The Distributed Cloud as the Future of the Cloud and the Internet

Rick McGeer, Chief Scientist, US Ignite

Motivation

The primary motivation for moving Information Technology services from dedicated on-site, enterprise-owned hardware to virtualized, rented, scalable hardware over the network ("The Cloud") was economic. Various aspects included automation of low-level and routine IT tasks, such as hardware replacement and backup, the economies of centralized management, protection against hardware and facility failures, and easy scalability in the event of surges in demand. The key enabler for the Cloud was virtualization - the ability to build a customized software environment for a service and deploy it on rented hardware -- and easy migration of configured environments.

The key problem of the Cloud is the network. Every service, whether local or distributed, goes over the network, and it instantly becomes the bottleneck for IT services and applications. For an enterprise, the comparison is to having the service onsite. Service latency goes from microseconds to tens of milliseconds -- equivalent, in human time scales, from message delivery going from seconds to several hours (if 1 microsecond is equivalent to a second, 10 milliseconds is about 3 hours -- and 50 milliseconds, roughly coast-to-coast one way in the US -- is about 15 hours).

A further problem is that bandwidth goes from being unlimited and free to restricted and costly. Even low-end switches on a local area network will have bisection bandwidth of at least 50 gigabits per second (Gb/s), and personal capacity of at least 1Gb/s and often 10 Gb/s. Standard border routers, which define the bandwidth for the enterprise campus, are generally sized for 1-10 Megabits/second/person. Moving IT services to the Cloud represents a bandwidth reduction between service and user of about 1000x. Bandwidth is often described as "speed", but this is a misnomer; it is really flow, and thus is best visualized with fluid flow and pipe capacity. Thus, if 1 Mb/s is visualized as a sink faucet of about 1 cm diameter (3 square cm cross-section area), then 1 Gb/s should be visualized as a very large pipe, about 30 cm in diameter. So visualize the network of a typical enterprise as a very large collection of very large pipes running throughout the building, fed by a very small faucet coming in from the street. This makes it obvious that almost all of the "water" flowing through the building must be internally generated and consumed, and it would take a very long time for a substantial amount of water to arrive from outside. The Cloud, therefore, paradoxically reduces the flow of data within an enterprise, and this shows up in practical applications for people every day. 10 Megabyte presentations, for example, are common today. At 1 Mb/sec, a 10 Megabyte presentation would take about 80 seconds to download; at 1 Gb/sec, its delivery would be effectively instantaneous.

In sum, the Cloud makes IT services much more dependent on the network -- while dramatically increasing the delay and reducing the capacity of the network to deliver those services.

There is some solution in increasing network capacity and reducing delays, but there are strict limits to both. The speed of light (200 km/msec in fiber) forms a very hard limit on the latter. It is 4000 km coast-to-coast in the US, and network paths rarely follow straight lines all the way across the continent; the practical distance is more like 6000 km. This is 30 msec, one way, even if there are no delays due to queuing or repeaters. As for increasing network capacity, while there are no fundamental physical limits to capacity across the wide area, there is a fundamental tradeoff between the length of a fiber run and its bandwidth, generally expressed as the material's *bandwidth-distance product*.[Senior2008] For this reason, high-bandwidth is expensive across the wide area. The very high-performance networks in the US and Europe (Geant, ESNet, LHC Net, NORDUNet, etc) all have much less bandwidth than a single modern edge switch.[Geant2014]

The expense and limits in bandwidth shows up in all sorts of ways. Over 6 billion hours of YouTube video are watched per month[YouTube2014]. YouTube video requires anywhere between 400 kb/s to 20Mb/s, depending on desired resolution[Whistleout2014]. There are 720 hours in a month, so 6 billion viewing hours works out to about 8 million videos being viewed simultaneously at any time. If we assume a low resolution and average bandwidth of 1 Mb/s, this gives a demanded bandwidth, for this service alone, of over 8 Tb/s every second of every day. To return to our water-faucet analogy, this should be visualized as a pipe about 60 m in diameter, or, put in human terms, a pipe the width of a football pitch. This is, as might be expected, prohibitively expensive at commercial pricing. Indeed, in 2009 Credit Suisse estimated that YouTube was losing almost US$470 million per annum, largely due to bandwidth costs [Manjoo2009]. These pipes are very rare in the physical world, and they are quite rare in the cyberworld, as well. In fact, Google uses a form of distributed cloud – the Google Global Cache[Zink2008]. Between the Google Global Cache and a number of commercial arrangements, Google's actual bandwidth costs have been estimated to be nearly zero[Singel2009].

Bandwidth expenses are significant for enterprise cloud customers, as well. Simple experiments with Amazon's cost calculator showed that even with modest bandwidth requirements (10 Mb/s-100 Mb/s), bandwidth costs dominated cloud deployment costs, forming between 75% and 99% of the average bill.[1]

Moreover, this mismatch between bandwidth supply and demand will grow, and not shrink, over time, so long as the current network and cloud architecture remains unchanged. Bandwidth demand is driven by the availability of electronic content, and this in turn is driven by the capability of endpoint devices (displays, etc) and sensors. The capacities of endpoint devices of all sorts are doubling every year, and the number of such devices is on a dramatic and increasing curve. Today, a simple cellphone camera can easily saturate not simply the phone's, but the home's internet connection, even with aggressive compression. Similar calculations can be made for programs, disks, and so on. Moreover, this discrepancy can be expected to grow over time. Network devices and endpoint devices improve on the same fundamental doubling curve; however, the network is a linear infrastructure, whereas endpoint devices are point products. The latter improves much faster than the former; the reason is that a point product improves in isolation (if a person buys a faster computer, his computations speed up), whereas linear

[1] Available at http://calculator.s3.amazonaws.com/index.html

infrastructures improve in concert. This is why 5G, proven in the lab, will not be deployed until 2020 [Garner2014]; there are lots of cell towers to upgrade.

The *only* thing which will reduce latency and bandwidth demand is ubiquitous computation. The effect of programs is to reduce data; the essence of a program is to extract information from a large amount of data. The general class of application for the Distributed Cloud is using "locavore computation", in US Ignite CTO Glenn Ricart's apt phrase, to reduce bandwidth and latency demands on the network.

Applications of the Distributed Cloud

In this section, we explore applications of the Distributed Cloud which fit the general class described above: sending programs to the users, or programs to the data, in order to enhance useability and reduce resource demands on the network.

A somewhat artificial, but instructive example, is the visual representation of the Mandlebrot set. A video of exploration of this set is of arbitrary size, depending on the region being transmitted, the scale factor, and the level of detail encoded. However, the program to generate the video is of trivial size, and almost any reasonable implementation will consume no more than a few hundreds of kilobytes. For example, David Madore's popular set of explorations of the Mandlebrot set [2] are about 200 MB in size, compressed; the single program to generate them (and many more) is under 500 KB, a compression factor of 400:1. While we wouldn't build an infrastructure to enable efficient transmission of visualizations of Mandlebrot or Julia sets, these are simple examples of a very widespread phenomenon: complex, rich visualizations generated by simple programs. There are many similar examples in the physical sciences, primarily simulations of physical phenomena, from particle physics to cosmology to genomics to molecular biology to advanced chemistry and materials engineering to meteorology and earth science -- this is a partial enumeration of a very long list. The same is true in entertainment; in particular, machinimas are an early example of what could be *in situ* rendering of entertainment videos.

A second compelling application is the reduction *in situ* of sensor data. There are a wealth of applications which are enabled by deployment of large scale sensor suites; these are variously described as Smart Cities, Internet of Things, and the Ubiquitous Web[Weiser1995, Hui2008, Mattern2010]. All amount to essentially the same idea: using ubiquitous sensors to gather real-time information about objects and environments, then using computation to optimize various environmental parameters. In some cases, networked actuators tune devices to effect an automatic response.

This is of course well-known and widely-hyped. It's also one of the rare cases where the hype genuinely understates the effect. The transformation wrought by the deployment of cheap, ubiquitous sensing coupled to computation will dwarf the changes that occurred because of the widespread adoption of personal computers and instantaneous communication. If one considers the economic effect of the "computer revolution" to date, it has primarily been to effect radical efficiencies in three areas: transmission of static information (e.g., dictionaries, encyclopedias, newspaper articles, etc); retail commercial transactions; and personal and enterprise clerical and administrative tasks. These are no small things, as witness the enormous transformation in human society which resulted from these innovations; a quantitative measure is that three of the

---

[2] http://www.madore.org/~david/math/mandelbrot.html

four most valuable companies on earth are Apple, Microsoft, and Google[Dullforce2014]. None of these companies existed 40 years ago, and all are enmeshed in this revolution.

For all of that, these are relatively minor areas of human activity. The Internet of Things promises to encompass a very large portion of human actions and enterprises, with correspondingly greater impact. There is literally no aspect of our environment or our lives which will remain unaffected by this combination of ubiquitous sensing combined with arbitrary computational resources. To take one simple, and small, example, Monsanto recently purchased the Climate Corporation for $1 billion. The Climate Corporation had accumulated 150 billion soil observations and coupled these to 10 trillion weather simulation points[Upbin2013]; Monsanto intends to use these data to choose the right seeds, planted at the right places, to increase crop yields. Monsanto estimates that maize yields in American fields can be increased by 30-50 bushels per field[Wigington2013].

The web of sensors, computation, and actuators that comrprise the Internet of Things entails significant network challenges. First, of course, there will be very many of them; ABI estimates that 30 billion devices will connect to the Internet of Things by 2020[ABI2013]. This is roughly an order of magnitude more than the number of connected devices today. Further, the sensors can be quite high-bandwidth, and many applications have real-time latency requirements. A specific example of a network of high-bandwidth sensors is the Collaborative Atmospheric Sensing Apparatus (CASA) project in the United States [Zink2009]. CASA's mission is precise observation and prediction of strongly localized severe weather events, notably tornadoes. It is difficult to predict tornadoes with precision today because there are relatively few weather radars, and thus the nearest weather radar is many tens of kilometers removed from the center of severe weather activity. Due to the curvature of the earth, the atmosphere in the storm area below several hundred meters in altitude is invisible to the radar - and it is in this region that tornadoes form.

CASA addresses this by deploying a large network of small weather radars, relatively closely spaced. If the grid spacing is 20 km, the effective floor on visibility is zero; one can see the entire atmosphere. The data from the radars is then sent to a supercomputing center to make accurate predictions of tornado activity. The ability to predict tornadoes accurately is a literal lifesaver: even a few minutes' warning can give residents time to evacuate or seek shelter. A specific application, NOWCasting,[Zink2011] offers customized weather reports on a tightly location-specific basis.

The critical piece for CASA is, unsurprisingly, the network. The output of each radar is the familiar weather-radar image; broadcasting those images from a radar grid of any reasonable size will easily overwhelm any conceivable network, and transmitting most data back to a processing center is pointless; most of the time most of the radars are looking at nothing of interest. Similarly, NOWCasting cannot rely on a standard hub-and-spoke network model; the report for a specific region shouldn't leave the region

What's needed for CASA is in-situ processing -- at least, preliminary processing -- of the output of the weather radars, to determine if the events warrant forwarding to the high-performance data-processing center, and localized processing to prepare and distribute local NOWCasting reports. Further, the network must be highly-reconfigurable on demand, in order to offer dedicated bandwidth to the processing center for radars capturing events of interest. For this reason, the CASA experiment leaders have embraced the GENI project in the US, which is a prototype of

the Distributed Cloud: a network of small Clouds at over 50 points-of-presence throughout the United States, connected by a nationwide software-defined layer-2 network. .[Krishnakappa2012] Another compelling and current application of the Distributed Cloud is Distributed Real Time Interactive Simulation (DRTIS).[Boyd2011, Smith2012]. This technology is moving far beyond its roots in video games to become an indispensable tool for distributed learning, educational assessment, and maintenance and troubleshooting[Freedberg2012, Quinn2012]. The applications of this platform class are certain to grow dramatically as its capabilities are explored. A specific driver of this platform class is the dramatically enhanced capabilities of HTML5. These capabilities permit DRTIS applications to be hosted in the Cloud and rendered in the browser, offering easy sharing and collaboration and easy, inexpensive construction of shared simulations. Much as the original HTML enabled true electronic document interchange by radically reducing the cost of document preparation and transmission, so HTML5 and the newly-developed tools which exploit its capabilities make the creation of this class of application newly practical. The US Departments of Labor and Education currently are administering a $2 billion program in using RTIS for educational assessments at the Community College (immediate post-secondary) level; the US Department of Defense has a smaller, but still substantial, pilot effort in its Advanced Distributed Learning (ADL) Lab.
There is a strong preference to host DRTIS systems in the Cloud to reduce dependence on client software, enforce Digital Rights Management, and control user access to information about simulation (for example, in an assessment scenario, one wants to ensure the assessee can't cheat). A significant barrier to DRTIS applications is network latency; user experience and system efficacy drops dramatically when network latencies increase beyond 20-30 msec. This was one of the principal reasons for the failure of OnLive, which attempted to host gaming applications in the Cloud. OnLive required a single physical machine for each player, and attempted to produce their own single-application distributed cloud with thousands of servers. [Hollister2012]Further, rich simulations often require substantial bandwidth from the Cloud host to the client.
The solution is to deploy the DRTIS application in the distributed Cloud, and US Ignite is currently partnering with the ADL Lab, Lockheed-Martin's Training and Simulation division, and SAP America Labs' Communication Design Group to deploy a demonstration DRTIS system this year. This deployment was successfully demonstrated at the US Ignite Application Summit in July 2014, and will be field-tested in an educational setting in September 2014. The system, the Distributed Virtual Worlds Framework, uses replicated computation to ensure that participants in a simulation see perfectly consistent simulation, while minimizing required bandwidth throughout the simulation.

Distributed Cloud

In the previous section, we outlined a number of existing and emergent applications which require distributed Clouds. For this reason, it's unsurprising that a number of distributed Clouds have begun to emerge, particularly in the US and EU research communities. The first such distributed Cloud was the PlanetLab Consortium, a joint venture of Intel, HP, Princeton University, the University of California at Berkeley, and the University of Washington[Peterson2002]. PlanetLab eventually spread to more than 1,000 nodes at over 300 sites worldwide. As it grew, distributed and decentralized management became critical, and the

European nodes in PlanetLab became independently managed as ONELab[Antoniadis2010]. ONELab and PlanetLab then formed the first federated, distributed Cloud, with independent management, separate user bases and resources, and agreements and tools which permitted users of each federate to acquire and use resources across the federation.

In some sense, the Distributed Cloud is nothing complex: it is simply a standard Infrastructure-as-a-Service platform like EC2, Rackspace, HP Cloud Services, and so on, but distributed across the wide area, optionally with deeply programmable networking between sites. In a similar sense, the World Wide Web was nothing complex: it was simply a protocol by which one computer could send a file to another (HTTP) and a standard format for documents (HTML). However, the surface simplicity masked many complex details of implementation, which have given rise to a very large industry in web technologies.

Similarly, the apparent simplicity of the Distributed Cloud masks significant complexities in implementation. The most important of these is authentication and authorization. Centralized clouds generally are each a single administrative domain. While it's possible that a few large commercial Distributed Clouds will emerge (consider Akamai), the more likely scenario is something akin to the Web: the Distributed Cloud will emerge as an *ad hoc* federation of many individual Clouds. This is the true Intercloud: much as the Internet emerged as a federation of independent networks, so the Intercloud, or Distributed Cloud, is likely to emerge as a federation of many independent, localized Clouds. Sheer economics dictates this outcome: it would be hideously expensive for a single entity to build a globe-spanning distributed Cloud, much as it would have been hideously expensive for a single entity to build a globe-spanning distributed electronic document interchange system. It is, on the other hand, extremely cheap for such a system to grow bottom-up from existing systems. And in IT, cheap means easy means this is the way it will happen. All that is needed is a small number of overarching building-block concepts, the Cloud equivalent of HTTP and HTML.

Existing and Emergent Distributed Clouds

PlanetLab and ONELab invented many of the concepts that have become standard in the emerging Distributed Cloud community. The key concept is the "Slice": a virtual network of virtual machines, with specific locations for each of the virtual machines[Peterson2002]. The critical distinction between a Slice and the more generic Cloud concept of a collection of Virtual Machines, or "instances", is that each virtual machine (or "sliver") in a slice has a specific location. And this leads to the key difference between the centralized cloud of today and the distributed cloud of tomorrow. In the centralized Cloud of today, the focus is on scalability: how can we take an essentially trivial service (e.g., Twitter) and scale it to millions of simultaneous users? But in the Distributed Cloud, the focus is on distribution and placement of VMs, and control of network between them. How can we appropriately place Virtual Worlds slivers so that every user on the planet is within 20 milliseconds of a sliver? How can we control the network so that updates between DVW hosts consume no more than 50 milliseconds? This is not to argue that a centralized cloud is antithetical to a Distributed Cloud. Far from it; these are complementary technologies and future systems will incorporate both. Indeed, the CASA experiment already does this. A centralized Cloud is used for heavy processing of events of interest: a distributed Cloud is used to filter events and for localized weather forecasting.

Other examples of the Distributed Cloud are emerging in the academic, commercial, and enterprise spaces. Indeed, the Distributed Cloud is a unifying overlay on a number of extant services, much as, in the early 1990's, the World Wide Web emerged as an overlay on a number of extant network services (FTP, Gopher, Archie, WAIS). As always, these services have been developed as ad-hoc solutions to point problems, and the new paradigm emerges as a unifying theme over the various individual systems.

One of the original Distributed Clouds, PlanetLab, grew as an institutionalization of then-common practice among computer science systems researchers. In order to test a distributed system (Content Distribution Network, wide-area store, multicast tree, etc) in the field, researchers had to call their colleagues at other institutions and request accounts on their local servers, then deploy their system to those servers; moreover, those servers were generally heterogeneous, requiring individualized specialization to each environment. The result was that it was much more time and effort to deploy an experiment than it was to design the system or actually run the experiment and analyze the data, and there was no way for another researcher to duplicate those results. The obvious solution was for each institution to set aside a small number of servers to run distributed systems experiments, agree on a common hardware and software platform to eliminate location-specific deployment issues, and put these servers on a common, cross-institutional administrative framework. This was PlanetLab, which has served as a workhorse testbed and deployment platform for more than a decade, and has served as an inspiration and model for a number of wide-area testbed and deployment platforms across the world, including the US GENI initiative, the EU FIRE initiative, Germany's GLab, and the EU's ONELab (which originally started as part of PlanetLab).

Some of the other systems and concepts that are examples of Distributed Clouds are: Content Distribution Networks, such as Akamai and LimeWire, among many others. These are the oldest examples of this general paradigm, appearing shortly after the commercialization of the Internet in the mid-nineties.

In a similar fashion, we expect that the Distributed Cloud will continue to arise on an *ad hoc* basis as pragmatic solutions to enterprise and other needs. Some emergent examples are already emerging in both the public and private spheres.

The "Science DMZ" is an IT construct popular on US University campuses for the movement of large science data[Dart2011, Dart2012]. Its classical instantiation is as a large File Transfer Protocol (FTP) server bank installed outside the institution's firewall. This permits large science users to employ efficient data transfer processes without the restrictions that the firewall imposes. R. Ricci of the University of Utah and S. Corbato of the Utah Education Network have observed that this is an ideal place to create virtual machines to host computation from other sites, since it reduces the need for massive data transfer and imposes no risk on the host institution, and have implemented a Distributed Rack Node, federated with the NSF Global Environment for Network Innovations (GENI) the University of Utah. The concept, called "Science Slices" is in the next generation of Science DMZ at Utah[Corbato2014]

Many US campuses are now adopting an internal Infrastructure-as-a-Service Cloud to consolidate research computing. It's long been observed that many on-campus researchers put computers in labs and offices, in difficult-to-manage and maintain locations, for the convenience of administering their own research infrastructure. IaaS offers researchers the ability to do this much more easily and cheaply, and free up space in their labs. This approach, which is essentially a campus version of Amazon AWS, is generically known as "computing Condos". J. Bottum of Clemson University has proposed a multi-campus version, which he calls the "Condo of Condos"[Bottum2013]; this would not only permit efficient use of IT resources, it is a ready-made Distributed Cloud, since it would necessarily involve opening on-campus network resources to an off-campus community

Akamai is offering a limited Distributed Cloud with their Edge computing platform, a Java2 Enterprise Edition Platform-as-a-Service wherever Akamai has a content delivery data center. This is a natural extension to Akamai's core business of serving up web pages, because a large number of web pages are served dynamically by Java2-based services. This technology base has now broadened considerably, and one expects that Akamai will broaden their services accordingly.

A number of large-scale application providers (notably Google, Facebook and Microsoft) run internal distributed clouds over a large number of data centers to efficiently serve a worldwide customer base. While details of the operation of these Clouds are confidential, enough information has been revealed to conclude that they share the essential features of the public distributed clouds discussed here: virtual machines spread across the wide area, interconnected by deeply-programmable private networks. We discuss one such public case study in the next subsection.

Distributed Clouds as Research Testbeds

The most prominent Distributed Cloud today is the US National Science Foundation's Global Environment for Network Innovations, or GENI, and it acts as a case study in the features of future Distributed Clouds[Berman2014]. GENI today consists of 50 small clouds distributed at Universities and research network providers across the United States. A map of the current GENI network is shown in Figure 1

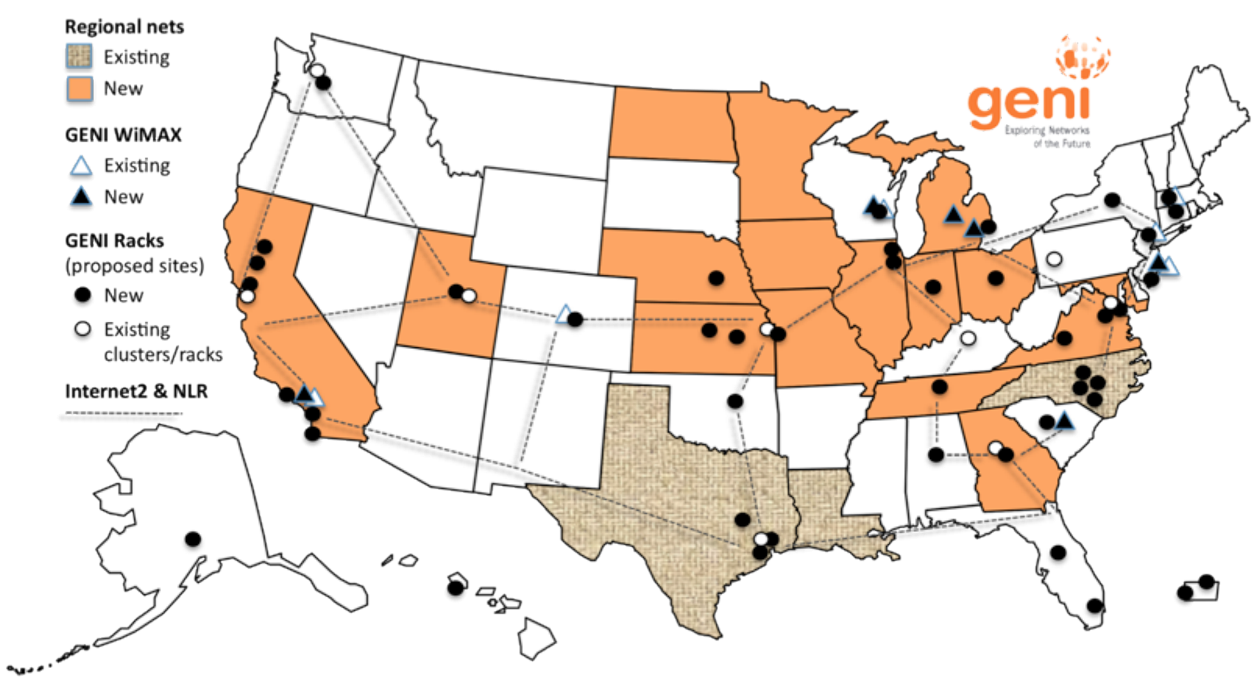

Figure 1: Current GENI Network

A GENI node consists of a “rack”: a Cloud with 80-160 or so cores, and a few terabytes of storage. Each rack is capable of supporting well over 100 virtual machines, even given the limited hardware deployment on a rack. The racks are interconnected by dedicated “Layer-2 networks”. In layman’s terms, this means that, in terms of network protocols, the entire facility looks like a single data center. The network itself is programmable on a “slice”, or application-level, basis. This means that someone using the GENI Cloud not only has an opportunity to allocate Virtual Machines anywhere on the GENI network, but also specify precisely how they are interconnected, which traffic should be prioritized, and other network functions. Moreover, the specific networking technology used in the GENI infrastructure, the OpenFlow protocol developed at Stanford and now maintained by the Open Networking Foundation, permits applications to reprogram the network during the course of the application.

This ability to program the network -- essentially bringing networking resources under the same level of application control as computation and storage -- is a key feature of the Distributed Cloud, and indeed of all distributed applications going forward. Until now, applications have generally run on a single computer, or at least within a collection of computers within the same building. Networking concerns emerged, but were far from dominant. Today, networking concerns dominate application performance, and a new level of control over the network is needed. In particular, modern applications require bandwidth and latency guarantees between application hosting sites, admission control to a private, application-specific network, and control of traffic routing. This is very difficult to obtain on the current network, since it is application-agnostic. What is specifically required is the ability to program the network to recognize traffic for a specific application and apply special-purpose rules to achieve the application’s bandwidth and latency needs, admission control and routing constraints.

This has been achieved by the OpenFlow protocol[McKeown2008]. Under OpenFlow, each network device operates under the governance of a network-wide centralized controller, which instructs each switch how to forward and handle each packet. The devices can be reprogrammed by the controller during the course of the application, which permits the network to exhibit dynamic behavior on an application-specific basis.

This is a gross oversimplification of OpenFlow, and programmable networks in general. There are many added features and concerns, to do with efficiency of implementation, isolation of application-specific networks, and security. These are areas of active investigation. However, OpenFlow to date has been successful enough that Google and Microsoft [McGillicuddy2013, McGillicuddy2013a] have announced that their internal Cloud networks run on OpenFlow, and most major telephone network operators are planning trial or operational deployments on their networks.

It seems clear that every future Distributed Cloud will follow GENI and make programmable networking an option, at least, for its users.

It should be noted that the computational resources at any site on the GENI network are somewhat limited; any specific application will only be able to command a few virtual machines at any site in this shared environment. This is a key difference between centralized and distributed cloud: in a distributed cloud the focus is on distribution and placement of VMs, and control of network between them, not scalability.

GENI is one of a number of research testbeds around the world which serve as exemplars of the Distributed Cloud. We have already mentioned OneLab in Europe, an offshoot of PlanetLab.

Canada has a nationwide research network, Smart Applications on Virtual Infrastructures (SAVI)[Leon-Garcia2010], which is a network of OpenStack installations with programmable networking internally and between sites. JGN-X in Japan features a strongly GENI-like infrastructure, with a locally-designed GENI Rack, the V-Node[Nakao2012]. The German G-Lab was a multisite, highly successful Distributed Cloud infrastructure[Mueller2010]; most of the sites remain active, though the project itself has wound up. The EU Future Internet Research (FIRE) initiative spawned a number of active research testbeds. Under the leadership of B. Vermuelen of iMinds, these have been federated into a Distributed Cloud under the FED4FIRE initiative[Wauters2014].

Applications of the Distributed Cloud

As with most infrastructures, the compelling services and applications enabled by the Distributed Cloud will only become apparent after the Distributed Cloud has emerged. No one predicted email, blogs, Google, Facebook, Amazon, and Twitter before or at the emergence of the Web. Fundamentally, the construction of infrastructure before the emergence of applications is a bet on human ingenuity: give people a tool and they'll use it in ways that can't be imagined in advance. While one doesn't know the specifics, the history of innovation in infrastructure enables one to say with great confidence that people will build services that, in retrospect, will appear indispensable.

While one can't identify with precision the compelling services that will be enabled by the Distributed Cloud (or, more precisely, the few that can will be very handsomely rewarded for their perspicacity) from the strengths of the Distributed Cloud one can deduce their general characteristics. Essentially, the Distributed Cloud permits computation to move closer to data and to users, both by physical movement of computation and by using software-defined networking technologies such as OpenFlow to optimize network transit for specific applications and services. The single most compelling class of service that requires in-situ computation is analysis of the output of high-bandwidth sensors (e.g., cameras). The "Internet of Things", the catchall phrase for ubiquitous networked sensors and actuators, promises to make the human environment far more responsive, efficient, and dynamic. However, this promise is reliant on software analysis of the output of networked sensors; even trivial back-of-the-envelope calculations will show that the current Internet cannot meet the demand implied by any reasonable network of sensors. Further, no evolutionary extrapolation of current Internet technologies can meet this demand. Attempting to meet the demands of the Internet of things simply by laying more fiber is exactly analogous to meeting the demands of a growing number of automobiles and people by building more roads. Ultimately, it is a lot easier and cheaper to build automobiles than it is to build roads, and any road-building program will soon be overwhelmed; similarly, it's easier and cheaper to build sensors than it is to lay the fiber to carry their output. In both cases fundamental changes to the architecture are required, and the Distributed Cloud is the emergent solution in the case of the Internet.

A second compelling class of application is collaboration around Big Data. This obviously intersects with the Internet of Things, because this worldwide network of sensors and actuators will be the primary source of Big Data. However, it is far from the only source. Many others are currently in existence. The most obvious are large-scale scientific instruments, including particle accelerators, large telescopes, medical imaging instruments, genomic sensors, and so on. Further, and particularly susceptible to optimization through the Distributed Cloud, are computational sources of Big Data. Simulators of all sorts are the most obvious example, but there are many others. D. Lary of the University of Texas at Dallas, with R. McGeer and G. Ricart of US Ignite, have developed the concept of the "virtual sensor". The intuition behind the virtual sensor is that measurements of specific physical quantities can be inferred from a combination of other measurements. This is particularly important when a physical sensor for the inferred quantity cannot be deployed for cost or logistical reasons. A specific example is an air pollution monitor, which must measure particulates of several microns in diameter. Such fine sensing is extremely expensive, and so cannot be ubiquitously deployed. However, it can be deployed in a number of laboratory and field stations, and its output correlated with the output of a wide battery of ubiquitously-deployed, cheap sensors (temperature and pressure sensors, satellite images, and so forth). Once the correlations produce a model of high fidelity, the model can be used to predict what the physical sensor would have output.

Collaboration around Big Data is compelling because the data must be analyzed by a number of specialists, many of whom are not physically co-located. This has been a significant problem for all scientific communities for a number of years, and this will grow as the volumes of data that all fields generate grows.

A Regulatory and Legal Environment

Many of the obstacles to the development of the Distributed Cloud are regulatory and legal in nature, and therefore are amenable to modifications in the regulatory framework. Part of this is simply assumptions: historically, this field is used to accounting for computation and storage, whereas data transmission is "free"; it isn't allocated by the job. This makes less sense now that data transmission is the dominant cost in large IT jobs than it did when the current accounting models were put in place. But another part is related to liability. At the moment, the owner of IT equipment are liable for any damage to third parties that a tenant does using his equipment, unless that tenant has explicitly agreed to indemnify the owner. Such indemnification is a routine part of the agreements that users of others' IT equipment routinely sign, on everything from wireless networks to Clouds. This method of making an explicit agreement between an owner and tenant assumes, however, that there are relatively few IT owners, so explicit agreements can be made. This is a logistical challenge in the case where there is a large number of IT owners and tenants. Permitting agreements to be made, and, in particular, assignment of liability in the absence of explicit contracts, is a function of regulation, and an area where careful and enlightened regulation can make a positive difference. It is a popular fallacy that an absence of regulation is a possibility. So long as liability attaches, there will be regulation. The sole question is whether the regulation will be done by a body tasked with regulating this specific set of activities, or on an ad-hoc basis by the courts.

Such a regulatory environment must encourage innovation, particularly in the network space, while ensuring transparency and equal access. This is a difficult balance, as can be seen from the current debate on "network neutrality" in the United States. "Net neutrality" is essentially a prohibition on over-the-top service providers purchasing downstream bandwidth from telephony providers. At first blush, there would seem to be no reason why an over-the-top provider shouldn't be able to rent a chunk of the network as he does a VM today. The concern of net neutrality proponents are essentially twofold: that the bandwidth has already been sold to the subscriber, who would prefer to choose his own priorities on services; and, second, that bandwidth will not be sold to upstream providers on a transparent, open, and equal basis. It is not clear that the current model -- that all traffic must be passed on an equal basis -- is the only or the correct one. It is clear that if last-mile bandwidth is to be sold to upstream providers, it must be done on a basis that is transparent, open, and ensures equal access to all comers. Pricing must be open and transparent, and not favor large scale, or pricing the network will lead to dominance by a few large players and discourage innovation in service delivery.

Conclusion

The Distributed Cloud offers enormous efficiencies to the operation of the network, and will prove vital for the large data collaborations that are the defining characteristics of the next generation of the Internet. Prototype distributed clouds are already emerging, for both special purpose applications and as testbeds in the US, Europe, Canada, and Japan. Significant progress will be made on the infrastructure, services, and applications front in the next few years, and in particular significant integration with software-defined networking technologies.

Acknowledgements

The author wishes to thank Glenn Ricart and Patrick Scaglia for numerous conversations around these topics and for insightful comments on early drafts of this manuscript.

References

[ABI2013] ABI. More Than 30 Billion Devices Will Wirelessly Connect to the Internet of Everything in 2020. At: https://www.abiresearch.com/press/more-than-30-billion-devices-will-wirelessly-conne. Retrieved July 2014

[Atoniadis2010]. P. Antoniadis, S. Fdida, T. Friedman and V. Misra. Federation of virtualized infrastructures: sharing the value of diversity. ACM CoNEXT, Philadelphia (USA): December 2010

[Baron2012] PLoïc Baron, Jordan Augé, Timur Friedman, Serge Fdida. Towards an integrated portal for networking testbed federation: an open platform approach*, FIRE Engineering Workshop* -- November 6-7, 2012 -- Ghent, Belgium

[Berman2014] Mark Berman, Jeffrey S. Chase, Lawrence Landweber, Akihiro Nakao, Max Ott, Dipankar Raychaudhuri, Robert Ricci, Ivan Seskar GENI: A federated testbed for innovative network experiments. *Computer Networks, Vol 61, 14*. March 2014, pp 5-23

[Bottum2013] James B. Bottum, Ruth Marinshaw, Henry Neeman, James Pepin, and J. Barr von Oehsen. 2013. The condo of condos. In *Proceedings of the Conference on Extreme Science and Engineering Discovery Environment: Gateway to Discovery* (XSEDE '13). ACM, New York, NY, USA, , Article 62 , 2 pages. DOI=10.1145/2484762.2484775 http://doi.acm.org/10.1145/2484762.2484775

[Boyd2011]. Richard Boyd and David A. Smith, The Virtual World Ecosystem Framework. *Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2011.* Retrieved at:

http://ncnewschools.org/2012stemconference/CONF_PDFs/The_Virtual_World_Ecosystem_Framework.pdf, July 2014
[Corbato 2014] Steve Corbato. Perspectives on campus and regional cyberinfrastructure development. *Broadening the Reach Workshop*, Salt lake City, UT. At http://www.internet2.edu/presentations/btr201405/20140506-corbato-Campus%20Infrastructure.pdf . Retrieved July 2014.
[Dart2011] Dart, Eli; Metzger, Joe (June 13, 2011). The Science DMZ. *CERN LHCOPN/LHCONE* workshop
[Dart2012] Dart, Eli; Rotman, Lauren (Aug 2012). The Science DMZ: A Network Architecture for Big Data. *LBNL-report*
[Dullforce2014]. Dullforce, Anne-Britt. The Global 500 2014. In *The Financial Times*, March 2014. Available at: http://www.ft.com/intl/cms/s/0/988051be-fdee-11e3-bd0e-00144feab7de.html#axzz37ytPw9kE. Retrieved July, 2014.
[Freedberg 2012]. Sydney Freedberg. Army Goes World of Warcraft; Virtual Training To Soften Budget Cut Pain. *Breaking Defense*, February, 2012. Retrieved at http://breakingdefense.com/2012/02/army-goes-world-of-warcraft-will-emphasize-virtual-training-be/, July 2014.
[Garner2014] Garner, Jamie. EU plans to fast track 5G rollout. July 3, 2014. At: http://www.daisygroupplc.com/2014/07/eu-plans-fast-track-5g-rollout/
[Geant2014] http://www.geant.net/Network/The_Network/Pages/default.aspx. Retrieved July 2014.
[Hollister2012]. Sean Hollister. OnLive lost: how the paradise of streaming games was undone by one man's ego. *The Verge*, August 2012. http://www.theverge.com/2012/8/28/3274739/onlive-report. Retrieved July, 2014
[Hui2008] Hui, J., Culler, D.: IP is Dead, Long Live IP for Wireless Sensor Networks. *Proc. 6th Int. Conf. on Embedded Networked Sensor Systems (SenSys), pp. 15–28 (2008)*
[Krishnappa2012] Dilip Kumar Krishnappa, Eric Lyons, David Irwin, and Michael Zink, Network Capabilities of Cloud Services for a Real Time Scientific Application. In *Proceedings of the 2012 IEEE Conference on Local Computer Networks (LCN)*, Clearwater, FL, USA, October 2012
[Krishnappa2012a]Dilip Kumar Krishnappa, Eric Lyons, David Irwin, and Michael Zink, CloudCast: Cloud Computing for Short-term Mobile Weather Forecasts. In *Proceedings of the IEEE IPCCC 2012, Austin, TX, USA, December 2012.*
[Leon-Garcia2010] A. Leon-Garcia, Smart Applications on Virtual Infrastructures, *Future Internet Canada-EU Workshop*. Waterloo, Ont., March 2010.
[Manjoo2009] Fahrad Manjoo. Do You Think Bandwidth Grows on Trees? *Slate*, April 14, 2009. At: http://www.slate.com/articles/technology/technology/2009/04/do_you_think_bandwidth_grows_on_trees.html . Retrieved July, 2014
[Mattern2010] Mattern, Friedemann; Christian Floerkemeier (2010). From the Internet of Computers to the Internet of Things. *Informatik- Spektrum* **33** (2): 107–121. Retrieved 3 February 2014.
[McGillicuddy2013] Shamus McGillicuddy. Vint Cerf: At Google, OpenFlow Now Runs In All Data Center Networks. *TechTarget*. 17 April 2013. Available at: http://searchsdn.techtarget.com/news/2240181909/Vint-Cerf-At-Google-OpenFlow-now-runs-in-all-data-center-networks
[McGillicuddy2013a] Shamus McGillicuddy. Microsoft uses OpenFlow SDN for network monitoring and analysis. *TechTarget*. 17 April 2013. Available at: http://searchsdn.techtarget.com/news/2240181908/Microsoft-uses-OpenFlow-SDN-for-network-monitoring-and-analysis

[McKeown2008] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. 2008. OpenFlow: enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev. 38, 2* (March 2008)
[Mueller2010] P. Mueller. G-Lab: The German Approach Towards Future Internet Research. *7th GENI Engineering Conference*. At: http://groups.geni.net/geni/raw-attachment/wiki/Gec7Agenda/G-Lab%20Project%20GEC7.pdf
[Nakao2012] A. Nakao. Vnode: A Deeply Programmable Network Testbed Through Network Virtualization. *Proceedings IEICE*, At: http://www.ieice.org/~nv/05-nv20120302-nakao.pdf
[Peterson2002]. L. Peterson, T. Anderson, D. Culler, and T. Roscoe . A Blueprint for Introducing Disruptive Technology into the Internet.. *(HotNets-I '02),* October 2002
,
[Quinn2012] Kristin Quinn. New Tool Aims to Slash Costs of Creating Virtual Worlds. In *Defense News*, January 27, 2012. At: http://www.defensenews.com/print/article/20120127/TSJ01/301270004/New-Tool-Aims-Slash-Costs-Creating-Virtual-Worlds, retrieved July, 2014.
[Senior2008] Senior, John. (2008). Optical Fiber Communications: Principles and Practice, 3rd ed. Prentice Hall. ISBN 978-0130326812
[Singel2009] Ryan Singel. YouTube's Bandwidth Bill Is Zero. Welcome to the New Net. *Wired*, October 10, 2009. At: http://www.wired.com/2009/10/youtube-bandwidth/. Retrieved July, 2014
[Smith2012] David Smith. The Virtual World Framework. *Dynamic Languages Symposium, SPLASH 2012*.
[Upbin2013] Upbin, Bruce. Monsanto Buys Climate Corp For $930 Million. In Forbes, October 2, 2013. At http://www.forbes.com/sites/bruceupbin/2013/10/02/monsanto-buys-climate-corp-for-930-million/, retrieved July 2014.
[Wauters2014] Tim Wauters, Brecht Vermeulen, Wim Vandenberghe, Piet Demeester, Steve Taylor, Loïc Baron, Mikhail Smirnov, Yahya Al-Hazmi, Alexander Willner, Mark Sawyer, David Margery, Thierry Rakotoarivelo, Felicia Lobillo Vilela, Donatos Stavropoulos, Chrysa Papagianni, Frederic Francois, Carlos Bermudo, Anastasius Gavras, Dai Davies, Jorge Lanza, Sueng-Yong Park. Federation of Internet experimentation facilities: architecture and implementation. *Proceedings of the European Conference on Networks and Communications,* 2014.
[Weiser1995] Weiser, M.: The Computer for the 21st Century. *Scientific American 265(9):66–75* (1991)
[WhistleOut2014]. Whistleout.com. Broadband Usage Guide. Available at: http://www.whistleout.com.au/Broadband/Guides/Broadband-Usage-Guide. Retrieved July 2014.
[Wigington2013] Wigington, Dane. Monsanto Buys Weather Company Climate Corporation. *GeoEngineering Watch*, October 10, 2013. At http://www.geoengineeringwatch.org/monsanto-buys-weather-company-climate-corporation/, Retrieved July 2014
[YouTube2014] YouTube. YouTube Statistics. Available at: https://www.youtube.com/yt/press/statistics.html. Retrieved July, 2014
[Zink2008] Michael Zink, Kyoungwon Suh, Yu Gu and Jim Kurose. Watch Global, Cache Local: YouTube Network Traffic at aCampus Network - Measurements and Implications. 2008. *Computer Science Department Faculty Publication Series, Paper 177*. Retrieved at http://scholarworks.umass.edu/cgi/viewcontent.cgi?article=1178&context=cs_faculty_pubs, July 2014
[Zink2009] Michael Zink, Eric Lyons, David Westbrook, Jim Kurose, and David Pepyne, Closed-loop Architecture for Distributed Collaborative Adaptive Sensing of the Atmosphere: Meteorological Command & Control. *International Journal for Sensor Networks (IJSNET)*(Vol.6 No.4). 2009

[Zink2011] Michael Zink. Nowcasting: UMass/CASA Weather Radar Demonstration. At *Euroview 2011*, August, 2011. Available at:
http://www.euroview2011.com/fileadmin/content/euroview2011/slides/slides_distribution_zink.pdf
Retrieved July 2014.