Cloud Data Sharing Platform

Final Project Report

S-67G

Power Systems Engineering Research Center

Empowering Minds to Engineer
the Future Electric Energy System

Cloud Data Sharing Platform (S-67G)

Final Project Report

Project Team

Carl Hauser, Project Leader
Anjan Bose
Ming Meng, Graduate Student
Dave Anderson, Graduate Student
Washington State University

Ken Birman
Theodoros Gkountouvas, Graduate Student
Weijia Song, Graduate Student
Cornell University

PSERC Publication 16-05

April 2016

For information about this project, contact

Carl Hauser, Project Leader
Associate Professor
School of Electrical Engineering and Computer Science
Washington State University
P.O. Box 642752
Pullman, WA 99164-2752
Phone: (509) 335-6470
Email: hauser@eecs.wsu.edu

Power Systems Engineering Research Center

The Power Systems Engineering Research Center (PSERC) is a multi-university Center conducting research on challenges facing the electric power industry and educating the next generation of power engineers. More information about PSERC can be found at the Center's website: http://www.pserc.org.

For additional information, contact:

Power Systems Engineering Research Center
Arizona State University
551 E. Tyler Mall
Engineering Research Center #527
Tempe, Arizona 85287-5706
Phone: 480-965-1643
Fax: 480-965-0745

Notice Concerning Copyright Material

Acknowledgements

This is the final report for the Power Systems Engineering Research Center (PSERC) research project titled "Cloud Data Sharing Platform" (project S-67G). We express our appreciation for the support provided by PSERC's industry members and by the National Science Foundation under the Industry / University Cooperative Research Center program.

We wish to specifically thank Eugene Litvinov and ISO New England for their sponsorship of this project, and to thank Eugene, Xiaochuan Luo, Qiang Zhang, and Feng Ma at ISO NE for their technical support and guidance throughout the project. Thanks also to industry advisors Jay Giri of Alstom Grid and George Stefopoulos and Atena Darvishi of NYPA.

Executive Summary

This final report details the findings of a study conducted by WSU, Cornell and ISO NE aimed at understanding the practical challenges associated with using cloud computing infrastructures to monitor PMU devices in a deployment intended to be similar in scope and scale to the Northeastern regional power pool. The core idea was to set up a cloud computing system secured in ways fully responsive to the relevant ISO security policies, and then send simulated PMU data to the cloud system over secured communication links. State estimation was then carried out in real-time, with the results relayed back to the ISO for visualization.

This study built on the GridCloud project sponsored by the U.S. Dept. of Energy, ARPA-E GENI program, in which Cornell University (Cornell) and Washington State University (WSU) developed technologies supporting real-time sharing of utility operational data (esp. PMU data streams) using cloud computing. GridCloud technologies overcome some apparent limitations of commercial cloud offerings, so that utilities can take advantage of the cost savings that cloud computing offers for dynamic computing load conditions.

In addition to the careful study of security requirements, options, and their degree of match, this project quantified latency, from when PMU data was captured to when the corresponding state estimate became available, and compared latency and data quality when the identical system was deployed in duplicate with one instance running on an Amazon data center on the US Northeast and the second running remotely in the Pacific Northwest.

Our main findings were as follows:

1. The ISO security team requested changes to some aspects of the GridCloud deployment as originally undertaken in the ARPAe research prior to our PSERC study. *At the ISO's request, we modified GridCloud to support a new system configuration that the ISO felt more comfortable with*. The change involved a "relay" computer owned and operated purely by the ISO, playing a firewall role: PMU data from the ISO was relayed to the GridCloud system through the relay machine, and results of the LSE computation were made available to the ISO control center through connections from visualization software running in the ISO to data relay applications on the relay computer. The ISO specified that all data sent to or from the cloud should be encrypted and suggested either transport-level security (TLS) or HTTPS; we ultimately employed tunnels over the Secure Shell (SSH) protocol because they proved to be the most convenient. Directionality of these relay connections was important to the ISO: they wanted only connections initiated from the ISO system to the cloud, with none initiated from the cloud to components owned by the ISO.
2. In our prior ARPAe study, each PMU was separately connected to three data collectors in each data center running GridCloud. ISO NE preferred a simpler scheme in which a single connection was made from the relay machine to a single data collector within GridCloud in each datacenter, We implemented this method and carried out experiments using it, but we *concluded that this configuration is not ideal and that a*

configuration more like the earlier ARPAe approach should be favored in future experiments.

3. With two GridCloud instances running in different datacenters we found that the instances received 100% identical raw data, and gave nearly identical outputs. We observed single-point data discrepancies only in one very artificial situation, namely when we "looped" the 11s of replayed data to create a longer 30 minute run. Since the input data was in fact identical we are certain that these oddities were an artifact of the manner in which the looped reset operation was implemented and that *GridCloud should be capable of perfect fidelity between two cloud computing data centers.* When looped, the data contain a discontinuity that would not occur in a real power system. Since the output differed only twice and only at the exact point where this discontinuity occurred we believe that the problem is not fundamental. We did not have adequate time to track down our bug before funding was exhausted.

4. End-to-end latency was approximately 300ms with a nearby data center running the LSE, to which should be added the internet delay from the PMU to the data center, approximately 105ms each-way in our cross-country experiments. Thus, *the ISO was able to receive LSE outputs at 5Hz with a delay of about 300ms from a nearby data center, and with a delay of about 510ms with LSE running on a data center on the opposite coast of the United States.* As explained below, a surprisingly high portion of this delay is apparently attributable an artificial delay during data alignment in the LSE. We believe the latency could be reduced by as much as 180-200ms with more time and effort since the linear state estimation task itself is not computationally expensive for this amount of data.

5. The relay machine was found to add approximately 8-15ms of latency to the LSE computation, with an additional 2ms when using SSH for the required security. The extra delays for security are extremely modest given that the overall end-to-end latency was 300ms even with a nearby Amazon data center in Virginia.

6. During our experiments, Cornell did some network upgrades which caused high loss rates and high delays on November 16 (see Appendix B). The GridCloud experiment was not impacted by these problems, confirming that the system is highly fault tolerant. We note that in prior ARPAe studies with redundant TCP connections, three from each PMU to each data collector, we never saw situations in which all three connections simultaneously exhibited high latency.

7. Although we did not experiment with node crashes during the ISO study (due to a lack of time) we did experiment with complete data center shutdowns. *Restart required approximately 175s, during which no data was lost because the Oregon data center continued to operate while the Virginia one was recovering.* The vast majority of this time was covered by booting the Windows State Estimation instance. Then, the delays of reconfiguration and initial setup of the SE instance are non-significant compared to 175s. Thus at the end of the 175s period, the GridCloud system was back to full function (state estimation included) and full redundancy. In ARPAe experiments we also explored cases where individual compute nodes were crashed; in all such situations, the built-in CloudMake manager restarted the failed component within a similar delay.

8. *Microbenchmarks revealed that the cost of AES 256 encryption is in the noise.* We experiment with and without encryption both on network links from the ISO to the cloud and for network model and SE outputs stored to disk by the GridCloud data historian and in both cases, the cost of encryption was so small as to be completely negligible.
9. The ISO requested that we extend the platform to report the LSE output as a series of IEEE C17.118 data streams for subscription and visualization within the ISO data center. This extension was easily accomplished, confirming that the GridCloud system as a whole is quite easily modified and extended with new functionality.
10. The ISO requested that when using Amazon storage (S3) encryption, we take steps to ensure that the encryption key would never be stored anywhere in the Amazon infrastructure. We were not able to provide this guarantee: *Amazon's normal setup does retain the given key, in a special-purpose secure key repository maintained by the company.* We recommend that ISO NE negotiate with the Amazon government cloud organization to have this feature added, if ISO NE security officers feel that it is a strong requirement.
11. The work reveals a series of next steps, which include bringing a second user into the system (expected to be NYPA under funding from NYSERDA), modifying the data path from the PMU owner to the cloud so as to reduce a form of "fate sharing" that was found to be a performance and security limitation, and starting to use the data historian capabilities of the system.

We conclude that the experiment was highly successful and that the ISO NE security requirements can be satisfied at negligible overhead. The ISO's original goal was to carry out real-time LSE with a maximum delay of 2-5s; we achieved far better performance with latency as low as 300ms when the data center is reasonably near the PMU data sources. Data consistency was perfect, the system was confirmed to be highly fault-tolerant, and is easily extended.

One limitation concerned the Amazon secure key management model, and was discussed above. A second limitation arose from the ISO's request to limit the relay computer to using a single connection between the relay machine and the cloud infrastructure. This created an undesirable form of "fate sharing": if this one TCP connection suffers a dropped data segment, the entire flow of data from the full set of ISO-managed PMUs is uniformly and negatively impacted until TCP's retransmission mechanism has recovered the dropped segment. In our prior ARPAe experiment we employed multiple side by side connections, three per PMU, and found this to be a more performant configuration. We recommend that in future work, the ISO shift to this more stable, more robust configuration option.

Project Publications:

[1] Anderson, Dave; Dave Bakken, Ken Birman, Anjan Bose, Carl Hauser, Theo Gkountouvas, Robbert van Renesse, and Weija Song. "Hosting the SmartGrid: The GridCloud Platform for Monitoring and Controlling the Bulk Power Grid." Submitted to *IEEE Power and Energy*. November 2015.

[2] Anderson, Dave; Dave Bakken, Ken Birman, David Bindel, Anjan Bose, Alan Ettlinger, Theo Gkountouvas, Carl Hauser, Eugen Litvinov, Edgardo Luzcando, Xiaochuan Luo, Weija Song, Robbert van Renesee, Frankie Zhang. “Enabling a Smarter Bulk Power Grid.” To be submitted to *Communications of the ACM*, March 2016.

Student Theses:

[1] Meng, Ming. “Two-Level Three Phase State Estimation.” Ph.D. dissertation, Washington State University. Expected August, 2016.

[2] Gkountouvas, Theodoros. “Dynamic Management of Distributed Systems.” Ph.D. dissertation. Cornell University. Expected May 2018.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Background

In the GridCloud project sponsored by the U.S. Dept. of Energy, ARPA-E GENI program, Cornell University (Cornell) and Washington State University (WSU) developed technologies supporting real-time sharing of utility operational data (esp. PMU data streams) using cloud computing. GridCloud technologies overcome some apparent limitations of commercial cloud offerings, so that utilities can take advantage of the cost savings that cloud computing offers for dynamic computing load conditions. In this project we are adapting the existing GridCloud technology to the current reality of the northeastern US grid, focusing on appropriate scaling, use of multiple data centers, and basic cybersecurity, in order to enhance the technology readiness level of the platform to enable scenarios such as depicted in Figure 1, where data are delivered to computations taking place in the compute cloud and the results are delivered to the operating grid.

1.2 Overview of the Problem

During our ARPAe effort, the main goal in GridCloud project was to evaluate extreme scaling of the platform. To this end, we demonstrated a WSU-developed linear state estimator running in the cloud, and receiving 4,632 concurrent PMU data streams, each sent in triplicate (so in fact 13,896 data flows were generated). Our system computed and reported state estimate results at 3 solutions per second. The emphasis in the current project represents a shift to focus on cybersecurity and flexible data delivery as well as performance and cost evaluation at a smaller scale reflective of current PMU deployments. The system is delivering incoming PMU data streams to applications running in the cloud (state estimation and historian), and delivering both incoming and computed result data streams back to the control center. We have quantified latencies and costs as well as incremental latencies associated with cryptographic cyber security mechanisms, and have assessed the impact on latency of operating the cloud data sharing platform in two different Amazon EC2 data centers.

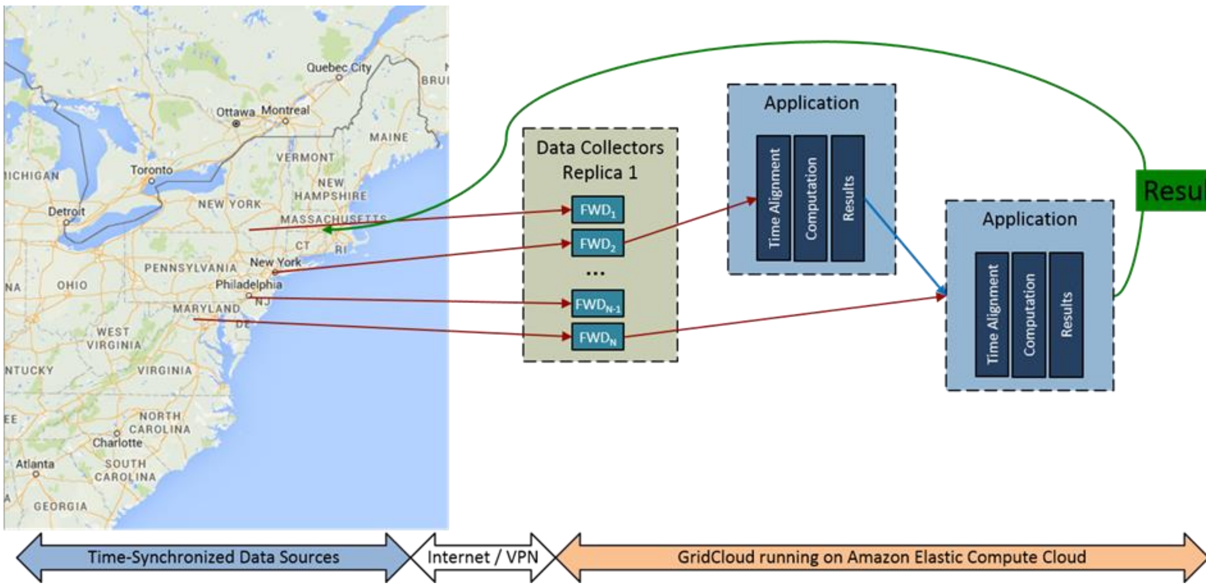


Figure 1: Overall System Concept

1.2.1 Main Issues

The detailed relationships between the components leading up to the linear state estimator (for the demonstration) are shown in Figure 2. In the demonstration, 73 30 Hz PMU data are delivered over a secure connection to a collection and distribution process running in the cloud, which further delivers the data to multiple cloud-hosted applications. In our experiments the applications are a hierarchical linear state estimator, itself made up of many processes, and a data archiving process (historian).

The configuration of Figure 2 allows evaluation of basic factors such as the latency cost associated with using Amazon Virtual Private Cloud (VPC) vs Amazon Elastic Compute Cloud (EC2), encryption vs no encryption on the ground-to-cloud link and encryption vs no encryption for data archiving.

As suggested in Figure 1, the design depicted in Figure 2 supports instantiating additional applications that receive streaming data from the FWD nodes in this picture, and when we are finished, also receiving data from the output of the control center SE.

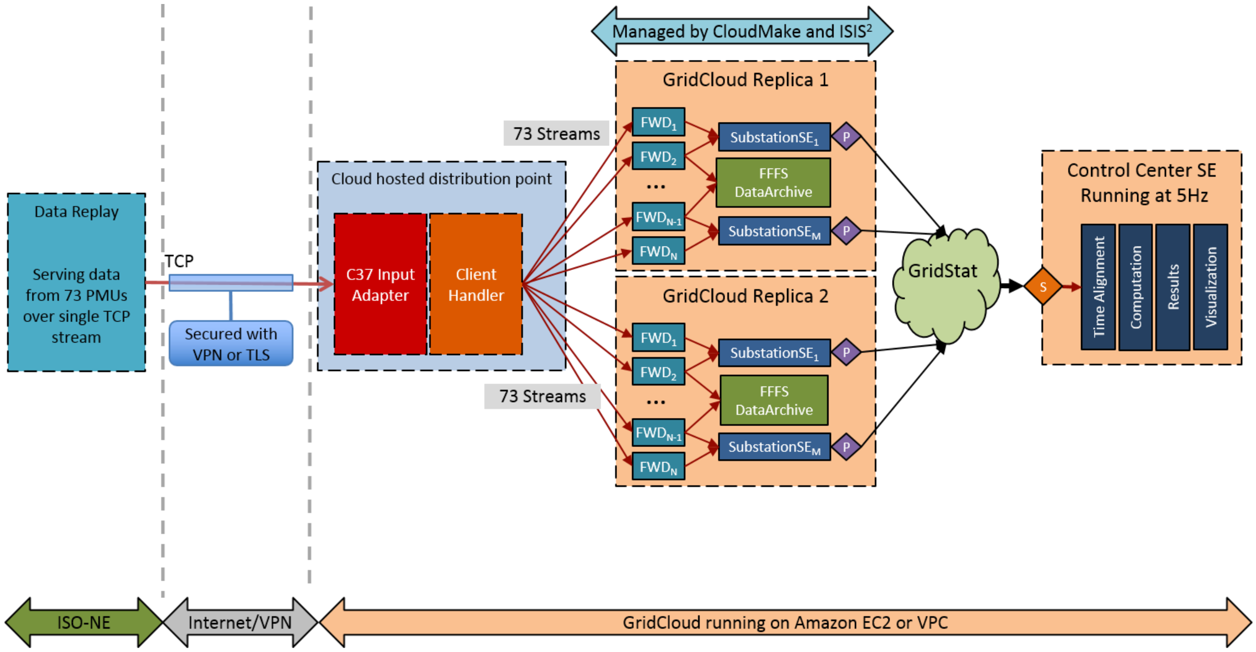


Figure 2: Detailed Components of the System (Single Data Center)

ISO NE asked us to set up and evaluate a more complicated, but more resilient, configuration as seen in Figure 3. It replicates the *entire system*. Thus whereas the diagram above showed one data center, the scenario below involves running the identical system on two Amazon AWS data centers, for geographic fault-tolerance.

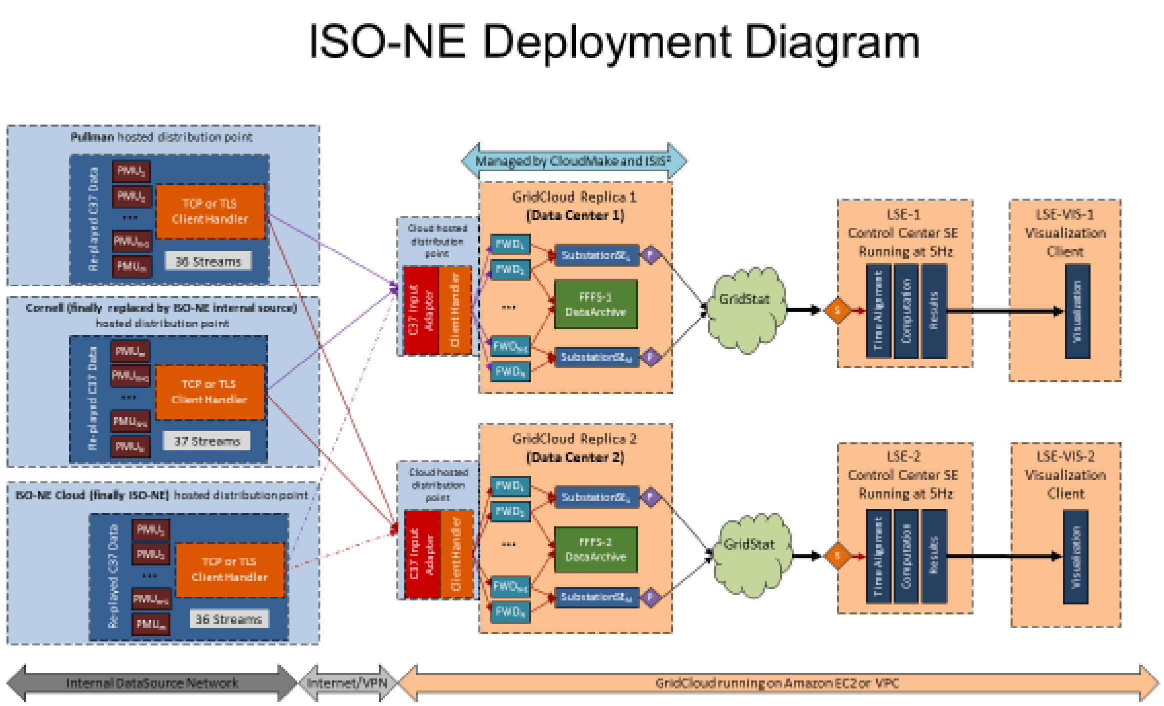


Figure 3: Evaluated System Using Multiple Data Sources and AWS Data Centers

1.2.2 Secondary Issues

Initially, Washington State University completed deployment and measurement of the GridCloud platform on Amazon EC2 using ISO-NE's system configuration with data sources at Pullman and Ithaca. This also incorporated the linear state estimator running in the EC2. (To avoid certain cybersecurity concerns, the PMU data are actually recordings that are played back at the same rate that the actual PMU would have sent them, rather than data from PMUs made at the current time.) We developed techniques for latency measurements in the absence of good clock synchronization between the data sources and the machines in EC2, and we measured latency and jitter to use as a baseline for comparison against what was achieved in the next step when we added encryption to the system. Ming Meng separated the state-computation part of the state estimator from the graphic visualization part in preparation for making states computed by the SE available for use by other applications; and a design choice was made to make these states available in C37.118 format, so that the states could be published to a network stream and visualized using the openPDC PMU Connection Tester.

At Cornell, Theo Gkountouvas enabled the cloud management infrastructure's (CloudMake) ability to configure secured TCP connections from the SE outputs back to the ISO NE visualization software running in the ISO NE control center.

The second Cornell activity included both Theo Gkountouvas and Weijia Song who designed and created the archiving data storage system and historian, which we call the Freeze Frame File System. FFFS has exceptional real-time accuracy, security and also guarantees a property called logical snapshot consistency. Here the main effort has focused on planning: Theo's data collection tool already creates an archive responsive to the requirements of the PSERC effort. But as we look beyond the end of this project, ISO NE is keen to develop a variety of "forensic analysis" tools for PMU data sets, and Weijia will be working with the ISO NE team to ensure that FFFS is ready and able to play the needed roles.

Turning to ISO-NE: the technical team at ISO-NE produced several use cases for the cloud data sharing platform that were used in their discussions with the ISO's security team about security requirements for the platform. Guidance was provided to WSU and Cornell about security requirements from the security team's perspective. The ISO team was also active in discussions about ways to make use of both stored and real-time streaming data in the platform and gained approval to host a test data source inside the ISO-NE firewall.

We finished the experimental phase of the project in mid-December, 2015, providing the ISO NE technical team with a demonstration of the functional, security, and fault-tolerance capabilities of the platform in their operations along with experimental evaluation of performance and analysis of design choices regarding security and fault tolerance. In January and February 2016 we have made presentations to the ISO-NE team and to the ISO's security team to show what has been accomplished and start gaining their insights about what it means for the next stage of experimentation and development involving actual data sharing between entities in the northeastern grid.

2. Detailed Approach

Our project began just as WSU and Cornell University completed a three-year ARPAe-funded project that yielded GridCloud, a cloud-hosted system designed to capture PMU data, archive the data into a historical repository, carry out state estimation using the WSU linear state estimator (LSE), and display the results using a PMU data visualization tool. The GridCloud system consists of a data collection layer, software to relay PMU data between system components (based on the WSU-developed GridStat pub-sub infrastructure), a stripped-down version of OpenPDC used as a PMU registry, a new system called CloudMake that performs 24x7 system monitoring and availability management, and a version of the WSU LSE ported to run in a cloud environment. The ARPAe validation experiments focused on data from the IEEE WEC model, which was simulated and then replayed using high-fidelity clock-synchronized data servers which passed the data over Internet links to the GridCloud platform hosted on an Amazon data center. The ARPAe experiment used approximately 6000 simulated PMUs representing a national scale deployment.

In this project, additional funding was provided by ISO New England through PSERC to reconfigure GridCloud in order to mock up a scenario of specific interest to the ISO. In this scenario, the network model was based on an ISO NE planning model (one considered to be safe for sharing with students under non-disclosure agreements), and a simulation of the network state was carried out for 11 seconds and then looped to create a steady state in which 73 PMU data streams transmitted to the GridCloud system at 30Hz. We configured the GridCloud platform to perform LSE using this data, and modified the LSE system to report bus-by-bus state as a set of IEEE C37.118 PMU feeds. Half the data was sent to GridCloud from ISO NE and the other half from Cornell, to mimic a situation in which two or more ISOs might collaborate to share data via the cloud platform.

The overall experimental design is seen below in Figure 4. We have a network deployed in the US NE with a collection of PMU units relaying data at 30Hz in IEEE C17.118 format. This data is captured by a set of data collectors, with a single collector per PMU (here we depart from the prior ARPAe experiment, which relayed data in triplicate and hence had three data collector banks side-by-side in the same data center, offering redundancy and increasing tolerance of real-time disruptions). The data are then processed by application software (two abstracted applications are shown, corresponding roughly to an LSE in the middle and a visualizer on the right), and results are also relayed back to the ISO in the form of additional C37.118 feeds, now representing the reconstructed bus states after LSE converges.

Figure 4 illustrates a dual-data-center setup on which our experiments focused, with one data center (DCA) on top and the second (DCB) below. In our experiments, DCA was near ISO NE on the US Northeast, but DCB was located in Amazon's Pacific Northwest region, 2500 miles away. As noted, 31 of the 73 PMU data streams originated at ISO NE and 42 originated at Cornell.

The figure also depicts the paths over which round-trip latencies were measured: ground-to-cloud (labelled L3raw – does not include SE time), full SE roundtrip (labelled L3se), and SE roundtrip internal to the cloud (labelled L2). (A one-way path from the data source to the cloud application that we called L1 was measured early in the project but since it was impossible to adequately synchronize clocks between the two ends of the path those preliminary results are not reported.).

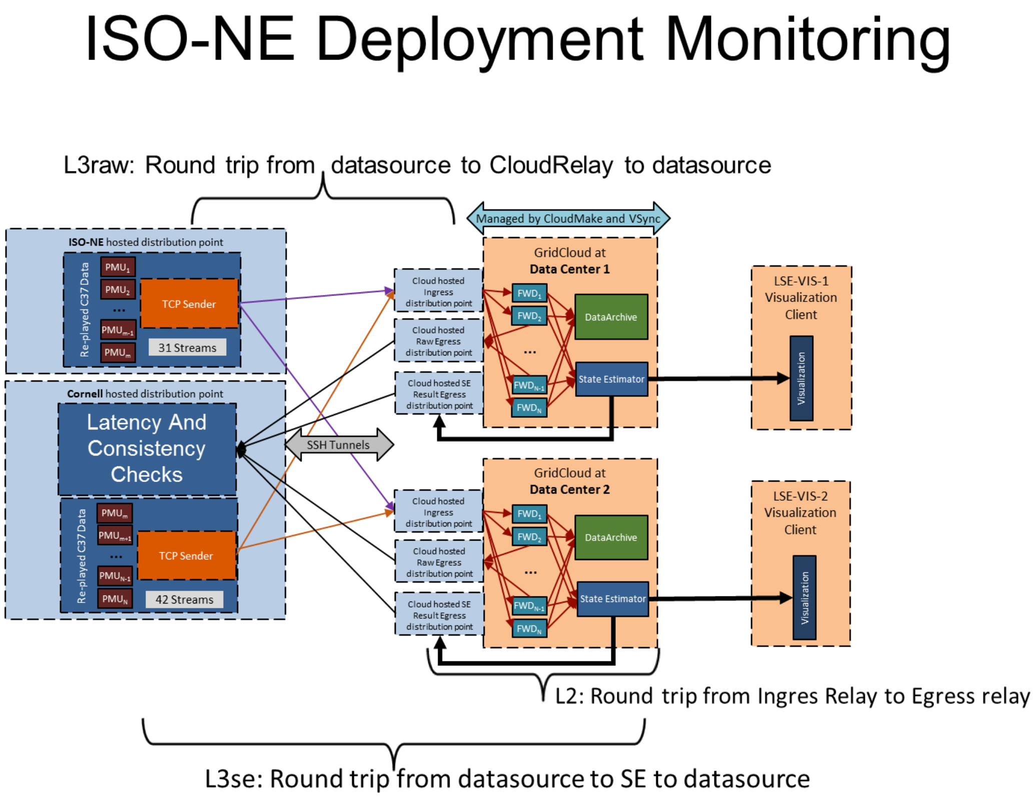


Figure 4: Measurement of 3 Latencies: Internal, Ground-to-Cloud, Publication-to-Receipt-of-State-Estimate

We extensively measured L2, L3raw and L3se latency figures over a period of 4 hours. Our tests were performed using datasource machines at Cornell and ISO NE over SSH tunnels.

We sampled 4 raw feeds and two SE feeds from each datacenter:

- Lowest numbered PMU from each datasource (ISO-NE and Cornell)
- Highest numbered PMU from each datasource
- PMUs send to the cloud in order from the datasource, this helps show us the spread of data from first to last measurement sent per round
- Lowest and Highest SE result

The resulting data is presented in the following slides as histograms and table of overall statistics

- Histograms only cover highest numbered PMU/SE as they have the highest variability
- Full stats and analysis included in companion excel file

Here we see that round-trip latencies averaged 65ms from ISO NE to the Virginia Amazon DC, but rose to 130ms when using the Oregon DC (left graph, blue and red). Cornell numbers were similar but more stable on November 15 when the middle graph was produced. On the right we see the LSE output for the full 73-stream dataset, with Virginia giving about 350ms total latency and Oregon, 425ms.

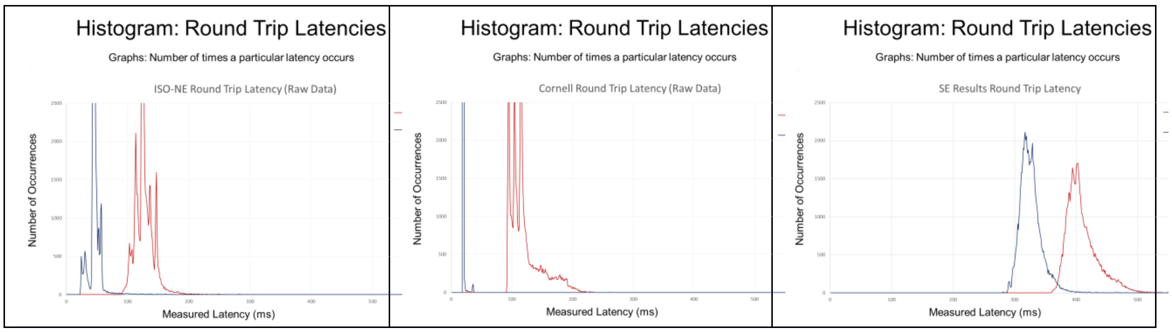


Figure 5: Latency Histograms

3. Discussion of Experimental Findings

We first measured performance without any application costs at all and obtained "EC2 Latency" for a pure data bus moving data around but with no computation occurring at all.

- Average = 245ms
- 1st Percentile = 211ms
- 99th Percentile = 255ms

The ISO wished to know how much Amazon "virtual private cloud" adds to the EC2 Latency:

- Average = 261ms
- 1st Percentile = 228ms
- 99th Percentile = 270ms

… thus the measured delta is approximately +15ms. Adding SSH tunnels for the ingress links added a RTT of less than 2ms. But notice that 245ms of latency is thus incurred even with no real application running at all. This strikes us as high and seems to be arising from some sort of scheduling delay in the data relaying logic we use internal to GridCloud. We believe it can be reduced by 180ms-200ms with additional effort, but did not have time or funding to carry out the needed careful investigation. On paper, though, we see no reason that this particular relaying task should take longer than 45ms.

We also measured the dollar price for Amazon instances as configured for testing:

- 13 instances total per datacenter (Vizualizer, CloudRelay, CloudMakeLeader, StateEstimator, 3xRawArchiver, 4xSEArchiver, 2xForwader)
- $2.47/hr to run per datacenter

4. Conclusions and Future work

We do not wish to burden the reader with a recap of the same summary given in the introduction, which listed the following main findings:

1. At the request of ISO NE, we modified the GridCloud security configuration, and conducted experiments with the new model.
2. We discovered that although most ISO NE changes had no impact, one configuration change was problematic, namely the request that all PMU data "tunnel" through a single shared HTTPS connection to each cloud. This was found to cause problematic delays and also creates an unexpected security concern, namely that an intruder who compromises the HTTPS link would see all the data. In future work we should revert to one connection per data flow (per PMU, PDC, etc).
3. By running in two data centers we gained a high quality of backup redundancy.
4. End-to-end latency was low enough for use by human operators.
5. The relay machine deployed by the ISO did not impose significant delays.
6. The system tolerated even an unplanned network disruption event.
7. Data center shutdowns followed by restart were successful, with full backup redundancy restored within 5 minutes.
8. Encryption overheads were negligible.
9. The ISO NE requests for supporting new data formats were easily accommodated.
10. ISO NE posed one concrete question about AWS key management that seems to require direct dialog with Amazon AWS; the question ("does Amazon ever store user-supplied keys on any form of persistent medium?") was not something our team could answer, and because we were using an unsupported free AWS research account, we do not have access to Amazon AWS senior personnel who could do so. But we see no obvious reason that Amazon itself couldn't answer the ISO's question.

Accordingly, we deem the project to have been highly successful. With respect to next steps:

1. We should bring a second user into the system (expected to be NYPA under funding from NYSERDA). This does not require any obvious changes, but dialog with the two security organizations will be required.
2. We should modify the data path from the PMU owner to the cloud so as to reduce a form of "fate sharing" that was found to be a performance and security limitation (see point 2). This should be a simple task.
3. We should begin to use the data historian capabilities of the system, and to experiment with analytics that compare current system state with past states.
4. We should begin to explore applications beyond continuous online state estimation.

The work done in this project mainly addressed technical feasibility of using cloud computing while meeting ISO NE's policy requirements for security during the experiments. As the additional technical work described above progresses we should also be investigating how the need to meet NERC CIP standards in an eventual actual deployment will affect some of the technical choices.

Appendix 1: Overall Timing Data

Table 1 Breaks down these numbers and gives full details:

Table 1: Latency Measurement Statistics

	Virginia	Virginia-Internal	Oregon	Oregon-Internal
	L3 Raw		L3 Raw	
ISONE Raw-Low Min	20		88	
ISONE Raw-Low 1st Percentile	22		89	
ISONE Raw-Low Average	**25**		**102**	
ISONE Raw-Low 99th Percentile	58		152	
ISONE Raw-Low Max	611		696	
ISONE Raw-High Min	22		90	
ISONE Raw-High 1st Percentile	25		99	
ISONE Raw-High Average	**46**		**127**	
ISONE Raw-High 99th Percentile	82		179	
ISONE Raw-High Max	612		697	
Cornell Raw-Low Min	17		90	
Cornell Raw-Low 1st Percentile	17		91	
Cornell Raw-Low Average	**18**		**115**	
Cornell Raw-Low 99th Percentile	20		191	
Cornell Raw-Low Max	49		407	
Cornell Raw-High Min	18		91	
Cornell Raw-High 1st Percentile	18		92	
Cornell Raw-High Average	**19**		**120**	
Cornell Raw-High 99th Percentile	20		199	
Cornell Raw-High Max	49		413	
	L3 SE	L2	L3 SE	L2
SE Results Min	279	242	351	240
SE Results 1st Percentile	294	267	370	273
SE Results Average	**325**	**300**	**409**	**317**
SE Results 99th Percentile	384	348	490	393
SE Results Max	911	469	962	642

Output can also be visualized using the OpenPDC Manager (Visualizer) as in Figure 6.

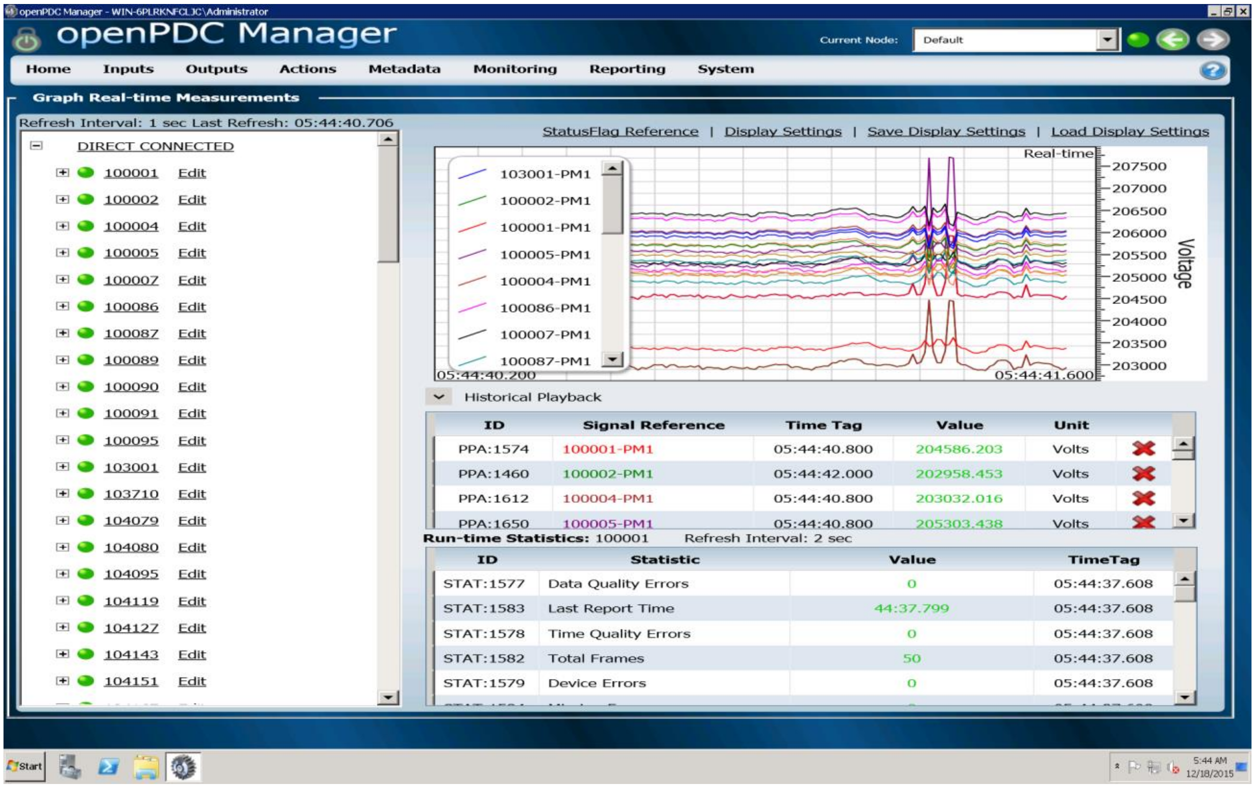
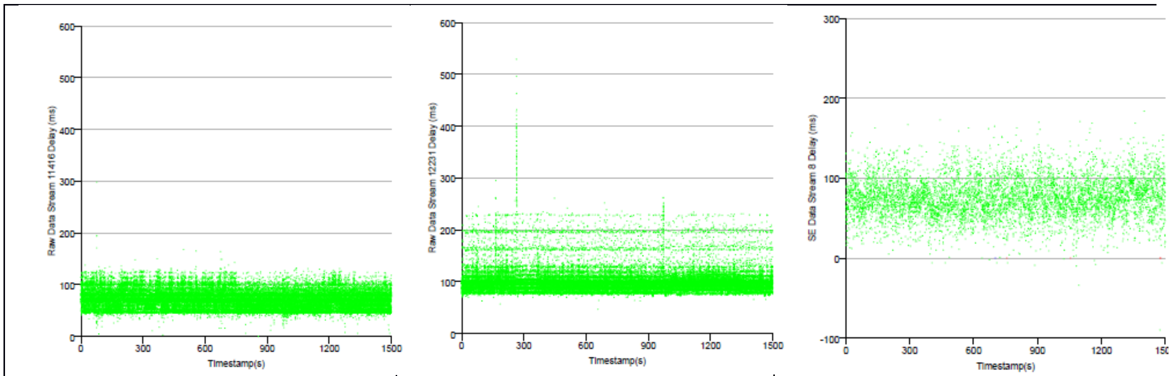


Figure 6: Visualization of PMU Data in openPDC Manager

Appendix 2: Consistency and round-trip latency data



DCA-DCB for raw ISO NE data	DCA-DCB for Cornell (Nov 16)	DCA-DCB LSE output

Figure 7: Recorded Latency Differences (scatterplots) with Consistency Indicators

The ISO also requested that we measure the cost of data encryption in our data historian. We obtained the following figures:

Write Data (30 minutes)

- Encryption: 204:499s
- Non-Encryption : 216:509s

Read Data (30 minutes)

- Encryption: 64:524s
- Non-Encryption: 64:632s

As we can see, the cost of encryption is small enough that other randomness in the EC2 performance dwarfs it and in fact the encrypted runs are actually faster than the non-encrypted runs. Of course this is not a meaningful speedup; the real point is that the cloud computing systems are somewhat unpredictable (they are shared, and might not have identical hardware setups) and so different runs can be slightly faster or slower. So we conclude only that encryption is very cheap.

Finally, we report statistics from our latency and consistency checks. For this experiment we ran the “LCC” test defined in our attached spreadsheet for 30 minutes.

Raw Data: here we compared raw data in to the two data centers and found 100% match.

LSE output: here we actually saw slight inconsistencies:

- Same data from both DCs: 99.925%
- Different data: 0.005%
- Missing DC-A data: 0.013%

- Missing DC-B data: 0.057%

We investigated and concluded that the inconsistency occurs because there was some sort of flaw in the “reset” mechanism used to loop the 21s of PMU data into a 30 minute test. As noted earlier, our approach was to replay 21s of data again and again, almost 90 times in total for 30m of data. But because the PMU data resets, the LSE sees a discontinuity once every 21s. This isn’t a natural step and as a result, we sometimes saw slight differences in the LSE output for the first PMU data points of each 21s interval. There wasn’t adequate funding to track down the exact cause of this LSE reset problem, but we concluded that it was minor.

Appendix 3: Detailed Experimental definitions

Background	
We created a program called the latency and consistency check (LCC) program for SE output. It compares the outputs of the two SEs (5 outputs per second) for a particular bus, and reports whether each output is present and whether the two SE outputs are the same. The LCC is launched N times to analyze SE on N buses. The LCCs, running at Cornell, WSU, or ISO-NE connect to a data connector node running in the cloud in order to receive data produced by the SE processes.	A key goal is to measure latency and consistency when GridCloud runs on two data centers: DC-A in the Northeast, near ISO NE, and DC-B in the PNY Oregon region. We estimated that round-trip delay to send PMU data to DC-B and receive SE output back would be about 85-100ms higher than for DC-A. Beyond this, we wanted to detect if data loss occurred due to timeouts if the SE cycles (this occurs every 200ms) but is missing data. In such cases, no SE output is produced. (Our LSE is tolerant of missing data but we decided to configure it to not produce SE output for this experiment unless all PMU inputs are received before the deadline expires). Our LCC program runs once per SE output (per bus), so there will be N copies running in an experiment with N estimated bus states. Each LCC writes a csv file with (1) the clock time in ms, which for time in seconds Ts will be Ts.000, Ts.200, Ts.400, Ts.600, and Ts.800), (2) a color code: -1=orange, 0=red, 1=green, 2=blue as discussed below, (3) a delta in ms: suppose that at time Ta the LCC received a particular SE input from DC-A and at time Tb, from DC-B. Then V is 0 if either input was missing, and if both inputs were received V is Tb-Ta. The color is -1 (orange) if both inputs were received but the value differed. The color is 0 if no input for this time was received from DC-A, but an input was received from DC-B. The color is 1 if either both sent no input (here V=0), or if both sent identical inputs (here, V=Tb-Ta). The color is 2 if an input was received from DC-B but none was received for that time from DC-A. We also evaluated latencies for raw data as well as the SE outputs. The raw data have 30 samples per second instead of 5 per second for the SE outputs.

Latency Tests

#	Test Name	Preconditions	Test Step Description	Duration
1a	Single Data Center Internal State Estimation Latencies	Data sources running at WSU & ISO-NE each producing 1/2 of the PMU data streams All GridCloud components (cloud connectors,	Collect latency measurements between cloud ingress point and SE completion	30 minutes

Latency Tests				
#	Test Name	Preconditions	Test Step Description	Duration
		forwarders, collectors, SE) running in DC-A Data receiver running in same machine at ISO NE as the data source program		
1b	Single Data Center Round-trip Raw Data Latencies	Data sources running at WSU & ISO-NE each producing 1/2 of the PMU data streams All GridCloud components (cloud connectors, forwarders, collectors, SE) running in DC-A Data receiver running in same machine at ISO NE as the data source program	Collect latency measurements between transmission time of a measurement by the data source, represented by the PMU timestamp, and receipt of that PMU measurement back at the data receiver	30 minutes
1c	Single Data Center Round-trip SE Results Latencies	Data sources running at WSU & ISO-NE each producing 1/2 of the PMU data streams All GridCloud components (cloud connectors, forwarders, collectors, SE) running in DC-A Data receiver running in same machine at ISO NE as the data source program	Collect latency measurements between transmission time of a measurement (as in #1b) and receipt of the computed SE data back at the data receiver	30 minutes

Latency Tests				
#	**Test Name**	**Preconditions**	**Test Step Description**	**Duration**
2a	Differential Raw Data Latencies for Two Data Centers	While experiment is active	We run LCC at Cornell or inside ISO NE, then generate a graph of the color-coded delays as documented at the top. The X axis shows GPS time for the PMUs. The Y axis is numbered 1..N for the N buses in the model, and shows a positive or negative histogram bar colored per the scheme described above and with amplitude (+/-) giving the V value as defined earlier. The lines on the Y axis will need to be separated enough to allow us to avoid "overlap" between the different parallel lines. We can also "focus in" on specific buses if desired, e.g. graphing just a subset of the LCC outputs. This experiment is for raw PMU data streams delivered via both data centers.	30 minutes
2b	Differential SE Results Latencies for Two Data Centers	While experiment is active	We run LCC at Cornell or inside ISO NE, then generate a graph of the color-coded delays as documented at the top. The X axis shows GPS time for the PMUs. The Y axis is numbered 1..N for the N buses in the model, and shows a positive or negative histogram bar colored per the scheme described above and with amplitude (+/-) giving the V value as defined earlier. The lines on the Y axis will need to be separated enough to allow us to avoid "overlap" between the different parallel lines. We can also "focus in" on specific buses if desired, e.g. graphing just a subset of the LCC outputs. This is experiment compares latencies for SE results computed and delivered via both data centers.	30 minutes

Latency Tests				
#	Test Name	Preconditions	Test Step Description	Duration
3a	Data Archiving Time w/o encryption	Aggregated while running experiment	Measure total time spent writing history archive files for raw PMU data and SE outputs, without AWS encryption.	30 minutes
3b	Data Archiving Time with encryption	Aggregated while running experiment	Measure total time spent writing history archive files for raw PMU data and SE outputs, with AWS encryption.	30 minutes
4a	Historical Data Retrieving Time w/o encryption	After running experiment	Measure time needed to read the full set of raw PMU files and SE files without AWS encryption	N/A
4b	Historical Data Retrieving Time with encryption	After running experiment	Measure time needed to read the full set of raw PMU files and SE files with AWS encryption	N/A

Fault Tolerance Tests				
#	Test Name	Preconditions	Test Step Description	Duration
1	Data Center Failure Test	DC-A, DC-B both running	Shut down DC-A	5 minutes
2	Data Center Recovery Test	DC-B up, DC-A down.	Restart DC-A	

Consistency Tests				
#	Test Name	Preconditions	Test Step Description	Duration
1	Raw Data Consistency	After running experiment	Compare raw PMU data files at DC-A with raw data files at DC-B	N/A
2	SE Output Consistency	After running experiment	Compare SE output data files at DC-A with SE output files at DC-B	N/A