GridCloud: Infrastructure for Cloud-Based Wide Area Monitoring of Bulk Electric Power Grids

Dave Anderson, Theo Gkountouvas, Ming Meng, Ken Birman, *Fellow, IEEE*, Anjan Bose, *Fellow, IEEE*, Carl Hauser, Eugene Litvinov, *Fellow, IEEE*, Xiaochuan Luo, *Senior Member, IEEE*, and Qiang Zhang

***Abstract*—The continuing rollout of phasor measurement units enables wide area monitoring and control (WAMS/WACS), but the difficulty of sharing data in a secure, scalable, cost-effective, low-latency manner limits exploitation of this new capability by bulk electric power grid operators. GridCloud is an open-source platform for real-time data acquisition and sharing across the jurisdictions that control a bulk interconnected grid. It leverages commercial cloud tools to reduce costs, employing cryptographic methods to protect sensitive data, and software-mediated redundancy to overcome failures. The system has been tested by ISO New England and the results reported here demonstrate a level of responsiveness, availability, and security easily adequate for regional WAMS/WACS, with the capacity for nation-wide scalability in the future.**

***Index Terms*—Power grids, power system control, power system security, wide-area monitoring systems.**

I. Introduction

TODAY'S bulk power grid lacks a wide-area monitoring infrastructure. Instead, the grid is structured as a federation of Reliability Coordinators (RC). Each region is monitored and managed by an independent system operator (ISO) or Regional Transmission Organization (RTO), which coordinates with transmission owners (TOs) to ensure a secure, reliable and efficient regional power system.

In current systems, data sharing occurs across regional "borders" to the extent needed to accomplish this objective, but on a need-to-know basis defined by narrowly-scoped peering agreements. Thus, in today's bulk grid, each ISO or TO's control system is running a separate supervisory control and data acquisition (SCADA) software, separate state estimation software to track voltages, power flows, transformer tap positions, breaker statuses, etc. Data is independently displayed using multiple visualization tools.

As the bulk power grid evolves in response to changing patterns of power generation and use, this approach to coordination and cooperation will also need to evolve. Wide area monitoring is needed to enable regional grid coordination, management, and problem solving. The GridCloud system, built on commercial cloud computing technologies and described here, represents a new capability, introducing a shared data system through which distinct utilities, ISOs and TOs can cooperate, as seen in Figure 1. By so-doing, it enables a new class of control paradigms that require joint actions by more than one ISO or TO, and enables distributed problem solving based on data from widely-dispersed sources.

This paper describes and evaluates the GridCloud prototype, key elements of which are seen in the figure. On the left, data flows into the shared platform from authenticated sources. GridCloud, running on a cloud-computing platform, captures and archives the data into a standard data collection infrastructure, which can include standard databases and grid-specific solutions such as OpenPDC [14], [20]. Applications track system state in real-time and can perform a variety of analyses (new analytic tools can easily be added). On the right, collaborating operators and systems draw upon the shared data and analytic outputs to manage the grid. The entire infrastructure is tightly secured against unauthorized access. Data and code replication is possible both within a single cloud data center and across any desired number of data centers in different geographic locations.

Our prototype demonstrates a state estimation and visualization application: it takes data collected from a regional deployment of phasor measurement devices, carries out a linear state estimation using the input, then displays the state estimator output. Additionally, state estimation enables oscillation detection, wide-area protection (RAS/SPS) and wide-area control. The system's historical data capture ability is intended to support off-line applications such as event reconstruction, parameter estimation, model development and scenario simulation. Overall, the goal is to support applications aimed at facilitating desired forms of regional cooperation and coordination, through a mix of on-line analysis and forensic data exploration.

Manuscript received April 5, 2017; revised September 6, 2017; accepted November 6, 2017. Date of publication January 8, 2018; date of current version February 18, 2019. This work was supported in part by ISO New England under PSERC Project S–67G and in part by the U.S. Department of Energy ARPA-E GENI Program under Award Number DE-AR0000230. Paper no. TSG-00470-2017. *(Corresponding author: Anjan Bose.)*

D. Anderson, M. Meng, A. Bose, and C. Hauser are with the School of EECS, Washington State University, Pullman, WA 99164-2752 USA (e-mail: daveanderson@wsu.edu; ming.meng@email.wsu.edu; bose@wsu.edu; chauser@wsu.edu).

T. Gkountouvas and K. Birman are with the Department of Computer Science, Cornell University, Ithaca, NY 14853-7501 USA (e-mail: tg294@cornell.edu; ken@cs.cornell.edu).

E. Litvinov, X. Luo, and Q. Zhang are with Business Architecture and Technology, ISO New England, Holyoke, MA 01040 USA (e-mail: elitvinov@iso-ne.com; xluo@iso-ne.com; qzhang@iso-ne.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSG.2018.2791021

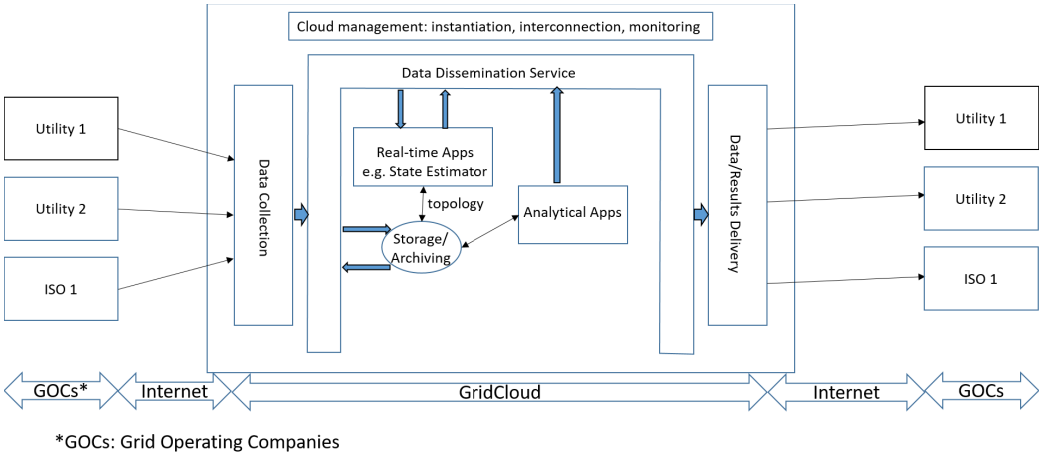

Fig. 1. Conceptual Architecture of the GridCloud system.

II. New Data Sources

Today's SCADA and Energy Management (EMS) systems track power system state with noteworthy efficiency and effectiveness, but future systems will require new approaches [19]. In particular, today's state estimation systems track the state of the power grid using SCADA, with scan rates of seconds and using data that is not time-stamped to high accuracy, resulting in state calculations that are error-prone and infrequent. The relatively recent large-scale deployment of phasor measurement units (PMU) [5], [11] enables precise tracking of the state of the grid at data rates of 30Hz or higher. PMU devices are thus capable of revealing transients and other signs of system stress that would be impractical to detect with the older style of SCADA monitoring [16]. Rapid, accurate tracking of the state of the grid opens up many new possibilities for downstream applications, the most promising of which are wide-area protection and control applications.

Because the PMU scan rates are much faster than SCADA, the methods currently used for exchanging SCADA data between grid operating companies are far too slow, especially if the results might trigger protection or control actions over a wide area that spans several jurisdictions. It is difficult at this stage in the evolution of PMU-using applications to give precise specifications for data delivery latency and data quality: the application space is still evolving rapidly. What is clear, however, is that with lower latency and higher data quality come more opportunities for innovative applications. The NASPInet specification identifies three classes of applications with projected latency requirements ranging from 50ms to 1s [26]. (Two additional off-line classes require only best-effort latency.) Bakken *et al.* [27] provide a detailed analysis of projected quality-of-service requirements for eight classes of applications with latency requirements ranging from 5ms to more than a second.

GridCloud solves this problem in a cost-effective way, leveraging the commercial cloud yet achieving a high degree of performance, security and robustness. We show that GridCloud could potentially track the full state of the U.S. national power grid with delays of about 250ms, can track regional state with far smaller delays, and is robust to a wide range of disruptive conditions and failures.

Although GridCloud focuses on wide area monitoring of PMU data and PMU-based state estimation, down the road we hope to expand the system to capture other forms of relevant data. For example, micro-PMU devices and wall-plug voltage monitoring devices [18] have been proposed for monitoring within distribution networks. Advanced digital substations can report a tremendous amount of operational state data. The list goes on, yielding a rich collection of data streams that could potentially be exploited. Some aren't tracked at all today, while others are monitored only for narrow purposes.

Beyond the uses discussed above, collecting more (and more diverse) data would enable development of machine intelligence applications that could rapidly sense instabilities, detect signs of attack, recommend possible adjustments to the network operating parameters, etc. GridCloud is intended as a first cautious step down this path. But the goal also implies that security will be of tremendous importance even if the platform won't be playing mission-critical operational roles anytime soon (indeed, even if it never plays such roles). For example, the network model and PMU state measurements are highly sensitive data under NERC critical infrastructure protection guidelines. If loss of SCADA comes to be viewed as a contingency that ISOs must anticipate and address, as some have recommended in the wake of SCADA damage as part of an attack against the bulk power grid in Ukraine during 2015 [16], GridCloud could be used to backup SCADA, bringing even stronger security requirements.

III. Why the Cloud?

Currently, data sharing for power grid operations largely takes place using the Inter Control Center Protocol (ICCP, also known as TASE.2). Using ICCP, a utility or ISO/TSO can grant its neighbors selective access to specific data fields in the utility's energy management system. The approach implicitly presumes that sharing will be limited to a relatively small number of specific data points representing the current system state. Were the ICCP model applied in operating regimes that require operators to have shared understanding of

the wider system state, including historical state, scaling and administrative overheads seemingly become prohibitive.

Data sharing between ISOs could be supported with a modest shared computing cluster, jointly operated by the ISOs, but we see several reasons that this is not likely to occur. First, inter-company administrative complexities associated with system acquisition and operation represent a serious barrier to this model. Second, on a technical level, the interconnectivity and operational systems required for such a cluster would yield a solution quite similar to what already exists in today's cloud. In recent years practitioners and researchers have noted the possible applicability and advantages of using cloud computing resources in this field, for example [23] and [24]. Besides addressing the difficulties with purpose-built solution mentioned above a cloud-based approach has the further advantage of not facing scaling boundaries as it grows to monitor most or all of the interconnected bulk grid on the North American continent.

Zweigle's analysis of full PMU visibility of the North American grid [19] suggests that at full scale the aggregated GridCloud input data rate might reach 40 to 100Gb/s, or more if a wider range of inputs are tracked [3], [4]. In other domains within the computing industry, such problems are commonly solved with cloud computing: a widely decentralized infrastructure for collection and analysis of streaming data, operating redundantly for fault-tolerance at geographically distributed locations. However, the cloud evolved over a long period of time, at great expense. This leads directly to the main argument for a cloud-based solution: *the cost of creating a new computing infrastructure dedicated to managing the national power grid from scratch would be prohibitive.* Indeed, the NASPINet effort proposed a ground-up architecture [11], but it quickly became clear that implementing NASPINet as a standalone entity would require billions of dollars and a decades-long effort. Accordingly, although inspired by NASPINet, we decided not to adhere to the specific architecture proposed by that effort.

GridCloud was created to validate the hypothesis that a new kind of mission-critical real-time solution can be hosted on today's cloud, by a mixture of careful attention to security policy together with additional technology aimed at high-availability and real-time performance. The approach required only a modest incremental development cost, and yielded an open software platform that can be shared freely without licensing restrictions and readily extended by users.

The core questions that arose were these. First, today's cloud is optimized for a style of use that differs deeply from our goals. To give just one example, today's cloud is famous for its relaxation of consistency in favor of fast response at massive scale [12]. We see this when Facebook briefly serves up stale profile pictures after a user updates his or her photo. By displaying stale data rather than pausing to obtain the most current copy, cloud systems achieve rapid end-user responsiveness and reduce load on authoritative databases that track the most current data, but would never be able to keep up with the full rate of real-time queries (driven by Web page requests). In contrast, bulk grid operators need strong consistency: different viewers should see the same data, and updates to state should be promptly evident; otherwise, those operators might work at cross-purposes or even take actions that could damage the grid. Thus, in building GridCloud we were forced to retrofit stronger forms of consistency into a framework that doesn't currently offer the needed options.

Similarly, configuring cloud security policies to fit a nationally critical infrastructure use-case isn't trivial. Today's cloud has a tremendous range of security options, including cryptographic data protection for connections into the cloud, firewalls, and even cryptographic storage: in the Amazon Web Services (AWS) framework on which we focused our work, there are operators with access to security keys and technicians with access to storage devices, but no individual has both. Our approach is to view the GridCloud system as a hybrid: a cloud-hosted data sharing platform that would be used by applications that might integrate the shared data with in-house systems having access to data too sensitive for upload. Even so, the security requirement is far stronger than for most cloud uses.

Related to security, there is the question of trust between the entities that are sharing data. Research such as that in [25] addresses this using cryptographic means. On the other hand, operating entities already share data based on NERC's Operating Reliability Data Confidentiality Agreement—essentially adopting a legal/contractual solution to the trust problem rather than a technological one [22].

IV. GridCloud: Architecture and Key Components

The GridCloud concept, as depicted in Fig. 1, is a platform that supports *collection* of real-time data from grid operating companies (e.g., utilities and ISOs) in the cloud; delivery of those data to *real-time applications* through a *data dissemination service* that also supports delivery of results from the applications to other applications and ultimately back to the GOCs; a *historian and archiving service* that stores the real-time data and makes it available for later use by *analytical applications*. Real-time applications may also use the storage service to obtain non-real-time data such as system models and topologies that are not delivered as part of the real-time data streams. Cloud management components in the platform configure and monitor the computational and network resources that it uses and support fault-tolerant operation across multiple cloud data centers.

The depiction in Fig. 1 highlights how GOCs interact with GridCloud and how applications can take advantage of GridCloud services. There is, of course, more complexity under the covers. As alluded to earlier, commercial cloud computing offerings are tuned for a rather different kind of application than those envisioned for GridCloud.

GridCloud uses *replication* to enhance availability, performance, and scalability of the system. A first strategy is *sharding:* data is spread over multiple physical resources, and saved redundantly to ensure that even if some number of faults occur, the system remains available. Sharding can be applied to the data collection service, to the storage and archiving service, and to applications themselves, and

thus contributes to scalability, availability and performance. Additionally, multiple instances of a service (or shard of a service) may be deployed. This can take the form of multiple service instances within a single cloud data center or complete replication of an entire GridCloud system at multiple provider data centers. As noted earlier, the cloud is not notable for its consistency guarantees, and yet replication creates acute data consistency requirements. Within GridCloud, a virtual synchrony service [1], [2] is used to provide applications with consistent data snapshots and to help the cloud management service maintain the GridCloud computing instances and their interconnections in a consistent way. The discussion that follows explores these aspects, starting at a low level and working upward.

A. The GridCloud Data Collection Layer

GridCloud users use robust networks to acquire PMU sensor outputs within their service areas. To forward this data to GridCloud, they deploy *data relay nodes*. These are under control of the utility but outside its firewall, and are employed only to ensure that the IT security team can maintain control of both the internal data sources as well as the first hop external to the corporate firewall. On arrival at the cloud, the incoming data pass through the cloud operator's firewall and then are received by a component of GridCloud called the *data collector* [8].

The data relaying path needs to be both secure (via cryptographic methods) and redundant, for two reasons: fault-tolerance and performance. Cloud data are ultimately routed over the TCP protocol, and hence subject to erratic throughput and potential connectivity outages. Accordingly, GridCloud allows use of redundant connections from the relay node to the data collection layer. Thus, to tolerate K-1 possible delays or faults, a single piece of data would be sent K times to each cloud data center, with each link terminating in a different node within the data collector layer. The data collectors can write received data to the archiving service and relay them to online applications that prefer streams of input.

It follows that with D data feeds, and with GridCloud replicated on R data centers, and with a redundancy factor of K, a data relay might make as few as R*K connections (if sending multiple data streams over a single shared link), or as many as D*R*K connections (if each data feed is streamed on its own connection). Because the data collector programs can be co-hosted, we would need at least K cloud computers for the data collection role, and at most D*K of them. In practice, we use far fewer than the maximum number.

The NERC CIP requires that the number of network connections between an ISO and any external system be minimal, and requires the ISO to document and monitor each connection. Thus from an IT security point of view, one would ideally prefer that $D = 1$ and $K = 1$ for each cloud system, all the PMU data would tunnel through a single link. We tested that approach, using a standard protocol called SSH for the secure tunnel. A contrasting approach would be to have D*K redundant connections: K from *each* PMU sensor, connected to at least K separate cloud data collectors, using a separate secure TCP link for each (secured by the TLS protocol). We tried this too, using $K = 2$ for an experiment that looked at the ISO New England system, and also using $K = 3$ in a larger experiment that had over 4500 simulated PMUs and mimicked a west-coast deployment scenario.

Our experiments reveal that GridCloud has the best real-time behavior when greater redundancy is used, and (perhaps surprisingly), that having more TCP connections did not measurably increase overheads or compute costs. The benefit of redundancy is that it eliminates a form of fate sharing that became highly evident with a shared TCP connection, while communication latency turned out to be insignificant relative to the latency of state estimation (which we did not need to replicate). Actual dollar costs of redundancy are low: in the cloud, there are no charges for streaming data into the platform, and the data collectors run on shared, inexpensive cloud instances. Although disabling security for our data links would clearly violate our security objectives, we also measured the costs of the cryptographic protection. As it turns out, cryptographic security does not add any measureable delay.

We noted before that an ISO takes on administrative and monitoring obligations for each connection it makes to an external compute service. It is therefore worth stressing that there is actually a security *benefit* to having D*K connections in the case of denial-of-service attacks: if the data collection system were ever attacked, the attacker would need to carry out a much larger scale of intrusion to significantly disrupt the system. In our view this consideration, and the strong security for the individual connections afforded by independent cryptographic protection, justifies having multiple connections for this particular case.

B. GridCloud's Archival Data Storage Subsystem

The GridCloud data collectors operate by writing the received data into files and simultaneously forwarding the data for state estimation processing (in effect, rather than deciding between a file-oriented compute model and a streaming model, we provide both). A third option that was considered but rejected is the so-called *data warehousing model*, popular in database systems, whereby data is captured, cleaned to remove data seemingly inconsistent with the majority of the input, reformatted into standard database representations, and then stored into relational tables. Although data warehousing is widely used in the cloud computing industry, GridCloud's goals include cases ill-matched to a database model.

For example, suppose that an intruder were to break into a SCADA infrastructure and to begin to experiment with small disruptive actions, as a first step towards some eventual plan to later undertake a major attack. It would be very hard to also break into collection of PMU devices deployed both by the ISO itself and by its neighbors. Thus were we to use GridCloud to continuously compare a PMU state estimate with the SCADA data, we would be in a position to immediately notice any departures between the SCADA/EMS output and the PMU-based state estimation. System operators, investigating the deviation, would have a reasonable chance of discovering the SCADA intrusion and rectifying

the problem long before the intruder could undertake the full-scale attack. Of course an intruder could break in and then lie low, but without having experimented with an exploit, would have reduced confidence that it works. Thus, a redundant monitoring capability increases security against attack.

When an unexplained deviation from the expected system state is noticed, operators would often engage in "data mining" to try and understand the root cause. Here we see the benefits of archiving as much raw data as possible. Whereas a typical data warehousing approach would reduce redundant data to a single input stream, in doing so the system might scrub evidence of the intrusion. In contrast, by keeping raw data in the original form, we also preserve the option of going back to that data with some form of new analysis that responds to an unexpected contingency occurring later.

The example highlights a kind of flexibility and power that comes from delaying any kind of decision that might delay (as for time-alignment), reformat, combine or discard legitimate data (indeed, it even makes sense to log data in duplicate or triplicate, for fault-tolerance within our cloud archive). Accordingly, GridCloud is conceived as the start of what could become a new style of "big data" curation and analytic platform for the bulk grid, in which extensive historical data archives could be used more and more frequently in a variety of ways: for real-time collaboration among regional operators, for planning studies, after-the-fact forensic investigations, etc.

GridCloud stores data in a variety of formats, favoring standards. For example, we store streams of PMU data into append-only logs in the IEEE C37.118 format [21]. We store network state estimates as a table, in which each row describes an element of the network model for the grid and gives that element's estimated state. Our current data sources all have embedded GPS timestamps: our storage format records both the original GPS timestamp in the PMU data, and the platform time at which the data was actually archived. By so doing, we can sense situations in which a GPS time receiver malfunctions and streams data with incorrect timestamps.

Some applications run on continuous, real-time data streams, but many analytics would run on the temporal archive. The problem arises of bridging from a collection of files to the representation convenient for analytic purposes. Power grid computations are best expressed as tensor or matrix problems and are most often carried out using libraries that originated in the HPC arena. Thus we need a simple way to describe the mapping of archived data in the desired format. Here, we are working to make a wide set of popular data analytic tools available within our system, including widely used computational ones such as MATLAB. The intent is to make it as easy as possible for the ISO to express applications in high level form, and then map the application directly and cleanly to reads and writes against our file system. Because the incoming data forms a temporal stream, a typical read operation accesses a particular record *at a specific time.* Further, since we are looking at states that evolve within milliseconds or tens of milliseconds, the quality of temporal data access will shape the quality of our output. This suggests a style of file system access often described as a "snapshot" model, in which computations run against (time-consistent) snapshots of past states.

There were a number of candidate file systems for this archival storage role, but upon review, we identified limitations that precluded use of existing solutions: the prior work on snapshots embodied temporal inaccuracies, performance issues, and inconsistencies when files are being updated rapidly (see [15] for details of the systems we reviewed). Accordingly, we developed a new solution: the Freeze Frame File System (FFFS) [15], which offers a quick way to access any desired instant in time, from any point in the past or present history of the system, and employs a memory-mapped access structure and hardware assisted data transfers (RDMA) to achieve high performance in support of smart grid computational tasks. FFFS also provides logical consistency, as defined by Chandy-Lamport [7].

By using a file system for data capture, we sought to make GridCloud flexible without locking its users into particular platforms, so that users can share data but still make distinct choices about which products will ultimately be used for analysis and other tasks. GridCloud can also be easily extended to capture types of data outside the classic power systems options. FFFS offers a direct solution to the core problem (scalable and highly fault-tolerant file storage with fast temporal search capabilities), and yet is also easily integrated with existing platforms.

In most settings where GridCloud will be used, we anticipate that in addition to performing state estimation and related analysis immediately upon freshly captured data, data would be copied to other platforms (often ones physically hosted in the control centers of the operators sharing the GridCloud solution). Thus, an ISO that prefers to use vendor-supported data analytic solutions such as OSISoft's PI server (and its cloud-hosted data sharing tools), or AspenTech (the AspenOne system), would do so, and yet we avoid imposing the obligation for our users to adopt any particular product. This avoids a barrier to adoption: in our target domain, each operator typically expects to make its own choices of data analytic products and technologies.

C. Cloud Manager

With so many moving parts, the question arises of how to keep the whole system running, particularly as users bring new applications into GridCloud. The existing cloud system managers focus mostly on an elasticity model quite different from our 24x7 real-time needs. Accordingly, we created our own robust system management tool. Cloud Manager (CM) [9] is an extension of the Unix "Makefile" infrastructure. The tool supports the standard dependency-triggered model used in Make (and the standard syntax, too), but extends the model to also react to system events such as nodes failing or joining, events being sensed in the incoming data stream, etc. We encode such events as XML files and when something occurs, the file update triggers reactive behavior by CM. Actions are similarly initiated: CM outputs a file, and we use this to drive the desired response. We use an off-the-shelf constraint solver to carry out the optimizations needed for purposes such as

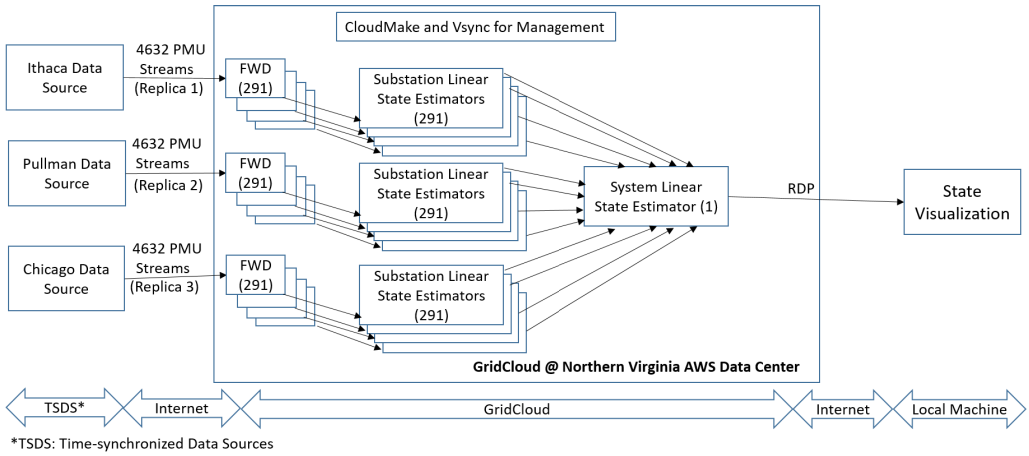

Fig. 2. GridCloud system as built for ARPA-E research illustrating use of extensive sharding and within-data-center replication for performance, scalability and reliability.

job placement on the available cloud nodes, deciding which data collector will handle which data stream, etc.

CM also manages the desired suite of applications. At present, CM is focused on a state estimator and visualization tool, but new applications can easily be added to the system and either configured to run continuously, 24×7, or on demand. CM can also automatically configure applications with redundancy: in our experiments, we've worked with either one or two Amazon data centers, and in one case, we went further and also triple-replicated our entire cloud infrastructure within one Amazon AWS system, to understand the extent to which doing so could conceal scheduling and other latency factors.

D. Vsync Library: Resilient Software and Data Replication Layer

Many parts of our system require fault-tolerance. Vsync [2] is a new version of an older style of software library: the Isis Toolkit, which supports virtual synchrony (an execution model closely related to the Paxos "state machine replication" model, but with additional features to support higher performance where application update semantics can be leveraged [1], [10]). We use Vsync to replicate key components of the GridCloud system, coordinate actions within CM, etc.

E. Linear State Estimator (LSE)

We treat the LSE as an application running on GridCloud, not one of its core components. In our experiments, we used an LSE created at WSU [16]. The LSE can be deployed either as a single application (which can be replicated) or as a hierarchical application in which a state estimator is deployed for each substation and those results combined to produce the state estimate for the whole system. This is a complex application, that includes its own internal communication bus (GridStat: a pub-sub system with built-in understanding of PMU data streams), a database (OpenPDC, which tracks PMU deployments, the properties of each PMU device and its configuration status, *etc.*), and a state estimator program. CM is used to track the health status of the LSE components and, if anything fails, can cleanly shut down and then reboot the entire LSE application. The current LSE, adopted and adapted from an earlier project, duplicates internally some services that could be obtained directly from GridCloud if the LSE application were to be re-implemented. From a performance and size-of-code perspective the current situation is not ideal, but on the other hand it does provide a fairly significant test of the CM's ability to manage a complex application.

F. GridCloud Instantiations

GridCloud was created under an ARPA-E contract that emphasized demonstration on simulated data at large scale—i.e., for systems with thousands of data streams—while maintaining low-latency in the loop of measurement, data delivery to the cloud, analysis in the cloud, and delivery of results from the cloud to power system control centers. The ARPA-E-funded work was directed at the long-term opportunities associated with cloud computing for power grid operations. Later, a grant from the Power Systems Energy Research Center (PSERC) enabled us to customize GridCloud as required to accept data from the ISO NE data sources.

Figure 2 illustrates the initial GridCloud configuration that we used for the ARPA-E research. In a single cloud (AWS) data center, we replicate (most of) GridCloud three times, sending data in triplicate from our PMU data sources (left), to replicated (3x) substation-level state estimators. The collection layer (FWD in the diagram) and the substation decomposition are examples of sharding. The replicated substation state estimators feed a single system-level state estimator that uses the first data to arrive for each substation as the basis of its state estimation. This configuration handled over 4500 PMU data streams and performed state estimation for a roughly 6000-bus simulated WECC-like system. This implementation demonstrated the value of replication and the first-to-arrive strategy for improving both performance (latency) and consistency. However, the high replication levels bring also high cloud computing costs.

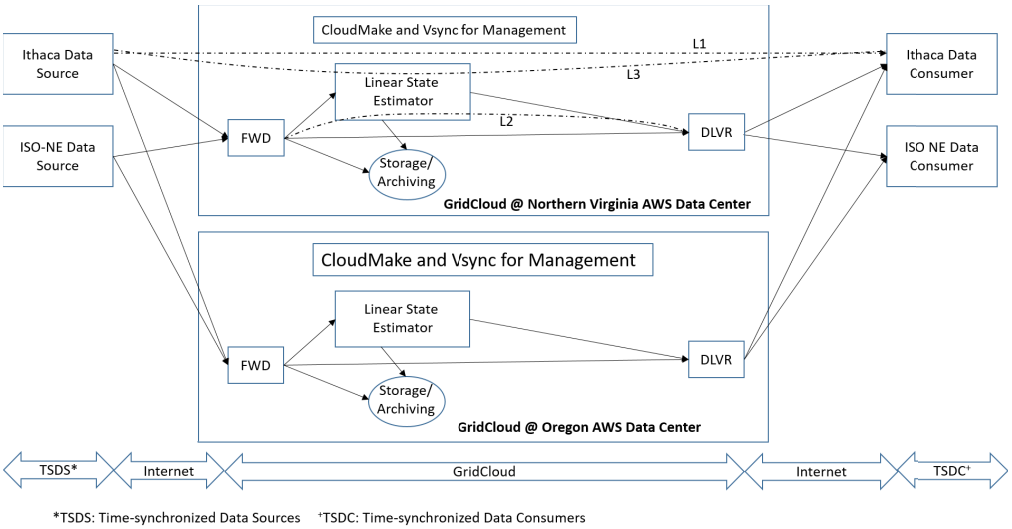

Fig. 3. GridCloud as built for ISO-NE, scaled for current needs and illustrating use of replication across two data centers. Labels L1, L2, and L3 identify several paths over which latency is measured (see text).

Figure 3 illustrates a GridCloud configuration that we created jointly with ISO NE to demonstrate and evaluate the concepts for more immediate application. The evaluated system involved data streams from 73 PMUs that collectively provide observability of the ISO-NE 345kV transmission system. These data streams, along with a model of the system, are sufficient to allow the LSE to compute the state of the ISO-NE 345kV system. The cloud part of the system is replicated between data centers in Virginia and Oregon.

The evaluated system delivers 30-sample-per-second data from the 73 PMUs, sharded across two data sources to the cloud. GridCloud components deliver these data to the Data Archive running in the cloud, and to a monolithic LSE where they are down-sampled to 5 times per second. Results from the LSE are delivered as synthetic PMU data streams in C37.118 format to consumers in the cloud and, via the delivery nodes (DLVR in Fig. 3), "on the ground".

V. Experimental Findings

In order to perform accurate latency measurements and to overcome cybersecurity concerns that would be associated with using data from the live transmission system, the evaluated system differs from how GridCloud would be deployed operationally in the following ways:

1. The PMU data are taken from a historical recording of the ISO-NE PMUs; the recorded data are replayed into the GridCloud system at 30 samples per second with accurate GPS timestamps, in C37.118 format.
2. The system topology used in the system model is fixed and the recorded data correspond to a time when that topology accurately described the system.
3. Rather than sending the data from 73 separate playback sources, the data are sent from two senders in the role of the data relays described above. The two senders are at different locations in order to emulate what is expected when GridCloud is fed with data from different entities. The two locations were ISO-NE (Holyoke, MA) and Cornell University (Ithaca, NY).
4. Results from the computations and communications conducted in the cloud are delivered back to the Cornell data sending machines to accurately compute round-trip latencies.

The evaluated system nevertheless is a useful model of the system as it would be in an operational deployment: the data rates are the same and the path lengths are similar. An actual deployment might have a greater diversity of paths if data are sent directly from the PMUs to the cloud, or it might be very similar to the evaluated system if PMU streams are funneled through a utility gateway node. Additional functionality to support topology changes would complicate the system but would not add latency to individual state estimation computations.

Having created the system, we undertook an evaluation of its behavior under realistic configurations, varying replication factors and the locations of the data centers used to host the platform. The incoming data are sent in duplicate to GridCloud instances running on two Amazon AWS data centers: one in Virginia and a second one in Oregon. The geographic diversity is intentionally extreme in order to investigate performance limitations both when widely separated entities are sharing data through GridCloud and when GridCloud resources are moved away from an anticipated or actual impact zone of a natural disaster. (For example, a utility on the East Coast could easily move its GridCloud resources across the country in anticipation of an event like Hurricane Sandy at the cost of additional latency).

Specific measurements of interest were:

1. L1: the round-trip latency from the Data Source machines to the cloud delivery node. Half of L1 approximates the time delay from when a measurement is made until when it can be used in a computation in the cloud. L1 is independent of the specific application and is essentially the contribution of the Internet and AWS network infrastructure to the total delay.

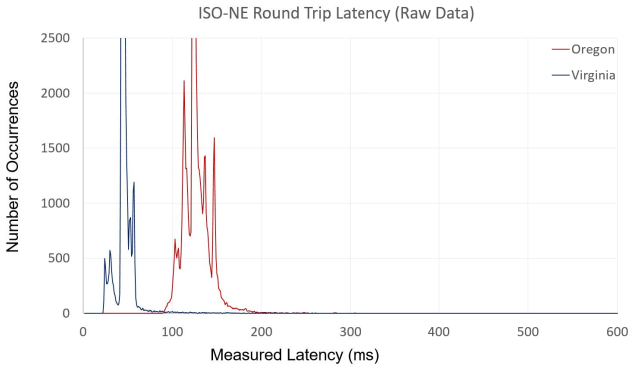

Fig. 4. L1 latency to a nearby data center (blue) and to one across country (red): In effect, a round-trip "ping" to GridCloud.

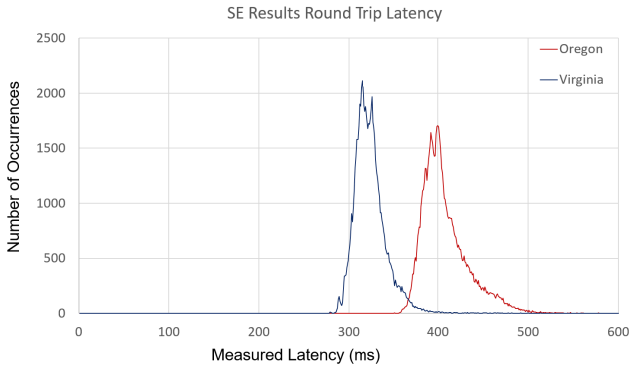

Fig. 5. L3 latency with a nearby data center (blue) and to one across country (red). Relative to Figure 4, the costs of writing data into the archive and performing state estimation are now included. Instrumentation of GridCloud internals suggests that time-alignment within the SE is the dominant source of delay (that is, the task of collecting a full set of inputs matched to a specific instant in time).

2. L2: the latency from when data arrives in the cloud until the corresponding LSE output is ready. L2 is LSE-specific and gives a sense of the lower limits for using the LSE in the cloud environment as it takes into account delays within the cloud but not delays associated with getting data to and from the cloud.
3. L3: round-trip latency for obtaining results from the LSE back at the Data Source machines. L3 is also LSE-specific and gives a sense of the latencies that entities using results from LSE in the cloud could expect. The distributions of L3 latencies are presented in Figure 5.

In addition to these latency measures we also studied: the additional latency caused by using encryption on the data streams, relative to using no encryption; the time taken to automatically restart GridCloud resources and reconnect them following a failure; and the ability of two GridCloud instances running simultaneously to deliver consistent results. Our primary findings were as follows:

- Metric L1: with all PMU devices tunneled over a single shared SSH link, raw data reached the Virginia system within approximately 25ms (round-trip 50ms) and Oregon within 65ms (round-trip 125ms). In contrast, by connecting to each PMU using a separate TCP connection, mean round-trip delays drop to 27ms and 76ms, respectively. The shared tunnel accounts for the correlated delays. In the earlier ARPA-E experiments, better performance was achieved by sending each PMU stream on 3 side-by-side TCP connections, then running LSE as soon as an adequate data set was received; however, for operational convenience ISO-NE preferred the single, shared connection used here. The current experiment establishes the cost of that choice.
- Metric L2: Delay through the LSE is approximately 250ms. Data alignment in the LSE accounts for approximately 175ms of this delay: the LSE needs an adequate set of PMU inputs before computing each "cycle" of estimated system state. Data alignment is not computationally intensive, but requires idle waiting to ensure that all of the data are present. The number of state estimates computed per second is governed mainly by the computing resources devoted to the computation. Using an Amazon c4.xlarge instance (4 hyper-threads on a Xeon e5-2666v3) we found that we can track state at a 5 Hz rate within the region consuming less than 50% of the instance's available computing cycles. Improving the performance of the LSE provides the greatest opportunity for improving the overall system's performance as it is the component over which we have the most control.
- Metric L3: the round-trip latency for LSE computation is approximately 300ms—essentially the sum of L1 and L2—to the closer data center. The distributions of L3 latencies are presented in Figure 5. Together, Figures 4 and 5, convey a sense both of what latency has been achieved with GridCloud today (Fig. 5) and floor on what could be achieved through improvements in the LSE alone (Fig. 4).
- Since TCP, a lossless transport protocol, was used for delivering data to and from the cloud, there were no data losses at the transport level; however, delivery that occurs too late to be useful in the state estimation computation for a given time slot is lost as far as the application is concerned. We observed such losses at the application level at the rate of 0.013% at the Virginia data center and 0.057% at the Oregon data center.
- Encryption adds just 2-3ms of delay relative to an unencrypted connection, and an AWS VPC security perimeter adds approximately 16ms of delay, while file encryption added no measureable delay at all. Given this finding of minimal impact, and because data confidentiality is desirable, if not required, all remaining measurements were made with encryption enabled.
- A full shutdown and restart of the Virginia data center requires 175s from outage to restoration of full service (the Figure 3 system). In the experiments where we replicated GridCloud 3 times within a data center, we were able to fully mask TCP link failures, failures of individual nodes running the data collectors, or failures of an LSE replica (the Figure 2 system).
- Each data center of the Figure 3 system has 13 AWS instances and costs a total of $2.47/hour of operation; replication scales this up linearly.

VI. Discussion and Conclusion

GridCloud demonstrates that the smart grid (and other complex real-time applications with strong security and consistency needs) can be supported on AWS and similar cloud platforms. The system uses redundancy and replication to mask the various sources of delay and jitter, and adds new components to manage the system more aggressively than the norm within AWS itself, and for tamper-proof real-time archival storage of the collected data. The added cost of cryptographic security was surprisingly low.

It is interesting to note that the kinds of requirements cited above for low latency and high availability are not specific to the bulk power grid. Tomorrow's smart highways, augmented with technology to assist in safely guiding smart vehicles, would need a very similar structure. Smart urban landscapes that shape traffic flows by adjusting traffic light patterns in response to congestion would as well. There is much interest in smart buildings and office complexes. Indeed, we see a huge number of Internet of Things (IoT) use cases where our GridCloud approach might be useful. Thus, over time, a cloud more friendly to mission-critical wide-area monitoring and control applications could certainly emerge, with GridCloud as an early example, but ultimately, one of many.

References

[1] K. P. Birman, *Guide to Reliable Distributed Systems. Building High-Assurance Applications and Cloud-Hosted Services*. (Texts in Computer Science). London, U.K.: Springer. 2012.

[2] K. P. Birman. (Feb. 2010). *Vsync: Consistent Data Replication for Cloud Computing*. [Online]. Available: http://vsync.codeplex.com.

[3] K. P. Birman, L. Ganesh, and R. van Renesse, "Running smart grid control software on cloud computing architectures," in *Proc. Workshop Comput. Needs Next Gener. Elect. Grid*, Apr. 2011, pp. 1–28.

[4] K. Birman, M. Jelasity, R. Kleinberg, and E. Tremel, "Building a secure and privacy-preserving smart grid," *SIGOPS Oper. Syst. Rev.*, vol. 49, no. 1, pp. 131–136, Jan. 2015, doi: 10.1145/2723872.2723891.

[5] *Phasor: Power Engineering*. Accessed: Jan. 19, 2017. [Online] Available: https://en.wikipedia.org/wiki/Phasor#Power_engineering

[6] (2015). *DOE Quadrennial Technology Review*. [Online]. Available: http://energy.gov/qtr

[7] K. M. Chandy and L. Lamport, "Distributed snapshots: Determining global states of a distributed system," *ACM Trans. Comput. Syst.*, vol. 3, no. 1, pp. 63–75, Feb. 1985.

[8] T. Gamage *et al.*, "Mission-critical cloud computing for next-generation power applications," in *Smart Grids: Clouds, Communications, Open Source, and Automation*, D. Bakken and K. Iniewski, Eds. Boca Raton, FL, USA: CRC Press, 2014.

[9] T. Gkountouvas, K. Birman, and D. Anderson, "CloudMake: Adaptive configuration of distributed systems," Dept. Comput. Sci., Cornell Univ., Ithaca, NY, USA, Rep. CS-TR-10-3-15, Oct. 2015. [Online]. Available: htttp://gridcloud.codeplex.com

[10] P. J. Marandi, S. Benz, F. Pedone, and K. Birman, "Practical experience report: The performance of Paxos in the Cloud," in *Proc. 33rd IEEE Symp. Rel. Distrib. Syst. (SRDS)*, Nara, Japan, Oct. 2014, pp. 41–50.

[11] (2008). *North American Synchrophasor Initiative (NASPI)*. [Online]. Available: https://www.naspi.org/

[12] S. S. M. Shim, "(Editor for special issue). The CAP theorem's growing impact," *IEEE Comput. Mag.*, vol. 45, no. 2, pp. 21–22, Feb. 2012.

[13] Z. Teo, V. Kutsenko, K. Birman, and R. van Renesse, "IronStack: Performance, stability and security for power grid data networks," in *Proc. 44th Annu. IEEE/IFIP Int. Conf. Depend. Syst. Netw. (DSN)*, Atlanta, GA, USA, Jun. 2014, pp. 792–797.

[14] *OpenPDC: The Open Source Phasor Data Concentrator*. Accessed: Jan. 19, 2017. [Online]. Available: http://openpdc.codeplex.com/

[15] W. Song, T. Gkountouvas, K. P. Birman, and R. van Renesse, "The freeze-frame file system," in *Proc. ACM Symp. Cloud Comput.*, Clara, CA, USA, Oct. 2016, pp. 307–320.

[16] T. Yang, S. Hongbin, and A. Bose, "Two-level PMU-based linear state estimator," in *Proc. IEEE/PES Power Syst. Conf. Exposit. (PSCE)*, Seattle, WA, USA, Mar. 2009, pp. 1–6.

[17] R. M. Lee, M. J. Assante, and T. Conway, *Analysis of the Cyber Attack on the Ukrainian Power Grid*, document, E-ISAC and SANS-ICS, Bethesda, MD, USA, Mar. 2016, accessed: Feb. 27, 2017. [Online]. Available: https://ics.sans.org/media/E-ISAC_SANS_Ukraine_DUC_5.pdf

[18] Y. Liu. *FNET/GridEye: A Low-Cost, GPS-Synchronized Wide-Area Power System Frequency Measurement Network*. Accessed: Jan. 19, 2017. [Online]. Available: http://powerit.utk.edu/fnet.html

[19] G. Zweigle, "Emerging wide-area power applications with mission critical data delivery requirements," in *Smart Grids: Clouds, Communications, Open Source, and Automation*, D. Bakken and K. Iniewski, Eds. Boca Raton, FL, USA: CRC Press, 2014.

[20] *OpenPDC*. Grid Protect. Alliance, Chattanooga, TN, USA, accessed: Feb. 27, 2017. [Online]. Available: https://github.com/GridProtectionAlliance/openPDC

[21] *IEEE Standard for Synchrophasor Data Transfer for Power Systems*, IEEE Standard C37.118.2-2011, 2011.

[22] *Confidentiality Agreement for Electric System Reliability Data, Version 3*, North American Elect. Rel. Coporat., Atlanta, GA, USA, 2009, accessed: Aug. 18, 2017. [Online]. Available: http://www.nerc.com/comm/OC/Operating%20Reliability%20Data%20Confidentiality%20Agreeme/NERC-ORD-Version-3-BOT-approved-050609.doc

[23] S. Rusitschka, K. Eger, and C. Gerdes, "Smart grid data cloud: A model for utilizing cloud computing in the smart grid domain," in *Proc. 1st IEEE Int. Conf. Smart Grid Commun.*, Gaithersburg, MD, USA, 2010, pp. 483–488.

[24] K. P. Birman, L. Ganesh, and R. van Renesse, "White paper running smart grid control software on cloud computing architectures," in *Proc. Comput. Needs Next Gener. Elect. Grid Workshop*, 2011, accessed: Aug. 21, 2017, pp. 1–39. [Online]. Available: https://energy.gov/sites/prod/files/FINAL_CompNeeds_Proceedings2011.pdf

[25] Y. Tong, J. Deyton, J. Sun, and F. Li, "S^3A: A secure data sharing mechanism for situational awareness in the power grid," *IEEE Trans. Smart Grid*, vol. 4, no. 4, pp. 1751–1759, Dec. 2013.

[26] *Phasor Gateway Technical Specifications for North American Synchro-Phasor Initiative (NASPInet)*, U.S. Dept. Energy, Washington, DC, USA, May 2009, accessed: Aug. 25, 2017. [Online]. Available: https://www.naspi.org/sites/default/files/reference_documents/73.pdf?fileID=590

[27] D. E. Bakken, A. Bose, C. H. Hauser, D. E. Whitehead, and G. C. Zweigle, "Smart generation and transmission with coherent, real-time data," *Proc. IEEE*, vol. 99, no. 6, pp. 928–951, Jun. 2011.

Dave Anderson is a Lead Programmer for grid-related software projects with Washington State University and the Lab Infrastructure Manager for the GridStat Research Group. He oversaw the design and implementation of the communication portions of GridCloud and managed the deployment and evaluation of the platform in the Cloud.

Theo Gkountouvas is currently pursuing the Ph.D. degree with the Department of Computer Science, Cornell University.

Ming Meng is currently pursuing the Ph.D. degree in electric power engineering with the School of Electrical Engineering and Computer Science, Washington State University.

Ken Birman (F'15) is the N. Rama Rao Professor of computer science with Cornell University, where he leads a group that works on fault-tolerant distributed computing solutions for critical-infrastructure applications. He was a recipient of the 2009 IEEE Kanai Award. He is a fellow of ACM.

Anjan Bose (F'89) is currently a Regents Professor and holds the endowed Distinguished Professor in power engineering with Washington State University, where he also served as the Dean of the College of Engineering and Architecture from 1998 to 2005. He is a member of the U.S. National Academy of Engineering, and a Foreign Fellow of the Indian National Academy of Engineering.

Carl Hauser is a Clinical Associate Professor of computer science with the School of Electrical Engineering and Computer Science, Washington State University.

Eugene Litvinov (F'13) is the Chief Technologist with ISO New England. He has over 35 years of professional experience in the area of power system modeling, analysis and operation, electricity markets design, implementation, and operation, and information technology.

Xiaochuan Luo (SM'17) is a Technical Manager with ISO New England, where he is responsible for the technology strategy, research, and development in power system planning and operations.

Qiang (Frankie) Zhang is a Senior Analyst with the Department of Business Architecture and Technology, ISO New England. His responsibilities include synchrophasor data and their applications and power system stability analysis and control.