21, rue d'Artois, F-75008 PARIS
http : //www.cigre.org

Type here your Paper number
6 characters, 9 for joint meetings
(to be centred)

CIGRE 2018

Cloud-based Data Exchange Infrastructure for Wide Area Monitoring of Bulk Electric Power Grids

D. ANDERSON,
S. SAHASRABUDDHE,
A. BOSE, C. HAUSER
Washington State University
USA

T. GKOUNTOUVAS,
W. SONG, Y. LIAO, K. BIRMAN
Cornell University
USA

E. LITVINOV, X. LUO, F. ZHANG
ISO New England
USA

A. DARVISHI, G. STEFOPOULOS,
A. ETTLINGER,
New York Power Authority
USA

SUMMARY

The GridCloud system is designed to leverage scalable and inexpensive cloud computing resources in support of bulk power grid monitoring, control, and coordinated operations. Basic functionality includes a data collection layer that can capture data from sensors such as synchrophasor measurement units (PMUs), a novel file system into which data are archived for extremely fast and reliable retrieval using temporal indexing, and a cloud manager used to configure the system and repair it after server failures. In the new work reported here, ISO-New England and New York Power Authority shared real-time data through GridCloud, streaming PMU data over secure links, then requesting state estimation. We illustrate this sharing by running a continuous state estimation in the cloud, yielding output that can be visualized on the cloud platform, securely exported to operator control rooms, or used as input for further analysis. State estimation is just one of many options, and the entire platform is designed to be easily extended. To underscore this opportunity, we show how we bridged from GridCloud's data archive to the Spark/Databricks analytics infrastructure, which offers a mix of classic computational tools, modern machine-learning packages, and big data analytics. We illustrate the resulting capability by applying a standard machine learning technology to the real-time data stream captured by the system (specifically, we train a convolutional neural network to predict bus voltages in a wide-area power network, and show how it can be used to estimate network state even when a major weather event is causing a severe disruption that has compromised normal monitoring). The broader point is that GridCloud can support a diversity of tools, including both standard power grid control applications and novel ones that use the extensive programming capabilities, machine-learning tools, and data analytics support available in modern cloud settings.

KEYWORDS

Synchrophasor measurement units, power system monitoring, data analytics, smart grid, GridCloud, convolutional neural network.

bose@wsu.edu

Introduction

Today's bulk power grid is structured as a federation of ISO/TSO/RTOs. Each region is responsible for its own system security and reliability. Each runs its own supervisory control and data acquisition (SCADA) infrastructure, selecting its own preferred vendors, then customizing the selected control products. This flexibility extends to choices of software for state estimation, voltage tracking, power flow control, transformer tap positions, breaker status, data archiving, data visualization, etc.

Diversity is in many ways beneficial. Because each organization makes its own decisions, each control system can be customized to address its specialized needs. Were a product to have a technical limitation or security flaw, the scope of impact would be limited to control centers that use it. Thus, today's arrangement promotes autonomy and offers resilience and security through diversity. Yet diversity is also a barrier to real-time tracking and control of the bulk grid state at multi-region scale. In fast-evolving situations (such as during disruptive storms), operators in distinct regions lack a common shared view of state, which is needed for remediating issues in a consistent and coordinated manner. There have been several episodes in which lack of consistency led to errors.

The industry lacks a single repository that could collect, archive or share data collected across wide multi-region areas. Absent such a repository, wide-area control tasks occur in a decentralized way. Need-to-know peering agreements define data that can be shared, the format that will be used, how the shared data may be used, and the obligations of each party with respect to security, reliability, and coordination. The shared responsibility for grid stability is carried out without real-time consistency: operators in different centers make decisions using local versions of the system state, yet know that their counterparts might be observing somewhat different data at the same instant in time.

Beyond simply holding data, a shared repository could host applications for data analysis and to assist in the tasks of day-to-day regional grid management, contingency response and planning. To the extent that shared tools are used, operators would be guaranteed consistency. On the other hand, since each control center has its own preferred analytic tools, the repository would also need ways to export data security and with low delay. In this manner, each of the organizations sharing the repository could easily extend the core capabilities of the platform itself with their own new applications, and could integrate its output into the tools they normally rely upon, in a manner open to a diversity of vendors. A diverse set of security, fault-tolerance (high availability), scalability and performance requirements are raised by this vision, and must be addressed by any solution. Cost of ownership is also an important pragmatic consideration.

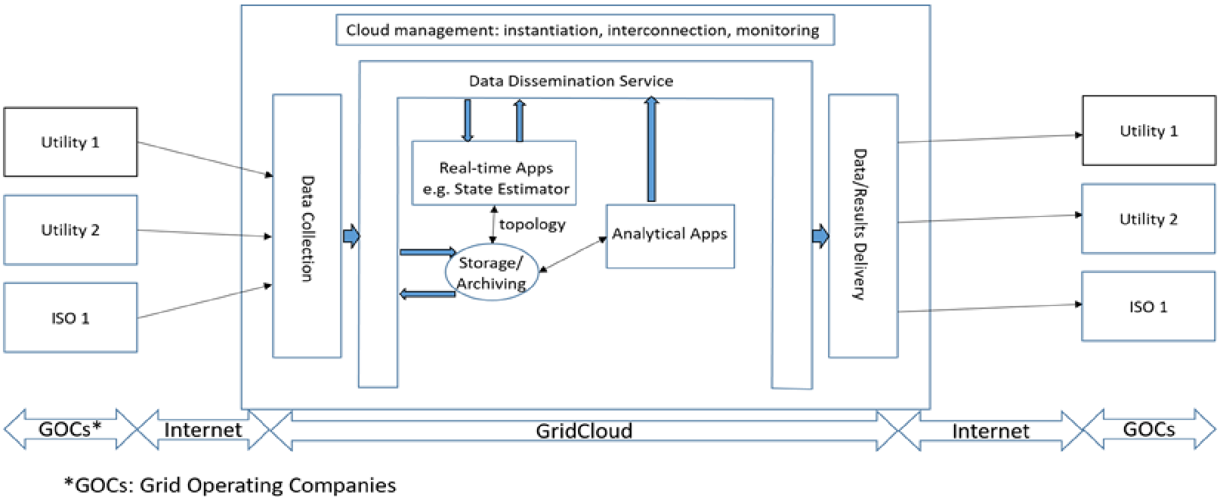

Figure 1: Conceptual architecture of the GridCloud system

Were the industry to adopt a shared "consensus" repository for global state, the resulting platform could be useful in many situations. During normal operation, it would facilitate day-to-day coordination, enabling operators to explore power sharing models that require such a capability. In the wake of a disruptive event, it would assist in understanding that event and correcting root causes. There is growing attention to the risks of large-scale manipulation or attacks on the power system; a wide-area monitoring and data mining capability would be a requirement for a strong response to attacks. More broadly, the power community has been increasingly intrigued by the potential of big data analytics and machine learning. Today, such tools are wedded to the current cloud computing infrastructure, which employs standards and tools quite different from those seen in control centers. Bridging from a grid data repository to cloud data analytic platforms would provide access to them.

The GridCloud platform (Figure 1), was created to enable secure data sharing between operators using inexpensive and scalable cloud computing resources. GridCloud is intended as an adjunct to today's systems, running side-by-side with standard SCADA, and assisting operators who remain in full control. The platform is open source, free and vendor-neutral. The results reported in this paper were obtained by implementing this platform on Amazon Web Services (AWS).

By deciding to host GridCloud on a commercial cloud, the cost of operation is greatly reduced, but a number of security, performance and reliability challenges arise. These issues are discussed in [5][3][14] and have also been raised by other researchers[13][2]. The present paper goes beyond our own prior work [1][7], looking at a new set of questions that were raised by two significant new steps. First, our previous work supported just a single operator, ISO-NE, hence only one set of security policies needed to be considered. In our new work, New York Power Authority (NYPA) also shares data through the platform, and the system is thus required to comply with both security frameworks.

A second innovation is a new data pipeline connecting GridCloud to a data archiving framework, then bridging into an analytics infrastructure. In our prior work data reaching GridCloud was relayed using a message bus, processed using a linear state estimator (LSE), and then visualized. In the activity reported here, we also archive data and export it to the Spark/Databricks analytics platform[16], with delays of just a few milliseconds from when data reaches our data collection infrastructure to when analytic tools can access it. This enables real-time use of a wide variety of solutions that would traditionally have only been available for offline purposes [6].

Our bridge to Spark/Databricks is new, and we present it carefully here. In contrast, our actual use of the Spark offerings is preliminary, and the example employed below is intended purely as an illustration of the potential of the technology and a demonstration that the pipeline we've created really works as claimed. A substantial amount of future research will be needed to understand how best to use the diversity of tools that are suddenly at our disposal through this pathway.

Accordingly, the contributions of this paper are:

- We evaluate GridCloud in a setting where data from multiple operators are shared, with strong consistency, using standard cloud security mechanisms and cryptographic communication options in a manner compatible with two distinct security policies.
- We connect GridCloud to the Spark/Databricks analytics framework, by creating Spark "resilient distributed data" (RDD) objects from real-time snapshots offered by GridCloud's new Freeze Frame File System (FFFS) [11]. Delays from when data is received in our platform to when these tools can operate on it vary in the AWS setting: sub-millisecond latencies are common, but we have observed latencies as high as 25ms under normal conditions, and even higher when a fault arises in the datacenter network, a machine fails, or a failed machine is restarted.
- To illustrate how the linkage can be used, we play simulated data for a bulk grid into GridCloud, store it into an archive in FFFS, extract RDDs, and then show how a "convolutional" neural network (a standard machine-learning model) can be trained to predict states for unmonitored buses. We then mimic a highly disruptive weather event (a tornado that cuts communication to a great many PMU devices), and show that the trained neural network can still estimate the state of individual buses with reasonable accuracy. However, we offer this demonstration purely as an

illustration of the technology, and do not undertake a deep exploration of the robustness of such an approach, or a comparison with a more standard linear state estimation approach.

- A number of issues will need to be explored carefully before the potential of machine learning and big-data analytics can safely be leveraged by the power community. We highlight open issues, limitations, and questions for future study.

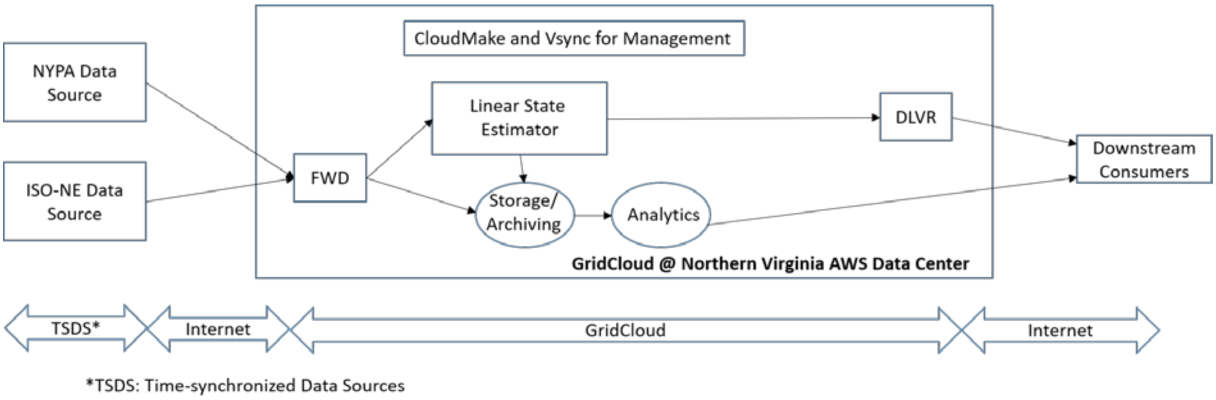

Figure 2: GridCloud as configured for ISO-NE and NYPA experiments.

In this section, we offer a very brief review of GridCloud. Further details of our prior work appear in [1][7]. GridCloud was created with support from the U.S. Department of Energy ARPA-E program as a tool for data sharing between regional bulk grid operators. The key goal was to understand whether, by leveraging cloud computing infrastructures, we could achieve a low-cost solution without sacrificing performance and security relative to custom-designed solutions such as the NASPInet architecture, a design for a wide-area PMU-based grid state monitoring and control system that was created as part of the National Synchrophasor Initiative [10] that was never fully implemented or deployed (primarily, for reasons of cost).

In the prior work, we showed that cloud communication and data-security features are adequate to protect this style of computing, but also identified technology deficiencies in the standard cloud, which we remedied. Solutions we created as elements of GridCloud include its storage system (the Freeze Frame File System, FFFS [11]), which offers substantially better real-time and consistency properties than were possible in the standard cloud, has better scaling and raw speed, and is fault-tolerant. Our CM “cloud management” tool proves real-time self-repair after disruptive failures, and the GridCloud data collection tools interoperate with these elements. We demonstrated GridCloud by connecting it to data sources at ISO-NE, then using WSU’s Linear State Estimator (LSE) [14] to track network state. For fault-tolerance against loss of an entire cloud data center, we mirrored the entire GridCloud system at two locations. To explore scaling, we ran experiments in which the system tracked as many PMUs as would be needed in a national-scale deployment.

However, none of these experiments pulled data from multiple regions, or shared data between operators in distinct regions. In the work on which we focus here, we expand the deployment of GridCloud to also accept data from NYPA. The NYPA data become part of a shared data repository and are also used to create a consensus state estimate. The state estimate can then be shared back to ISO-NE and NYPA, offering them ways to simultaneously visualize system state, or to use the LSE output as input to their own local decision making.

In our previous work with ISO-NE, the evaluated system involved data streams from 73 PMUs that collectively provided observability of the ISO-NE 345kV transmission system. In this work ISO-NE added data from 12 more PMUs within New England and NYPA added data from 31 PMUs in New York, resulting in a total of 894 synchrophasor measurements being transmitted to the cloud in every measurement period (30 periods per second). The system configuration is shown in Figure 2. The data used in the experiments were recorded over a 1-hour period in 2017 and are played back continuously as if they were current. Playback is performed by data source machines located respectively at ISO-NE and NYPA. The datasource machines are equipped with clocks synchronized to GMT with an accuracy of 5ms or less, using NTP. Thus the data streams sent to GridCloud resemble, to the best of our ability, real-time synchrophasor data streams from actual PMUs. The data are received in the cloud by two data concentrators (without time alignment). In the cloud, GridCloud components deliver the data to the Data Archive and the LSE. The LSE down-samples the data to 5 times per second and the results are delivered as synthetic PMU data streams in C37.118 format to consumers in the cloud and, via the delivery nodes (DLVR in Fig. 2), to consuming applications "on the ground". 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 and portions of the NYPA 765kv, 345kv, and 230kv systems. The SE computes results for between 104 and 113 buses in these systems depending on which valid synchrophasor measurements that it receives in each interval (which varies over the course of the hour).

For geographic fault-tolerance, GridCloud can send all the data to two or more GridCloud instances running in different AWS data centers. In previous experiments we used data centers in Virginia and Oregon [1]. Such geographic diversity supports both sharing between widely separated entities and a use case in which GridCloud resources are moved away from an anticipated or actual impact zone of a natural disaster. For example, at the cost of additional latency, a utility on the East Coast could easily move its GridCloud resources across the country in anticipation of an event like Hurricane Sandy. The current experiments were conducted using only the Virginia data center.

Experimental Findings

The experiments reported here were designed to test the behavior of GridCloud with disparate data feed qualities from the ISO-NE and NYPA. Relative to previous results [1], improvements to core components resulted in a higher level of reliability and better performance. All of the results reported here were obtained over the course of a 100 minute run of the system. During this time, the difference in data feed quality between the two sources is seen in Figure 3 where histograms of the round-trip latencies are displayed. Latencies were measured over the same SSH tunnels as were used to carry the PMU data. Note the sharp lower bound on the NYPA latencies (27.7ms) and their somewhat higher average (28.8ms) compared with the ISO-NE average round-trip time (25.2ms).

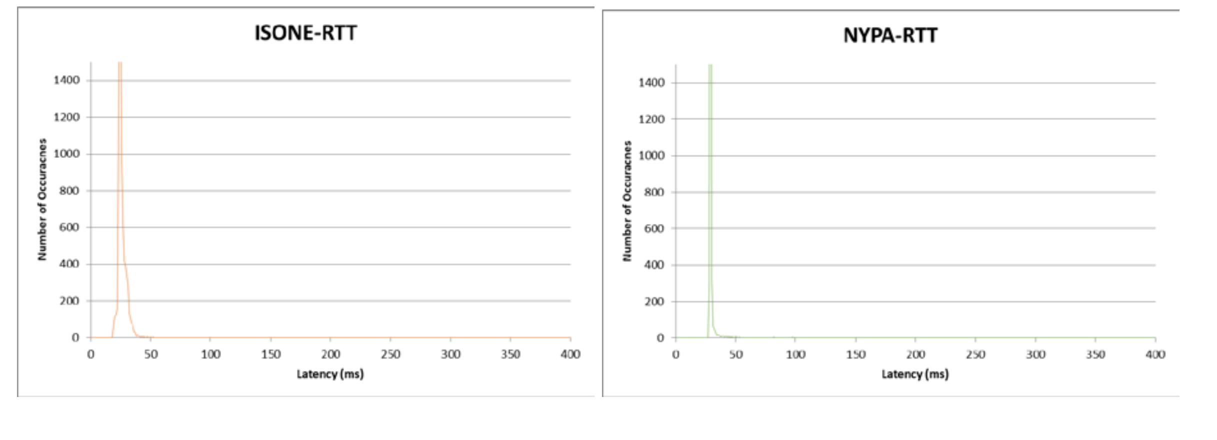

Figure 3: Measured round-trip times during the experiment.

In contemplating use of GridCloud technology, two concerns are variations in the time it takes to deliver results from one period to the next and variations in the time it takes to deliver results for the same period to different users. While we are not yet able to provide measurements for the latter, we studied the former by examining the variation in the inter-round delays during the experimental run. The results are presented in Figure 4. In these measurements, the no-jitter case appears as a single line at 200ms. As seen in the figure, we observe variation of up to 40ms from this. (Note that a measurement that is 40ms later than expected is likely followed by one that is on-time, which will appear as 160ms on this graph.)

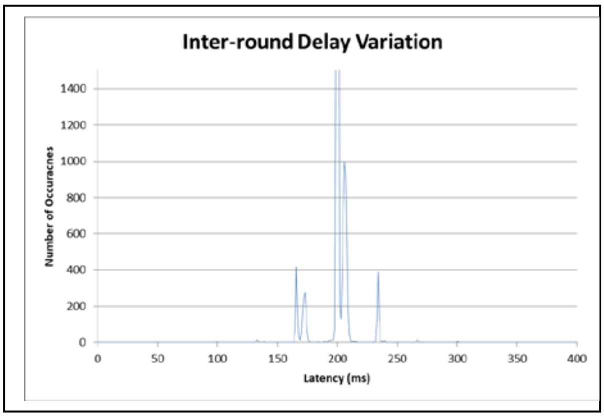

Figure 4: Inter-round delay variation.

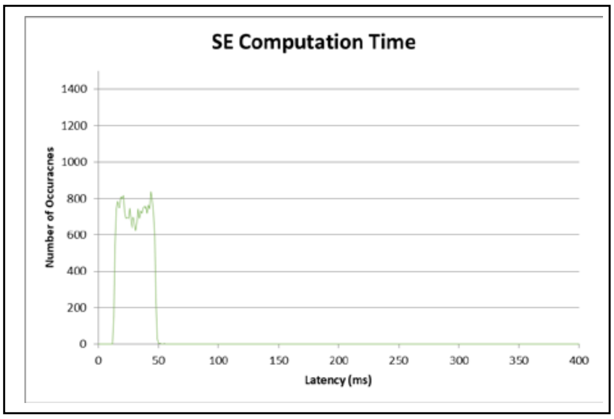

Figure 5: State estimator computation times.

The LSE code has been improved relative to the work reported previously [1] by introducing a new data alignment algorithm; this singe change reduces the latency between when data are collected and when the state estimation computation begins by nearly 200ms. In addition, reduced computational cost was achieved by more carefully managing the numerical library's use of parallelism—for this computation the overhead of setting up to use a high degree of parallelism swamps the benefits achieved by performing computation in parallel. The result of these two improvements is that latencies that were previously in excess of 250ms are, in the current experiments, less than 50ms. Furthermore, CPU utilization that was previously about 50% of an Amazon c4.xlarge instance (4 hyper-threads on a Xeon e5-2666v3) is now under 5% on that same machine, despite computing a larger set of states.[1]

Another experiment evaluates self-repair. We capture data from 32 PMU sources, replicating each stream 3 times to achieve fault-tolerance to overcome Internet performance issues. The data is captured by 8 different data collector nodes in the cloud. Each collector can handle at most 12 streams; together they can handle all 96 data streams. In Figure 6 we see time (X axis) and the level of redundancy for each PMU stream (color coded, Y axis). At T=3, we kill 2 collectors (red dashed lines). Some data streams are now inadequately replicated. CloudMake must maximize the replication factor for data from all the sources using the available resources, but without violating the peak load constraint on data collectors. Accordingly, it connects each PMU to 2 or 3 distinct data collectors, then balances load (T=3.. T=5). At T=24, the faulty collectors recover (blue dashed lines) and CloudMake can rebalance again (second black dashed line). This takes a bit longer: 5

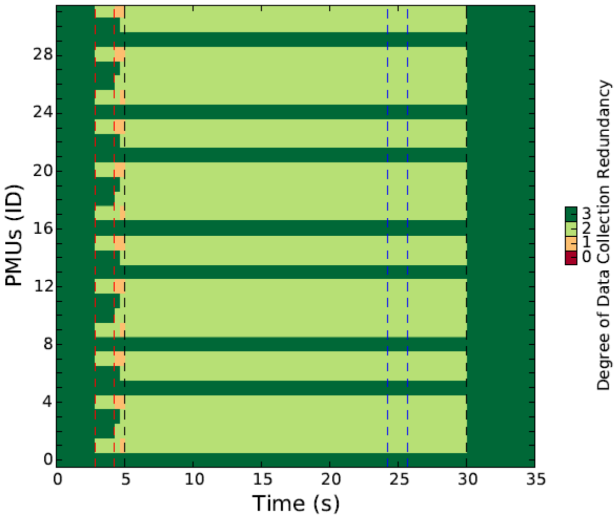

Figure 6: Self-Repair Experiment.

[1] Note that if comparing Figure 5 with Figure 5 in [1], the roundtrip latency from Figure 3 needs to be added. (The security setup required in the current experiments did not allow direct measurement of round-trip latency for state estimates as was reported in Figure 5 of [1].)

seconds (the recovery triggers an initial static analysis by CloudMake, followed by a change to the CloudManager group membership in a library called Vsync, and finally setup of the restarted collectors occurs). We should note that the experiment used to create Figure 6 mimics the AWS configuration, but did not run in AWS: we used 8 m1.small virtual machine instances (1 CPU 2.9 GHz, 4GB RAM, ~1Gb/s NIC), but ran them on a dedicated experimental cluster. Data flows were as in our AWS experiments, but originated on a single workstation computer.

New Data Analytics Capability

In support of the data analytics objectives described earlier, we also extended GridCloud to integrate with a widely used standard for big-data analytics: the Spark/Databricks infrastructure. Originally created at UC Berkeley as a research project that sought to speed up the MapReduce/Hadoop framework by caching data in-memory [16], Spark has grown to become a popular solution for modern cloud data analytics. Spark is now offered commercially under the name Databricks (https://databricks.com). Databricks hosts a wide variety of applications implemented in languages like Tensor Flow: machine learning tools for building models, training them, and then using them to classify new data or to make decisions; visualization products, etc.

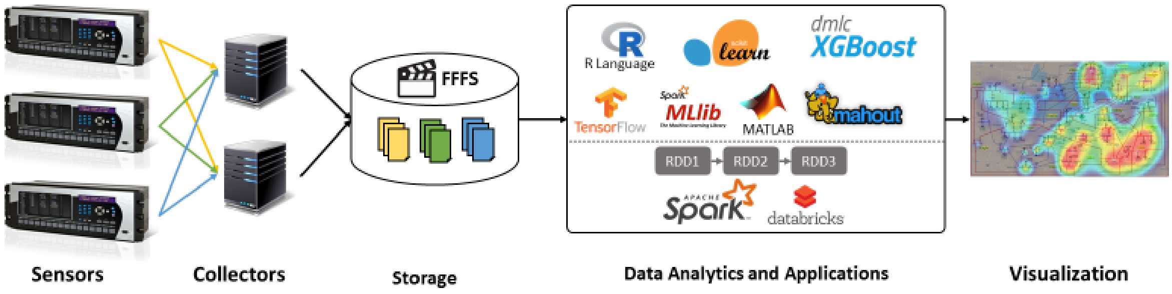

Figure 7: GridCloud "pipeline" from data collection (left) to Spark/Databricks (middle).

The extensions to GridCloud are illustrated in the right-most boxes of Figure 7. We tap into the GridCloud data collection infrastructure's data archive. We created software to transform our data archive files (which use IEEE C37.118 data format) into the Resilient Distributed Data (RDD) format employed by Spark/Databricks. RDDs can represent a wide range of objects including raw data, tensors and other forms of tabular data. Further, Spark offers a direct way to express transformations and computations directly on RDDs, evaluating the resulting expressions on demand and caching the results for efficient reuse. Spark also permits programmers to treat RDDs like variables or files, making it easy to develop new programs that treat them like other program variables.

As suggested by Figure 7, this configuration gives us access to a wide range of data analytics solutions. These would include any of the mature technologies employed by today's power systems operators, in standard form. But the platform would also facilitate creation of completely new analytic tools, using powerful packages that are widely popular in today's "big data" analytics communities. These include the R language, several major big data packages, the "Tensor Flow" analytics language, Matlab, the M-Lib language, Hadoop, etc. A number of machine-learning tools are available for a wide range of scenarios. Spark also includes powerful visualization tools.

One practical problem that arises in our setting is that PMU data has an important real-time dimension. Although Spark has a file system with a real-time "snapshot" capability, we evaluated it and found that this (and other such features in other systems) tend to "blend" data from distinct times, creating a noisy mashup. Accordingly, we created a new GridCloud file system, FFFS (seen in the storage "box"): it manages files, but indexes them by time, accurate to the millisecond, and using time extracted from the PMU format. Prior time alignment is not required, in keeping with GridCloud's general philosophy of pushing time alignment as close to applications as possible and using only as much time alignment as necessary. To retrieve time-aligned data, it suffices to read from an FFFS "file system snapshot." FFFS can provide such snapshots efficiently, rapidly, and with high temporal precision and strong causal consistency guarantees [12]. We then connect this to Spark RDDs.

To demonstrate this new capability, we selected a challenge that is of interest in the power community: estimating the state of a bus that is not directly instrumented with a PMU, under conditions in which the SCADA and LSE systems lack adequate observability. For example, during a major storm, connectivity to system state sensors can be so compromised that only a few PMUs are accessible. Prior research has suggested that in such a scenario, a machine learning tool might be able to step in when the standard options are no longer viable [4][6][8].

For this experiment, we first replay synthetic bulk power grid data drawn from a public data set available from the Overbye Research Group at University of Illinois. (We plan to repeat using actual data from ISO-NE and NYPA in the future.) The full data set consists 30Hz simulated PMU output for a 42-bus network, with 17 high-voltage buses, and a further 25 low-voltage buses. We trained a collection of 42 convolutional neural networks (one per bus, using a minimal absolute percentage error (MAPE) criteria), employing half of the data for training. Then we reran our neural networks, classifying a stream of incoming data, but using just the 17 high-voltage buses as input. We replayed the original training data first, and then experimented with completely new scenarios. We could then graph the "loss" function (the accuracy of the 42 models), and could visualize the resulting output. Note that our CNN was trained to predict voltages, not full bus states (other CNN technologies would be able to learn full bus states, in which case total vector error (TVE) would be a more suitable training criteria).

In Figure 8 we see the results of this experiment. On the left is data associated with training the CNN. The Y axis shows the mean average percentage loss (MAPE) for the learned model as a function of the number of training iterations (X axis). The red data shows the MAPE for the training data (we trained on this data, then reran the CNN on the same data); blue shows a different portion of the data file, with the MAPE computed by taking the difference between the predicted results and the truth. The MAPE is expected to decline for the training set, then eventually rise again as the model becomes over-trained, and this is precisely what we observe. 40 iterations gave the best MAPE. On the right, we use the CNN to estimate network state and visualize it; the image shows a single frame from the resulting 30s movie. The colors denote under-voltage (blue) and over-voltage (red) during a simulated tornado that causes extensive damage, tripping multiple lines.

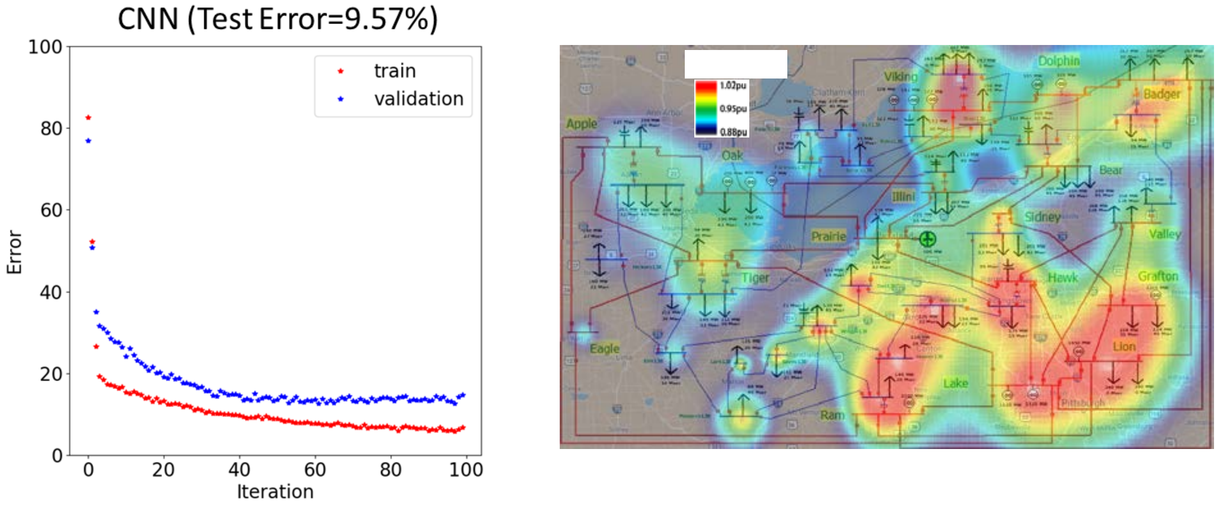

Figure 8: (Left) Model accuracy for a convolutional neural network as a function of training iterations. (Right): Voltage visualization shortly during a "severe store" scenario.

Limitations of Machine Learning

Although our experiment yielded an encouraging outcome (the convolutional neural network is able to estimate individual network bus states with mean accuracy of less than 10% on training data after 40 iterations of the training process, and its visualization of system state during the extreme storm scenario is in fact fairly accurate), it is important not to overgeneralize from this single example. Examples of open questions include the following:

- Would an LSE or SCADA solution yield an accurate state estimate for this case? How does a state estimate of this kind compare with a more classical SCADA or LSE approach?
- The 42 neural network models are treated here (and in prior work) as independent, hence even if they produce accurate predictions of bus state for the 25 uninstrumented buses, there is no certainty that the "joint" state corresponding to that set of 25 predictions is a possible joint state for the network model. Is there a way to link the models to ensure that their predictions are as accurate as possible, but also mutually consistent?
- There are many ways to configure a neural network, and many neural network training packages. For our example, we employed a set of CNNs, one per bus, but we could have trained one CNN to predict the full vector of network states. Further, our CNN only predicted voltage, whereas network bus states are actually complex numbers. The question then arises of what the most appropriate fit of neural network technologies to power scenarios should be. Indeed, a meta-question must also be asked: how can we rigorously answer this question of "best" neural network technology? We noted earlier that Total Vector Error (TVE) is the usual metric of quality for a state estimator output, but the larger questions here go far beyond this issue of error metric.

An additional set of questions reflect the evolving technical options for large-scale grid management:

- Can we develop a rigorous science that would quantify the precise conditions under which it might be appropriate to use methods such as these? When is it safe to trust the output?
- What is the proper notion of "security" for a shared data infrastructure that tracks wide-area system state? Do today's rigorous security definitions for smaller regions apply directly in wide-area contexts, or is there some risk that doing so might incorrectly classify an insecure wide-area state? One reason that this arises as a new and interesting question is that in existing systems, security is always discussed with relation to a single control center that owns the regional network and makes all the relevant decisions. At the scale of a wide-area, multi-regional network, we must anticipate that multiple control centers share data yet make independent control decisions. Thus, observation of a single state might still trigger inconsistent control actions. Further, there may be cases in which every regional network appears to be "locally secure" and yet the wide-area network state is actually insecure because of conditions only evident at large scale.

Despite the many open questions, we also see this demonstration as an exciting hint of things to come. By making a wide variety of powerful data analytic tools available to the bulk power community, GridCloud opens the door to a huge range of new options for system management and control, exploratory data analysis, forensic studies, and eventually, computer-assisted wide-area bulk power sharing and management with improved efficiency.

Conclusion and Future Work

GridCloud demonstrates that the smart grid (and other complex real-time applications with strong security and consistency needs) can be supported on commonly available 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. In work still underway, we plan to bring additional data from NY-ISO into the platform. At that point, our data set should be sufficiently rich to diagnose and detect problems with individual PMUs and to tackle some of the more ambitious data analytic opportunities enabled by the integration with Spark/Databricks.

BIBLIOGRAPHY

[1] D. Anderson, T. Gkountouvas, M. Meng, K. Birman, A. Bose, C. Hauser, E. Litvinov. X. Luo, and F. Zhang. “GridCloud: Infrastructure for cloud-based wide area monitoring of bulk electric power grids” (IEEE Transactions on Smart Grid 2018 to appear)

[2] J. Baek, Q. Vu, J. Liu, X. Huang, and Y. Xiang. “A secure cloud computing based framework for big data information management of smart grid.” (IEEE Transactions on Cloud Computing 3:2 2015 pages 233-244)

[3] K. Birman, L. Ganesh, and R. van Renesse. “Running smart grid control software on cloud computing architectures” (Workshop on Computational Needs for the Next Generation Electric Grid. Cornell University, Ithaca, NY, April 19-20 2011)

[4] M. Biserica, Y. Besanger, R. Caire, O. Chilard, and P. Deschamps. “Neural networks to improve distribution state estimation—Volt VAR control performances” (IEEE Transactions on Smart Grid 3:3 2012 pages 1137-1144)

[5] A. Bose. “Smart transmission grid applications and their supporting infrastructure” (IEEE Transactions on Smart Grid 1:1 2010 pages 11-19)

[6] M. Khan, M. Li, P. Ashton, G. Taylor, and J. Liu. “Big data analytics on PMU measurements” (11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD) 2014 pages 715-719)

[7] E. Litvinov, X. Luo, Q. Zhang, K. Birman, T. Gkountouvas, D. Anderson, C. Hauser, and A. Bose. “A cloud-hosted synchrophasor data sharing platform” (Power System Grid Operation Using Synchrophasor Technology, Springer-Verlag, 2017)

[8] A. A. Majeed, C. Kattmann, S. Tenbohlen, and R. Saur. “Usage of artificial neural networks for pseudo measurement modeling in low voltage distribution systems” (IEEE PES General Meeting| Conference & Exposition 2014 pages1-5)

[9] E. Manitsas, R. Singh, B. C. Pal, and G. Strbac. “Distribution system state estimation using an artificial neural network approach for pseudo measurement modelling” (IEEE Transactions on Power Systems 27:4 2012 pages 1888-1896)

[10] North American Synchrophasor Initiative (NASPI), 2008. https://www.naspi.org/

[11] A. Onwuachumba, M. Musavi. “Eigenvalue-based optimal placement of PMUs in large power systems” (North American Power Symposium 2015 pages 1-6)

[12] W. Song, T. Gkountouvas, Q. Chen, Z. Xiao, and K. Birman. “The Freeze-Frame File System” (ACM Symposium on Operating Systems Principles, Santa Clara, CA, October 2016)

[13] S. Uludag, K.S. Lui, W. Ren, and K. Nahrstedt. “Secure and scalable data collection with time minimization in the smart grid” (IEEE Transactions on Smart Grid 7:1 2016 pages 43-54)

[14] Y. Wang, Y. Pradeep, and A. Bose. “Decentralized communication and control systems for power system operation” (IEEE Transactions on Smart Grid 6:2 2015 pages 885-893)

[15] T. Yang, S. Hongbin, and A. Bose. “Two-level PMU-based linear state estimator” (Power Systems Conference and Exposition 15-18 March 2009 pages 1-6)

[16] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, I. Stoica. “Spark: Cluster Computing with Working Sets.” (2nd USENIX Conference on Hot Topics in Cloud Computing. Boston, MA. June 22-25, 2010)