Running Smart Grid Control Software on Cloud Computing Architectures

Kenneth P. Birman, Lakshmi Ganesh, and Robbert van Renesse[1]

Abstract

There are pressing economic as well as environmental arguments for the overhaul of the current outdated power grid, and its replacement with a Smart Grid that integrates new kinds of green power generating systems, monitors power use, and adapts consumption to match power costs and system load. This paper identifies some of the computing needs for building this smart grid, and examines the current computing infrastructure to see whether it can address these needs. Under the assumption that the power community is not in a position to develop its own Internet or create its own computing platforms from scratch, and hence must work with generally accepted standards and commercially successful hardware and software platforms, we then ask to what extent these existing options can be used to address the requirements of the smart grid. Our conclusions should come as a wakeup call: many promising power management ideas demand scalability of a kind that only cloud computing can offer, but also have additional requirements (real-time, consistency, privacy, security, etc.) that cloud computing would not currently support. Some of these gaps will not soon be filled by the cloud industry, for reasons stemming from underlying economic drivers that have shaped the industry and will continue to do so. On the other hand, we don't see this as a looming catastrophe: a focused federal research program could create the needed scalability solutions and then work with the cloud computing industry to transition the needed technologies into standard cloud settings. We'll argue that once these steps are taken, the solutions should be sufficiently monetized to endure as long-term options because they are also of high likely value in other settings such as cloud-based health-care, financial systems, and for other critical computing infrastructure purposes.

1. Introduction: The Evolving Power Grid

The evolution of the power grid has been compared, unfavorably, with the evolution of modern telephony; while Edison, one of the architects of the former, would recognize most components of the current grid, Bell, the inventor of the latter, would find telephony unrecognizably advanced since his time [40]. It is not surprising, then, that the power grid is under immense pressure today from inability to scale to current demands, and is growing increasingly fragile, even as the repercussions of power outages grow ever more serious. Upgrading to a smarter grid has escalated from being a desirable vision, to an urgent imperative. Clearly, the computing industry will have a key role to play in enabling the smart grid, and our goal in this paper is to evaluate its readiness, in its current state, for supporting this vision.

[1] Department of Computer Science, Cornell University, Ithaca NY 14853. This work was supported by a grant from Lawrence Berkeley Laboratories. Emails {ken,lakshmi,rvr}@cs.cornell.edu.

EXECUTIVE SUMMARY

HIGH ASSURANCE CLOUD COMPUTING REQUIREMENTS OF THE FUTURE SMART GRID

- **Support for scalable real-time services.** A *real-time service* will meet its timing requirements even if some limited number of node (server) failures occurs. Today's cloud systems do support services that require rapid responses, but their response time can be disrupted by transient Internet congestion events, or even a single server failure.

- **Support for scalable, consistency guaranteed, fault-tolerant services.** The term *consistency* covers a range of cloud-hosted services that support database ACID guarantees, state machine replication behavior, virtual synchrony, or other strong, formally specified consistency models, up to some limited number of server failures. At the extreme of this spectrum one finds Byzantine Fault Tolerance services, which can even tolerate compromise (e.g. by a virus) of some service members. Today's cloud computing systems often "embrace inconsistency" [31][37], making it hard to implement a scalable consistency-preserving service.

- **Protection of Private Data.** Current cloud platforms do such a poor job of protecting private data that most cloud companies must remind their employees to "not be evil". Needed are protective mechanisms strong enough so that cloud systems could be entrusted with sensitive data, even when competing power producers or consumers share a single cloud data center.

- **Highly Assured Internet Routing.** In today's Internet, consumers often experience brief periods of loss of connectivity. However, research is underway on mechanisms for providing secured multipath Internet routes from points of access to cloud services. Duplicated, highly available routes will enable critical components of the future smart grid to maintain connectivity with the cloud-hosted services on which they depend.

Figure 1: Summary of findings. A more technical list of specific research topics appears in Figure 6.

We shall start with a brief review to establish common terminology and background. For our purposes here, the power grid can be understood in terms of three periods [34],[10]. The "early" grid arose as the industry neared the end of an extended period of monopoly control. Power systems were owned and operated by autonomous, vertically-integrated, regional entities that generated power, bought and sold power to neighboring regions, and implemented proprietary *Supervisory Control And Data Acquisition* (SCADA) systems. These systems mix hardware and software. The hardware components collect status data (line frequency, phase angle, voltage, state of fault-isolation relays, etc.), transmit this information to programs that clean the input of any bad data, and then perform *state estimation*. Having computed the optimal system configuration, the SCADA platform determines a *control policy* for the managed region, and then sends instructions to actuators such as generator control systems, transmission lines with adjustable capacity and other devices to increase or decrease power generation, increase or decrease power sharing with

neighboring regions, shed loads, etc. The SCADA system also plays key roles in preventing grid collapse by shedding busses if regional security[2] requires such an action.

The "restructuring" period began in the 1990's and was triggered by a wave of regulatory reforms aimed at increasing competitiveness [19]. Regional monopolies fragmented into power generating companies, *Independent System Operators* (ISOs) responsible for long-distance power transmission and grid safety, and exchanges in which power could be bought and sold somewhat in the manner of other commodities (although the details of power auctions are specific to the industry, and the difficulty of storing power also distances power markets from other kinds of commodity markets). Small power producers entered the market, increasing competitive pressures in some regions. Greater inter-regional connectivity emerged as transmission lines were built to facilitate transfer of power from areas with less expensive power, or excess generating capacity into regions with more costly power, or less capacity.

One side effect of deregulation was to create new economic pressures to optimize the grid, matching line capacity to the pattern of use. Margins of excess power generating capacity, and excess transmission capacity, narrowed significantly, hence the restructured grid operates much nearer its security limits. SCADA systems play key roles, performing adjustments in real-time that are vital for grid security. The cost of these systems can be substantial; even modest SCADA product deployments often represent investments of tens or hundreds of millions of dollars, and because federal regulatory policies require full redundancy, most such systems are fully replicated at two locations, so that no single fault can result in a loss of control.

This review was prepared during the very first years of a new era in power production and delivery: the dawn of the "smart" power grid. Inefficient power generation, unbalanced consumption patterns that lead to underutilization of expensive infrastructure on the one hand, and severe overload on the other, as well as urgent issues of national and global concern such as power system security and climate change are all driving this evolution [40]. As the smart grid concept matures, we'll see dramatic growth in green power production: small production devices such as wind turbines and solar panels or solar farms, which have fluctuating capacity outside of the control of grid operators. Small companies that specialize in producing power under just certain conditions (price regimes, certain times of the day, etc.) will become more and more common. Power consumers are becoming more sophisticated about pricing, shifting consumption from peak periods to off-peak periods; viewed at a global scale, this represents a potentially non-linear feedback behavior. Electric vehicles are likely to become important over the coming decade, at least in dense urban settings, and could shift a substantial new load into the grid, even as they decrease the national demand for petroleum products. The operation of the grid itself will continue to grow in complexity, because the effect of these changing modalities of generation and consumption will be to further fragment the grid into smaller regions, but also to expand the higher level grid of long-distance transmission lines. Clearly, a lot of work is required to transition from the 50-year-old legacy grid of today to the smart grid of the future. Our purpose in this paper is to see how far the computing industry is ready to meet the needs of this transition.

[2] *Security* here is to mean the safety and stability of the power grid, rather than protection against malice.

2. The Computational Needs of the Smart Grid

We present a few representative examples that show how large-scale computing must play a key role in the smart power grid. In the next sections, we shall see whether current computing platforms are well suited to play this role.

i. **The smart home.** In this vision, the home of the future might be equipped with a variety of power use meters and monitoring devices, adapting behavior to match cost of power, load on the grid, and activities of the residents. For example, a hot-water heater might heat when power is cheap but allow water to cool when hot water is unlikely to be needed. A washing machine might turn itself on when the cost of power drops sufficiently. Air conditioning might time itself to match use patterns, power costs, and overall grid state. Over time, one might imagine ways that a SCADA system could reach directly into the home, for example to coordinate air conditioning or water heating cycles so that instead of being random and uniform, they occur at times and in patterns convenient to the utility.

ii. **Ultra-responsive SCADA for improved grid efficiency and security.** In this area, the focus is on improving the security margins for existing regional control systems (which, as noted earlier, are running with slim margins today), and on developing new SCADA paradigms for incorporating micro-power generation into the overall grid. One difficult issue is that the power produced by a wind farm might not be consumed right next to that farm, yet we lack grid control paradigms capable of dealing with the fluctuating production and relatively unpredictable behavior of large numbers of small power generating systems. One recent study [2] suggested that to support such uses, it would be necessary to create a new kind of grid-stat system, tracking status at a fine-grained level. Such approaches are likely to have big benefits, hence future SCADA systems may need to deal with orders of magnitude more information than current SCADA approaches handle.

iii. **Wide area grid state estimation.** Blackouts such as the NorthEast and Swiss/Italian blackouts (both in 2003), originated with minor environmental events (line trips caused by downed trees), but that snowballed through SCADA system confusions that in turn caused operator errors (see “Northeast_Blackout_of_2003” and “2003_Italy_blackout" in Wikipedia). Appealing though it may be to blame the humans, those operator errors may have been difficult to avoid. They reflected the inability of regional operators to *directly observe* the state of the broader power grids to which their regions are linked; lacking that ability, a hodgepodge of guesswork and telephone calls are often the only way to figure out what a neighboring power region is experiencing. Moreover, the ability to put a telephone call through during a spreading crisis that involves loss of power over huge areas is clearly not something one can necessarily count upon in any future system design. As the power grid continues to fracture into smaller and smaller entities, this wide area control problem will grow in importance, with ISOs and other operators needing to continuously track the evolution of the state of the grid and, especially important, to sense abnormal events such as bus trips or equipment failures. Data about power contracts might inform decisions, hence the grid state really includes not just the data captured from sensors but also the *intent* represented in the collection of power production and consumption contracts.

What are the computational needs implied by these kinds of examples?

i. **Decentralization.** Information currently captured and consumed in a single regional power system will increasingly need to be visible to neighboring power systems and perhaps even visible on a national scale. An interesting discussion of this topic appears in [2].

ii. **Scalability.** Every smart grid concept we've reviewed brings huge numbers of new controllable entities to the table. In some ideas, every consumer's home or office becomes an independent point for potential SCADA control. In others, the homes and offices behave autonomously but still must tap into dynamic data generated by the power provider, such as pricing or load predictions. Other ideas integrate enormous numbers of small power producing entities into the grid and require non-trivial control adjustments to keep the grid stable. Thus scalability will be a key requirement – scalability of a kind that dwarfs what the industry has done up to now, and demands a shift to new computational approaches [25][26][2][40].

iii. **Time criticality.** Some kinds of information need to be fresh. For example, studies have shown that correct SCADA systems can malfunction when presented with stale data, and some studies have even shown that SCADA systems operated over Internet standards like the ubiquitous TCP/IP protocols can malfunction [25][26][2][12], because out-of-the-box TCP delays data for purposes of flow control and to correct data loss. Future smart-grid solutions will demand real-time response even in the presence of failures.

iv. **Consistency.** Some kinds of information will need to be consistent [5][6][7][8][25][19], in the sense that if multiple devices are communicating with a SCADA system at the same time, they should be receiving the same instructions, even if they happen to connect to the SCADA system over different network paths that lead to different servers that provide the control information. Notice that we're not saying that control data must be computed in some sort of radically new, decentralized manner: the SCADA computation itself could be localized, just as today's cloud systems often start with one copy of a video of an important news event. But the key to scalability is to replicate data and computation, and consistency issues arise when a client platform requests data from a service replica: is this really the *most current* version of the control policy? Further, notice that consistency and real-time guarantees are in some ways at odds. If we want to provide *identical* data to some set of clients, failures may cause delays: we lose real-time guarantees of minimal delay. If we want *minimal delay*, we run the risk that a lost packet or a sudden crash could leave some clients without the most recent data.

v. **Data Security. S**everal kinds of data mentioned above might be of interest to criminals, terrorists, or entities seeking an edge in the power commodities market. Adequate protection will be a critical requirement of future SCADA systems.

vi. **Reliability.** Power systems that lose their control layer, even briefly, are at grave risk of damage or complete meltdown. Thus any SCADA solution for the future smart grid needs to have high reliability.

vii. **Ability to tolerate compromise.** The most critical subsystems and services may need to operate even while under attack by intruders, viruses, or when some servers are malfunctioning. The technical term for this form of extreme reliability is Byzantine Fault Tolerance; the area is a rich one and many solutions are known, but deployments are rare and little is known about their scalability.

3. The Evolving Computing Industry: An Economic Story

We shall now describe the current state of the computing industry, and examine its ability to provide the properties described above for the future smart grid. We begin by giving a brief history of the computing industry and the economic drivers of its evolution. These same drivers are likely to determine whether the power community can use current computing platforms for its needs, or not.

Prior to the late 1990's, the computing industry was a world of client computers that received data and instructions from servers. Client-server computing represented a relatively wrenching transition from an even earlier model (mainframe computing), and the necessary architecture and tools were slow to mature; in some sense, the excitement associated with the area anticipated the actual quality of the technology by five to ten years. Yet the client-server architecture slowly gained acceptance and became the basis of widely adopted standards, until finally, within the last decade or so, software tools for creating these kinds of applications have made it possible for a typical programmer to create and deploy such applications with relative ease.

Today, client-server computing is the norm, yet the power industry retains legacies from the mainframe computing era. For example, SCADA systems use high performance computing (HPC) techniques but play roles similar to SCADA solutions in older mainframe architectures, which featured a big computer in the middle of a slaved network of sensors and actuators. This is in contrast to cloud architectures, which take the client-server model and push it even further: the client is now supported by multiple data centers, each of which might be composed of a vast number of relatively simple servers, with second and even third tiers of support layered behind them. But the issues are also social: power is a critical infrastructure sector – one that affects nearly every other sector – and understandably, the power community is traditionally risk-averse and slow in adopting new technology trends.

The computing industry has seen three recent technical epochs, each succeeding the prior one in as little as five years. Looking first at the period up to around the centennial, we saw a game-changing transition as the early Internet emerged, blossomed, briefly crashed (the .com boom and bust), and then dramatically expanded again. That first boom and bust cycle could be called the early Internet and was dominated by the emergence of web browsers and by human-oriented Internet enterprises. The Internet architecture became universal during this period. Prior to the period in question, we had a number of networking technologies, with some specialized ones used in settings such as wireless networks, or in support of communications overlaid on power transmission lines. Many power companies still use those old, specialized, communication technologies. But today, the Internet architecture has become standard. This standardization is useful. For example modern power companies visualize the status of sensors and actuators through small web pages that provide quick access to parameter settings and controls. Software on those devices can be quickly and easily patched by upgrading to new versions over the network. But these same capabilities have also created the potential for unintended connectivity to the Internet as a whole. Attackers can exploit these opportunities: we saw this in the widely publicized "Eligible Receiver" exercises, in which the government demonstrated that a technically savvy but non-expert team could use publicly available information to take control of power systems and inflict serious damage on transformers, generators, and other critical equipment [39].

We now arrive at a period covering roughly the past five years, which witnessed a breathtaking advance in the penetration and adoption of web technologies. Standardization around web protocols and the ease of adding web interfaces even to older mainframe or client-server applications meant that pretty much any computing entity could access any other computing entity, be it hardware or software. Outsourcing boomed as companies in India, China, and elsewhere competed to offer inexpensive software development services. Penetration of the Internet into the public and private sector triggered explosive revenue growth in all forms of Internet advertising. New computing platforms (mobile phones, tablet computers) began to displace traditional ones, triggering a further boom associated with mobility and "app" computing models. Rarely have so many changes been compressed into so short a period of time.

Perhaps most unsettling of all, completely new companies like Facebook and Google displaced well established ones like IBM, HP, and Microsoft, seemingly overnight. One might reasonably argue that the power industry should be immune to this sort of turmoil, yet the impact of restructuring has caused an equal shakeup on the business side of the power community, even if the technical side remains less impacted. And there is good reason to believe that this will soon change. For example, the team that created Google is prominent among industry leaders promoting a smarter power grid. It is hard to imagine them being content to do things in the usual ways.

Cloud computing, our primary focus in this paper, is an overarching term covering the technologies that support the most recent five-years or so of the Internet, with different specific meanings for different cloud operators. The term means different things to different cloud owner/operators, but some form of cloud computing can be expected in any future Internet. A recent document laying out a Federal Cloud Computing Strategy, drafted by the CIO of the United States government (Dr. Vivek Kundra) recently called for spending about $20 billion of the $80 billion federal IT budget on cloud computing initiatives [28] and urged all government agencies to develop Cloud-based computing strategies. About a third of the cost would come from reductions in infrastructure cost through data center consolidation.

The perspective that sheds the most light on the form that cloud computing takes today starts by recognizing that cloud computing is an intelligent response to a highly monetized demand, shaped by the economics of the sectors from which that demand emanated [8]. These systems guarantee the properties needed to make money in these sectors; properties not required (or useful only in less economically important applications) tend not to be.

What are these requirements? Perhaps the most important emerges from the pressure to aggregate data in physically concentrated places. The rise of lightweight, mobile devices, and of clients who routinely interact with multiple devices, shifts the emphasis from personal computing (email on the user's own machine, pictures in my private folder, etc.) towards data center hosting models, for example Hotmail, Gmail, Flickr, and YouTube. Social networking sites gained in popularity, for example Facebook, YouTube, Flickr, and Twitter; they revolve around sharing information: my data and your data need to be in the same "place" if we're to share and to network in a sharing-driven manner. Moreover, because cloud platforms make money by performing search and placing advertising, cloud providers routinely need to index these vast collections of data, creating pre-computed tables that are used to rapidly personalize responses to queries.

Thus, cloud computing systems have exceptional capabilities for moving data from the Internet into the cloud (web crawlers), indexing and searching that data (MapReduce [16], Chord [3], Dynamo [17], etc.), managing files that might contain petabytes of information (BigTable [13], the Google File System [20], Astrolabe[35]), coordinating actions (Chubby [12], Zookeeper [26], DryadLINQ [38]), and implementing cloud-scale databases (PNUTS [15]). These are just a few of many examples.

Massive data sets are just one respect in which cloud systems are specialized in response to the economics of the field. Massive data centers are expensive, and this creates a powerful incentive to drive the costs down and to keep the data center as busy and efficient as possible. Accordingly, cost factors such as management, power use, and other considerations have received enormous attention [21]. Incentives can cut both ways: social networking sites are popular, hence cloud computing tools for sharing are highly evolved; privacy is less popular, hence little is known about protecting data once we move it into the cloud [29].

It should not be surprising that cloud computing has been shaped by the "hidden hand of the market," but it is important to reflect on the implications of this observation. The specific attributes of the modern data center and its cloud computing tools are matched closely to the ways that companies like Amazon, Microsoft, Google and Facebook use them: those kinds of companies invested literally hundreds of billions of dollars to enable the capabilities with which they earn revenue. Cloud computing emerged overnight, but not painlessly, and the capabilities we have today reflect the urgent needs of the companies operating the cloud platforms.

How then will we deal with situations in which the power grid community needs a cloud capability lacking in today's platforms? Our market-based perspective argues for three possible answers. If there is a clear reason that the capability is or will soon be central to an economically important cloud computing application, a watch and wait approach would suffice. Sooner or later, the train would come down the track. If a capability somehow would be costly to own and operate, even if it were to exist, it might rapidly be abandoned and actively rejected by the community. We'll see that there is an instance of this nature associated with *consistency.* Here, only by finding a more effective way to support the property could one hope to see it adopted in cloud settings (hence, using the same economic metrics the community uses to make its own go/no-go decisions). Finally, there are capabilities that the commercial cloud community would find valuable, but hasn't needed so urgently as to incentivize the community to actually create the needed technology. In such cases, solving the problem in a useful prototype form might suffice to see it become part of the standards.

4. The Case for Hosting the Smart Grid on Cloud Computing Infrastructures

Cloud computing is of interest to the power community for several business reasons. Some parallel the green energy considerations that have stimulated such dramatic change in the power industry: cloud computing is a remarkably efficient and green way to achieve its capabilities. Others reflect pricing: cloud computing turns out to be quite inexpensive in dollar terms, relative to older models of computing. And still others are stories of robustness: by geographically replicating services, companies like Google and Microsoft are achieving fraction of a second responsiveness for clients worldwide, even when failures or regional power outages occur. Cloud systems can be managed cheaply and in highly automated ways, and protected against attack more easily than traditional systems [31]. Finally, cloud computing offers astonishing capacity and elasticity: a modern cloud

computing system is often hosted on a few data centers any one of which might have more computing and storage and networking capacity than all of the world's supercomputing centers added together, and can often turn on a dime, redeploying services to accommodate instantaneous load shifts. We shall enumerate some of the issues in the debate about using the cloud for building the smart grid.

4.1. The Cloud Computing Scalability Advantage

The cloud and its transformation of the computing industry have resulted in the displacement of previous key industry players like Intel, IBM, and Microsoft by new players like Google, Facebook, and Amazon. Technology these new-age companies created is becoming irreversibly dominant for any form of computing involving *scalability*: a term that can mean direct contact with large numbers of sensors, actuators or customers, but can also refer to the ability of a technical solution to run on large numbers of lightweight, inexpensive servers within a data center. Earlier generations of approaches were often abandoned precisely because they scaled poorly. And this has critical implications for the smart grid community, because it implies that to the extent that we launch a smart grid development effort in the near term, and to the extent that the grid includes components that will be operated at large scale, those elements will be built on the same platforms that are supporting the Facebooks and Amazons of today's computing world. In Figure 2 and Figure 3, we look at the scalability needs of two scenarios representative of the future smart grid.

4.2. The Cloud Cost Advantage

The Smart Grid needs a national-scale, pervasive network that connects every electricity producer in the market, from coal and nuclear plants to hydroelectric, solar, and wind farms, and small independent producers, with every electricity consumer, from industrial manufacturing plants to residences, and to every device plugged into the wall. This network should enable the interconnected devices to exchange status information and control power generation and consumption. The scale of such an undertaking is mind boggling. Yet, the key enabler, in the form of the network itself, already exists. Indeed, the Internet already allows household refrigerators to communicate with supermarkets and transact purchases [30]. It won't be difficult to build applications ("apps") that inform the washing machine of the right time to run its load, based on power pricing information from the appropriate generators. Whatever their weaknesses, the public Internet and cloud offer such a strong cost advantage that the power community cannot realistically ignore them in favor of building a private, dedicated network for the smart grid.

4.3. Migrating High Performance Computing (HPC) to the Cloud

We noted that SCADA systems are instances of "high performance computing" applications. It therefore makes sense to ask how the cloud will impact HPC. Prior to the 1990s, HPC revolved around special computing hardware with unique processing capabilities. These devices were simply too expensive, and around 1990 gave way to massive parallelism. The shift represented a big step backward for some kinds of users, because these new systems were inferior to the ones they replaced for some kinds of computation. Yet like it or not, the economics of the marketplace tore down the old model and installed the new one, and HPC users were forced to migrate. Today, even parallel HPC systems face a similar situation. A single cloud computing data center might have storage and computing capabilities tens or hundreds of times greater than all of the world's

Scenario one: National Scale Phasor Data Collection

A *phasor* is a complex number representing the magnitude and phase angle of a wave. Phasors are measured at different locations at a synchronized time (within one microsecond of one another). The required accuracy can be obtained from GPS. For 60 Hz systems, each Phasor Measurement Unit (PMU) takes about 10 to 30 such measurements per second. The data from various (up to about 60) PMUs is collected by a Phasor Data Concentrator (PDC) (transmitted over phone lines), and then forwarded along a Wide Area Measurement System (WAMS) to a SCADA system. The SCADA system must receive the data within 2 to 10 seconds.

It has been suggested that as the future power grid becomes increasingly interconnected to promote sharing so as to reduce wasted power and smooth the regional impact of erratic wind and solar power generation, we will also expose the grid to rolling outages. A possible remedy is for the regional operators to track the national grid by collecting phasor data locally and sharing it globally. We now suggest that the scale of the resulting problem is similar to the scale of computational challenges that motivated web search engines to move to the modern cloud computing model.

Simple back-of-the-envelope-calculations lead to a cloud computing model: Today's largest PMU deployment has about 120 PMUs, but for the purposes outlined here, one could imagine a deployment consisting of at least 10,000 PMUs. If we have 25 PMUs per PDC, then such a system would require 400 PDCs. Each PDC would deliver 30 measurements per second. If a measurement is 256 bytes in size (including magnitude, phase angle, timestamp, origin information, and perhaps a digital signature to protect against tampering or other forms of data corruption), then each PDC would deliver 25 x 256 x 30 = 192 KBytes/sec. The 400 PDCs combined would contribute about 77 Mbytes/sec, or about 615 Mbits/sec. The data would probably have to be shared on a national scale with perhaps 25 regional SCADA systems, located throughout the country, hence the aggregate data transmission volume would be approximately 15 Gbit/sec, more than the full capacity of a state of the art optical network link today[3].

While it would be feasible to build a semi-dedicated national-scale phasor-data Internet for this purpose, operated solely for and by the power community, we posit that sharing the existing infrastructure would be so much cheaper that it is nearly inevitable that the power community will follow that path. Doing so leverages the huge investment underway in cloud computing systems to distribute movies and Internet video; indeed, the data rates are actually "comparable" (a single streamed HD DVD is about 40 Mbits/second). But it also forces us to ask what the implications of monitoring and controlling the power grid "over" the Internet might be; these questions are at the core of our study (we pose, but don't actually answer them).

Figure 2: Tracking Phasor Data on a National Scale

[3] The 10Gbit rate quoted is near the physical limits for a single optical network link operated over long distances (as determined by the Shannon coding theory). But it is important to keep in mind that Internet providers, having invested in optical networking capacity, can often run multiple side-by-side optical links on the same physical path. Thus, the core Internet backbone runs at 40Gbits, and this is achieved using 4 side-by-side 10Gbit optical links. Moreover, network providers often set aside dedicated bandwidth under business arrangements with particular enterprises: Google or MSN, for example, or Netflix. Thus even if the future power grid runs "over" the Internet, this does not imply that grid control traffic could be disrupted or squeezed out by other kinds of public traffic.

Scenario Two: Power Aware Appliances in a Smart Home

According to the most recent US government census report, the United States had approximately 115 million households in 2010. Appliance ownership is widely but variably estimated. Reports on the web suggest that more than 95% of all households have major kitchen equipment such as a refrigerator and range, that 40 to 60% own a dishwasher, between 60 and 95% have a dedicated washer and dryer, and that as many as 80% or more have their own hot water heaters (the quality of these statistics may be erratic). These homes are heated, air conditioned, artificially lighted, and contain many powered devices (TVs, radios, etc.). Some will soon own electric vehicles.

Such numbers make clear the tremendous opportunity for smart energy management in the home. Current industry trends suggest the following mode: the consumer will probably gravitate towards mobile phone "apps" that provide access to home energy management software, simply because this model has recently gained so much commercial traction through wide adoption of devices such as the iPhone, BlackBerry, and Android phones, all of which adopt this particular model; apps are easy to build, easy to market, have remarkable market penetration, and are familiar to the end user. As they evolve, power-aware apps will coordinate action to operate appliances in intelligent ways that reduce end-user costs but also smooth out power demands, reduce load when the grid comes under stress, etc.

Thus, one might imagine a homeowner who loads the dishwasher but doesn't mind it running later, needs hot water early in the morning (or perhaps in the evening; the pattern will vary but could be learned on a per-household basis), etc. Ideally, the local power grid would wish to "schedule" these tasks in a price-aware, capacity-aware, energy efficient manner.

In one popular vision the grid simply publishes varying prices, which devices track. But this approach is poorly controlled: it is hard to know how many households will be responsive to price variability, and while one could imagine a poorly subscribed service failing for lack of popularity, one can also imagine the other extreme, in which a small price change drives a massive load shift and actually destabilizes the grid. Some degree of "fine grained" control would be better.

Thus, we suspect that over time, a different model will emerge: utilities will be motivated to create their own power management "apps" that offer beneficial pricing in exchange for direct grid control over some of these tasks: the grid operator might, for example, schedule dishwashing and clothes washing at times convenient to the grid, vary household heating to match patterns of use, heat water for showers close to when that hot water will be needed, etc.

But ***these are cloud computing concepts***: the iPhone, Blackberry, and Android are all so tightly linked to the cloud that it is just not meaningful to imagine them operating in any other way. Smarter homes can save power, but the applications enabling these steps must be designed to run on cloud computing systems, which will necessarily handle sensitive data, be placed into life-critical roles, and must be capable of digital "dialog" with the utility itself. All of these are the kinds of issues that motivate our recommendation that the power community start now to think about how such problems can be solved in a safe, trustworthy, and private manner.

Figure 3: Power-Aware Home Using Cloud-Hosted Power Management Applications ("Apps")

supercomputing facilities combined. Naturally, this incentivizes the HPC community to look to the cloud. Moreover, to the extent that HPC applications do migrate into the cloud, the community willing to pay to use dedicated HPC (non-cloud HPC) shrinks. This leaves a smaller market and, over time, represents a counter-incentive for industry investment in faster HPC systems. The trend is far from clear today, but one can reasonably ask whether someday, HPC as we currently know it (on fast parallel computers) will vanish in favor of some new HPC model more closely matched to the properties of cloud computing data centers.

The big challenge for HPC in the cloud revolves around what some call the *checkpoint barrier.* The issue is this: modern HPC tools aren't designed to continue executions during failures. Instead, a computation running on *n* nodes will typically stop and restart if one of the *n* fails. To ensure that progress is made, periodic checkpoints are needed. As we scale an application up, it must checkpoint more often to make progress. But checkpointing takes time. It should be clear that there is a number of nodes beyond which all time will be spent checkpointing and hence no progress can be made at all. On traditional HPC hardware platforms, the checkpoint barrier has not been relevant: failure rates are low. But cloud computing systems often have relatively high rates of node and storage server failures: having designed the systems to tolerate failures, it becomes a cost-benefit optimization decision to decide whether to buy a more reliable, but more costly server, or to buy a larger number of cheaper but less reliable ones. This then suggests that HPC in the current form may not migrate easily to the cloud, and also that it may not be possible to just run today's standard SCADA algorithms on large numbers of nodes as the scale of the problems we confront grows in response to the trends discussed earlier. New SCADA solutions may be needed in any case; versions matched closely to the cloud model may be most cost-effective.

4.4. High Assurance Applications and the Cloud Computing Dilemma

The cloud was not designed for high-assurance applications, and therefore poses several challenges for hosting a critical infrastructure service like the smart grid. One complicating factor is that many of the cost-savings aspects of the cloud reflect forms of sharing: multiple companies (even competitors) often share the same data center, so as to keep the servers more evenly loaded and to amortize costs. Multiple applications invariably run in a single data center. Thus, whereas the power community has always owned and operated its own proprietary technologies, successful exploitation of the cloud will force the industry to learn to share. This is worrying, because there have been episodes in which unscrupulous competition within the power industry has manifested itself through corporate espionage, attempts to manipulate power pricing, etc. (ENRON being only the most widely known example). Thus, for a shared computing infrastructure to succeed, it will need to have ironclad barriers preventing concurrent users from seeing one-another's data and network traffic.

The network, indeed, would be a shared resource even if grid operators were to run private, dedicated data centers. The problem here is that while one might imagine creating some form of separate Internet specifically for power industry use, the costs of doing so appear to be prohibitive. Meanwhile, the existing Internet has universal reach and is highly cost-effective. Clearly, just as the cloud has inadequacies today, the existing Internet raises concerns because of its own deficiencies. But rather than assuming that these rule out the use of the Internet for smart grid applications, we should first ask if those deficiencies could somehow be fixed. If the Internet can be enhanced to

improve robustness (for example, with multiple routing paths), and if data is encrypted to safeguard it against eavesdroppers (using different keys for different grid operators), it is entirely plausible that the shared public Internet could emerge as the cheapest and most effective communication option for the power grid. Indeed, so cost-effective is the public Internet that the grid seems certain to end up using it even in its current inadequate form. Thus, it becomes necessary to undertake the research that would eliminate the technical gaps.

We've discussed two aspects of the cloud in enough detail to illustrate the mindset with which one approaches these kinds of problems, using a market-based perspective to understand why cloud computing takes the form it does, and then using that same point of view to conceive of ways that technical improvements might also become self-sustaining cloud computing options once created, evaluated, and demonstrated in a convincing manner. But it is important to understand that these were just two of many such issues. Let's touch briefly on a few other important ones. Cloud computing is also peculiar in its access control and privacy capabilities [18][27][33]. Google's motto is "Don't be Evil", because in the cloud, the providers all must be trusted; if Google (or any of its thousands of employees) actually *are* evil, we may already be in a difficult situation. The cloud just doesn't have a serious notion of private data and, indeed, many in the industry have gone to lengths to point out that in a detailed, technical, legally binding sense, terms like privacy are very much up in the air today [33]. What precisely does it mean to ensure the privacy of an email, or a video, in a world where people casually send unencrypted messages over the public network, or share details of their personal histories with "friends" they know only as user-names on Facebook?

So extreme is this situation, and so pervasive the reach of the cloud, that it is already possible that any technical remedy could be out of reach. At minimum, the law lags the technology [29]. An editorial in the New York Times goes further, suggesting that the era of individual privacy may already be over [27], a sobering thought for those who hope to live unobserved, private lives.

Today's cloud technology is also weak in the area of reliability: the cloud is always up, but data centers often suffer from brief episodes of amnesia, literally forgetting something as soon as they learn it, and then (perhaps) rediscovering the lost information later. Sometimes, data is uploaded into a cloud, promptly lost, and never rediscovered at all. This can lead to a number of forms of *inconsistency*, a term used in the distributed computing community to refer to a system that violates intuitive notions of server correctness in ways that reveal the presence of multiple server replicas that are acting in uncoordinated ways, or using stale and incomplete data [4]. A consistency-preserving guarantee would eliminate such issues, but today's cloud systems manage well enough with weak consistency (after all, how much consistency is really required for a search query, or to play a video?) By imposing weak consistency as an industry standard, the cloud platforms become simpler and hence cheaper to build and to manage. Thus, yet again, we see economic considerations emerging as a primary determinant of what the cloud does and does not offer.

The issue goes well beyond service consistency. Cloud computing also places far greater emphasis on the robustness of the data center as a whole than on the robustness of any of the hundreds of thousands of servers it may have within it: data centers casually shut servers down if they seem to be causing trouble. No reliability assumptions at all are made about client systems, in part because viruses, worms, and other malware have hopelessly compromised the technologies we run on client platforms. By some estimates [14][18], fully 80% of home computers are slaves in one or more

Botnets, basically seeming normal (maybe slow) to the owner yet actually under remote control by shadowy forces, who can use the hijacked machines as armies in the Internet's version of warfare (for example, Estonia and Ukraine have both been taken off the network in recent years [14]), use them as host sites for illicit materials, or simply harness them as sources for waves of spam. In his fascinating analysis of the cyber-attack risks associated with network-based terrorism, Richard Clarke discusses the risks to today's power grid at some length [14]. In a nutshell, he shows that power control systems are poorly secured and can be attacked via the Internet or, using public information, attacked by cutting wires. Either outcome could be disastrous. Worst among his scenarios are attacks that use "logic bombs" planted long ahead of the event; he conjectures that such threats may already be widely disseminated in today's power grid control systems.

Clearly, this situation will need to change. The smart grid will play a wide range of safety and life-critical roles, and it is completely reasonable to invest more money to create a more robust technology base. For example, it is possible to use automated code verification techniques to prove that modest sized computing systems are correct. We can use hardware roots of trust to create small systems that cannot be compromised by viruses. By composing such components, we can create fully trustworthy applications. Such steps might not work for the full range of today's cloud computing uses (and might not be warranted for the cloud applications that run Twitter or Facebook), but with targeted investment, the smart grid community can reach a point of being able to create them and to deploy them into cloud environments.

To summarize, let's again ask what cloud computing is "really about". The past few pages should make it clear that the term is really about many things: a great variety of assumptions that can seem surprising, or even shocking, when stated explicitly. We have a model in which all data finds its way into one or more massive storage systems, which are comprised of large numbers of individually expendable servers and storage units. Cloud platforms always guarantee that the data center will be operational, and try to keep the main applications running, but are far weaker in their guarantees for individual data items, or individual computations. The cloud security and privacy guarantees are particularly erratic, leaving room for cloud operators to be evil if they were to decide to do so, and even leaving open the worry that in a cloud shared with one's competitors, there might be a way for the competition to spy on one's proprietary data or control activities. Yet there seem to be few hard technical reasons for these limitations: they stem more from economic considerations than from science. Given the life-critical role of the power grid, some way of operating with strong guarantees in all of these respects would be needed, at least for the grid and for other "safety critical" purposes.

SUMMARY OF CLOUD PROPERTIES

CHARACTERISTICS OF TODAYS CLOUD COMPUTING AND INTERNET INFRASTRUCTURE

- **Inexpensive to own and operate.** Economies of scale, sharing, and automation are pervasive within cloud systems and central to the model.
- **Emphasis on rapid response and scalability.** Modern cloud computing systems are designed to ensure that every request from the client to the cloud receives a timely response, even if the response might be "incorrect".
- **Self-Managed, Power-Efficient, Self-Repairing.** Cloud computing systems are astonishingly green: they use power efficiently, keep machines busy, and dynamically adapt under all sorts of stresses, including load surges, failures, upgrades/downgrades, etc.
- **Weak Consistency Guarantees.** The embrace of the CAP theorem (see Section 6.4) has been used to justify a number of weak guarantees [31][37]. In a nutshell, most cloud services are capable of using stale data to respond to requests and the client is expected to deal with this. Cloud services are also unable to hide failures: the client must anticipate sudden faults and should reissue requests or otherwise compensate to mask such events.
- **Internet as a weak point.** The modern Internet experiences a surprising number of brief outages. Cloud computing systems are expected to ride them out. Multi-homing is offered for the cloud but not the average client (a cloud can be addressed by two or more distinct IP addresses), but we lack true multi-path routing options, so even with multi-homing, some clients may experience long periods of disrupted connectivity.

Figure 4: Summary of Assurance Properties

5. Three styles of Power Computing

We now concretize the foregoing discussion by grouping smart grid computing into three loosely defined categories. These are as follows:

i. **Applications with weak requirements.** Some applications have relatively relaxed needs. For example, because it takes a long time to install new transmission lines, applications that maintain maps of the physical infrastructure in a power delivery region will change relatively rarely, much as road maps rarely change. They can be understood as systems that provide guarantees but against easy constraints. Today's cloud is well matched to these uses.

ii. **Real-time applications.** This group of applications needs extremely rapid communication, for example to move sensor readings or SCADA control information fast enough to avoid actions based on stale data. Some studies suggest that for many SCADA control policies, even 50ms of excess delay relative to the minimum can be enough to result in incorrect control decisions [23][25][1]. Today's cloud is tuned to provide fast responses, but little attention has been given to maintaining speed during failures of individual server nodes or brief Internet connectivity disruptions.

iii. **Applications with strong requirements.** A final class of applications requires high assurance, strong access control and security policy enforcement, privacy, fault-tolerance, consistent behavior over collections of endpoints at which actions occur, or other kinds of properties. We will argue that the applications in this class share common platform requirements, and that those differ (are incomparable with) the platform properties needed for real-time purposes [4][5][36][23]. Today's cloud lacks the technology for hosting such applications.

We've argued that the cloud takes the form seen today for economic reasons. The industry has boomed, and yet has been so focused on rolling out new competitively exciting technologies and products that it has been limited by the relative dearth of superb engineers capable of creating and deploying new possibilities. The smart grid would have a tough time competing head to head for the same engineers who are focused on inventing the next Google, or the next iPad. However, by tapping into the academic research community, it may be possible to bring some of the brightest minds in the next generation of researchers to focus on these critical needs.

Figure 5 summarizes our observations. One primary conclusion is that quite a bit of research is needed simply to clarify the choices we confront. Yet the broader picture is one in which a number of significant technology gaps clearly exist. Our strong belief is that these gaps can be bridged, but we also see strong evidence that today's cloud developers and vendors have little incentive to do so and, for that reason, that a watch-and-wait approach would not succeed.

	Fits current cloud computing model	*Poses demanding HPC computing challenges*	*Collects data at many locations*	*Takes actions at many locations*	*Requires rapid real-time response*	*Requires strong consistency*	*Needs to protect personal or proprietary data*	*Must protect against attack*
Smart home	**Varies (1)**						✓	**? (3)**
Next generation SCADA with alternative power generation		✓	✓	✓	✓	**? (3)**		**? (3)**
Support for wide-area power contracts		✓	✓	✓		✓	✓	**✓(2)**
Grid protection			✓	✓	✓	**? (3)**		✓
Grid status monitoring	**? (3)**		✓			**? (3)**		**✓(2)**

Notes:

(1) Some prototypical "smart home" systems operate by using small computing devices to poll cloud-hosted web sites that track power pricing, then adapt actions accordingly. However not all proposed home-adaptation mechanisms are this simple; many would require closer coordination and might not fit the current cloud model so closely.

(2) Concerns here include the risk that disclosure of too much information could give some producers opportunities to manipulate pricing during transient generation shortages, and concerns that without publishing information about power system status it may be hard to implement wide-area contracts, yet that same information could be used by terrorists to disrupt the grid.

(3) Further research required to answer the question.

Figure 5: Cloud-Hosted Smart Grid Applications: Summary of Assurance Requirements

6. Technical Analysis of Cloud Computing Options

Some technical questions need more justification than was offered in the preceding pages. This section undertakes a slightly deeper analysis on a few particularly important issues. We reiterate claims made earlier, but now offer a more specific explanation of precisely why these claims are valid and what, if anything, might be done about the issues identified.

6.1. Rebooting a cloud-controlled smart grid

One place to start is with a question that many readers are no doubt puzzled by: the seeming conundrum of implementing a smart grid control solution on top of an Internet that would be incapable of functioning without power. How could one restart such a system in the event of a loss of regional power? There are two basic elements to our response. First: geographic diversity. Cloud computing makes it relatively easy to replicate control functionality at two or more locations that operate far from one another and hence, if one is lost, the other can step in. As for the Internet, it automatically reroutes around failures within a few minutes. Thus, for many kinds of plausible outages that impact a SCADA system at one location, having a software backup at a modest distance is sufficient: shipping photons is cheap and fast. In the Internet, nobody knows if their SCADA system is running next door, or two states over. Geographic diversity is also interesting because, at least for cloud operators, it offers an *inexpensive* way to obtain redundancy. Rather than building dual systems, as occurs in many of today's SCADA platforms for the existing power grid, one could imagine cloud-hosted SCADA solutions that amortize costs in a similar manner to today's major cloud applications, and in this way halve the cost of deploying a fault-tolerant solution.

But one can imagine faults in which a remote SCADA platform would be inaccessible because the wide-area network would be down, due to a lack of power to run its routers and switches. Thus, the second part of the answer involves fail-safe designs. The smart grid will need to implement a safe, "dumb" mode of operation that would be used when restarting after a regional outage and require little or no fine-grained SCADA control. As the system comes back up, more sophisticated control technologies could be phased back in. Thus, the seeming cycle of dependencies is broken: first, one restores the power; next, the Internet; last, the more elaborate forms of smart behavior.

6.2. Adapting standard cloud solutions to support more demanding applications

We've repeatedly asserted that the cloud is cheap. But why is this the case, and to what extent do the features of today's cloud platforms relate to the lower cost of those platforms?

Cloud computing can be understood as an approach that starts with client-server computing as its basis, and then scales it up dramatically – whereas server systems of the past might have run on 32 nodes, cloud systems often have hundreds of thousands of machines, each of which may have as many as 8 to 16 computational cores. Thus a cloud computing system is a truly massive structure. Some are as large as 4-5 football fields, packed so densely with computing and storage nodes that machines are purchased by the container-truck load and the entire container is literally "plugged in" as a unit. Yet as vast as these numbers may be, they are dwarfed by the even larger number of client systems. Today, it is no exaggeration to say that every laptop, desktop, pad, and even mobile telephone is a cloud-computing client system. Many have literally dozens of cloud applications running at a time. Thus the cloud is a world of billions of end user systems linked, over the Internet,

to tens of millions of servers, residing in data centers that individually house perhaps hundreds of thousands or millions of machines.

The cost advantage associated with this model relates to economies of scale. First, simply because of their scale, cloud computing systems turn out to be remarkably inexpensive to own and operate when compared with a small rack of servers such as one finds in most power industry control centers. James Hamilton, in his widely cited blog at http://mvdirona.com, has talked about the "cost of a cloud." He concludes that relative to other types of scalable infrastructure, the overall cost of ownership is generally a factor of 10 to 15 lower when all costs are considered (human, infrastructure, servers, power, software development, etc.). This is a dramatic advantage. Cloud systems also run "hot": with buildings packed with machines, rather than humans, the need for cool temperatures is greatly reduced. The machines themselves are designed to tolerate these elevated temperatures without an increased failure rate. The approach is to simply draw ambient air and blow it through the data center, without any form of air conditioning. Interior temperatures of 100°+F are common, and there has been talk of running clouds at 120°F. Since cooling costs money, such options can significantly reduce costs.

Furthermore, cloud systems often operate in places where labor costs and electric power costs are cheap: if a large power consumer is close to the generator, the excess power needs associated with transmission line loss are eliminated and the power itself becomes cheaper. Thus, one doesn't find these systems in the basement of the local bank; they would more often be situated near a dam on a river in the Pacific Northwest. The developers reason that moving information (such as data from the client computing system) to the cloud, computing in a remote place, and moving the results back is a relatively cheap and fast option today, and the speed and growth trends of the Internet certainly support the view that as time passes, this approach might even do better and better.

6.3. The Internet as a weak link

We've asserted that the Internet is "unreliable," yet this may not make sense at first glance; all of us have become dependent on a diversity of Internet-based mechanisms. Yet upon reflection, the concern makes more sense: anyone who uses an Internet radio, or who owns a television adapter that supports watching movies on demand, quickly realizes that while these technologies "usually" are quite robust, "sometimes" outages do occur. The authors of this white paper own a number of such technologies and have sometimes experienced multiple brief outages daily, some lasting just seconds, and others perhaps minutes. Voice over IP telephony is a similar experience: users of Skype think nothing of needing to try a call a few times before it goes through. Moreover, all of these are consequences of mundane issues: studies reveal that the Internet glitches we've been talking about are mostly triggered by operator error, brief load surges that cause congestion, or by failures of the routers that support the network; a typical network route today passes through 30 or more routers and when one goes offline, the Internet may need as much as 90 seconds to recover full connectivity. Genuinely long Internet outages have occurred more rarely, but they do happen from time to time, and the root causes can be surprising: in one event, an undersea cable got severed off Egypt, and India experienced disrupted network connectivity for some several days [1].

When the Internet has actually come under attack, the situation is much worse. Experience with outright attacks on the network is less limited than one might realize: recent events include so-called distributed denial of service attacks that have taken entire small countries (such as Estonia) off the

network for weeks, disrupted government and military web sites, and harassed companies like Google (when that company complained about China's political policies recently). A wave of intrusions into DoD classified systems resulted in the theft of what may have been terabytes of data [14]. Researchers who have studied the problem have concluded that the Internet is really a very fragile and trusting infrastructure, even when the most secure protocols are in use. The network could be literally shut down, and there are many ways to do it; some entirely based on software that can be launched from anywhere in the world (fortunately, complex software not yet in the hands of terrorists); other attacks might deliberately target key components such as high-traffic optical cables, using low-tech methods such as bolt cutters. Thus any system that becomes dependent upon the Internet represents a kind of bet that the Internet itself will be up to the task.

Thus the Internet is one "weak link" in the cloud computing story. We tolerate this weak link when we use our web phones to get directions to a good restaurant because glitches are so unimportant in such situations. But if the future smart grid is to be controlled over a network, the question poses itself: would this be the Internet, in a literal sense? Or some other network to be constructed in the future? On this the answer is probably obvious: building a private Internet for the power grid would be a hugely expensive proposition. The nation might well contemplate that option, but when the day comes to make the decision, we are not likely to summon the political will to invest on the needed scale. Moreover, that private Internet would become an extension of the public Internet the moment that some enterprising hacker manages to compromise even a single machine that has an Internet connection and also has a way to talk to the power network.

This is why we've concluded that the best hope is for a technical advance that would let us operate applications that need a secure, reliable Internet over today's less secure, less reliable one. Achieving such a capability would entail improving handling of failures within today's core Internet routers (which often are built as clusters but can be slow to handle failures of even just a single router component), and also offering enhanced options for building secure routes and for creating redundant routes that share as few links as possible, so that if one route becomes disrupted or overloaded, a second route might still be available. In addition, the power grid can make use of leased connections to further improve reliability and performance.

6.4. Brewer's CAP Conjecture and the Gilbert/Lynch CAP Theorem

We've discussed the relatively weak consistency properties offered by today's cloud computing platforms and even commented that cloud providers "embrace inconsistency" as a virtue [31][37]. Why is this the case, and can we hope to do anything about it? Cloud computing systems are so massive (and yet built with such relatively "weak" computers) that the core challenge in building cloud applications is to find ways to scale those applications up, so that the application (a term that connotes a single thing) might actually be implemented by thousands or even tens of thousands of computers, with the user's requests vectored to an appropriate machine.

How can this form of scaling be accomplished? It turns out that the answer depends much on the extent to which different user systems need to share data:

- At the easiest end of the spectrum we find what might be called "shared nothing" applications. A good example would be the Amazon shopping web pages. As long as the server my computer is communicating with has a reasonable approximation of the state of

the Amazon warehouse systems, it can give me reasonable answers to my queries. I won't notice if a product shows slightly different popularity answers to two identical queries reaching different servers at the same time, and if the number of copies of a book is shown as 3 in stock, but when I place my order suddenly changes to 1, or to 4, no great harm occurs. Indeed, many of us have had the experience of Amazon filling a single order twice, and a few have seen orders vanish entirely. All are manifestations of what is called "weak consistency" by cloud developers: a model in which pretty good answers are considered to be good enough. Interestingly, the computations underlying web search fall solidly into this category – so much so that entire programming systems aimed at these kinds of computing problems have become one of the hottest topics for contemporary research; examples include MapReduce [16] and other similar systems, file systems such as Google's GFS [20] and the associated BigTable database layered on top of it [13], etc. These are systems designed with loose coupling, asynchronous operation and weak consistency as fundamental parts of their model.

- A slightly harder (but not much harder) problem arises in social networking sites like Twitter or Facebook where groups of users share data, sometimes in real-time. Here, the trick turns out to be to control the network routing protocols and the so-called Domain Name Service (DNS) so that people who share data end up talking to the same server. While a server far away might pull up the wrong version of a page, or be slow to report a Tweet, the users talking to that single server would be unaware that the cloud has split its workload into perhaps millions of distinct user groupings.
- Gaming and Virtual Reality systems such as Second Life are similar to this second category of systems: as much as possible, groups of users are mapped to shared servers. Here, a greater degree of sophistication is sometimes needed and computer gaming developers publish extensively on their solutions: one doesn't want to overload the server, and yet one does want to support games with thousands of players. eBay faces a related challenge when an auction draws a large number of bidders. Such systems often play small tricks: perhaps not every bidder sees the identical bid sequence on a hotly contended-for item. As long as we agree on the winner of the auction, the system is probably consistent enough.
- Hardest of all are applications that really can't be broken up in these ways. Air Traffic Control would be one example: while individual controllers do "own" portions of the air space, because airplanes traverse many such portions in short periods of time, only an approach that treats the whole airspace as a single place and shows data in a consistent manner can possibly be safe. The "my account" portion of many web sites has a similar flavor: Amazon may use tricks to improve performance while one shops, but when an actual purchase occurs, their system locks down to a much more careful mode of operation.

The trade-offs between consistency and scalable performance are sometimes summarized using what Eric Brewer has called the Consistency Availability and Partitioning (CAP) theorem [11]. Brewer, a researcher at UC Berkeley and co-founder of Inktomi, argued in a widely cited keynote talk at PODC 2000 that to achieve high performance and for servers to be able to respond in an uncoordinated, independent manner to requests they receive from independent clients, those servers must weaken the consistency properties they offer. In effect, Brewer argues that weak consistency scales well and strong consistency scales poorly. A formalization of CAP was later proved under certain weak assumptions by MIT's Gilbert and Lynch, but data centers can often make

stronger assumptions in practice, and consequently provide stronger properties. Moreover, there are many definitions of consistency, and CAP is only a theorem for the specific definition that was used in the proof. Thus CAP is something of a folk-theorem: a convenient paradigm that some data centers cite as a reason for offering weak consistency guarantees (guarantees adequate for their own needs, although inadequate for high assurance purposes), yet not a "law of nature" that cannot be circumvented under any circumstances.

We believe that more investigation is needed into the scalability and robustness options that weaker consistency models might offer. CAP holds under specific conditions; perhaps data centers can be designed to invalidate those conditions most closely tied to the impossibility result. Hardware assistance might be helpful, for example in supporting better forms of cloud security. Thus CAP stands as an issue, but not one that should discourage further work.

6.5. Hidden Costs: Security Implications of Weak Consistency

Cloud security illustrates one of the dangers of casual acceptance of the CAP principles. We build secure systems starting with specifying a security policy that the system is expected to obey. Typically, these policies consist of rules and those rules are represented as a kind of database; the data in the database gives the logical basis for making security decisions and also identifies the users of the system and the categories of data. As the system runs, it can be thought of as proving theorems: Joe is permitted to access Sally's financial data because they are a couple; Sandra can do so because she is Sally's banker. John, Sally's ex-husband, is not permitted to access those records. The data evolves over time, and correct behavior of the system depends upon correct inference over the current versions of the underlying rules and the underlying data.

Cloud systems have real difficulty with these forms of security, because the same embrace of weak consistency that makes them so scalable also implies that data may often be stale or even outright wrong when the system tries to operate on it. Perhaps some node will be slow to learn about Sally's divorce – maybe it will never learn of it. Cloud systems don't provide absolute guarantees about such things, on the whole, and this makes them easier to scale up. But it also makes them deeply – perhaps fundamentally – untrustworthy.

The term "trustworthy" deliberately goes beyond security. Suppose that a smart grid control device needs to handle some event: perhaps line cycles drop or increase slightly, or a current surge is sensed. To coordinate the reaction appropriately, that device might consult with its cloud server. But even if connectivity is not disrupted and the cloud server is running, we run into the risk that the server instance that responds – perhaps one of a bank of instances that could number in the thousands – might have stale data and hence respond in an incorrect manner. Thus it is entirely possible for 99 servers to "know" about some new load on the grid, and yet for 1 server to be unaware of this, or to have data that is incorrect ("inconsistent") in a plethora of other ways.

Cloud systems are also quite casual about restarting servers even while they are actively handling client requests – this, too, is part of the scalability model (it reduces the human cost of management, because one doesn't need to gracefully shut things down before restarting them or migrating them). Thus our smart grid control device might find itself working off instructions that reflect faulty data, or deprived of control in an abrupt, silent manner, or suddenly talking to a new controlling server with no memory of the recent past.

7. Pretty Good is Sometimes Good Enough

Cloud computing is a world of very large scale systems in which most components are working correctly even if a few are lagging behind, working with stale data, restarting after an unplanned and sudden outage, or otherwise disrupted. Yet it is vital to realize that for many purposes these properties are good enough. Facebook, Youtube, Yahoo, Amazon, Google, MSN Live – all are examples of systems that host vast numbers of services that work perfectly well against this sort of erratic model. Google's difficulties repelling hacker attacks (apparently from China) do give pause; this event illustrates the downside of the cloud model; it is actually quite hard for Google to secure its systems for the same reasons we discussed earlier: security seems to be at odds with the mechanisms that make those systems scalable. Moreover the cloud model would seem to create loopholes that hackers can exploit (including the massive and remote nature of the cloud centers themselves: ready targets for agents of foreign powers who might wish to intrude and introduce virus or other undesired technical components).

The frustration for many in the field today is that we simply don't know enough about what can be solved in the standard cloud model. We also don't know enough about mapping stronger models onto cloud-like substrates or onto the Internet. Could the same hardware that runs the Internet not host software that might have better network security and reliability characteristics? One would be foolish to assert that this cannot be done. Could the same platforms we use in cloud settings not support applications with stronger properties? Very possibly. We simply don't know how to do so, yet, in part for the reason just cited: Google, Microsoft, Yahoo, and others haven't had much need to do this, and so the huge investment that gave us the cloud hasn't seen a corresponding investment to create a highly assured cloud for mission-critical roles.

Moreover, one can turn the problem on its head and ask whether control of the future smart grid actually requires consistency and coherency. Very possibly, one can control a smart grid in a manner that relies on a "mostly consistent" behavior by huge numbers of relatively loosely coupled, autonomous control agents. Perhaps centralized servers aren't even needed or, if they are needed, they don't need to behave in a manner one would normally think of as reflecting central control – terminology that already evokes the image of a single entity that makes the control decisions.

Finally, it is worthwhile to recognize that while the smart grid community may be confronting these problems for its own reasons, the community is certainly not alone. A future work of increasingly automated health care systems will surely have similar needs (imagine, for example, a substantial community of elderly home-care diabetic patients who depend upon remote control of their insulin pumps: the picture is comparable and the same concerns apply). Electronic medical health records will demand a strong model, at least as far as security, privacy, and rapid accurate data reporting are concerned. The same is true of banking systems, systems controlling infrastructure such as water or traffic lights, and indeed a plethora of socially sensitive, critical applications and services. Cloud computing beckons through its attractive price-point, but to benefit from that price point, we need to learn to move applications with sensitive requirements onto the cloud.

8. A Research Agenda

This paper was written to expose a problem, but not to solve it. The problem, as we've now seen, is that many of the most exciting ideas for the future smart grid presuppose models of computing that

have become outmoded and are being replaced by cloud computing. Others require a kind of scalability that only cloud computing can offer. And even mundane ideas sometimes have failed to grapple with the implications of an industry shift in which cloud computing has become a universal answer to every need: a commodity standard that is sweeping all other standards to the side. Familiar, successful computing models of the recent past may be the unsupported legacy challenges of the near-term future.

Yet cloud computing, as we've shown, lacks key properties that power control and similar smart grid functionality will need. These include security, consistency, real-time assurances, ways to protect the privacy of sensitive data, and other needs.

A doom-and-gloom story would, at this point, predict catastrophe. But the authors of this survey believe that every problem we've noted can probably be solved. The key is to incentivize researchers to work on these problems. Somewhat astonishingly, that research is not occurring today. With the exception of work on computer security, the government has largely pulled back from funding what could be called "basic systems" research, and there are no major research programs focused on highly assured cloud computing at NSF, DARPA, or other major government research agencies today. In effect, we're making a wager that industry will solve these problems on its own. Yet as noted above, cloud computing systems are under at most modest economic incentives to tackle these needs. They don't impact the bottom line revenue stream in major ways, and cloud computing has been shaped, up to now, by the revenue stream. To us this suggests that such a wager might fail.

Accordingly, we recommend that the nation embark on a broad-reaching and multi-faceted research effort. This effort would have elements specific to the smart electric power grid, but other elements that are cross-cutting and that would seem equally beneficial in future medical systems, banking systems, and a wide range of other application areas:

i. **Quantify the kinds of guarantees that cloud computing solutions can offer.** The goal of this effort would be to create a scientific foundation for cloud computing, with the mathematical and practical tools one associates with any scientifically rigorous foundation.
ii. **Quantify the kinds of guarantees that are required for a new generation of smart grid control paradigms.** This effort would seek to develop new strategies for control of a smart power grid, perhaps including such elements as decentralized control points and some degree of autonomous local control for smaller devices such as home units that might adapt their power consumption to better exploit off-peak power and reduce peak needs. It would then look at various ways to implement those strategies on cloud platforms.
iii. **Learn to reintroduce strong trust properties in cloud settings.** Perhaps the conclusion from these first efforts would be that today's CAP-conjecture-based cloud is ideally suited to some new style of weakly consistent control paradigm. But we may also find that some applications simply require cloud applications that can scale well and be administered cheaply, and yet that offer strong guarantees of security, consistency, availability, fault-tolerance, etc. If so, it will be incumbent upon us to learn to host such applications in cloud settings.
iv. **Better quantify the possible attacks against a computer-controlled smart grid.** We've seen that energy producers might be motivated to manipulate power markets (cf. the Enron

situation of some years ago), and Clarke's book points to the possibility of hackers or even foreign powers that might single out the power grid as their target. Needed are a careful analysis of the threats – all forms of threats – and a considered strategy for building systems that might defend against such attacks.

v. **Learn to build an Internet with better availability properties, even under attack.** Today's Internet has one primary role: it needs to get the data from the sender to the receivers and while reliability isn't a need on a packet-by-packet basis, we are learning that reliability *does* matter for "flows" that occur over longer periods of time. But the Internet isn't reliable in this second sense, and is easily attacked. We need to find ways to evolve the network to have much higher reliability for packet flows that need stronger assurance properties, and to do so even when the network comes under attack.

vi. **Improve attack tolerance.** If we are to build nationally critical infrastructures on the Internet, the day may come when adversaries attack those infrastructures. Today, this would result in serious disruption; tomorrow, as the dependencies enlarge, the results may be devastating. Thus it is obligatory to learn to build attack-tolerant versions of the key components of the future infrastructure. This is a tall order, but short of rejecting the Internet and the cloud as inappropriate for critical use, there really is no alternative but to find ways to secure what we build against major, deliberate, coordinated, sophisticated attacks.

It would not be honest to offer such a list without also observing that this is a tremendously ambitious, difficult agenda. Saying that such-and-such a problem "must be solved" is easy; estimating the time and resource needs to solve it is another matter. Worse still, the topics we've listed aren't typical of the areas receiving the most research energy and enthusiasm today.

A further observation, of a similar nature, is that computer security has been a source of frustration for decades; we've made huge progress and yet the landscape has shifted beneath our feet in such a way that the problems we've solved seem like issues that haven't mattered in decades, while the problems of the day seem far beyond reach. So to say that we need to "find a way" to create trustworthy cloud computing applications is facile and perhaps unrealistic. It may be that we will never reach a point at which computing can really be trusted in the senses required!

Yet it would also seem premature to give up. While there is a CAP theorem, we've commented that it holds only under very weak assumptions and there is no hard-and-fast reason that data centers can't make stronger assumptions. For this reason, CAP is more of a folk theorem: those who wish to build weakly consistent systems use CAP to justify their approach, elevating it to the status of a theorem perhaps as much to justify their own endorsement of weak properties as for any mathematically rigorous reason. Meanwhile, the theory community points to the theorem as an impossibility result, seemingly unaware that many cloud systems wouldn't match the assumptions used in proving the result, and hence aren't "bound" by it. And this same comment could be made in a much broader way: There is little concrete evidence that the obstacles to highly assured cloud computing are even all that hard. Perhaps all that is needed is new, talented minds and new kinds of applications, such as the smart grid, to help motivate the work and to ground it in reality. Lack of funding has impeded this entire area for almost a decade (triggered by a DARPA pull-back under the Bush administration). Thus, with more resources, an exciting and important problem, and perhaps some really bright young researchers, it may actually be possible to move mountains.

Summary of Highest Priority Research Topics

1. **Quantify the kinds of guarantees that cloud computing solutions can offer.**
2. **Quantify the kinds of guarantees that are required for a new generation of smart grid control paradigms.**
3. **Learn to reintroduce strong trust properties in cloud settings.**
4. **Better quantify the possible attacks against a computer-controlled smart grid.**
5. **Learn to build an Internet with better availability properties.**
6. **Improve attack tolerance.**

Figure 6: Summary of the most urgent research topics

9. Conclusions

The smart grid challenges us today: creating it could be the first and perhaps most important step towards a future of dramatically improved energy efficiency and flexibility. The Internet and the Cloud Computing model around which it has coalesced appear to be natural partners in this undertaking, representing the culmination of decades of work on high-productivity, low-cost computing in a distributed model. But only if the gap between the needs of the smart grid and the properties of the cloud can be bridged can these apparent opportunities be safely realized.

References

[1] Murad Ahmed. India Suffers Massive Internet Disruption After Undersea Cables Break. The Times. December 19, 2008.

[2] David E. Bakken, , Anjan Bose, Carl H. Hauser, Edmond O. Schweitzer III, David E. Whitehead, and Gregary C. Zweigle. Smart Generation and Transmission with Coherent, Real-Time Data. Technical Report TR-GS-015. School of Electrical Engineering and Computer Science. Washington State University; Pullman, Washington, USA. August, 2010

[3] Hari Balakrishnan, M. Frans Kaashoek, David Karger, Robert Morris, Ion Stoica. Looking up data in P2P systems. February 2003 Communications of the ACM, Volume 46 Issue 2

[4] Kenneth Birman. Reliable Distributed Systems Technologies, Web Services, and Applications. P. 2005, XXXVI, 668 p. 145 illus., Hardcover ISBN: 0-387-21509-3.

[5] Ken Birman. History of the Virtual Synchrony Replication Model. Appears in Replication: Theory and Practice. B. Charron-Bost, F. Pedone, A. Schiper (Eds) Springer Verlag, 2010. LNCS 5959, pp. 91–120, 2010.

[6] Ken Birman and Robbert van Renesse, eds. *Reliable Distributed Computing with the Isis Toolkit.* IEEE Computer Society Press, 1994, Los Alamitos, Ca.

[7] Ken Birman, Gregory Chockler, Robbert van Renesse. Towards A Cloud Computing Research Agenda. SIGACT News Distributed Computing Column. June 2009.

[8] Ken Birman, Coimbatore Chandersekaran, Danny Dolev, and Robbert van Renesse. How the Hidden Hand Shapes the Market for Software Reliability. In Proceedings of the First IEEE Workshop on Applied Software Reliability, Philadelphia, PA. June 2006.

[9] Kenneth P. Birman, Jie Chen, Kenneth M. Hopkinson, Robert J. Thomas, James S. Thorp, Robbert van Renesse, and Werner Vogels. Overcoming Communications Challenges in Software for Monitoring and Controlling Power Systems. Proceedings of the IEEE. Vol. 9, No. 5. May 2005.
[10] Anjan Bose. Electric Power Systems. Chapter in The Electrical Engineering Handbook, Wai Kai Chen Ed., Elsevier Academic Press, 2005.
[11] Eric A. Brewer. Towards Robust Distributed Systems Distributed Systems. Keynote presentation, ACM PODC, July 2000. www.cs.berkeley.edu/~brewer/cs262b-2004/PODC-keynote.pdf
[12] Mike Burrows. The Chubby lock service for loosely-coupled distributed systems. OSDI '06, November 2006.
[13] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber. Bigtable: A Distributed Storage System for Structured Data. Transactions on Computer Systems (TOCS) , Volume 26 Issue 2, June 2008.
[14] Richard Clarke and Robert Knake. Cyber War: The Next Threat to National Security and What to Do About. Ecco (April 20, 2010).
[15] Brian F. Cooper, Raghu Ramakrishnan, Utkarsh Srivastava, Adam Silberstein, Philip Bohannon, Hans-Arno Jacobsen, Nick Puz, Daniel Weaver, Ramana Yerneni. PNUTS: Yahoo!'s hosted data serving platform. Proceedings of the VLDB, Volume 1 Issue 2 August 2008.
[16] Jeffrey Dean, Sanjay Ghemawat. MapReduce: a flexible data processing tool. January 2010 Communications of the ACM, Volume 53 Issue 1.
[17] Joseph DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall, Werner Vogels. Dynamo: Amazon's highly available key-value store. SOSP 2007, Oct 2007.
[18] Whitfield Diffie, Susan Eva Landau. *Privacy on the Line: The politics of wiretapping and encryption.* MIT Press, 1999.
[19] Kira Fabrizio, Nancy Rose, Catherine Wolfram. Do Markets Reduce Costs? Assessing the Impact of Regulatory Restructuring on the U.S. Electric Generation Efficiency. American Economic Review. 2007.
[20] Sanjay Ghemawat, Howard Gobioff, Shun-Tak Leung. The Google File System. SOSP '03 (December 2003), Brighton Beach, England.
[21] James Hamilton. Internet Scale Efficiency. Keynote presentation at LADIS 2008, the 3rd Workshop on Large Scale Distributed Systems (IBM Hawthorne Research Facilities, Sept 2008). Presentation materials available at http://www.cs.cornell.edu/projects/ladis2008/materials/JamesRH_Ladis2008.pdf
[22] Maya Haridasan, Robbert van Renesse. Defense Against Intrusion in a Live Streaming Multicast System. In Proceedings of the 6th IEEE International Conference on Peer-to-Peer Computing (P2P2006), Cambridge, UK, September 2006.
[23] Chi Ho, Robbert van Renesse, Mark Bickford, Danny Dolev. Nysiad: Practical Protocol Transformation to Tolerate Byzantine Failures. USENIX Symposium on Networked System Design and Implementation (NSDI 08). San Francisco, CA. April 2008.
[24] Kenneth Hopkinson, Kate Jenkins, Kenneth Birman, James Thorp, Gregory Toussaint, and Manu Parashar. Adaptive Gravitational Gossip: A Gossip-Based Communication Protocol with User-Selectable Rates. IEEE Transactions on Parallel and Distributed Systems. December 2009 (vol. 20 no. 12) pp. 1830-1843
[25] Hopkinson, Ken M.; Giovanini, Renan; Wang, Xaioru; Birman, Ken P.; Coury, Denise V.; Thorp, James S. IEEE Transactions on Power Systems. EPOCHS: Integrated Cots Software For Agent-Based Electric Power And Communication Simulation (Earlier version published as Winter Simulation Conference.7-10 of December 2003, New Orleans, USA.)
[26] Flavio P. Junqueira, Benjamin C. Reed. The life and times of a Zookeeper (Talk abstract). PODC '09, August 2009.
[27] Paul Kirn. Little Brother is Watching. New York Times, Oct. 15 2010.

[28] Vivek Kundra. Federal Cloud Computing Strategy. http://www.cio.gov/documents/Federal-Cloud-Computing-Strategy.pdf. February 2011.

[29] Laurence Lessig. Code and Other Laws of Cyberspace (2000) ISBN 978-0-465-03913-5.

[30] LG Appliances. Internet Refrigerator. http://us.lge.com/www/product/refrigerator_demo.html

[31] Dan Pritchett. BASE, an ACID Alternative. ACM Queue, July 2008.

[32] Fred B. Schneider and Ken Birman. The Monoculture Risk Put into Context. IEEE Security & Privacy. Volume 7, Number 1. Pages 14-17. January/February 2009.

[33] Polly Spenger. Sun on Privacy: "Get Over It.". Wired Magazine, Jan 26, 1999.

[34] Robert J Thomas. *Issue Papers on Reliability and Competition*: Managing Relationships Between Electric Power Industry Restructuring and Grid Reliability. PSERC technical report Aug 2005. http://www.pserc.wisc.edu/documents/publications/papers/2005_general_publications/thomas_grid_reliability_aug2005.pdf. School of Electrical and Computer Engineering, Cornell University.

[35] Robbert van Renesse, Kenneth Birman, Werner Vogels. Astrolabe: A Robust and Scalable Technology for Distributed System Monitoring, Management, and Data Mining. ACM Transactions on Computer Systems, May 2003, Vol.21, No. 2, pp 164-206.

[36] Robbert van Renesse, Rodrigo Rodrigues, Mike Spreitzer, Christopher Stewart, Doug Terry, Franco Travostino. Challenges Facing Tomorrow's Data center: Summary of the LADiS Workshop. Large-Scale Distributed Systems and Middleware (LADIS 2008). September 2008.

[37] Werner Vogels. Eventually Consistent. ACM Queue. December 2008.

[38] Yuan Yu, Michael Isard, Dennis Fetterly, Mihai Budiu, Ulfar Erlingsson, Pradeep Kumar Gunda, Jon Currey. DryadLINQ: A System for General-Purpose Distributed Data-Parallel Computing Using a High-Level Language. Symposium on Operating System Design and Implementation (OSDI), San Diego, CA, December 8-10, 2008.

[39] US Department of Defense. Eligible Receiver. June 9-13, 1997. http://www.globalsecurity.org/military/ops/eligible-receiver.htm

[40] US Department of Energy. The Smart Grid: An Introduction. 2008. http://www.oe.energy.gov/DocumentsandMedia/DOE_SG_Book_Single_Pages(1).pdf