AXaaS: Case for Acceleration as a Service

Nathaniel Powers*, Alexander Alling*, Regina Gyampoh-Vidogah†, Tolga Soyata*
*Dept. of Electrical and Computer Engineering
University of Rochester
Rochester, NY 14627
{npowers, aalling, soyata}@ece.rochester.edu
† Independent Researcher
Wolverhampton, UK
rgyampoh@hotmail.com

Abstract—**The ubiquity and the range of utility of ”smart” devices is ever increasing. Device limitations have lead developers to leverage cloud-offloading to gain performance for their applications. As users become aware of the expanding utility of their devices through these powerful applications, they tend to demand more from them. However, developers' intent on providing state-of-the-art applications will undoubtedly hit performance barriers for emerging products due to the inherently high latency of the prevailing mobile-cloud architecture. This paper proposes a new type of service architecture called AXaaS (Acceleration as a Service) that will empower developers to satisfy user demand for greater application performance and fully realize new computationally-intensive applications that would be otherwise impossible or impractical. While Telecom Service Providers (TSP) already provide data and bandwidth services, we introduce a new paradigm in which the TSP may charge subscribers for computational acceleration of complex applications by outsourcing computational tasks to larger cloud operators. We provide an exposition of the performance potential of such a service by examining its theoretical impact upon an open-source-based Face Recognition application. We also examine a sample instantiation of cloud resources via Amazon Web Services, and estimate the return on investment for a TSP implementing AXaaS. We find the TSP-side ROI to be quite favorable, which means that AXaaS is a viable new aaS alternative.**

I. Introduction

The development of consumer technologies is a push-pull affair: Users constantly want more speed, more power, bigger and more responsive programs. Developers try to fulfil these wants and provide a superior product in an economical fashion. Once consumers have their wishes granted and see what is possible, they of course want more. In recent years, cloud-based computing has become a popular service that attempts to fulfil the demand for more and better applications. Services such as Microsoft Azure [1], Amazon Web Service [2], and Google Cloud Platform [3] allow users and businesses to offload their resource-intensive applications to the cloud.

Many mobile applications are not computationally-intensive to necessitate accessing cloud computational resources. However, a family of applications, such as real-time Face Recognition, demand short bursts of intense computation and the mobile processor simply cannot supply this demand. When these applications access the cloud server for additional computational power, a cloud server can only provide its maximum peak GFLOPS (billion FLOating point OPerations per Second). As we will demonstrate in this paper, a single cloud server cannot satisfy the computational intensity of these highly-parallelizable applications within a reasonable amount of time, necessitating the utilization of multiple cloud servers in a burst (100's of servers in some cases). This problem is exacerbated due to the fact that, these short bursts of computation (termed *acceleration*, and stylized *AX*), must be provided at a very low latency, typically too low to be achievable for a cloud operator.

This paper presents the idea of *Acceleration as a Service*, or *AXaaS*, which can be provided by the Telephone Service Provider (TSP). TSP customers are already familiar with throughput and data storage plans. A *data computation plan* is the next logical step in enhancing the abilities of mobile phones. Our vision is for TSPs to rent computational resources from cloud operators [1]–[3] and provide AXaaS as a bundled service within the user's already existing monthly services. A number of real-life mobile apps could benefit significantly from AXaaS. Notable examples include many forms of image-processing-intensive applications, privacy-preserving medical cloud applications [4]–[7], graphics rendering, real-time object (specifically face) recognition, real-time language translation, and augmented reality. In this paper, we will analyze a Face Recognition application in detail [8] to provide a meaningful discussion of the potential capabilities of AXaaS. We will show that, the ROI of the AXaaS service offering is good enough for the TSPs to adopt it. This paper is organized as follows: In Section II, background information is provided on existing aaS models and the lack of a model similar to AXaaS is shown. Our proposed AXaaS model and its business evaluation are provided in Section III. We evaluate its implications to the user and TSPs in Section IV and provide concluding remarks in Section V.

II. Background

An “as a service” (aaS) offering is a collection of products that provide a unit-rate based pricing structure appropriate for a specific item delivered over the internet rather than locally [9]. There are a number of models available for enabling convenient, on-demand network access to a shared pool of configurable resources such as:

Software as a Service (SaaS) allows customers to run a software through an internet connection [10], and receive periodic updates seamlessly (e.g., Salesforce.com). **Platform as a Service (PaaS)** is the capability provided to the consumer to deploy onto cloud infrastructure using programming languages, libraries, services and tools supported by the provider [11]. **Infrastructure as a service (IaaS)** is the

978-1-4799-7470-2/14/$31.00 ©2014 IEEE

provision of applications and resources where the consumer is able to deploy and run arbitrary software [12]. IaaS removes the need for users to manage their own hardware. **Desktop as a Service (DaaS)** is the hosting of customers' entire desktop environment through a Cloud Service Provider. Customers are able to access applications, email, data storage/online backup, etc., as they would a normal computer [13]. **Monitoring as a Service (MaaS)** provides the option to offload a large majority of system monitoring costs by running them as a service as opposed to a fully invested in-house tool [14]. MaaS enables a pay-as-you-go utility model for state monitoring and minimizes the cost of ownership. **Communication as a Service (CaaS)** provides consumers with enterprise level VoIP, VPNs, PBX and Unified Communications without the costly investment of purchasing, hosting and managing the infrastructure. [15]. **Network as a Service (NaaS)** provides consumers with access to application resources forming a network. This can be used to implement the normal functions of a local network based on application needs [16].

Existing services focus on distributing software and making hardware environments available to the user. Services that increase and optimize the user's data transmission and data storage abilities are already available. Missing is a focus on enhancing a user's data computation abilities. IaaS partially fills this niche, but is limited in that the consumer is connected on a one for one basis with their cloud instance. In many cases, it is not economical or practical for a user to have access and control of an entire computer system, when all they want is a burst in computational power. This forms the basis of our formulation for acceleration as a service (AXaaS).

III. AXaaS: Acceleration as a Service

Acceleration as a Service (AXaaS) is a monthly subscription service offered by TSPs for computational acceleration to run mobile applications that require acceleration due to resource limitations of the mobile device [8], [17]–[20]. We make the clear distinction between *computation* and *acceleration* in this paper. While the term *computation* does not necessarily have a time-related connotation, *acceleration* implies performing an intense computational task and returning the result to the requester in a very short period of *time*. Mobile applications such as real-time Mobile-Cloud Face Recognition [8] require acceleration (AX).

AX service would be best implemented by a TSP because: 1) TSPs offer the lowest communication latency to mobile device users due to their direct communication interface, 2) TSPs have the capacity to aggregate and outsource intense computational tasks from multiple users to larger cloud operators. This outsourcing allows the TSP to take advantage of the economies of scale that cloud operators can offer on the commoditized computational resources. While (1) and (2) satisfy the intense computational requirements of *acceleration*, TSPs can also benefit from their established customer base, since 3) they are familiar with the specific needs of their customers, and 4) they already possess a captive audience for targeted advertising of new products and services.

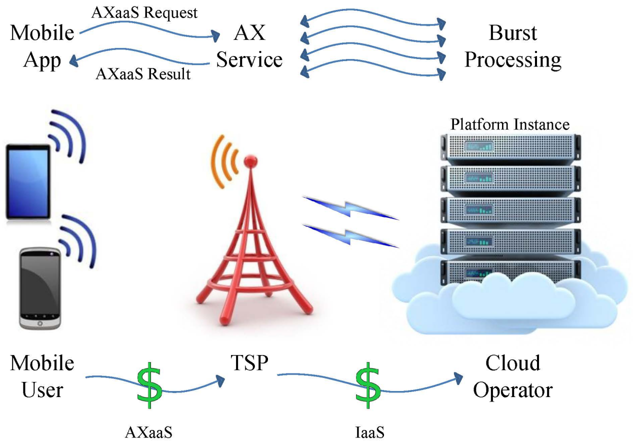

Fig. 1. The Acceleration as a Service (AXaaS) model.

A. AXaaS Operational Structure

The AXaaS model illustrated in Figure 1 consists of a Platform Instance (termed **PI**) or cluster of Instances with considerable compute-capability accessible only to the TSP and is transparent to the user. The end-user gains access to the power of the PI through a set of Application Programming Interfaces (API) provided by their TSP. The computational power of these APIs would scale based on the AXaaS plan that the user would subscribe to. The TSP itself is renting the PI from a cloud server operator as a form of Infrastructure as a Service (IaaS). Computation plans may be introduced in the same vein as data plans where the subscriber pays for allocations of computation (FLOPs) as an enhancement of an existing data usage plan. Once a device is registered and authorized, client-side applications may be downloaded and used within the AXaaS framework. Application-specific data may be transferred through the TSP, between the client device and the PI where compute-intensive processes are performed.

The PI would contain host-side applications tailored to AXaaS-oriented real-time applications such as FR, language translation and augmented reality. Burst-processes would run in the PI and the desired results would be transmitted back down to the end-user, satisfying their demand for fast results on complex processes. Storage requirements must be satisfied in order to support application hosting and functionality. The computational capacity of the PI must also be scaled with enough headroom to provide consistent and reliable performance for a reasonable saturation of application requests.

B. TSP-side view of AXaaS

AXaaS model allows the TSP to rent instances of hardware from the cloud operator. This is a highly economical model with a good degree of flexibility. Amazon Web Services offers a multitude of PI types with different fees and usage structures [21]. For our example, we chose a Heavy-Utilization Reserved c3.8xlarge PI because it maximizes the *availability* of compute power to customers while minimizing long-term costs to the TSP. Currently, the cost to reserve this PI is $1,352.73 monthly which would provide 23,887,872 TFLOPs of computation to be distributed among subscribers. For expandability, up to twenty of these Reserved PIs may be launched in each of the three U.S. Regions.

Table I shows an example distribution of 500,000 subscribers (merely 0.17% of the 290M American wireless customers [22]) across five computation tiers that would be available to subscribers for a monthly fee. A total monthly allocation of approximately 2.8B TFLOPs would require the support of 117 c3.8xlarge PIs. Gross monthly revenue in excess of $11M at a margin of 98.64% is possible for this scenario. Network bandwidth between the TSP and PIs should be maximized in order to support acceleration. Measurements by [23] evaluate the average local upload rates from Amazon EC2 instances to S3 buckets in Virginia, California and Oregon as 3.8, 9.4 and 14.6 MB/s respectively. Note that, these calculations are strictly a back-of-the-envelope feasibility analysis to show the viability of AXaaS, rather than a precise business report.

Tier TFLOP/mo	**50**	**500**	**4k**	**10k**	**30k**
Monthly AXaaS Fee	$5	$10	$20	$40	$60
# Subscribers	25k	100k	250k	100k	25k
TFLOP Alloc.	1.25M	50M	1B	1B	750M
Gross Revenue	$125k	$1M	$5M	$4M	$1.5M

TABLE I
BALANCED DISTRIBUTION OF 500,000 SUBSCRIBERS.

C. User-side view of AXaaS

We conceptualize AXaaS as a monthly subscription-based service offered by the TSP through some form of tiering, much like the currently existing data plans, as shown in Table I. Monthly allowed acceleration total (in TFLOPS) would depend on the monthly fee. Subscription to a service such as AXaaS must be justified to the user by offering benefits that outweigh the cost. Subscription to AXaaS would involve employing an API that enables the use of applications requiring acceleration. Applications would be downloaded, launched and managed just as any other non-accelerated application, ensuring ease and familiarity. The user may have no indication during run-time that their data is being offloaded for processing. What matters to them is the apparent application response time. To the user, the difference in computation tiers would translate to a number of accelerated requests that may be made in a time interval.

For a resource-intensive application such as Face Recognition, based on the model we have extrapolated from extensive testing, the estimated computational requirement is 1.675 TFLOP per-query with a database of 10,000 images. Subscribers to each tier in Table I would be able to perform respectively 1, 10, 80, 200 and 600 FR requests per day. If we assume there are 8,500 subscribers to one PI and each performs only 50 requests per day, the PI must process on average 4.92 requests per second, which is within its capacity.

AXaaS allows the offloading of not only *intensive computation*, but also the *energy consumption* as a consequence. According to our simulations, we determine AXaaS to result in a multiple orders-of-magnitude energy savings in the mobile device for compute-intensive applications. This allows AXaaS to be subscribed to by institutions to run compute-intensive applications *in the field*, e.g., wildlife monitoring [24], [25] due to their low energy budget [26]. This can be viewed as institutions "purchasing energy" from the cloud.

IV. Performance Evaluation

In this section, we will evaluate the potential of the AXaaS model using a specific Face Recognition (FR) application. The FR application begins with an end-user utilizing a mobile device which uploads image data (or detected face data for Detection-enabled devices) to a dedicated AXaaS Platform Instance (PI). Within the PI, a series of FR functions are performed and an identification result is returned to the device. The entire query process time consists of image-data upload time, frame-process time and result-data download time.

A. Experimental Setup

A combination of a low latency client-host connection (i.e., Verizon 4G LTE) and an acceleration methodology on a per-frame basis using a short burst of intense computational power is necessary to provide a *real-time* response. Without loss of generality, we define *real-time* as 1000 ms per frame. In our experiments with FR, a Windows platform with an Intel i7-4770K CPU and a Nvidia GTX 760 GPU performed the three steps of the FR algorithm: 1) Detection, 2) Projection and 3) Search functions on image frames from an LG Nexus4. Utilizing the GTX 760, per-frame Detection (step 1) occurred at an average of 50 ms regardless of the number of faces detected (due to the Face Detection algorithm being able to take advantage of the already-computed results). For each face detected in the frame, Projection (step 2) occurred in a time interval proportional to the number of images contained by the database from which results are obtained. Finally, for each computed Projection (step 3), a Search is run on the database which occurs in a time interval that increases quadratically with database size. The Search routine at this time is implemented entirely by the CPU. By running the application over a wide variety of databases and scenes, we have been able to collect and compile enough performance data to generalize parametric dependencies on database size (n_{images}) along with the computational requirements of each stage of the FR process in terms of GFLOPS: To perform Detection, Projection or Search (*t* in ms):

$$
\begin{aligned}
GFLOPS_{Detect} &= 919,678 * t_d^{-1.606} \\
GFLOPS_{Project} &= (160 * n_{images} + 38,513) * t_p^{-1} \\
GFLOPS_{Search} &= \left(\frac{n_{images}^2}{2,797} + \frac{n_{images}}{9.83} + 9.77 \right) * t_s^{-1}
\end{aligned}
$$

where t_d, t_p, and t_s curve-fit approximated Detection, Projection, and Search times from our experiments. We generated estimations for potential AXaaS platforms with respect to the FR application by extrapolating this empirical equation.

B. Experimental Results

Verizon Wireless 4G LTE wireless broadband speeds range from 5-12 Mbps download (640-1536 kB/s) and 2-5 Mbps upload (256-640 kB/s) [27]. Using 8kB as an estimated result size for a 160x160 JPEG thumbnail and text containing identification and query confidence data, our expected result download time will range from 13 ms to 6 ms. Using 145kB as the image size of a compressed 800x480 JPEG frame, our expected image upload time will range from 566 ms to 227 ms. Process times for frames containing multiple faces will scale with the frame-face density. If the mobile device being used is capable of performing Detection, the uploaded data size per query could be reduced by 94%. For our example, if we allow the face data generated from the mobile device to be compressed to 8kB the expected face upload time would range from only 32 ms to 13 ms. In an application where the mobile performs face Detection, the total query time would be reduced further by circumventing the process load for Detection at the PI. The savings from this scheme become increasingly pronounced as the latency of the link increases. However, a significant consequence of this scheme is a limitation to 1 to 2 faces that can be detected in any given frame. We have found that the total FR query time is most heavily influenced by:

1) The floating point performance capabilities of the devices on which the FR algorithms are run.
2) The efficiency of the searching algorithm and structure of the database.
3) The frame-face density in the case of an image upload.
4) The size of the data types transmitted across the WAN.
5) The WAN communication upload/download speeds.

Using the GFLOPS-process time relationships defined above, performance estimates can be extrapolated: For instance, assume the PI is an AWS EC2 C3 HPC cluster. With a floating-point performance of approximately 484 TFLOPS, it ranks #76 in the Top500 supercomputer list [28]. Assuming maximum 4G LTE speeds between the mobile device and TSP, query times can be estimated for any given database size. Our estimations in comparison with performance data gathered on the Baseline platform can be seen in Figure 2.

The implication of this estimation is that the AWS cluster would be able to respond to a single FR query in "real-time" with a database of up to 947,600 images whereas the Baseline platform reaches the real-time threshold at only 10,270 images. Furthermore, at the 10,270 image mark the AWS cluster performs Projection and Search on a query in under 4.5 ms with a total query time of 22 ms; the majority of the delay is owed to communication time over the WAN. Since the processor load for a single query at the 10,270 database is 1.72 TFLOP, the AWS HPC can be expected to process 281 concurrent queries per second. This metric shows the impressive performance capability of configurable cloud resources in a high-demand application such as Face Recognition. To illustrate the impact of network speed on performance, Figure 3 shows estimated performance of the AWS cluster at the low end of the 4G LTE speed spectrum.

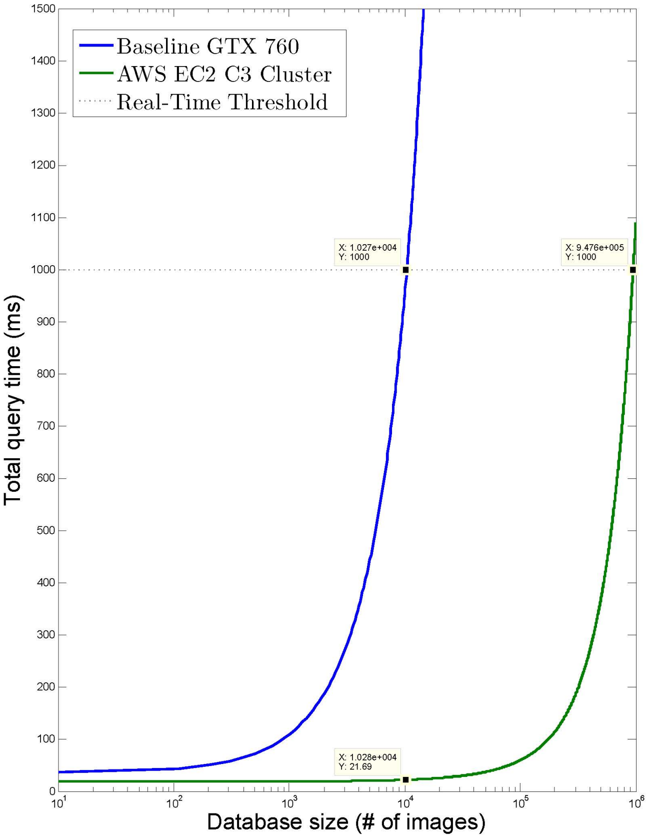

Fig. 2. Total query time of single face detected on mobile with ideal 4G LTE speeds, Baseline platform vs. Amazon EC2 C3 High Performance Cluster

At these average WAN speeds, the PI loses the capacity to respond in real-time for any database size if the frame-face density exceeds little more than 10 faces.

V. CONCLUSIONS AND FUTURE WORK

We have presented a mobile-TSP-cloud service model called *Acceleration as a Service (AXaaS)* whereby the TSP seamlessly and transparently connects its users to powerful cloud server instances capable of providing significant computation acceleration over short durations. AXaaS is able to provide a benefit to mobile applications where the end-user does not need to perform an operation continuously, but does need that operation completed as quickly as possible. The TSP charges its subscribers for this burst computation (i.e., *acceleration*), which is far beyond the capabilities of any mobile device.

We provided an analysis of the computational requirements of one Face Recognition (FR) application and found that an AWS EC2 C3 HPC cluster accessible to subscribers via 4G LTE would be capable of processing over 280 simultaneous FR requests on a 10,000 face database in real-time. Therefore, thousands of users could be serviced by the same Instance Cluster at a satisfactory performance.

We also provided an analysis of the business case for AXaaS in the context of the FR application. We found a TSP implementing AXaaS could potentially see annual profits in excess of $60M, with gross profit margins exceeding 90%.

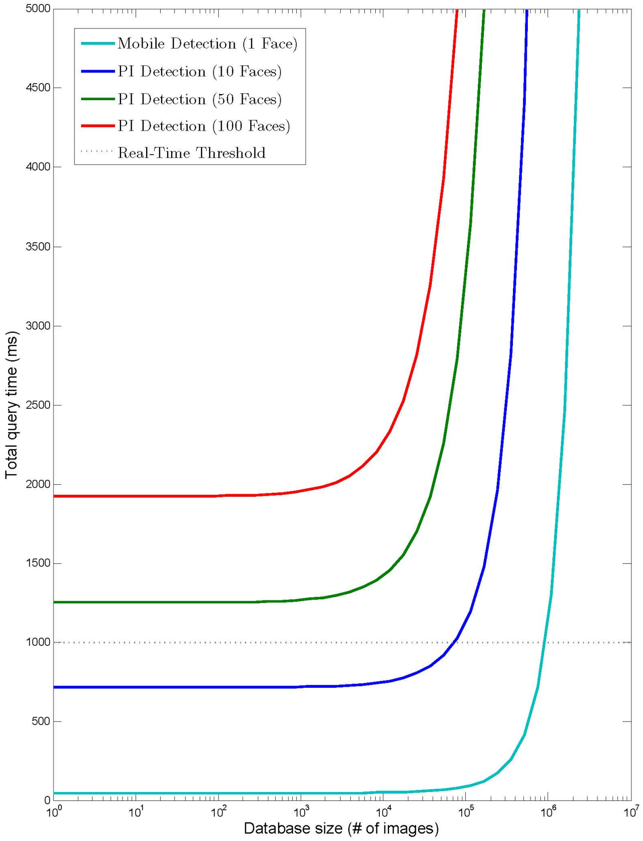

Fig. 3. FR query times of AWS HPC Cluster, *Min* 4G LTE

This estimate is not meant to be a detailed financial analysis, but, rather, a quick back-of-the-envelope calculation to prove the compelling potential of the AXaaS model.

Acknowledgments

This work was supported in part by the National Science Foundation grant CNS-1239423 and a gift from Nvidia Corp.

References

[1] "Microsoft windows azure," http://www.microsoft.com/windowazure.

[2] "Amazon web services," http://aws.amazon.com.

[3] "Google cloud platform," https://cloud.google.com/.

[4] Alex Page, Ovunc Kocabas, Scott Ames, Muthuramakrishnan Venkitasubramaniam, and Tolga Soyata, "Cloud-based secure health monitoring: Optimizing fully-homomorphic encryption for streaming algorithms," in *IEEE Globecom 2014 Workshop on Cloud Computing Systems, Networks, and Applications (CCSNA)*, Austin, TX, Dec 2014.

[5] Alex Page, Ovunc Kocabas, Tolga Soyata, Mehmet Aktas, and Jean-Philippe Couderc, "Cloud-Based Privacy-Preserving Remote ECG Monitoring and Surveillance," *Annals of Noninvasive Electrocardiology (ANEC)*, 2014.

[6] Ovunc Kocabas and Tolga Soyata, "Medical data analytics in the cloud using homomorphic encryption," in *Handbook of Research on Cloud Infrastructures for Big Data Analytics*, P. R. Chelliah and G. Deka, Eds., chapter 19, pp. 471–488. IGI Global, Hershey, PA, USA, Mar 2014.

[7] Ovunc Kocabas, Tolga Soyata, Jean-Philippe Couderc, Mehmet Aktas, Jean Xia, and Michael Huang, "Assessment of cloud-based health monitoring using homomorphic encryption," in *Proceedings of the 31st IEEE International Conference on Computer Design (ICCD)*, Ashville, VA, USA, Oct 2013, pp. 443–446.

[8] Tolga Soyata, Rajani Muraleedharan, Colin Funai, Minseok Kwon, and Wendi Heinzelman, "Cloud-Vision: Real-Time face recognition using a Mobile-Cloudlet-Cloud acceleration architecture," in *Proceedings of the 17th IEEE Symposium on Computers and Communications (IEEE ISCC 2012)*, Cappadocia, Turkey, Jul 2012, pp. 59–66.

[9] V Remenar, "Xaas services as modern infrastructure of e-government in the republic of croatia. international scientific conference," http://www.academia.edu/3143834/xaas_services_as_modern_infrastructure_of_e-government_in_the_republic_of_croatia.

[10] P Mell and T Grance, "The nist definition of cloud computing. recommendations of the national institute of standards and technology," http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.

[11] S Joshi, "What is platform as a service (paas)?," http://thoughtsoncloud.com/2014/02/what-is-platform-as-a-service-paas/.

[12] "Making infrastructure-as-a-service in the enterprise a reality," http://www.oracle.com/us/products/enterprise-manager/infrastructure-as-a-service-wp-1575856.pdf.

[13] "Desktop as a service with vmware and symantec," https://www.vmware.com/files/pdf/b-desktop_as_a_service_WP_en-us_08-11.pdf.

[14] S Meng and L Ling, "Enhanced monitoring-as-a-service for effective cloud management," http://www.istc-cc.cmu.edu/publications/papers/2013/maas-tc.pdf.

[15] A Hendryx, "Concepts: Iaas, paas, saas, maas, caas & xaas," http://www.zdnet.com/cloudy-concepts-iaas-paas-saas-maas-caas-and-xaas-4010024679/.

[16] P Costa, M Migliavacca, and P Pietzuch, "Naas: Network-as-a-service in the cloud," Proceedings of the 2nd USENIX conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services.

[17] Minseok Kwon, Zuochao Dou, Wendi Heinzelman, Tolga Soyata, He Ba, and Jiye Shi, "Use of network latency profiling and redundancy for cloud server selection," in *Proceedings of the 7th IEEE International Conference on Cloud Computing (IEEE CLOUD 2014)*, Alaska, USA, Jun 2014, pp. 826–832.

[18] Haoliang Wang, Wei Liu, and Tolga Soyata, "Accessing big data in the cloud using mobile devices," in *Handbook of Research on Cloud Infrastructures for Big Data Analytics*, P. R. Chelliah and G. Deka, Eds., chapter 18, pp. 444–470. IGI Global, Hershey, PA, USA, Mar 2014.

[19] Tolga Soyata, R. Muraleedharan, S. Ames, J. H. Langdon, C. Funai, M. Kwon, and W. B. Heinzelman, "Combat: mobile cloud-based compute/communications infrastructure for battlefield applications," in *Proceedings of SPIE*, May 2012, vol. 8403, pp. 84030K–84030K.

[20] Tolga Soyata, He Ba, Wendi Heinzelman, Minseok Kwon, and Jiye Shi, "Accelerating mobile cloud computing: A survey," in *Communication Infrastructures for Cloud Computing*, H. T. Mouftah and B. Kantarci, Eds., chapter 8, pp. 175–197. IGI Global, Hershey, PA, USA, Sep 2013.

[21] "Amazon ec2 pricing," http://aws.amazon.com/ec2/pricing/.

[22] "Vzw industry overview," http://www.verizon.com/investor/industryoverview.htm.

[23] "The aws olympics: Speed testing amazon ec2 and s3 across regions," http://www.takipiblog.com/2013/03/20/aws-olypmics-speed-testing-amazon-ec2-s3-across-regions/.

[24] Amal Fahad, Tolga Soyata, Tai Wang, Gaurav Sharma, Wendi Heinzelman, and Kai Shen, "SOLARCAP: super capacitor buffering of solar energy for self-sustainable field systems," in *Proceedings of the 25th IEEE International System-on-Chip Conference (IEEE SOCC)*, Niagara Falls, NY, Sep 2012, pp. 236–241.

[25] Moeen Hassanalieragh, Tolga Soyata, Andrew Nadeau, and Gaurav Sharma, "Solar-supercapacitor harvesting system design for energy-aware applications," in *Proceedings of the 27th IEEE International System-on-Chip Conference (IEEE SOCC)*, Las Vegas, NV, Sep 2014, pp. 280–285.

[26] Andrew Nadeau, Gaurav Sharma, and Tolga Soyata, "State-of-charge estimation for supercapacitors: A kalman filtering formulation," in *Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2013)*, Florence, Italy, May 2014, pp. 2213–2217.

[27] "4g lte speeds vs. your home network," http://www.verizonwireless.com/insiders-guide/network-and-plans/4g-lte-speeds-compared-to-home-network.

[28] "Amazon ec2 c3 instance cluster," http://top500.org/system/178321.