towards a virtualized sensing environment. In *Proceedings of the 6th International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities*, May 2010.

[11] K. Keahey and T. Freeman. Science clouds: Early experiences in cloud computing for scientific applications. In *Conference on Cloud Computing and its Applications*, Chicago, Illinois, October 2008.

[12] Unidata local data manager. http://www.unidata.ucar.edu/software/ldm/.

[13] Ming Li, Tingxin Yan, Deepak Ganesan, Eric Lyons, Prashant Shenoy, Arun Venkataramani, and Michael Zink. Multi-user data sharing in radar sensor networks. In *Proceedings of the 5th ACM Conference on Embedded Networked Sensor Systems (Sensys)*, Sydney, Australia, November 2007.

[14] S. Magnuson. New northern border camera system to avoid past pitfalls. In *National Defense Magazine*, September 2009.

[15] GENI Project Office. The geni system overview. Technical Report GENI-SE-SY-SO-02.0, BBN Technologies, September 2008.

[16] B. Plale, D. Gannon, J. Brotzge, K. Droegemeier, J. Kurose, D. McLaughlin, R. Wilhelmson, S. Graves, M. Ramamurthy, R.D. Clark, S. Yalda, D.A. Reed, E. Joseph, and V. Chandrasekar. Casa and lead: Adaptive cyberinfrastructure for real-time multiscale weather forecasting. *Computer Special Issue on System-Level Science*, 39(11):56–63, November 2006.

[17] Navin Sharma, David Irwin, and Prashant Shenoy. Multisense: Fine-grained multiplexing for steerable sensor networks. Technical Report UM-CS-2009-033, University of Massachusetts at Amherst, April 2009.

[18] Upendra Sharma, Prashant Shenoy, Sambit Sahu, and Anees Shaikh. Kingfisher: A system for elastic cost-aware provisioning in the cloud. Technical Report In Submission, University of Massachusetts at Amherst, January 2010.

[19] I.E. Sutherland. A futures market in computer time. *Communications of the ACM*, 11(6):449–451, June 1968.

[20] Timothy Wood, Ludmila Cherkasova, Kivanc Ozonat, and Prashant Shenoy. Profiling and modeling resource usage of virtualized applications. In *Proceedings of the ACM International Conference on Middleware*, Leuven, Belgium, December 2008.

[21] Timothy Wood, Prashant Shenoy, Arun Venkataramani, and Mazin Yousif. Black-box and gray-box strategies for virtual machine migration. In *Proceedings of the 4th Symposium on Networked Systems Design and Implementation (NSDI)*, pages 229–242, Cambridge, MA, April 2007.

[22] Timothy Wood, Gabriel Tarasuk-Levin, Prashant Shenoy, Peter Desnoyers, Emmanuel Cecchet, and Mark Corner. Memory buddies: Exploiting page sharing for smart colocation in virtualized data centers. In *Proceedings of the ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments (VEE)*, Washington, D.C., March 2009.

[23] Michael Zink, David Westbrook, Sherief Abdallah, Bryan Horling, Vijay Lakamraju, Eric Lyons, Victoria Manfredi, Jim Kurose, and Kurt Hondl. Meteorological command and control: An end-to-end architecture for a hazardous weather detection sensor network. In *Proceedings of the Workshop on End-to-End, Sense-and-Respond Systems, Applications, and Services*, June 2005.