Mission Critical Cloud Computing for Critical Infrastructures

Thoshitha Gamage[1], David Anderson[1], David Bakken[1], Kenneth Birman[2], Anjan Bose[1], Carl Hauser[1], Ketan Maheshwari[3], Robbert van Renesse[2]

[1]Washington State University, Pullman, WA. Email: tgamage@eecs.wsu.edu, daveanderson@wsu.edu, bakken@wsu.edu, bose@wsu.edu, hauser@eecs.wsu.edu
[2]Cornell University, Ithaca NY. Email: ken@cs.cornell.edu, rvr@cs.cornell.edu
[3]Argonne National Laboratory, Lemont, IL. Email: ketan@mcs.anl.gov

I. INTRODUCTION

The term "Cloud" is becoming prevalent in nearly every facet of day-to-day life, bringing up an imperative research question: How can the cloud improve future critical infrastructures? Certainly, cloud computing has already made a huge impact on the computing landscape and has permanently incorporated itself in almost all sectors of industry. The same, however, cannot be said of critical infrastructures. Most notably, the power grid, has been very cautious regarding cloud-based computing capabilities .

This is not a total surprise: the power industry is notoriously conservative about changing its operating model, and its rate commissions are generally focused on short-term goals. With thousands of moving parts, owned and operated by just as many stakeholders, even modest changes are difficult. Furthermore, continuing to operate while incorporating large paradigm shifts is neither a straightforward nor a risk-free process. In addition to industry conservatism, progress is slowed by the lack of comprehensive cloud-based solutions meeting current and future power grid application requirements. Nevertheless, there are numerous opportunities on many fronts – from bulk power generation through wide-area transmission to residential distribution, including at the micro-grid level – where cloud technologies can bolster power grid operations and improve the grid's efficiency, security, and reliability.

The impact of cloud computing is best exemplified by the recent boom in e-commerce and online shopping. The cloud has empowered modern customers with outstanding bargaining power in making their purchasing choices by providing up-to-date pricing information on products from a wide array of sources whose computing infrastructure is cost effective and scalable on demand. For example, not long ago air travelers relied on local travel agents to get the best prices on their reservations. Cloud computing has revolutionized this market, allowing vendors to easily provide customers web-based reservation services.

In fact, a recent study shows the online travel e-commerce skyrocketed from a mere $30 billion in 2002 to a staggering $103 billion, breaking the $100 billion mark for the first time in the U.S. in 2012 [1]. A similar phenomenon applies to retail shopping. Nowadays, online retail shops offer a variety of consumer electronics, clothing, books, jewelry, video games, to event tickets, digital media, and lots more at competitive prices. Mainstream online shops such as *amazon*, *ebay*, *etsy*, etc., provide customers with an unprecedented global marketplace to both buy and sell items. Almost all major U.S. retail giants, such as *walmart*, *macy's*, *bestbuy*, *target*, etc., have adopted a hybrid sales model providing online shops complementing the traditional in-store shopping experience. A more recent trend is flash sale sites (*fab*, *woot*, *deals2buy*, *totsy*, *myhabit*, etc.) which offer limited time deals and offers. All in all, retail e-commerce in the U.S. increased by as much as 15% in 2012, totaling $289 billion. To put this into perspective, the total was $72 billion 10 years earler. Such rapid growth relied heavily on cloud-based technology to provide the massive computing resources behind online shopping.

A. Cloud Computing

What truly characterizes cloud computing is its business model. The cloud provides on-demand access to virtually limitless hardware and software resources meeting the users' requirements. Furthermore, users only pay for resources they use, based on the time of use and capacity. National Institute of Standards and Technology (NIST) defines five essential cloud characteristics: on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service [2].

The computational model of the cloud features two key characteristics – *abstraction* and *virtualization*. The cloud provides its end users with well-defined APIs that support requests to a wide range of hardware and software resources. Cloud computing supports various configurations (CPU, memory, platform, I/O, networking, storage, servers) and capacities (scale) while abstracting away resource management (setup, startup, maintenance etc.), underlying infrastructure technology, physical space, and human labor requirements. The end users see only APIs when they access services on the cloud. For example, users of *dropbox*, the popular cloud-based online storage , only need to know that their stored items are accessible through the API; they do not need any knowledge of the underlying infrastructure supporting the service. Furthermore, end users relieved from owning large computing resources that are often underused. Instead, resources are housed in large data centers as a shared resource pool serving multiple users, in doing so, optimizing their use and amortizing the cost of maintenance. At the same time, end users are unaware of where their resources physically reside, effectively virtualizing the computing resources.

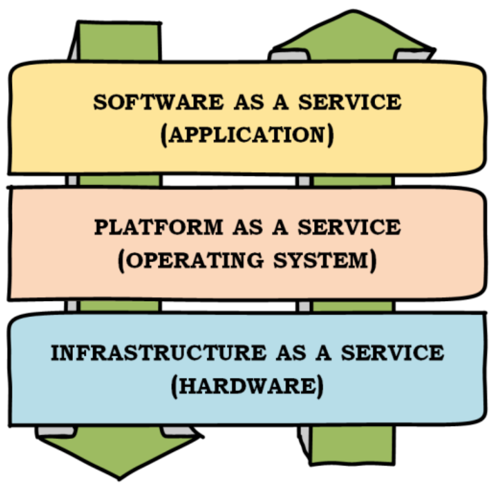


Figure 1 : Cloud Service Models as a Stack

Cloud computing provides three service models: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). Each of these service models provides unique APIs. Services can be purchased separately, but are typically purchased as a solution stack. The SaaS model offers end-point business applications which are customizable and configurable based on specific needs. One good example is the Google Apps framework that offers a large suite of end user applications (email, online storage, streaming channels, domain names, messaging, web hosting, etc.) which individuals, businesses, universities, and other organizations can purchase individually or in combination. Software offered in this manner has a shorter development life cycle resulting in frequent updates and up-to-date versions. The life cycle maintenance is explicitly handled by the service provider who offers the software on a *pay per use* basis. Since the software is hosted in the cloud, there is no explicit installation or maintenance process for the end users in their native environment. Some of the prominent SaaS providers include *Salesforce*, *Google*, *Microsoft*, *Intuit*, *Oracle*, etc.

The PaaS model offers a development environment, middleware capabilities, and a deployment stack for application developers to build tailor-made applications or host pre-purchased SaaS. Amazon Web Services (AWS), Google App Engine, and Microsoft Azure are a few examples for PaaS. In contrast to SaaS, PaaS does not abstract away development lifecycle support, given that most end users in this model are application developers. Nevertheless, the abstraction aspect of cloud computing is still present in PaaS where developers rely on underlying abstracted features such as infrastructure, operating system, backup and version control features, development and testing tools, runtime environment, workflow management, code security, and collaborative facilities.

The IaaS model offers the fundamental hardware, networking, and storage capabilities needed to host PaaS or custom user platforms. Services offered in IaaS include hardware level provisioning, public and private network connectivity, [redundant] load balancing, replication, data center space, and firewalls. IaaS relieves end users from operational and capital expenses. While the other two models also provide these features, here it is much more prominent since IaaS is the closest model to actual hardware. Moreover, since the actual hardware is virtualized in climate controlled data centers, IaaS can shield end users from eventual hardware failures, greatly increases availability and eliminating repair and maintenance costs. A popular IaaS provider, Amazon Elastic Compute Cloud (EC2), offers 9 hardware instance families in 18 types [3]. Some of the other IaaS providers include *GoGrid*, *Elastic Hosts*, *AppNexus*, and *Mosso* [4].

B. The Advanced Power Grid

Online shopping is just one of many instances where cloud computing is making its mark on society.. The power grid, in fact, is currently at an interesting crossroads in this technological space. One fundamental capability engineers are striving to improve is the grid's *situational awareness* – its real-time knowledge of grid state – through highly time-synchronized Phasor Measurement Units (PMUs), accurate Digital Fault Recorders (DFRs), Advanced Metering Infrastructure (AMI), smart meters, and significantly better communication. The industry is also facing a massive influx of ubiquitous household devices that exchange information related to energy consumption . In light of these new technologies, the traditional power grid is being transformed into what is popularly known as the *"Sm*art Grid" or the *"Advanced P*ower Grid".

The evolution of the power grid brings its own share of challenges. The newly introduced data has the potential to dramatically increase the accuracy, but only if processed quickly and correctly. True situational awareness and real-time control decisions go hand in hand. The feasibility of achieving these two objectives, however, heavily depends on three key features:

1. The ability to capture the power grid state accurately and synchronously
2. The ability to deliver grid state data timely and reliably over a (potentially) wide area
3. The ability to rapidly process large quantities of state data and redirect the resulting information to appropriate power application(s)
    a. To a lesser extent, the ability to rapidly acquire computing resources for on-demand data processing

Emerging power applications are the direct beneficiaries of rapid data capture, delivery, processing, and retrieval . One such example is the transition from conventional state estimation to direct state calculation. Beginning in the early 1960's, the power grid has been employing Supervisory Control and Data Acquisition (SCADA) technology for many of its real-time requisites such as balancing load against supply, demand response, and contingency detection and analysis. SCADA uses a slow, cyclic polling architecture where the decisions are based off unsynchronized measurements that may be several seconds old. Consequently, the estimated state lags the actual state most. Thus, state estimation gives very limited insight and visibility into the grid's actual operational status. In contrast, tightly time synchronized PMU data streams deliver data under strict Quality of Service (QoS) guarantees – low latency, high availability – allowing control centers to perform direct state calculations and measurements. The capabilities that come with the availability of status data make creating a real-time picture of the grid's operational state much more realistic.

There are also many myths surrounding the operations of power grid in conjunction with big data and its efficient use. The following is a non-exhaustive list of some of these myths:

1. Timeliness: “Real-time data” is a relative term. Often the application requirements dictate the timeliness needs. Modern software and hardware technologies provide many workarounds on the timeliness of data availability on wide area networks with average bandwidths. One of them is selective packet dropping. This technique guarantees a minimum quality of service while delivering information to recipients in a timely manner. Smart power grids will greatly benefit from these techniques.

2. Security and Safety: Security and safety are concerns often cited by decision-makers when considering new technologies. While absolute security is impossible, most concerns arising out of data security issues have been technically addressed. One large factor that affects security is human errors and oversights. Often, insufficient emphasis is given to this side of security. More and more emphasis is given to the communication channels. Securing an already secure channel only results in performance losses and overheads.

3. Cost: The cost of maintaining information infrastructures has become a major portion of budgets for large industries, and is a substantial challenge in running a sustainable, data centered architecture. Thanks to data centers and cloud computing infrastructures, these challenges are being successfully addressed. Clouds facilitate out-sourcing of large-scale computational infrastructures while achieving provably reliable quality of services.

II. Cloud Computing’s Role in the Advanced Power Grid

Cloud computing can play a vital role in improving the the advanced power grid’s vsituational awareness and the ability to derive better control decisions. As mentioned earlier, emerging power applications will leverage large amounts data in making control decisions affecting the stability and reliability of the grid. Analyzing and processing such large amounts of data requires data parallelism and massive computational capabilities well beyond general-purpose computing. Beyond data analysis, the future grid can greatly benefit from much more extensive simulation and analysis to remediate stressful situations. These are spontaneous special purpose applications (e.g., SIPS AKA RASs or SPSs) [6b[1]] each with different needs – real-time, scaling, and computational – that are triggered by grid disturbances such as frequency oscillations, voltage fluctuations, line overloads, and blackouts. Moreover, the number of power grid applications and their computational needs can only be expected to increase as the grid evolves. Managing this variety of applications and needs presents a challenge. Keeping these applications running idle on dedicated hardware until the specific condition is triggered is both inefficient and expensive.

An elegant solution is presented here which utilizes cloud computing and its rapid elasticity. Power grid applications can utilize the cloud to rapidly deploy an application-specific infrastructure using IaaS and PaaS to achieve new levels of availability and scalability. Availability and scalability are properties that are much harder to meet in a piecemeal fashion, but are inherent features of the cloud and easily adoptable. Cloud-based solutions also benefit entities at different levels of the control hierarchy, giving them the ability to perform independent, replicated analysis on same sensor data. The ability to elastically manage resources in the presence grid disturbance is extremely attractive comparison to in-house solutions, which could be over-provisioned or under-provisioned at the time of need. Another area where cloud computing performs well is in supporting the varying needs of the growing ecosystem of power applications. Both Paas and Saas will be useful for developing and supporting power applications.

PaaS for the power grid will need to encompass industry best practices, engineering standards, compliance requirements, and data privacy and security requirements as properties of the platform itself. The CAP theorem [7] argues that simultaneously achieving three key properties – consistency, availability, and partition-tolerance – is impossible in distributed systems. As a result, and especially since their apps are not mission critical, present

---

[1] PUBLISHER: Add a reference to this: S. Horowitz, D. Novosel, V. Madani, and M. Adamiak. System-wide protection, IEEE Power Energy Mag., vol. 6, no. 5, pp. 34–42, Sep. 2008.

day commercial clouds often sacrifice consistency in favor of availability. Cloud environments used for power applications must be able to guarantee high-assurance properties including consistency, fault-tolerance and real-time responsiveness in order to support the anticipated needs of power applications.

While PaaS enables power researchers and developers to expand the power application ecosystem, SaaS can abstract essential properties and requirements to provide end user application solutions. Grid incident-specific applications can be offered as SaaS , readily deployable by power utilities. The success of power grid SaaS depends heavily on the completeness and the richness of power grid PaaS. The overarching challenge lies in ensuring that power applications delivered across SaaS/PaaS models inherently carry the necessary high assurance properties. The subtle intricacies of high-assurance properties, which are often outside of the power engineering realm, will necessitate a different approach to cloud computing as well as a stronger mesh between power engineering and computer science.

A. Berkeley Grand Challenges and the Power Grid

The Berkeley view of the cloud [8] [9] outlines 10 grand challenges and opportunities for cloud computing. The following list reviews of some of these challenges and their implications on cloud-based power grid applications:

1. Ensuring High Service Availability: Consistency arguably one of the most critical requirements for cloud-based power applications [6], but availability is a close second. Many of the early adopters of cloud technology support availability as a liveness property, while smart-grid applications depend on availability as a safety property. Availability also relates to the issue of whether cloud-based power grid applications should follow stateful or stateless models. The ability to design stateful applications often depends on the availability of state information.

   Achieving high availability requires avoiding *single point of failure* scenarios and potential bottlenecks. The general consensus is that the cloud promotes and provides high availability. However, using cloud services from a single service provider allows a single point of failure [9]. Interoperability between different cloud vendors for the sake of availability for power grid applications is a farfetched ambition simply given the many proprietary and market advantages. Perhaps one solution would be for the power grid community to manage and operate its own cloud , either overlaying existing commercial clouds or as a private infrastructure with built-in replication at all levels. Such an initiative however would be dictated by the many economic drivers.

2. Eliminating Data Transfer Bottlenecks: Large amounts of high-frequency data must cross the cloud boundary to reach power applications running within the cloud. Application responsiveness is directly tied to the timelines with which data reaches its destination. The outermost layer of the cloud can have a dual role as a sensor data aggregator for sources outside the cloud and as a multiplexer toward the applications within the cloud . Thus, a sufficiently large number of replicated cloud-end points for sensor data must be provided in order to prevent a potential data transfer bottleneck.

3. Assuring Data Confidentiality and Availability: For a community historically notorious for a conservative *modus operandi*, sharing sensor data is a frightening proposition. Power grid entities operate under industry regulations and standards that can prevents data sharing in many circumstances. Additionally, companies are reluctant to share market-sensitive data that could give away economic advantage. Power application data traversing the cloud must be secured, meeting all compliance requirements so that it may not be used by unintended parties. Thus, the cloud will need to provide adequate data sanitization and filtering capabilities to protect application data.

4. Performance Predictability under Scaling: The enormous amount of data some power applications require, combined with the impacts of virtualized I/O channels and interrupts, lead to unpredictable

performance during elastic scaling. Different IaaS vendors exhibit different I/O performance, resource acquisition and release time characteristics. Computational performance on current cloud computing infrastructures also shows signs of strain under scaling [10]. High-end batch processing applications will require improved resource sharing and scheduling capabilities for virtual machines to ensure strict QoS demands.

III. A Model for Cloud-Based Power Grid Applications

Many of the cloud adoption challenges outlined in Section Cloud Computing's Role in the Advanced Power Grid are essentially about supporting highly scalable, highly assured behaviors and stringent communication guarantees. These are properties that are rarely found in today's commercial cloud infrastructures, which are optimized to support mobile applications and web-based e-commerce applications. The notions of real-time responsiveness, guaranteed consistency, data security, and fault-tolerance are significantly more forgiving in these applications than in infrastructure control , supplying little incentive for current commercial clouds to embrace the type of changes necessary to support critical infrastructure systems.

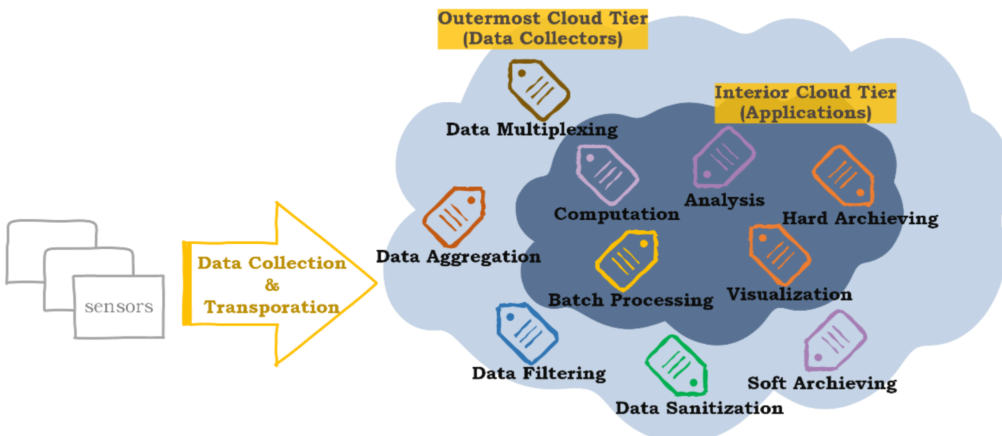


Fig 2: An Abstract Architectural Model for a Cloud-based Power Grid Application

Figure Fig visually represents an abstract architectural model for cloud-based power applications. The architecture includes three basic components:

1. A real-time data collection and transport infrastructure
2. A soft-state, elastic outer cloud tier that supports data collection, data pre-processing, temporary archiving, data sanitization and filtering, and multiplexing to services residing in interior tiers
3. Interior cloud tiers hosting services and applications, and supporting data processing, analysis, batch processing, persistent storage, and visualization functions

The data collection and transportation infrastructure sits between the physical sensors and the outermost tier of the cloud, and is the communication backbone of the overall architecture. This component is responsible for delivering data that are produced outside the cloud to the first-tier *cloud collectors* with strong QoS guarantees such as guaranteed delivery, ultra-high availability, ultra-low latency, and guaranteed latency. The soft state, outermost cloud tier provides the interface to data flowing to the applications hosted in interior tiers. The primary objective of this tier is to provide high availability, exhibit rapid elasticity,, and forward correct data to the appropriate applications. To aid in this process, this tier will also host auxiliary applications that provide data sanitization, filtering of bad data, buffering (or soft achieving), data pre-processing, and forwarding capabilities. Availability and fault-tolerance are heightened by replicated shards - nodes that collect data from a group of sensors – and by mapping sensor data sources to shards appropriately. The interior cloud tiers host the actual applications that consume data from the shards and perform analysis, computation, and batch processing tasks.

Additionally, the results of these deeper computations may be delivered at high rates to visualization applications residing inside and outside the cloud.

IV. GridCloud: A Capability Demonstration Case Study

An ARPA-E funded high-profile research collaboration between Cornell University and Washington State University is spearheading efforts to develop, prototype, and demonstrate a powerful and comprehensive software platformrealizing the cloud computing needs of the future power grid. Appropriately named ***GridCloud*** [11], this research project aims to bring together best-of-breed, already existing high assurance distributed system technologies as a basis to innovate new cloud architectural models for power systems monitoring, management, and control. The technologies integrated in this effort include ***GridStat*** [12] [13], ***Isis²*** [14] [15], ***TCP-R*** [16], and ***GridSim*** [17]. A brief description of each of these technologies is presented here:

A. GridStat

GridStat implements a data delivery overlay network framework designed from the bottom up to meet the challenging requirements of the electric power grid [CITE Chapter 4 in this book]. Power grids today are increasingly populated with high-rate, time-synchronized sensors that include Phasor Measurement Units (PMUs) and Digital Fault Recorders (DFRs), whose functionalities are actually blurring. High-rate, time-synchronized data are expected to form the basis of many monitoring and control applications with a wide variety of delivery requirements and configurations across such dimensions as geographic scope, latency, volume, and required availability [18]. These needs cannot be met by IP multicast, which forces all subscribers of a given sensor variable to get all updates at the highest rate that any subscriber requires. They also cannot be met by MPLS, which is not designed to provide per-message guarantees (only overall statistical guarantees) and also only has 3 bits (8 categories) by which to categorize the millions of different sensor flows that will likely be deployed in 5–10 years.

GridStat delivers rate-based updates of sensor variables with a wide range of QoS+ guarantees (latency, rate, availability) that include support for ultra-low latency and ultra-high availability, which are implemented by sending updates over redundant disjoint paths, each of which meets the end-to-end latency requirements for the given subscription. Additionally, GridStat enables different subscribers to a given sensor variable to require different QoS+ guarantees which can greatly reduce bandwidth requirements and improve scalability.

GridStat's data delivery plane is a flat graph of forwarding engines (FEs), each of which stores state for every subscription whose updates it forwards. FEs forward sensor updates on each outgoing link at the highest rate that any downstream subscriber requires. It drops updates that are not needed downstream, based on the expressed rate requirements of subscribers. GridStat's management plane is implemented as a hierarchy of QoS Brokers that can be mapped onto the natural hierarchy of the power grid. Each node in the hierarchy is designed to contain policies for resource permissions, security permissions, aggregation, and adaptations to anomalies. With these policies, the management plane calculates the paths required for the data delivery (with the given number of disjoint paths) and updates the forwarding tables in the FEs. Applications interact with GridStat using publisher and subscriber software libraries through which the applications' requirements for QoS are conveyed to the management plane. GridStat incorporates mechanisms for securing communication between the management plane entities and those of the data plane. Security mechanisms for end-to-end message security between publishers and subscribers are modular and configurable, allowing different data streams and applications to fulfill different security and real-time requirements [19]. GridStat in the power grid provides the opportunity to respond to different power system operating conditions with different communication configurations. GridStat provides a mechanism by which communication patterns can be rapidly changed amongst multiple pre-configured modes in response to anticipated power system contingencies.

B. Isis²

Isis² is a high-assurance replication and coordination technology that makes it easy to capture information at one location and share it in a consistent, fault-tolerant, secure manner with applications running at other locations – perhaps great numbers of them. This system revisits a powerful and widely accepted technology for replicating objects or computations, but with a new focus on running at cloud-scale, where the system might be deployed onto thousands of nodes and supporting new styles of machine learning algorithms. Isis² enables massive parallelism, strong consistency and automated fault-tolerance, and requires little sophistication on the part of its users. With Isis², all computational nodes and applications sharing the same data see it[the data?] evolve in the same manner and at nearly the same time with delays often measured in hundreds of microseconds. The system also supports replicated computation and coordination: with Isis² one could marshal 10,000 machines to jointly perform a computation, search a massive database or to simulate the consequences of control actions, all in a manner that is fast, secure against attack or intrusion, correct even if some crashes occur.

The form of assurance offered by Isis² is associated with a formal model that merges two important prior models – virtual synchrony [20] and Paxos [21]. Isis² embeds these ideas into modern object oriented programming languages. Isis² is used to create two new components for GridCloud: a version of the technology specialized for use in wide-area power systems networks, and support for high-assurance smart-grid applications that are hosted in cloud computing data centers.

The GridCloud researchers believe that Isis² can be used to support services that run on standard cloud infrastructures and yet (unlike today's cloud solutions) are able to guarantee continuous availability, automatically adapting under attack so that intruders cannot disrupt the grid even if a few nodes are compromised. They are also analyzing and demonstrating the best options for building cloud services that respond to requests in a time-critical manner.

C. TCP-R

GridCloud will tie together a very large number of components, including sensors, actuators, forwarding elements and aggregators, cloud-based services, and so on, using Internet standards. For best performance, it is important that related components communicate using persistent, stateful connections. Stateful connections reduce retransmissions, wasteful connections, and provide better flow control. The standard for stateful connections in the Internet is TCP. TCP provides network connections that provide reliable FIFO communication as well as fair flow provisioning using adaptive congestion windows.

Consider a cloud service that sends commands to a large number of actuators. The cloud service consists of a cluster of a few hundred servers. To keep actuators simple, and also to allow flexibility in evolving the cloud service, the cloud service should appear to the actuators as a single endpoint with a single TCP/IP address. While an actuator will receive commands from a particular server machine in the cluster, it appears to the actuators (and their software) as if the cloud service is a single, highly reliable and fast machine. It is desirable to maintain this illusion even when connections migrate between server machines for load balancing, for hardware or software upgrades, or when rebooting cloud servers. TCP connections, unfortunately, are between socket endpoints, and using current operating systems abstractions socket endpoints cannot migrate or survive process failures. Also, the cloud service would have to maintain a TCP socket for every actuator. This does not scale well as each TCP socket involves storing a lot of state information.

Replacing TCP with a radically different protocol would not be feasible today: Operating systems and even networking hardware implement TCP connections very efficiently. TCP is the dominant communication protocol on the Internet, and Internet routers have evolved to support TCP efficiently, easily scaling to many millions of simultaneous TCP connections. TCP-R proposes to support standard TCP connections, but extend them with a technology that addresses the shortcomings mentioned above. The essential idea is to extend the cloud service with a filter that intercepts and pre-processes TCP packets. The filter is scalable and maintains little state per

TCP connection (on the order of 32 bytes). It has only soft state (that is, it does not have to store its state persistently across crashes, much simplifying fault tolerance). The filter allows servers to migrate TCP connections, and TCP connections to survive server failure and recovery. Originally developed to maintain TCP connections between BGP (internet routing) servers across failures and subsequent recovery [22], TCP-R is extended into a scalable technology for a cluster serving client endpoints and also to “park” [pause?] connections that are not currently alive.

D. GridSim

GridSim is a real-time end-to-end power grid simulation package that in unique in its integration of a real-time simulator, data delivery infrastructure, and multiple applications all running in real-time. The goal of this project is to simulate power grid operation, control and communications at grid-wide scale (e.g. the western interconnect), as well as provide utilities a way to explore new equipment deployments and predict reactions to contingencies. The ability to simulate operation under different equipment deployment configurations includes large-scale configurations of phasor measurement units (PMUs). With the objective of simulating real world equipment usage, and usage in conjunction with readily available industry equipment, the GridSim simulation package uses the industry standard C37.118 data format for all streaming measurement data.

the first element in the GridSim platform is a transient power stability simulator, specially modified to output streaming data in real-time [“composed of” implies a list of things. Fix]. The output data is encoded into C37.118 and sent to a huge number of substation processes. At each of these processes, the data is aggregated as would be done in a real power utility substation. The data is also sent to any of the substation level power applications that are running. Both the raw substation data as well as any power application outputs are then published to GridStat.

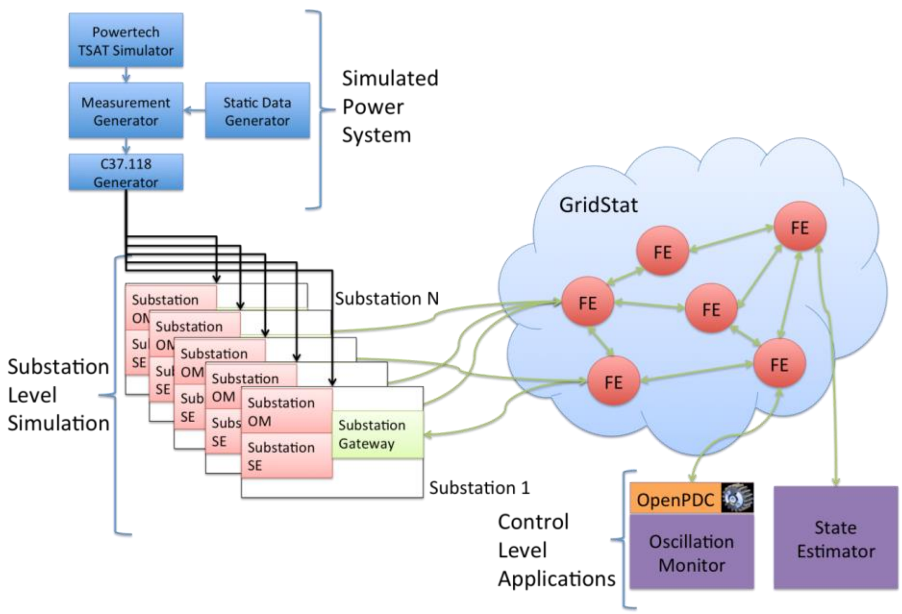
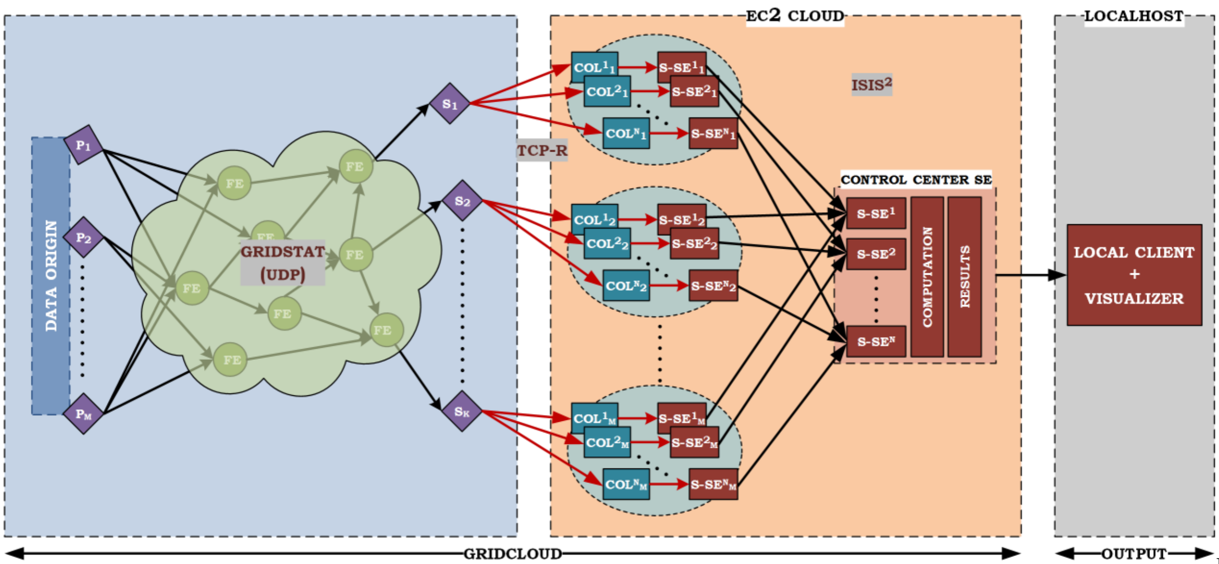


Fig : The GridSim Architecture [17]

GridStat allows the substation data to be distributed as it would in the real world. Published data can be sent via redundant paths, 1→many communication (publish-subscribe, whose degenerate version is network-level multicast), etc. The flexibility provided by the GridStat data delivery middleware allows subscription applications to be easily integrated into the system with minimal reconfiguration. Published data is available to any subscribers of GridStat, including the two applications included in the GridSim simulation, the Hierarchical State Estimator and the Oscillation and Damping Monitor.

E. GridCloud Architecture

GridCloud was designed with the expectation the developers of the advanced power grid will require easy access to large computing resources. Ttasks may require large-scale computation, or may involve such large amounts of data that simply hosting and accessing the data will pose a substantial scalability challenge. This leads to believe that cloud computing will play an important role in the future grid, supplementing the roles played by existing data center architectures. The compelling economics of cloud computing, the ease of creating “apps’’ that might control household power consumption[not a subject that's been mentioned yet], and the remarkable scalability of the cloud all support this conclusion.



Fi

g : GridCloud Architecture

Figure Fig shows the architecture of GridCloud. The representative application used in this case is a Hierarchical State Estimator (HSE) [17] [23] [24]. Data sources represent PMUs which stream data to data collectors across a wide-area using GridStat. The HSE comprises several substation level state estimators that aggregate, filter, and process PMU data before forwarding it to a control center level state estimator. The input and the first level computation are inherently sharded at substation granularity. Furthermore, computations are inherently parallel between substations. Thus, the HSE has a natural mapping in GridCloud with substation state estimators residing in the outermost tier of the cloud while the control center state estimator is moved to the interior tier. The substation state estimators are replicated to increase fault-tolerance and availability. Consistency of the replicas is managed through $Isis^2$. TCP-R is used to provide fail-over capabilities for connections across the cloud.

V. Conclusions

This chapter presents a roadmap of how the cloud computing can be used to support hc computational needs of the advanced power grid. Today’s commercial cloud computing infrastructure lacks the essential properties required by power grid applications. These deficiencies are explained and   a cloud-based power grid application architecture is presented with overcomes these difficulties using well-known distributed system constructs. Furthermore, the GridSim project is described which instantiates this model is presented as case study example.

VI. Bibliography

[1] G. Fulgoni, A. Lipsman and I. Essling, "State of the U.S. Online Retail Economy in Q4 2012," February 2013.

[Online]. Available: http://goo.gl/BCnbn.

[2] P. Mell and T. Grance, "The NIST Definition of Cloud Computing: Recommendations of the National Institute of Standards and Technology," The National Institute of Standards and Technology, Gaithersburg, MD, 2011.

[3] A. W. Services, "Amazon Elastic Compute Cloud: User Guide API ver. 2013-02-01," 01 02 2013. [Online]. Available: http://awsdocs.s3.amazonaws.com/EC2/latest/ec2-ug.pdf. [Accessed 03 06 2013].

[4] R. Prodan and S. Ostermann, "A Survey and Taxonomy of Infrastructure as a Service and Web Hosting Cloud Providers," in *10th IEEE/ACM International Conference on Grid Computing*, Banff, AB , 2009.

[5] D. Novosel, V. Madani, B. Bhargava, K. Vu and J. Cole, "Dawn of the Grid Synchronization," *IEEE Power & Energy Magazine,* vol. 6, no. 1, pp. 49-60, 2008.

[6] K. Birman, L. Ganesh and R. V. Renesse, "Running Smart Grid Control Software on Cloud Computing Architectures," in *Workshop on Computational Needs for the Next Generation Electric Grid*, Cornell, 2011.

[7] S. Gilbert and N. Lynch, "Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-tolerant Web Services," *ACM SIGACT News,* vol. 33, no. 2, pp. 51-59, 2002.

[8] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica and M. Zaharia, "A View of Cloud Computing," *Communications of the ACM,* vol. 53, no. 4, pp. 50-58, 2010.

[9] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing," Depatment of Electrical Engineering and Computer Sciences, University of California Berkeley, Report UCB/EECS 28, 2009.

[10] A. Iosup, S. Ostermann, M. Yigitbasi, R. Prodan, T. Fahringer and D. H. J. Epema, "Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing," *IEEE Transactions on Parallel and Distributed Systems,* vol. 22, no. 6, pp. 931-945, June 2011.

[11] K. Birman, A. Bose, D. Bakken and C. Hauser, "GridControl: A Software Platform to Support the Smart Grid," A Cornell University and Washington State University Research Collaboration, 2011. [Online]. Available: http://www.cs.cornell.edu/Projects/gridcontrol/index.html#gridcloud. [Accessed 03 June 2014].

[12] H. Gjermundrod, H. Gjermundrod, D. Bakken, C. Hauser and A. Bose, "GridStat: A Flexible QoS-Managed Data Dissemination Framework for the Power Grid," *IEEE Transactions on Power Delivery,* vol. 24, no. 1, pp. 136-143, 2009.

[13] C. Hauser, D. Bakken and A. Bose, "A Failure to Communicate: Next Generation Communication Requirements, Technologies, and Architecture for the Electric Power Grid," *IEEE Power and Energy Magazine,* vol. 3, no. 2, pp. 47-55, March-April 2005.

[14] K. P. Birman, D. A. Freedman, Q. Huang and P. Dowell, "Overcoming CAP with Consistent Soft-State Replication," *IEEE Computer,* vol. 45, no. 2, pp. 50-58, February 2012.

[15] K. Birman, "Isis2 Cloud Computing Library," 2013. [Online]. Available: http://isis2.codeplex.com/.

[16] A. Agapi, K. Birman, R. M. Broberg, C. Cotton, T. Kielmann, M. Millnert, R. Payne, R. Surton and R. van Renesse, "Routers for the Cloud: Can the Internet Achieve 5-Nines Availability," *IEEE Internet Computing,* pp. 72-77, September-October 2011.

[17] D. Anderson, C. Zhao, C. Hauser, V. Venkatasubramanian, D. Bakken and A. Bose, "A Virtual Smart Grid: Real-Time Simulation for Smart Grid Control and Communications Design," *IEEE Power and Energy Magazine,* vol. 10, no. 1, pp. 49-57, January-February 2012.

[18] D. E. Bakken, A. Bose, C. H. Hauser, D. E. Whitehead and G. C. Zweigle, "Smart Generation and Transmission with Coherent, Real-time Data," *Proceedings of the IEEE,* vol. 99, no. 6, pp. 928-951, 2011.

[19] E. Solum, C. Hauser, R. Chakravarthy and D. Bakken, "Modular Over-the-Wire Configurable Security for Long-Lived Critical Infrastructure Monitoring Systems," in *In Proceedings of the Third ACM International Conference on Distributed Event-Based Systems (DEBS)*, Nashville, TN, 2009.

[20] K. Birman, Reliable Distributed Systems: Technologies, Web Services, and Applications, Springer, 2005.

[21] L. Lamport, "Paxos Made Simple," *ACM SIGACT News,* vol. 32, no. 4, pp. 18-25, 2001.

[22] R. Surton, K. Birman, R. Broberg, T. Marian and R. v. Renesse, "Man-in-the-Middle TCP Recovery," Cornell University, [Online]. Available: http://goo.gl/9S4Jz. [Accessed 03 June 2013].

[23] W. Jiang, V. Vittal and G. Heydt, "A Distributed State Estimator Utilizing Synchronized Phasor Measurements," *IEEE Transactions on Power Systems,* vol. 22, no. 2, pp. 563-571, 2007.

[24] M. Zhao and A. Abur, "Multi Area State Estimation using Synchronized Phasor Measurements," *IEEE Transactions on Power Systems,* vol. 20, no. 2, pp. 611-617, 2005.