Mahadev Satyanarayanan *Carnegie Mellon University*

Editor: Carla Schlatter Ellis

A BRIEF HISTORY OF CLOUD OFFLOAD

A Personal Journey from Odyssey Through Cyber Foraging to Cloudlets

Every time you use a voice command on your smartphone, you are benefitting from a technique called *cloud offload.* Your speech is captured by a microphone, pre-processed, then sent over a wireless network to a cloud service that converts speech to text. The result is then forwarded to another cloud service or sent back to your mobile device, depending on the application. Speech recognition and many other resource-intensive mobile services require cloud offload. Otherwise, the service would be too slow and drain too much of your battery.

Research projects on cloud offload are hot today, with MAUI [4] in 2010, Odessa [13] and CloneCloud [2] in 2011, and COMET [8] in 2012. These build on a rich heritage of work dating back to the mid-1990s on a theme that is broadly characterized as cyber foraging. They are also relevant to the concept of *cloudlets* [18] that has emerged as an important theme in mobile-cloud convergence. Reflecting my participation in this evolution from its origins, this article is a personal account of the key developments in this research area.

It focuses on mobile computing, ignoring many other uses of remote execution since the 1980s such as distributed processing, query processing, distributed object systems, and distributed partitioning.

At the Dawn of Mobile Computing

In 1993, the editor of the "Hot Topics" department of IEEE Computer invited me to write a short thought piece on the brand new topic of mobile computing that was just starting to emerge. To the best of my knowledge this is the first place where the inherent resource poverty of mobile devices is identified as a key long-term constraint of mobile computing. To quote verbatim from that piece entitled "Mobile Computing" [15]:

> **Mobile elements are resource-poor relative to static elements. Regardless of future technological advances, a mobile unit's weight, power, size, and ergonomics will always render it less computationally capable than its static counterpart. While mobile elements will undoubtedly improve in absolute ability, they will always be at a relative disadvantage.**

Later, in 1995, I made this same point in a keynote talk entitled "Fundamental Challenges of Mobile Computing," to the ACM Principles of Distributed Systems Conference. An invited paper based on this talk was published in the following year's

conference proceedings [16] and has been widely cited since.

In the 20+ years since that prediction was made, it has remained consistently true. *Figure 1* from Flinn [5] illustrates the consistent large gap in the processing power of typical server and mobile device hardware over a 16-year period. This stubborn gap reflects a fundamental reality of user preferences. The most sought- after features of a mobile device are light weight, small size, and long battery life. By using remote execution on static infrastructure that does not suffer from these constraints, the mobile device can overcome its computational limitations.

FIGURE 1: Evolution of Hardware Performance

	Typical Server		Typical Handheld or Wearable	
Year	Processor	Speed	Device	Speed
1997	Pentium® II	266 MHz	Palm Pilot	16 MHz
2002	Itanium®	1 GHz	Blackberry 5810	133 MHz
2007	Intel® CoreTM	9.6 GHz 2 (4 cores)	Apple iPhone	412 MHz
2011	Intel® Xeon®	32 GHz X5 (2x6 cores)	Samsung Galaxy S2	2.4 GHz (2 cores)
2013	Intel® Xeon®	64 GHz E5 (2x12 cores)	Samsung Galaxy S4	6.4 GHz (4 cores)
			Google Glass OMAP 4430	2.4 GHz (2 cores)

(Source: adapted from Flinn [5])

A Decade of Exploration

We began exploring remote execution for mobile computing in the context of the Odyssey system. As described in 1997 by Noble *et al* [12], the Janus speech recognition application was modified to operate in one of three modes in Odyssey. One mode involved strictly local execution. The second mode involved strictly remote execution: the speech signal captured on the mobile client was shipped to a remote server for recognition, and the transcribed text was shipped back. The third mode was hybrid: a preliminary phase of speech processing was done locally, and the extracted information was shipped to a remote server for the completion of the recognition process. An important attribute of this implementation was Odyssey's ability to dynamically select the optimal execution mode based on runtime factors such as current network bandwidth. Odyssey was thus the technical forerunner of today's mobile speech-to-text systems such as Siri, as well as the recent mechanisms for adaptive cloud offload that were mentioned at the beginning of the paper.

Rudenko *et al* [14] were the first to suggest that remote execution could extend battery life on mobile devices, and to provide experimental evidence for this claim. At about the same time, Flinn [6] extended Odyssey to support energy-aware adaptation and showed that remote execution could indeed save energy in the Janus speech recognition application.

In 2001, I was invited to write an article for an IEEE Personal Communications special issue on pervasive computing. This article, entitled "Pervasive Computing: Vision and Challenges" [17] suggested many research directions that have since proved fruitful. Abstracting and generalizing from the work on remote execution, it emphasized the enduring problem of resource poverty of mobile devices and suggested that leveraging nearby resources in a principled way might be the best approach to addressing this problem. I used the metaphor of foraging for food in the wild:

> ***Cyber foraging*, construed as "living off the land", may be an effective way to deal with this problem. The idea is to dynamically augment the computing resources of a wireless mobile computer by exploiting wired hardware infrastructure.**
> …
> **When hardware in the wired infrastructure plays this role, we call it a surrogate of the mobile computer it is temporarily assisting.**

Implicit in the foraging metaphor is the notion of "nearby" resources. It seemed obvious that low latency and high bandwidth to the remote execution site was essential. However, the paper did not explicitly discuss proximity as an important criteria. This turned out to have unexpected consequences some years later, as described below. It is important to note that cyber foraging as envisioned in this article encompassed both remote execution and the staging of data nearby. The data staging concept was later expanded by Flinn *et al* [7]. Today's content delivery networks (CDNs) can be viewed as implementing a form of data staging. The work on fluid replication by Noble *et al* [11, 3] is related to the concept of cyber foraging for data, and the associated infrastructure concept of *WayStations* can be viewed as similar to surrogates.

The period from 2001-2008 saw vigorous research activity in this space. A detailed account of these efforts is provided in the excellent survey by Flinn [5].

The Cloud Appears

A nagging question from the very beginning that had never been satisfactorily answered was who would provide the infrastructure for cyber foraging? How would trustworthy hardware for remote execution be dispersed in the environment? How would mobile devices discover them? What business incentives would there be to deploy and maintain such infrastructure?

The emergence of cloud computing circa 2008 suddenly clarified these issues. Independent of mobile computing considerations, companies like Amazon were making computing resources that could be used for transient purposes available on the Internet. There was now a business model and hence incentive for

deploying and maintaining hardware for remote execution.

The emergence of Apple's cloud-based Siri speech recognition service both validated the remote execution concept for mobile devices at commercial scale, and simultaneously offered a viable deployment model. "In the cloud," was the obvious answer to the question, "Where should remote execution be performed?" In industry and in academia, the imminent convergence of mobile computing and cloud computing was heralded.

Alas, that celebration was premature. The economics of cloud computing strongly favor the centralization of infrastructure into a few large data centers. It is through economies of scale in operations and system administration that cloud computing wins. Unfortunately, global consolidation implies large average separation between a mobile device and its cloud. End-to-end communication then involves many network hops and results in high latencies. From the beginning, I had implicitly assumed low latency and high bandwidth between mobile device and remote execution site. But by early 2008, I realized that my implicit assumption of "nearby" in framing the cyber foraging concept was a mistake. I should have made explicit the importance of proximity.

Discussions with a number of senior researchers in mobile computing at the 2008 MobiSys conference in Breckenridge, CO convinced me that it was necessary to make the case for proximity explicit to the research community and to industry. In close collaboration, Victor Bahl from Microsoft, Roy Want from Intel, Ramon Caceres from AT&T, Nigel Davies from Lancaster University, and I articulated the need for a two-level architecture for mobile-cloud convergence.

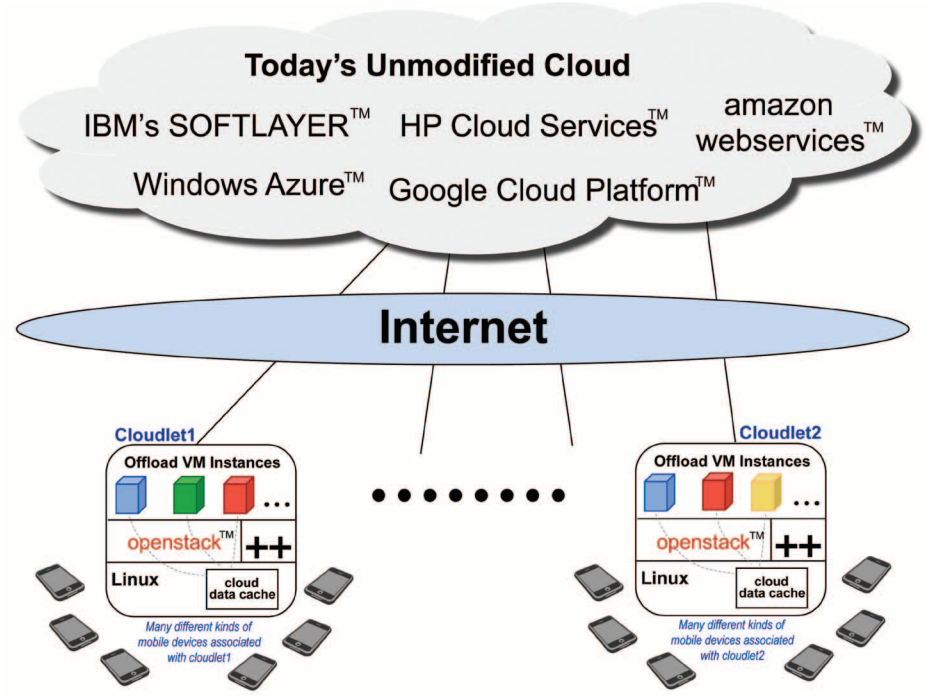


FIGURE 2. Two-Level Cloud-Cloudlet Architecture

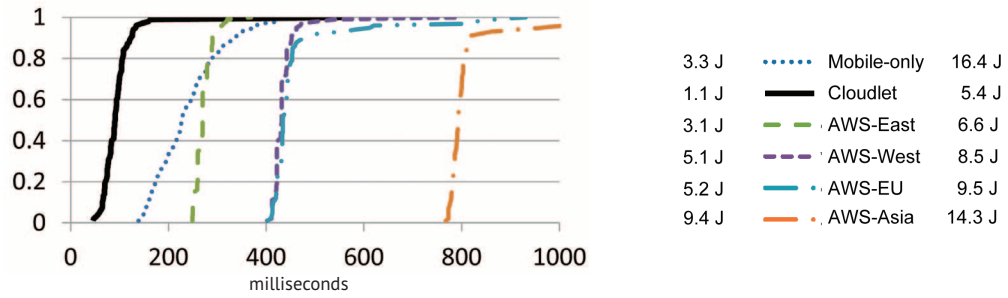
Figure 2 illustrates this architecture. The first level is today's unmodified cloud infrastructure. The second level consists of dispersed elements with no hard state called cloudlets. A cloudlet is effectively a "second-class data center" with soft state generated locally or cached from the first level. By using persistent caching instead of hard state, the management of cloudlets is kept simple in spite of their physical dispersal at the edge of the Internet. Replacing a cloudlet is conceptually similar to replacing a networking element such as router. We described the cloudlet concept in a paper that has proved to highly influential, receiving more than 550 citations in less than five years [18]. Cisco's recent concept of fog computing [1] is consistent with the cloudlet concept.

Proximity Really Matters

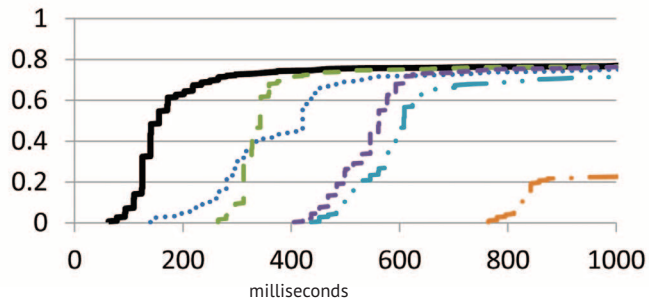
Alas, I discovered that just publishing a paper articulating the need for proximity in remote execution was not sufficient. There were still plenty of skeptics out there who needed much harder evidence that proximity was really necessary. Perhaps the best evidence of this skepticism is this verbatim quote from the 2009 panel summary of my proposal on cloudlets that was rejected by the National Science Foundation:

> **Many panelists do not agree with the premise of the proposal in which distant cloud computing incurs too high latency to be acceptable by mobile applications. They question the validity of such assumption as the proposal provides no real data to justify it. . . .**

THE EMERGENCE OF APPLE'S CLOUD-BASED SIRI SPEECH RECOGNITION SERVICE BOTH VALIDATED THE REMOTE EXECUTION CONCEPT FOR MOBILE DEVICES AT COMMERCIAL SCALE, AND SIMULTANEOUSLY OFFERED A VIABLE DEPLOYMENT MODEL.



(a) Augmented Reality



(b) Face Recognition

FIGURE 3. Response Time Distribution and Per-Operation Energy Cost

(Source: Ha et al [10])

To gain the necessary hard evidence, we invested substantial effort in obtaining latency-sensitive and compute-intensive applications such as face recognition, object recognition, and augmented reality. Extensive measurements have now proven beyond all reasonable doubt that running such applications on a nearby cloudlet gives much better response time and lower energy usage on a mobile device than running them in the cloud.

The importance of cloudlets can be seen in the results shown in Figure 3 for augmented reality and face recognition on a mobile device. Full details of these experiments and many others can be found in the paper by Ha et al [10]. An image from the mobile device (located in Pittsburgh, PA) is transmitted over a Wi-Fi first hop to a cloudlet or to an Amazon Web Services (AWS) data center.

The image is processed at the destination by computer vision code executing within a virtual machine (VM).

For augmented reality, buildings in the image are recognized and labels corresponding to their identities are transmitted back to the mobile device. For face recognition, the identity of the person is returned. Each curve in Figure 3 corresponds to the CDF of the observed response time distribution. The ideal curve is a step function that jumps to 1.0 at the origin. *Figure 3* shows that this ideal is best approximated by a cloudlet.

End-to-end latency plays a dominant role, as shown by the worsening response time curves corresponding to more distant AWS locations. Increasing response time also increases the per-operation energy consumption on the mobile device. This value is shown beside the corresponding label in the middle of the figure. For example, the mobile device consumes 1.1 J on average to perform an augmented reality operation on the cloudlet, but 3.1 J, 5.1 J, and so on when performing it on AWS-East, AWS-West, etc. Although these results were obtained on AWS, similar results can be expected with any offload service that is concentrated in a few large data centers.

The label "mobile-only" in *Figure 3* corresponds to a case where no offloading is performed, and the computer vision code is run on the mobile device. In spite of avoiding the energy and performance cost of Wi-Fi communication, the data shows that mobile-only does worse than using the cloudlet. Offloading is clearly important for these applications.

Closing Thoughts

Although simple in concept, cloudlets are a disruptive force in mobile computing. Their ability to provide low latency, high bandwidth access to energy-unlimited high-end computing within one wireless hop of mobile devices is transformative. Many valuable applications can be created using cloudlets.

What new kinds of mobile services might cloudlets make possible in the future? One possibility is wearable cognitive assistance, as described by Ha *et al* [9]. Such a service combines the video capture and sensing capabilities of wearable devices such as Google Glass with cloud offload to perform real-time scene analysis. To a user in cognitive decline (for example, an elderly person with Alzheimer's disease), such a service could offer helpful assistance, much as a GPS navigation system does today for a driver in an unfamiliar city. Crisp response time, below a few tens of milliseconds per offload operation, will be essential for avoiding user distraction. Cloudlets will be essential for such a service.

The early stages of the convergence of mobile computing and cloud computing are already under way. There is a small window of opportunity to shape this convergence so that it preserves openness and cohesiveness of software interfaces and network protocols in the new infrastructure that will emerge. This path can lead to the kind of explosive growth seen in the Internet itself. The next few years will be critical in shaping this future. ■

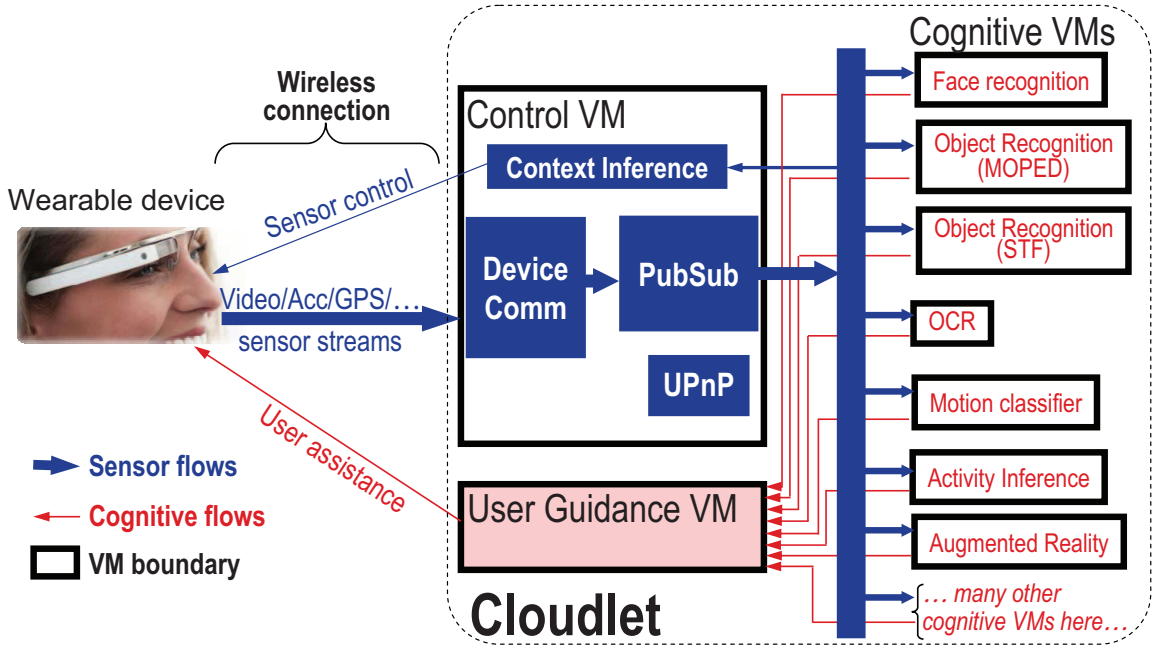


FIGURE 4. Cloudlet architecture of Gabriel, a wearable cognitive assistance system

(Source: Ha et al [9])

REFERENCES

[1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli. Fog Computing and Its Role in the Internet of Things. In *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, Helsinki, Finland, 2012.

[2] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti. CloneCloud: Elastic Execution between Mobile Device and Cloud. In *Proceedings of EuroSys 2011*, Salzburg, Switzerland, April 2011.

[3] L. P. Cox and B. D. Noble. Fast Reconciliations in Fluid Replication. In *Proceedings of the The 21st International Conference on Distributed Computing Systems*, 2001.

[4] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl. MAUI: Making Smartphones Last Longer with Code Ooad. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, San Francisco, CA, June 2010.

[5] J. Flinn. *Cyber Foraging: Bridging Mobile and Cloud Computing via Opportunistic Offload*. Morgan & Claypool Publishers, 2012.

[6] J. Flinn and M. Satyanarayanan. Energy-aware Adaptation for Mobile Applications. In *Proceedings of the 17th ACM Symposium on Operating Systems and Principles*, Kiawah Island, SC, December 1999.

[7] J. Flinn, S. Sinnamohideen, N. Tolia, and M. Satyanarayanan. Data Staging on Untrusted Surrogates. In *Proceedings of FAST'03: 2nd USENIX Conference on File and Storage Technologies*, San Francisco, CA, March 2003.

[8] M. S. Gordon, D. A. Jamshidi, S. Mahlke, Z. M. Mao, and X. Chen. COMET: Code Ooad by Migrating Execution Transparently. In *10th USENIX Symposium on Operating Systems Design and Implementation (OSDI 12)*, Hollywood, CA, October 2012.

[9] K. Ha, Z. Chen, W. Hu, W. Richter, P. Pillai, and M. Satyanarayanan. Towards Wearable Cognitive Assistance. In *Proceedings of the Twelfth International Conference on Mobile Systems, Applications, and Services*, Bretton Woods, NH, June 2014.

[10] K. Ha, P. Pillai, G. Lewis, S. Simanta, S. Clinch, N. Davies, and M. Satyanarayanan. The Impact of Mobile Multimedia Applications on Data Center Consolidation. In *Proceedings of the IEEE International Conference on Cloud Engineering*, San Francisco, CA, March 2013.

[11] B. Noble, B. Fleis, M. Kim, and J. Zajkowski. Fluid replication. In *Proceedings of the Network Storage Symposium*, Seattle, WA, October 1999.

[12] B. Noble, M. Satyanarayanan, D. Narayanan, J. Tilton, J. Flinn, and K. Walker. Agile Application-Aware Adaptation for Mobility. In *Proceedings of the 16th ACM Symposium on Operating Systems Principles*, Saint-Malo, France, October 1997.

[13] M. Ra, A. Sheth, L. Mummert, P. Pillai, D. Wetherall, and R. Govindan. Odessa: Enabling Interactive Perception Applications on Mobile Devices. In *Proceedings of the International Conference on Mobile Systems, Applications, and Services* (MobiSys 2011), Bethesda, MD, June 2011.

[14] A. Rudenko, P. Reiher, G. J. Popek, and G. H. Kuenning. Saving portable computer battery power through remote process execution. *Mobile Computing and Communications Review*, 2(1), January 1998.

[15] M. Satyanarayanan. Mobile computing. *IEEE Computer*, 26(9), September 1993.

[16] M. Satyanarayanan. Fundamental Challenges in Mobile Computing. In *Proceedings of the ACM Symposium on Principles of Distributed Computing*, Ottawa, Canada, 1996.

[17] M. Satyanarayanan. Pervasive Computing: Vision and Challenges. *IEEE Personal Communications*, 8(4), 2001.

[18] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. The Case for VM-Based Cloudlets in Mobile Computing. *IEEE Pervasive Computing*, 8(4), October-December 2009.