NSF Workshop Report on

Grand Challenges in Edge Computing

October 26, 2016

Washington, DC

Sponsored by

National Science Foundation

NSF Edge Computing Workshop Committee

Workshop Co-Chairs

- Mung Chiang, Princeton University Edge Lab
- Weisong Shi, Wayne State University

Steering Committee

- Helder Antunes, Cisco
- Victor Bahl, Microsoft Research
- Mung Chiang, Princeton University
- Bruce Maggs, Akamai Technologies/Duke University
- John Smee, Qualcomm
- Weisong Shi, Wayne State University

Breakout Session Leads

- Applications
    - Chunming Qiao, University at Buffalo (SUNY)
    - Athina Makopoulouo, University of California Irvine
- Architecture
    - Fred Douglis, Dell EMC
    - Lin Zhong, Rice University
- Capabilities/Services
    - Arun Iyengar, IBM Research
    - Steven Ko, University at Buffalo (SUNY)
    - Tian Lan, George Washington University

Executive Summary

The Internet has witnessed two radical changes in the past decade: rapidly growing cloud computing and pervasive mobile devices, sensors and Internet of Everything (IoE). Cloud Computing, an alternative to the traditional model of owning and managing private resources by customers, provided centralized computing services and pay-as-you-use convenience to the clients. However, due to the often unpredictable network latency, expensive bandwidth, and privacy concerns, especially in a mobile environment, cloud computing often cannot meet the stringent requirements of applications, for example latency-sensitive, security/privacy-sensitive, or geographically constrained applications. The growing amount of data generated by end devices, things and systems can also become impractical or resource-prohibitive to transport over networks to remote clouds.

To this end, "Edge computing" is a paradigm in which the resources for communication, computation, control and storage are placed at the edge of the Internet, in close proximity to mobile devices, sensors, actuators, connected things and end users.

New challenges and opportunities arise as the consolidation of cloud computing meets the dispersion of edge computing. On both data plane and control plane, the benefits of real-time capabilities, device-to-device and device-for-device communication, edge caching, client-centric control, and agile development need to be realized through evolving interfaces between the edge and the core. The challenges of security, trust and incentivization, of potential instability and inconsistency due to autonomous agents, and of the tradeoff between local and global control must also be addressed.

This workshop brought together experts from academia, national laboratories, government, and industry to assess the vision, recent trends, state-of-the-art research, and impending challenges of the edge computing. This report serves as a collection of such input to government funding agencies and interested parties in industry and academia.

We identified the top five grand challenges in each of the three areas, represented through the parallel breakout sessions in this one day workshop: Applications, Architecture, and Capabilities and Services. Examples of grand challenge from each of these three lists include:

- Support real time applications, from edge analytics in embedded AI to feedback control in cyber physical systems, through reduced and almost deterministic latency afforded by edge networking.
- Develop a framework for decomposition of computation over heterogeneous and volatile computing substrate.
- Create edge service API definitions and sustain an edge service ecosystem with incentivized participation from many stakeholders.

I: Overview

The Internet has witnessed two radical changes in the past decade: rapidly growing cloud computing and pervasive mobile devices, sensors and Internet of Everything (IoE). Cloud Computing, an alternative to the traditional model of owning and managing private resources by customers, provided centralized computing services and pay-as-you-use convenience to the clients. While there are many emerging issues to be solved, Cloud Computing has reaped its field from enterprises to personal end users.

Meanwhile, mobile devices, such as smartphones and tablets, have become pervasive and are driving the development of many new applications, powered by the ever-improving wireless networking and mobility support. According to Cisco's conservative estimate, there will be 50 billion connected devices by 2020, forming an Internet of Things. Things across all industry domains, from transportation to healthcare to manufacturing to smart cities to smart grids, are being connected to address a growing range of needs for businesses and consumers. Enabling these future Internet of Things imposes unique challenges. For example, many devices will have limited battery power and processing capabilities, and hence cannot support computational-intensive tasks.

Motivated by these two trends, a plethora of research has been conducted to support mobile cloud computing, which bridges the Cloud and mobile devices by leveraging both the powerful computing capability of the Cloud and the mobility support of mobile devices. Much of the work has been focused on how to effectively offload the computation-intensive tasks to the cloud and gets the results back promptly. In this way, the battery of mobile devices can be effectively preserved. However, due to the often unpredictable network latency, expensive bandwidth, and privacy concerns, especially in a mobile environment, cloud computing often cannot meet the stringent requirements of applications, for example latency-sensitive, security/privacy-sensitive, or geographically constrained applications. The growing amount of data generated by end devices and systems can also become impractical or resource-prohibitive to transport over networks to remote clouds.

To this end, "edge computing" is a paradigm in which the resources for communication, computation, control and storage are placed at the edge of the Internet, in close proximity to mobile devices, sensors, actuators, connected things and end users.

There have been many interesting examples over the past decade: "cloudlets" and "micro data centers" have been designed as these small, edge-located data centers, "crowdsensing" has leveraged a collaborative crowd of devices to sense the environment, "edge analytics" has focused on mining streaming data right here and right away, and "edge control overlay" has explored alternatives to LTE core networks for network management, from smart data pricing to 5G HetNets control. "Edge computing" or "edge networking" refers to a broad collection of these and more directions, covering both mobile edge and wireline edge.

Recently, another term, “fog,” has also been used to denote architectures that use a collaborative multitude of client or edge devices to carry out storage, communication and management closer to the end users along the cloud-to-things continuum.

The edge of the Internet is a unique place. For example, located often just one wireless hop away from associated mobile devices, it offers ideal placement for low-latency offload infrastructure to support emerging applications such as augmented reality, wearable cognitive assistance and cyber physical systems. It can be an optimal site for aggregating, analyzing and distilling bandwidth-hungry sensor data from devices such as video cameras. In the Internet of Things, it offers a natural vantage point for organizational access control, privacy, administrative autonomy and responsive analytics. In vehicular systems, it marks the junction between the well-connected inner world of a moving vehicle and its tenuous reach into the cloud.

New challenges and opportunities arise as the consolidation of cloud computing meets the dispersion of edge computing. On both data plane and control plane, the benefits of real-time capabilities, device-to-device and device-for-device communication, edge caching, client-centric control, and agile development need to be realized through evolving interfaces between the edge and the core. The challenges of security, trust and incentivization, of potential instability and inconsistency due to autonomous agents, and of the tradeoff between local and global control must also be addressed.

This workshop brought together experts from academia, national laboratories, government, and industry to assess the vision, recent trends, state-of-the-art research, and impending challenges of the edge computing.

- Foster the edge computing community. Increase interaction between academia and industry.
- Set the vision and identify grand challenges and open problems.
- Identify collaboration mechanisms among academia, industry and government.

This report serves as a collection of such input to government funding agencies and interested parties in industry and academia. We identified the top five grand challenges in each of the three areas, represented through the parallel breakout sessions in this one day workshop: Applications, Architecture, and Capabilities and Services.

II: Applications

While it is fairly easy to say that a datacenter is not an edge computing device, neither is a simple sensor that does analog to digital conversion (e.g. a transducer) and collects/sends data, it is difficult to pinpoint exactly what constitutes an edge device.

An edge computing system combines computing and communication, and typically performs sensing and actuating as well. We tried to define edge computing from multiple perspectives:

- Network location point of view: typically, edge refers to the part of the network outside the Core network, but could include Metro/Access networks.
- Data point of view: edge computing typically is done at or close to where the sensory data is generated, and/or where the users/consumers of the data are.
- Control-decision point of view: typically, if all the processing is done centrally, then it is not edge processing, which inherently involves some distributed computing.

Edge computing works on both downstream data on behalf of cloud services and upstream data on behalf of IoT services. An edge device is any computing or networking resource residing between data sources and cloud-based datacenters. For example, an edge device could be a smartphone sitting between body sensors and the cloud, or an mDC or cloudlet between a mobile device and the cloud. In edge computing, the end device not only consumes data but also produces data. And at the network edge, devices not only request services and information from the cloud but also handle computing tasks—including processing, storage, caching, and load balancing—on data sent to and from the cloud. The edge must be designed well enough to handle such tasks efficiently, reliably, securely, and with privacy in mind. It thus must support requirements such as differentiation, extensibility, isolation, and reliability.

Edge computing could yield many benefits. For example, researchers have shown that using cloudlets to offload computing tasks for wearable cognitive-assistance systems improves response times by between 80 and 200 ms and reduces energy consumption by 30 to 40 percent. CloneCloud technology reduces response times and power usage by 95 percent for tested applications, in part via edge computing. SkyEye in networked drone cameras can completely eliminate frozen screens in high definition streaming of live outdoor sports events.

Application Examples

Characteristics of applications that can benefit from edge computing are typically time-critical and require a very low computing and communication latency in an environment with limited bandwidth, limited computing power, possibly lots of data to be processed or limited energy that powers the edge computing device.

Below are some examples of edge computing applications that were discussed.

- Vehicular applications that generate and send safety warnings/alerts, suggest driving speeds, routes and other driving behaviors to improve travel efficiency or reduce fuel

consumption and emission, and provides telemetry and in-vehicle entertainment services.

- Video surveillance apps for public safety: wherein distributed cameras collect and process data for face/object recognition with real-time processing (based on e.g. video analytics).
- Smart city applications: to monitor and manage buildings, infrastructures and governance processes, in addition to transportation and public safety.
- Mobile and wearable applications: wherein smartphones and wearable devices provide cognitive personal assistance information gathering and decision making.
- Disaster-relief apps: wherein mobile devices such as drones can aid search and rescue, or damage/infrastructure assessment.
- Bedside clinical apps which combine measurements from devices attached to the patient with contextual information (e.g., medical history, and diagnosis) to assist in fast medical intervention
- Games that incorporate location information and require real-time response: examples include VR/AR technologies for sports and training.
- Smartgrid and microgrid applications: where distributed monitoring, management and control of various renewable energy sources, currents, voltage, and demands can yield a more efficient and resilient system.
- Privacy-protection for cloud storage: clients shred a file into many pieces and spread them over multiple public cloud storage infrastructures.
- CDN in the sky: a lot of imagery is coming from hi-res satellites. These are bandwidth-constrained, with sensitive info, restricted mobility, and limited power. Thus satellites may offer an environment that stresses many of the issues raised by these grand challenges.
- Camera intelligence: Today, cameras deployed in cities could capture a missing child's image. However, this camera data usually is not uploaded to the cloud because of privacy concerns or the cost of transferring the information. With edge computing, the data could be pushed to the many edge devices in a target area. They could search the data they receive and report the findings to the cloud, yielding the results much faster than using only cloud computing.

Five Grand Challenges in Applications of Edge Computing and Networking:

1. **Real-time processing and communications**: while most edge computing applications require this, it is particularly challenging to achieve real-time processing for video analytics, and activities analysis, due to limited computing and communication abilities of edge computing devices.

2. **Security and privacy**: on one hand, since edge is close to data and users, there is an opportunity for stronger security and privacy. On the other hand, the sheer number of heterogeneous devices which need to process data from and for nearby users makes it challenging to achieve those goals.

3. **Incentives and Monetization**: unlike traditional networking services (e.g, middleboxes) or cloud services, edge computing services still needs better-developed business models, and incentive schemes to encourage users to adopt the edge computing apps.

4. **Adaptive application development**: Given the variety of the application scenarios, it is challenging to develop applications which can adapt to various environment and handle graceful degradation of performance/services.

5. **Tools for the development and testing of apps in edge computing**: since edge computing apps are still in their infancy, it is both important that the infrastructure provides provide tools (and potentially open standards) and services to facilitate the development of such apps.

III: Architecture

There will be computational resources available at various points between the end users and the massive data centers operated by infrastructure providers, such as Microsoft, Google and Amazon. These range from macro data centers at the regional and national level (run by the major cloud infrastructure providers) to mini data centers at the municipal and local level (operated by CDN and communication infrastructure providers) to micro data centers at the institutional level (run by corporations and institutes) and finally to personal servers such as smartphones and home routers (operated by end users).

These computational resources provide services not only in the form of computation but also storage and software. By using the intermediate computational resources, it is possible to reduce latency and network bandwidth usage. We also recognize that there are three orthogonal dimensions along which a computation resource can be placed: administration (who owns and manages it), security/trust from the administrator's perspective, and security/trust from the users' perspective. Recognizing these dimensions helps us clarify the grand challenges as will be elaborated below.

Five Grand Challenges in Architectures of Edge Computing and Networking:

1. **Cage-Level Security.** The paramount challenge toward computational resources out of the massive data centers under complete control of the operator is how to guarantee the same level of security and safety as those under complete control of the operator. For example, under extreme situations, the operator may lock a computer inside a physical cage and deny access to anyone without a key to the cage. Is it possible to achieve the same security guarantees as a physical cage via only hardware and software support? What are the implications of this level of security as the resource moves to entirely to the edge? The recent reports of DDoS attacks stemming from traffic cameras and other internet-connected things are a reminder of the vulnerability of anything accessible over the internet, but some of these things will be physically accessible to attackers.

2. **Embracing Approximation.** Much in the realm of edge computing will involve high-volume data. For instance, the amount of data needed for a city-wide video surveillance network can generate substantial requirements for both networking and storage. The legal requirements to store the video created by vest-mounted cameras for law enforcement is taxing municipal budgets. How can this data be reduced and how can processing be improved? There is a natural tension between bandwidth usage reduction and the usefulness of the resulting data. The more the data is compressed close to the source, the less data has to travel through the network, but the less useful the data may be for end applications. There is a strong need to look beyond what is available from existing data compression schemes. How can the usefulness of data be analytically modeled or represented? How can the timeliness of data be analytically

modeled or represented as some data may have a lifetime or shelf life. The answers are likely to be application-dependent. One possible approach to helping with efficient processing of high-volume data is self-descriptive data, also known as smart objects. Having the data "help" the system to do more efficient processing will improve scalability. One could imagine the smart objects knowing they should simply self-destruct because they're not important enough. Data processing at the edge is likely to introduce uncertainty in the data itself. The grandest of the grand challenges here is: How can this uncertainty be expressed in a probabilistic manner? Probabilistic computing is not yet very mature, though there has been significant work in the past 15-20 years on stream processing, which does best-effort processing on data as it flows through a system.

3. **A Theorem for Tradeoffs**. Brewer's "CAP Theorem" has been an important design principle for distributed systems. What are the fundamental properties and design principles about tradeoffs for edge computing? It was suggested that there may be a fundamental tradeoff between the following four factors, i.e. one might be able to achieve any three of them but not all four, or there may be pairs in direct opposition: (1) Mobility, (2) Latency, (3) Capability, and (4) Privacy. The conflicts may be such things as: large capability implies long latency and improving privacy dictates increasing latency. A "CAP theorem for Edge Computing" would be an interesting intellectual challenge and potentially have real influence.

4. **Data Provenance.** Provenance refers to how data is generated: the input sources, programs, users involved, and other factors. There has been a lot of work in the scientific computing community about tracking the provenance of data. Within the systems community there has been work on a provenance-aware storage system at Harvard and elsewhere. But provenance has never been tackled at scale. In the context of edge computing, there is a second related aspect, which is how data will be used. For instance, law enforcement will want to know who has viewed a particular surveillance video, even if the video hasn't been modified. Because the data is likely to go through the computational resources of various administrative domains and different levels of trust, provenance is further complicated. Additionally, the integrity of the provenance data itself is critically important.

5. **Enabling QoS on Network Edge**. As an application/service leverages computational resources available at various points between the end users and the massive data centers, how can end-to-end quality-of-service (QoS) be guaranteed? As these resources are likely to be owned and operated by different providers, how should responsibilities (and profits) be divided amongst them? New mechanisms are necessary to incentivize providers for cooperation and to incentivize application developers to efficiently utilize resources. For instance, without providers cooperating amongst themselves, if a certain level of end-to-end latency is desired, the application developer must reason about the

application's computational and communication requirements, negotiate an SLA (service-level agreement) separately with each of providers, and correspondingly map different parts of it to different computation resources so that the end-to-end latency requirement is met. The resulting application deployment can be not only expensive and over-provisioned, but also rigid and likely non-optimal.

6. **Testbed**. How can application developers develop applications that may eventually be deployed across multiple domains of different ownerships and levels of trust? We need an open cross-domain application development environment, with appropriate standards and security APIs. For researchers to fully explore edge computing, we need testbeds at scale.

IV: Capabilities and Services

This session focuses on capabilities and services that edge computing can or should provide. However, since capabilities and services do not exist in a vacuum and need computing resources, we first discussed the following two questions.

- What types of computing resources are there in edge computing?
- Who provides the resources?

The first question seeks to understand what kinds of computing resources are available to edge computing services. It also seeks to establish a common terminology that refers to various types of computing resources to avoid any confusion in our discussion. In the end, we have identified the following three types that can be used for edge computing.

- **Edge devices**: These are the devices that end users use and interact with. Examples include smartphones, tablets, wearables, laptops, IoT devices, and sensors.
- **Backend clouds**: These are the backends that service providers own and use. Ultimately, edge devices interact with these backends to provide services to end users. Examples include Google services, Microsoft services, Facebook, Amazon clouds, etc.
- **Edge infrastructure**: This is the edge infrastructure that sits between edge devices and backend clouds. Examples range from current ones to more futuristic ones. Current examples include extreme-edge computing resources while more futuristic examples include in-car servers that are built in a vehicle or a building.

Identifying these three types of resources leads us to our second question, which seeks to understand who provides the edge infrastructure. While it is clear that edge devices are from end users and backend clouds are from service providers, it remains a question as to who provides and maintains edge infrastructure. We have identified three models for the edge infrastructure. Each of these models requires a business model as well. As we envision many edge services, it is important to consider societal impacts. This pertains to answering two questions---how can we enable end users to do more? And how can we help end users to do less?

- **Cloud service provider model**: This model is similar to the current telco model, where a cloud service provider owns, manages, and distributes edge infrastructure. The primary purpose and the business model for edge infrastructure is for cloud service providers to run their services better by using resources closer to end users.
- **Third party edge service provider model**: This model is similar to the current Akamai model, where a third party edge service provider owns and manages edge infrastructure. The primary purpose and the business model for the infrastructure is for third parties to run value-added services.
- **End user model**: This model is similar to the current mobile device model, where end users purchase and manage edge infrastructure and service providers run their services using end user's infrastructure. For example, an end user could purchase edge servers

for her/his home and allow cloud services to use the servers; a hotel could purchase a data center for their guests that run edge services; a vehicle could come with general-purpose computing resources for its passengers; etc. Since end users need to bring in their own infrastructure, there has to be incentives; we discuss this later in this summary.

Five grand challenges in capabilities and services of edge computing and networking:

1. **Naming, identifying, and discovering resources**: Since edge services are distributed at the edge, traditional challenges of distributed systems all apply. This includes naming, identifying, and discovering resources.

2. **Standardized APIs**: Getting edge computing resources to cooperate is quite challenging, particularly if they are from different providers/vendors. APIs need to be standardized to get proper communication and synergy between different edge services. Along this line, it is important to support computation migration. Computations can migrate in edge computing. For example, virtual machines and containers could migrate between different nodes of an edge computing network. Achieving efficient virtual machine and container migration is important.

3. **Intelligent Edge Services**: Edge services will have data processing and analytics capabilities, as well as intelligent cognitive capabilities. The development of intelligent edge services is a key research challenge. An example is location-based Communication: An edge computing application might have communications and computations that are dependent on the specific physical location of the device (e.g., a drone). At times, an edge-computing device might have proper connectivity with a major network (e.g. LTE). At other times, it may rely on communication with a wireless ad hoc network.

4. **Security and Trust**: Security is critically important, as is trust. There will be many edge services, and methods are needed to assess trustworthiness and deal with untrusted services. Since edge infrastructure may run critical services and are close to end users, it is also of critical importance to achieve high availability and reliability.

5. **Edge Service Ecosystem**: Having capabilities similar to app stores for edge services would incentivize end users to purchase and own edge computing resources (e.g., servers). This envisions an open ecosystem where (1) third parties package up and distribute their edge services, similar to mobile apps, and (2) end users browse, download, and install various edge services as they see fit on their own edge infrastructure, similar to the way end users use online app stores to download various apps. Edge service brokers may become important. An edge service broker will help a client pick the best services for its needs, taking cost into consideration.

List of Participants

Helder F Antunes	Cisco and OpenFog Consortium
Victor Bahl	Microsoft Research, Microsoft Corporation
Bharath Balasubramanian	Senior Inventive Scientist, ATT Labs Research
Suman Banerjee	UW-Madison
Ken Calvert	NSF
Rong Nickle Chang	IBM T.J. Watson Research Center
Songqing Chen	George Mason University
Xiuzhen Cheng	The George Washington University
Mung Chiang (Co-Chair)	Princeton University Edge Lab
Fred Douglis	Dell EMC
Schahram Dustdar	TU Wien, Austria
John Garofolo	NIST
Glenn A. Fink	Pacific Northwest National Lab
Jason Flinn	University of Michigan
Sangtae Ha	University of Colorado, Boulder
Robert J. Hall	AT&T Labs Research
James Horan	NIST IAD
Arun Iyengar	IBM Research
Samee U. Khan	NSF
Guenter Klas	Vodafone
Steve Ko	University at Buffalo (SUNY)
Tian Lan	George Washington University
Martin Lehofer	Siemens Corporate Technology
Qun Li	The College of William and Mary
Ling Liu	Georgia Institute of Technology
Bruce Maggs	Duke / Akamai
Athina Markopoulou	UC Irvine
Mimi McClure	NSF
Vincent Park	Qualcomm
Chunming Qiao	University at Buffalo (SUNY)
Michael Rabinovich	Case Western Reserve University
Pablo Rodriguez	Telefonica
Krishan Sabnani	Bell Labs/Nokia
Mahadev Satyanarayanan	Carnegie Mellon University
Eve M. Schooler	Intel
Prashant Shenoy	University of Massachusetts
Weisong Shi (Co-Chair)	Wayne State University
Dilma Da Silva	Texas A&M University
John Smee	Qualcomm Technologies Inc.

Yan Solihin	NSF
Dennis Strigl	Former CEO, Verizon Wireless/COO, Verizon
Andrew Weinert	MIT Lincoln Laboratory
Fan Ye	Stony brook university
John Zaleski	Bernoulli Enterprise
Honggang Zhang	UMass Boston
Yanyong Zhang	Winlab, Rutgers University
Lin Zhong	Rice University
Douglas N. Zuckerman	IEEE

Workshop Agenda

8:15	*Breakfast and registration*
8:45	**Welcome Remarks (Richmond)** Ken Calvert, Division Director of CNS, NSF
9:00-9:15	**Introduction by co-organizers** Mung Chiang, Princeton University Weisong Shi, Wayne State University
9:15-10:00	**Rapid fire** (<1 min per person stating an interesting point about Edge)
10:00-11:15	**Breakout session 1** Track 1: Applications: Roanoke Track 2: Architecture: Richmond Track 3: Capabilities/Services: Williamsburg
11:15-11:30	*Break (Commonwealth Foyer)*
11:30-12:15	**Keynote** Dennis Strigl, Former CEO of Verizon Wireless
12:15-1:30	*Lunch*
1:30-2:30	**Panel** Moderator: Mung Chiang, Princeton University Panelists: Helder Antunes, Cisco Victor Bahl, Microsoft Research Bruce Maggs, Akamai Technologies and Duke University Mahadev Satyanarayanan, Carnegie Mellon University Weisong Shi, Wayne State University John Smee, Qualcom
2:30-3:30	**Breakout session 2** Track 1: Applications: Roanoke Track 2: Architecture: Richmond Track 3: Capabilities/Services: Williamsburg
3:30-3:45	*Break*
3:45-4:15	**Breakout session report Back**
4:15-4:30	**Wrap up**