Towards Deploying Decommissioned Mobile Devices as Cheap Energy-Efficient Compute Nodes

Mohammad Shahrad and David Wentzlaff

Princeton University

Abstract

The performance of mobile phone processors has been steadily increasing, causing the performance gap between server and mobile processors to narrow with mobile processors sporting superior performance per unit energy. Fueled by the slowing of Moore's Law, the overall performance of single-chip mobile and server processors have likewise plateaued. These trends and the glut of used and partially broken smartphones which become environmental e-waste motivate creating cloud servers out of decommissioned mobile phones. This work proposes creating a compute dense server built out of used and partially broken smartphones (e.g. screen can be broken). This work evaluates the total cost of ownership (TCO) benefit of using servers based on decommissioned mobile devices and analyzes some of the architectural design trade-offs in creating such servers.

1 Introduction

The global sales of mobile devices surpassed PC sales in 2010 and is projected to grow. Smartphones alone are anticipated to have 6.3 billion subscriptions in 2021, double the number of subscribers in 2015 [1]. This growing market has created extraordinary competition for relevant industries and has lead to a great innovation rate.

In particular, two trends have become visible: the performance gap between mobile system on chips (SoCs) and commodity desktop and server processors is shrinking, and TCO for mobile SoCs remains much lower than typical CPUs. The HPC community has seen this same trend switching from vector to RISC machines and then from RISC to x86 machines [31]. Speculating that mobile SoCs will be the next cost-effective platform, academia, as well as industry, are building HPC and cloud systems using mobile SoCs [32, 19].

While systems can be custom-built using mobile SoCs to achieve the best possible performance, we propose the direct deployment of decommissioned (used) mobile devices in Infrastructure-as-a-Service (IaaS) clouds. The key idea is that after some time, mobile devices lose a significant portion of their value due to physical damage, slowdowns due to OS updates, etc.; whereas their processing power remains constant. Therefore, it makes economic sense to purchase cheap, low-power, used devices and only use their SoCs.

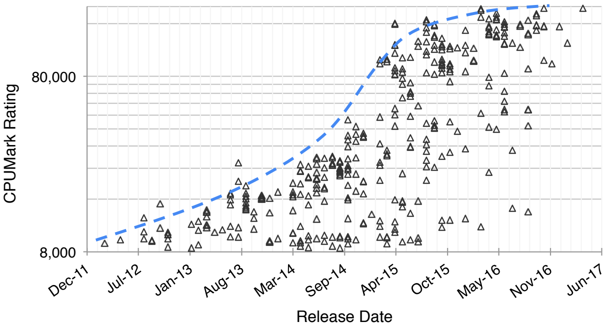


Figure 1: CPU performance trend for Android mobile devices.

Deploying decommissioned mobile devices can be a major move towards green computing. This is mostly due to the fact that most of the carbon footprint of those devices comes from their production [12]. Such deployment extends effective lifetime of mobile devices and decreases their average global warming potential (GWP), benefiting the environment.

Analyzing the mobile SoC performance for the last five years (Figure 1), we find two interesting trends. First, the end of Moore's Law is starting to kick in for mobile SoCs, which means soon, the relative performance gap of a new vs. a 3-year-old device will shrink. Second, the relative performance gap between high-end and low-end SoCs is shrinking, which enables having similar performance on a cheaper device. These trends reveal the promising opportunities for deploying used mobile devices in the near future.

In this paper, we explore the merits of deploying decommissioned mobile devices in data centers. We then investigate the possible applications that can benefit from them. We propose a design that efficiently integrates such devices in typical data centers and analyze the integration density. We finally compare the TCO of our proposed system to a performance- and density-comparable traditional server.

2 Background & Motivation

In this section, we describe the primary motivations for deploying decommissioned mobile devices in the cloud and mention some similar efforts.

2.1 Performance and Power

Although mobile SoCs are relatively slower than commodity servers, the performance "*gap is quickly being closed*" [31]. We used the publicly available data from PassMark's PerformanceTest Mobile benchmark [29] to understand the CPU performance trends in Android mobile devices. In particular, CPUMark is a thorough measure of CPU performance which fully utilizes all CPU cores [30].

Comparing the CPUMark ratings of 350 different models versus their release dates (Figure 1), two main trends are observed. First, the end of Moore's law is kicking in for mobile SoCs and their exponential rate of growth in computational performance has slowed down. Second, the relative performance gap between high-end and low-end mobile devices is shrinking.

Halpern et al. have observed that "*the mobile CPU's single-core thermal design point (TDP) has saturated at around 1.5 W.*" [16]. Moreover, newer smartphone generations have slightly lower energy efficiency as their power "*consumption increases outweigh performance gains*" [16]. Deploying decommissioned devices in the cloud, the former observation guarantees that the per-core power budget won't increase as devices scale, and the latter suggests that decommissioned devices have better overall energy efficiency.

2.2 State of Decommissioned Devices

Decommissioned mobile devices are anticipated to suffer from physical damage and/or reduced battery life. They might also be running legacy OSs, with fewer capabilities, or updated OSs, which usually run slower on older SoCs. Although all of these factors can seriously affect mobile user experience, they have no effect on the deployment of such devices as compute nodes, since their computational capacity is fixed.

2.3 Environmental Benefits

Designed to work under tight energy constraints, mobile devices are more energy-efficient than commodity servers. Therefore their adoption can help further improve the energy efficiency of modern data centers. In addition, using decommissioned mobile devices can have a more specific advantage. The production phase of a mobile device consumes significantly more energy than the device's energy consumption under use [9]. In fact, considering a three-year period, more than 80% of an average smartphone's carbon dioxide equivalent (CO_2e) is due to its production [12]. Therefore, extending the effective lifetime of such devices can decrease their average global warming potential (GWP) and benefit the environment.

3 Targeted Applications & Services

Due to their energy efficiency and improved performance, ARM-based architectures have recently gained substantial attention for HPC and cloud infrastructure deployment. ARM multicores deliver good energy proportionality [7] for server workloads [35]. Researchers have explored the possibility of building ARM-based HPC clusters [31, 32]. HP has built the Moonshot system using Texas Instrument's (TI) KeyStone II SoCs, which incorporates TI DSPs and ARM processors [19]. In March 2017, Microsoft announced a collaboration with Qualcomm to use their 48-core Centriq 2400 ARM-based SoC in its cloud platform and will support Windows Server for ARM.

The most promising applications for decommissioned mobile devices in the cloud include:

1) I/O-intensive applications. For I/O-intensive applications, the processing capacity remains underutilized due to I/O-induced idle cycles [36]. Considering the fact that smaller processors generally have a better I/O to computation ratio, researchers have proposed systems to use low-end CPUs to match I/O and computation. For instance, the Fast Array of Wimpy Nodes (FAWN) [4] uses embedded CPUs to provide two orders of magnitude more queries per Joule in a key-value storage system.

Modern mobile devices support high bandwidth I/O[1], which is usually enough to saturate their SoC I/O capacity. This together with their ample RAM size enables I/O-intensive applications to run with less I/O-CPU mismatch on them. Moreover, the shrinking performance gap between ARM SoCs and commodity servers [31] is resolving previous concerns about low quality-of-service of wimpy nodes [17, 18].

[1] USB 3.1 introduced in July 2013 supports transfer rates up to 10Gbps.

2) Low-end VM provisioning. Cloud infrastructure providers provision low-end virtual machines (VMs) to the clients. For instance, Amazon EC2's burstable `t2.nano` and `t2.micro` instances have 0.5GB and 1GB of memory, respectively. Considering that common hypervisors such as KVM and Xen support virtualization for ARM [8] and an average mobile device has more than 2GB of memory, IaaS providers can assign multiple of such miniature instances to each device.

3) GPU-accelerated dwarfs. GPU-accelerated instances offered by IaaS providers such as Amazon EC2 (P2 instances [3]), GCE [15], and the Azure N Series [24] use beefy accelerators, such as the NVIDIA K80. These instances are mainly designed to accelerate workloads with massive parallelism. However, mobile SoCs are equipped with smaller GPUs, primarily used for graphics rendering. This allows low-end GPU acceleration for small VMs to utilize platforms like OpenCL. Techniques such as Virtual Function I/O (VFIO) allow sharing an SoC's GPU between multiple tenants [25].

4) Low-end nodes for dynamic reliability platforms. Researchers have shown how heterogeneity of IaaS cloud resources can be utilized to serve different clients with various availability requirements [33]. Decommissioned devices can be used as cheap nodes with lower reliability in order to diversify the reliability heterogeneity of the infrastructure. While the diversity of ISAs can limit the potential gains, there had been recent efforts towards cross-ISA migration [27, 37, 5] and composable cores for IaaS systems [38].

4 Data Center Integration

4.1 Integration Density

Figure 2 shows our proposed server design that houses decommissioned mobile devices in standard server racks. As seen, mobile devices are held in cages with fixed size and don't have to be modified. Analyzing the dimensions of the 350 mobile device models used in Figure 1, we decided that each cage be $180 \times 80 \times 9mm$. This fits more than 75% of those models, with most of the tablets not fitting. Adding three rows of fans, a network router, and a power supply, allows room for 84 cages (smartphones) with the spacing of $5.5mm$ for airflow. We will use this number to estimate other design parameters.

The average number of CPU cores per device for the 350 mobile devices we studied is about 5.6 (M=5.55, SD=1.97). This means our 2U server could fit around 470 cores and the core density would be 235 cores/1U. This core density number can help better interpret TCO analysis results in Section 4.4.

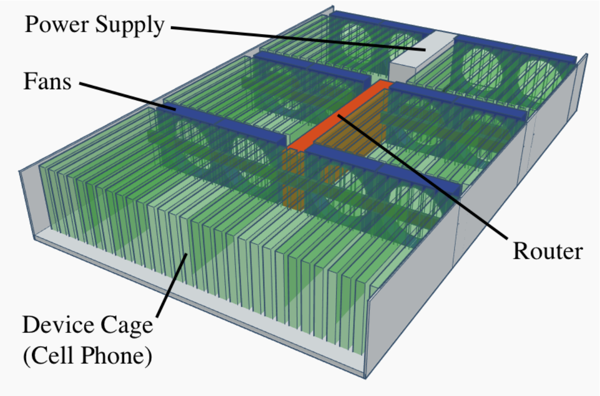


Figure 2: A standard 19-inch 2U server can gracefully lodge 84 typical mobile devices, 3 rows of fans, a network router, and a power supply.

Server weight might be one possible concern with the dense deployment of mobile devices in standard racks since such mobile devices have many extraneous components. The average weight for devices fitting in our cage is about $160g$. Together with other elements, we estimate the overall server component weight to be under $20kg$, which is below the typical 2U weight of $25kg$.

4.2 Networking

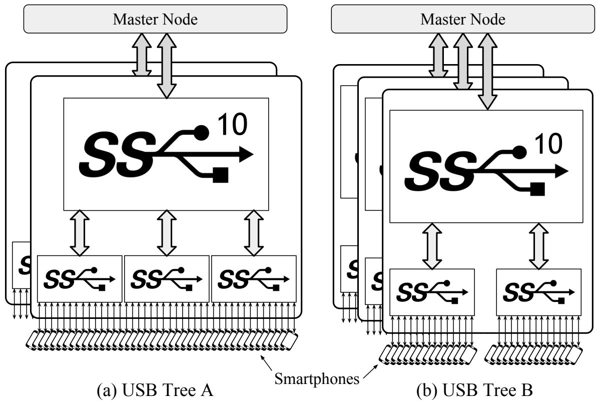
Mobile devices are equipped with multiple communication means such as USB ports, Wi-Fi, Bluetooth, etc. Büsching et al. has compared USB and WiFi to build a small smartphone cluster (6 devices) running the Linpack benchmark. They concluded that USB links scale slightly better [9]. Our proposed platform, however, is at a much larger scale, having 84 devices per-server. In this section, we investigate and compare a number of different possible network architectures. Table 1 compares the performance for those network solutions.

1. USB Tree with a Master Node: A USB connection is required to power each device. A USB tree can be formed to connect all mobile devices to a master node. The master can then establish virtual Ethernet interfaces for mobile devices [9]. Figure 3 shows how such USB trees can be built efficiently, by matching the hubs to the uplink/downlink bandwidths. Here, the root hubs support USB 3.1 and their underlying hubs support USB 3.0.

2. USB On-The-Go (OTG): USB OTG enables Ethernet over USB for mobile devices using USB-Ethernet adapters [11, 10]. Special cables should be used to support charging while communicating (Figure 4). As shown in Table 1, this solution can offer much higher network performance compared to the USB tree, depending on the capability of the network switch that mobile devices connect to. However, using more Ethernet ports requires more network switches, leading to a higher cost [6].

Solution	Connection to Infrastructure	Max. BW/Device	Avg. BW/Device
USB Tree A	2xUSB 3.1 to a Master Node	5Gbps	244Mbps
USB Tree B	3xUSB 3.1 to a Master Node	5Gbps	366Mbps
Ethernet via USB OTG	Mellanox Edgecore AS4610 1GbE	1Gbps	853Mbps
WiFi with 1Gbps Router	Ethernet to TOR	867Mbps (802.11ac)	12Mbps (Router Bottleneck)
WiFi with 10Gbps Router	Ethernet to TOR	867Mbps (802.11ac)	83Mbps (Channel Bottleneck)

Table 1: Peak and mean performance comparison of multiple proposed network architectures.



(a) USB Tree A

(b) USB Tree B

Figure 3: Two possible architectures for connecting mobile device using USB hubs in a server.

3. Intra-server WiFi communication: WiFi is another possible means of communicating with mobile devices in the server. This could be done by having a WiFi router inside each of these mobile farm servers. The metal case of the server attenuates the radio waves and the congestion risk with outside entities is reduced. However, due to the high density of transmitters and receivers, we anticipate the signal to noise (SNR) ratios to be very low and the channel to be highly congested. This means reduced bandwidth and more energy per bit for wireless communication. Although such a factor should be evaluated through precise empirical measurements, we think overcoming this issue with current WiFi standards can be fairly tricky. However, WiFi can still be used as a cheap complementary or redundant communication channel for deployed mobile devices.

4.3 Mobile Batteries as Distributed UPSs

Researchers have proposed using distributed UPSs [20] or batteries [2] to shave peak power in data centers. This allows installing more servers using the same power infrastructure and decreases the TCO.

The average battery capacity for devices we studied is slightly more than 3100mAh. Battery degradation depends on the number of charging cycles, but even assuming a capacity loss of 15% per year, a 3-year-old device would have ~61% of its capacity left (~1900mAh). Therefore, we can approximate a server with 84 used mobile devices to have a total capacity of ~160Ah. This much capacity on a single server is 8 times denser than batteries used by Aksanli et al. [2] and 4 times denser than distributed UPSs used by Kontorinis et al. [20].

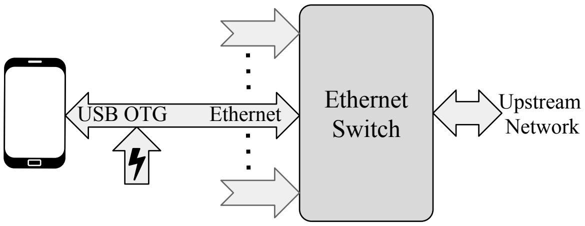


Figure 4: Mobile devices can connect to Ethernet via USB On-The-Go (OTG) and be charged simultaneously.

The important takeaway here is that such a high battery capacity is a byproduct of deploying mobile devices and comes at no extra cost. Distributed batteries effectively dampen temporal power demand variations; shaving the peak power under high utilization, while storing energy under low utilization. The high energy storage density enables more aggressive power capping of servers that are filled with used mobile devices.

4.4 Total Cost of Ownership Analysis

In order to have a fair comparison between our proposed system and existing servers, we select a server which has roughly similar performance as our 84-node server. We use Geekbench 4 [14], which is a cross-platform benchmark. We pick the Samsung Galaxy Note 4 as a three-year-old device. This phone's multicore Geekbench 4 performance grade is 3060. As seen in Table 2, to have a similar performance density, we select the Lenovo Flex System x880 X6 server [21] with two Intel Xeon E7-8890 v3 processors. Although this server supports up to 48 DDR3 memory DIMMs, we consider using only eighteen[2] 8GB DDR3 DIMMs (144 GB) to have a similar RAM density. Also, using two 800GB SSDs leads to a very similar storage density to the case where only internal storage of smartphones (32 GB each) is used.

Table 3 presents the details of our TCO analysis. We used the OuterVision Power Supply Calculator [28] to estimate the power consumption of non-CPU components of System A as listed in Table 2 and based our CPU power calculations on 90% of TDP. We also assumed our

[2] Although 19 DIMMs lead to a more similar number, for this specific server DIMMs should be used in pairs.

System	Server	CPU/SoC	Server Height	CPU per Server	Grade per CPU	Total RAM	Total Storage	Per Rack Unit (1U)		
								Grade	Memory	Storage
A	Lenovo Flex System x880 X6 [21]	Intel Xeon E7-8890 v3	1.2U	2	74482 [14]	144 GB (18×8)	1.56 TB (2×800GB)	124,137	120 GB	1.30 TB
B	Proposed Server	Snapdragon 805 (Samsung Galaxy Note 4)	2U	84	3060 [14]	252 GB (84×3)	2.63 TB (84×32GB)	128,520	126 GB	1.32 TB

Table 2: Platforms chosen for total cost of ownership (TCO) study.

TCO Component	Parameter	Server A Cost ($)	Server B Cost ($)
	Server Unit Price	$26,545	-
	Memory DIMMs	18×$95=$1,710	-
	Intel 800GB SSD	2×$620=$1,240	-
	Used Galaxy Note 4	-	84×$160=$13,440
	Power Supply	-	$50-$150
	Server Fans	-	12×$10 = $120
	USB Cabling	-	~$100
	Network Router(s)	-	~$300
	Server Case	-	~$50
	Master Node (Lenovo ThinkServer RS160 [22])	-	~$725
	2-port 10Gb Network Card	Included in Unit Price	~$400
CAPEX	Total	~$29,495	~$15,235
	CPU per Server	2	84
	TDP per CPU	165W [21]	3.5W [16]
	CPU Utilization	90% TDP	90% TDP
	Non-CPU Power	~200W [28]	-
	Router Power	-	20W [34]
	Fan Power	-	~25W
	Master Node Power (NIC Included)	-	~165W [28]
	Total Power	~500W [28]	~475W
	Monthly kWh	~360	~342
	Electricity Unit Cost [6]	$0.10/kWh	$0.10/kWh
	Electricity Cost	$36.0	$34.2
	Server Height (Rack Unit)	1.2	2 + 1(Master Node)
	Facility Space Depreciation [20]	$4.15	$10.38
	Infrastructure Depreciation [20]	$8.54	$8.54
	Monitoring, Engineering, and Installation	$9.13 [20]	2×$9.13=$18.26
	PUE Overhead (PUE=1.8 [6])	$28.8	$27.4
OPEX (monthly)	Total	$86.62	$98.74

Table 3: Capital expenditure (CAPEX) and operational expense (OPEX) for two systems described in Table 2.

proposed server to have twice the monitoring, engineering, and installation cost compared to a standard server.

Using the CAPEX and OPEX values calculated in Table 3 we performed TCO analysis, results of which are reflected in Figure 5. To include residual values after deployment lifetime for these systems, we considered annual depreciation rates (δ) of 45% and 77.5%, which translate into only 5% of initial value after 5 and 2 years, respectively. We also studied the case where servers have no residual value after deployment (δ=100%), which occurs when they cannot be resold. The right sub-figures in Figure 5 compare TCO when those two servers have different lifetimes. This analysis is essential for a fair comparison because we anticipate our proposed server to have a shorter lifetime compared to a new high-end server. It can be seen that with much shorter lifetimes, our proposed server can deliver better TCO values. It also shows how the equal-TCO margin (the line between light and dark areas) varies for different depreciation rates.

Server A that can achieve a similar density of performance, memory, and storage compared to our proposed server (Server B) has a much higher CAPEX, with slightly lower OPEX. Figure 5 indicates that for a typical deployment cycle of an expensive high-end server, CAPEX is the dominant term. That is the motivation behind deploying used mobile devices that have low CAPEX but can be used to build dense servers with low power consumption. The power consumption of our proposed server together with a master node is slightly lower than the high-end server we are comparing against.

5 Discussion and Future Work

The debate around “wimpy” (low-performance) versus “brawny” (high-performance) cores has been ongoing for the past decade with proponents on both sides [4, 26, 17, 18]. Undoubtedly, their relative merits highly depend on the application and environment. However, we

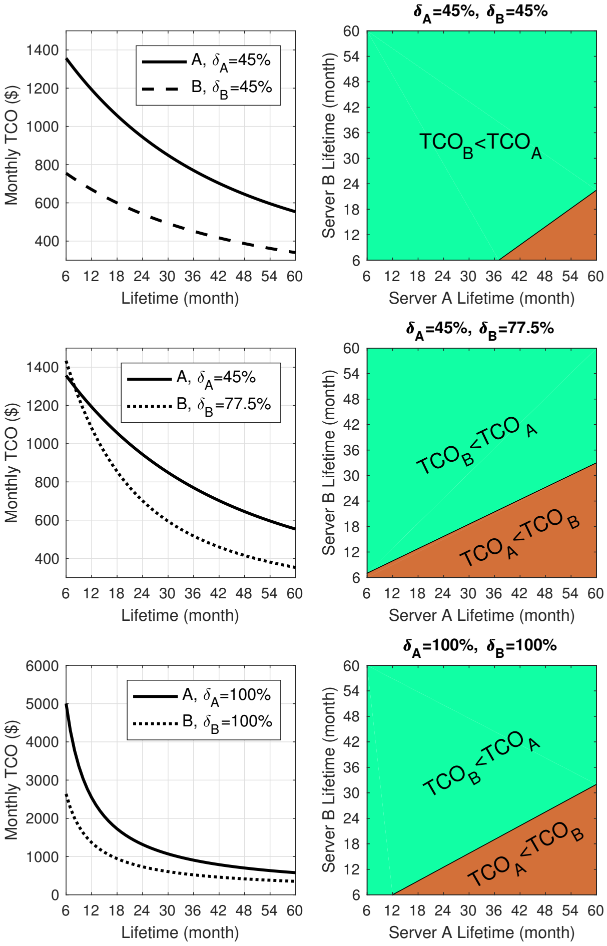


Figure 5: TCO analysis of sytems described in the Table 2 with different annual depreciation rate (δ) values. In the last case, $\delta = 100\%$ means no residual value.

believe some trends should be considered in that discussion. First, wimpy cores have evolved to catch up with higher performance processors, narrowing the gap; second, wimpy and brawny cores can be deployed heterogeneously; third, manycore architectures are widely used in data centers today [23] and utilizing high parallelism is not unusual; and finally, some emerging applications support very high parallelism [13]. We expect this work to fuel the wimpy versus brawny discussion regarding those aspects.

In this paper, we mentioned four classes of applications that can benefit from the proposed environment, some of which are unprecedented. In future work, we will explore additional killer applications for this platform.

Reliability of used mobile devices can be a concern since mobile SoCs and memories (DRAM and flash) might age faster under continuous data center workloads. Although we modeled this in our TCO analysis using a higher depreciation rate, empirical tests are required to precisely model smartphone aging under such workloads.

6 Conclusion

In this work, we proposed deploying decommissioned mobile devices in clusters as well as IaaS clouds. We argue that this makes economic sense considering the CPU performance trend for those devices and enumerate a few major motivations to do so, including decreasing the global warming potential of mobile devices by extending their lifetime. We listed a number of applications that can benefit from our proposed architecture. We discussed how those mobile devices can be integrated into the data center and performed a TCO analysis to compare our proposed server with a conventional server that has similar computational, memory, and storage densities. The results showed that our proposed system is cheaper even with a higher depreciation rate.

Acknowledgments

This work was partially supported by the NSF under Grants No. CCF-1217553, CCF-1453112, and CCF-1438980, AFOSR under Grant No. FA9550-14-1-0148, and DARPA under Grant No. N66001-14-1-4040. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of our sponsors.

References

[1] Ericsson mobility report. Tech. rep., Ericsson, Stockholm, Sweden, June 2016. https://www.ericsson.com/res/docs/2016/ericsson-mobility-report-2016.pdf.

[2] Aksanli, B., Pettis, E., and Rosing, T. Architecting efficient peak power shaving using batteries in data centers. In *2013 IEEE 21st International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems* (Aug 2013), pp. 242–253.

[3] Amazon Web Services. Amazon EC2 P2 instances. https://aws.amazon.com/ec2/instance-types/p2/, Accessed: 2017-03-12.

[4] Andersen, D. G., Franklin, J., Kaminsky, M., Phanishayee, A., Tan, L., and Vasudevan, V. FAWN: A fast array of wimpy nodes. In *Proceedings of the ACM SIGOPS 22nd Symposium on Operating Systems Principles* (New York, NY, USA, 2009), SOSP '09, ACM, pp. 1–14.

[5] Barbalace, A., Lyerly, R., Jelesnianski, C., Carno, A., Chuang, H.-R., Legout, V., and Ravindran, B. Breaking the boundaries in heterogeneous-ISA datacenters. In *Proceedings of the 22nd Int. Conference on Architectural Support for Programming Languages and Operating Systems* (New York, NY, USA, 2017), ASPLOS '17, ACM, pp. 645–659.

[6] Barroso, L. A., Clidaras, J., and Hölzle, U. The datacenter as a computer: An introduction to the design of warehouse-scale machines, second edition. *Synthesis Lectures on Computer Architecture 8*, 3 (2013), 1–154.

[7] Barroso, L. A., and Hölzle, U. The case for energy-proportional computing. *Computer 40*, 12 (Dec 2007), 33–37.

[8] Bugnion, E., Nieh, J., and Tsafrir, D. Hardware and software support for virtualization. *Synthesis Lectures on Computer Architecture 12*, 1 (2017), 1–206.

[9] Büsching, F., Schildt, S., and Wolf, L. DroidCluster: Towards smartphone cluster computing – the streets are paved with potential computer clusters. In *2012 32nd International Conference on Distributed Computing Systems Workshops* (June 2012), pp. 114–117.

[10] Choi, P., Gao, J., Ramanathan, N., Mao, M., Xu, S., Boon, C. C., Fahmy, S. A., and Peh, L. S. A case for leveraging 802.11p for direct phone-to-phone communications. In *2014 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)* (Aug 2014), pp. 207–212.

[11] Curti, M., Merlo, A., Migliardi, M., and Schiappacasse, S. Towards energy-aware intrusion detection systems on mobile devices. In *2013 International Conference on High Performance Computing Simulation (HPCS)* (July 2013), pp. 289–296.

[12] Ercan, M., Malmodin, J., Bergmark, P., Kimfalk, E., and Nilsson, E. Life cycle assessment of a smartphone. In *4th International Conference on ICT for Sustainability (ICT4S 2016)* (2016).

[13] Fouladi, S., Wahby, R. S., Shacklett, B., Balasubramaniam, K. V., Zeng, W., Bhalerao, R., Sivaraman, A., Porter, G., and Winstein, K. Encoding, fast and slow: Low-latency video processing using thousands of tiny threads. In *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)* (Boston, MA, 2017), USENIX Association, pp. 363–376.

[14] Geekbench Software. Geekbench browser. https://browser.primatelabs.com, Accessed: 2017-03-14.

[15] Google Cloud Platform. GPUs on compute engine. https://cloud.google.com/compute/docs/gpus/, Accessed: 2017-03-12.

[16] Halpern, M., Zhu, Y., and Reddi, V. J. Mobile CPU's rise to power: Quantifying the impact of generational mobile CPU design trends on performance, energy, and user satisfaction. In *2016 IEEE International Symposium on High Performance Computer Architecture (HPCA)* (March 2016), pp. 64–76.

[17] Janapa Reddi, V., Lee, B. C., Chilimbi, T., and Vaid, K. Web search using mobile cores: Quantifying and mitigating the price of efficiency. In *Proceedings of the 37th Annual International Symposium on Computer Architecture* (New York, NY, USA, 2010), ISCA '10, ACM, pp. 314–325.

[18] Kanev, S., Darago, J. P., Hazelwood, K., Ranganathan, P., Moseley, T., Wei, G. Y., and Brooks, D. Profiling a warehouse-scale computer. In *2015 ACM/IEEE 42nd Annual International Symposium on Computer Architecture (ISCA)* (June 2015), pp. 158–169.

[19] Keeton, K. The Machine: An architecture for memory-centric computing. In *Workshop on Runtime and Operating Systems for Supercomputers (ROSS)* (2015).

[20] Kontorinis, V., Zhang, L. E., Aksanli, B., Sampson, J., Homayoun, H., Pettis, E., Tullsen, D. M., and Rosing, T. S. Managing distributed UPS energy for effective power capping in data centers. In *2012 39th Annual International Symposium on Computer Architecture (ISCA)* (June 2012), pp. 488–499.

[21] Lenovo Inc. Lenovo Flex system X6 compute nodes (7196), 2017. https://lenovopress.com/tips1286.pdf.

[22] Lenovo Inc. Lenovo ThinkServer RS160, 2017. https://lenovopress.com/lp0553.pdf.

[23] McKeown, M., Fu, Y., Nguyen, T., Zhou, Y., Balkind, J., Lavrov, A., Shahrad, M., Payne, S., and Wentzlaff, D. Piton: A manycore processor for multitenant clouds. *IEEE Micro 37*, 2 (Mar 2017), 70–80.

[24] Microsoft Azure. Virtual machine series. https://azure.microsoft.com/en-us/pricing/details/virtual-machines/series/#n-series, Accessed: 2017-03-12.

[25] Motakis, A., Rigo, A., and Raho, D. Platform device assignment to KVM-on-ARM virtual machines via VFIO. In *2014 12th IEEE International Conference on Embedded and Ubiquitous Computing* (Aug 2014), pp. 170–177.

[26] Mudge, T., and Hölzle, U. Challenges and opportunities for extremely energy-efficient processors. *IEEE Micro 30*, 4 (July 2010), 20–24.

[27] Nider, J., and Rapoport, M. Cross-ISA container migration. In *Proceedings of the 9th ACM International on Systems and Storage Conference* (New York, NY, USA, 2016), SYSTOR '16, ACM, pp. 24:1–24:1.

[28] OuterVision. Outervision® power supply calculator. https://outervision.com/power-supply-calculator, Accessed: 2017-05-14.

[29] PassMark. PerformanceTest mobile. http://www.passmark.com/products/pt_mobile.htm, Accessed: 2017-03-07.

[30] Passmark Software. Interpreting your results from performancetest. http://www.passmark.com/support/performancetest/interpreting_test_results.htm, Accessed: 2017-03-14.

[31] Rajovic, N., Carpenter, P. M., Gelado, I., Puzovic, N., Ramirez, A., and Valero, M. Supercomputing with commodity CPUs: Are mobile SoCs ready for HPC? In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis* (New York, NY, USA, 2013), SC '13, ACM, pp. 40:1–40:12.

[32] Rajovic, N., Rico, A., Puzovic, N., Adeniyi-Jones, C., and Ramirez, A. Tibidabo: Making the case for an ARM-based HPC system. *Future Generation Computer Systems 36* (2014), 322 – 334.

[33] Shahrad, M., and Wentzlaff, D. Availability Knob: Flexible user-defined availability in the cloud. In *Proceedings of the Seventh ACM Symposium on Cloud Computing* (New York, NY, USA, 2016), SoCC '16, ACM, pp. 42–56.

[34] TPCDB. The power consumption database. http://www.tpcdb.com/list.php?type=11, Accessed: 2017-03-14.

[35] Tudor, B. M., and Teo, Y. M. On understanding the energy consumption of ARM-based multicore servers. In *Proceedings of the ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems* (New York, NY, USA, 2013), SIGMETRICS '13, ACM, pp. 267–278.

[36] Vasudevan, V., Franklin, J., Andersen, D. G., Phanishayee, A., Tan, L., Kaminsky, M., and Moraru, I. Fawndamentally power-efficient clusters. In *HotOS* (2009), vol. 9, pp. 22–22.

[37] Venkat, A., Shamasunder, S., Shacham, H., and Tullsen, D. M. HIPStR: Heterogeneous-ISA program state relocation. In *Proceedings of the 21st International Conference on Architectural Support for Programming Languages and Operating Systems* (New York, NY, USA, 2016), ASPLOS '16, ACM, pp. 727–741.

[38] Zhou, Y., and Wentzlaff, D. The sharing architecture: Subcore configurability for IaaS clouds. In *Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems* (New York, NY, USA, 2014), ASPLOS '14, ACM, pp. 559–574.