ASIC Clouds: Specializing the Datacenter

Ikuo Magaki+, Moein Khazraee, Luis Vega Gutierrez, and
Michael Bedford Taylor

+UC San Diego, Toshiba UC San Diego

ABSTRACT

GPU and FPGA-based clouds have already demonstrated the promise of accelerating computing-intensive workloads with greatly improved power and performance.

In this paper, we examine the design of ASIC Clouds, which are purpose-built datacenters comprised of large arrays of ASIC accelerators, whose purpose is to optimize the total cost of ownership (TCO) of large, high-volume chronic computations, which are becoming increasingly common as more and more services are built around the Cloud model. On the surface, the creation of ASIC clouds may seem highly improbable due to high NREs and the inflexibility of ASICs. Surprisingly, however, large-scale ASIC Clouds have already been deployed by a large number of commercial entities, to implement the distributed Bitcoin cryptocurrency system.

We begin with a case study of Bitcoin mining ASIC Clouds, which are perhaps the largest ASIC Clouds to date. From there, we design three more ASIC Clouds, including a YouTube-style video transcoding ASIC Cloud, a Litecoin ASIC Cloud, and a Convolutional Neural Network ASIC Cloud and show 2-3 orders of magnitude better TCO versus CPU and GPU.

Among our contributions, we present a methodology that given an accelerator design, derives Pareto-optimal ASIC Cloud Servers, by extracting data from place-and-routed circuits and computational fluid dynamic simulations, and then employing clever but brute-force search to find the best jointly-optimized ASIC, DRAM subsystem, motherboard, power delivery system, cooling system, operating voltage, and case design. Moreover, we show how data center parameters determine which of the many Pareto-optimal points is TCO-optimal. Finally we examine when it makes sense to build an ASIC Cloud, and examine the impact of ASIC NRE.

1. INTRODUCTION

In the last ten years, two parallel phase changes in the computational landscape have emerged. The first change is the bifurcation of computation into two sectors: cloud and mobile; where increasingly the heavy lifting and data-intensive codes are performed in warehouse-scale computers or datacenters; and interactive portions of applications have migrated to desktop-class implementations of out-of-order superscalars in mobile phones and tablets.

The second change is the rise of dark silicon [1, 2, 3, 4] and dark silicon aware design techniques [5, 6, 7, 8, 9, 10] such as specialization and near-threshold computation, each of which help overcome threshold scaling limitations that prevent the full utilization of transistors on a silicon die.

Accordingly, these areas have increasingly become the focus of the architecture research community. Recently, researchers and industry have started to examine the conjunction of these two phase changes. GPU-based clouds have been demonstrated as viable by Baidu and others who are building them in order to develop distributed neural network accelerators. FPGA-based clouds have been validated and deployed by Microsoft for Bing [11], by JP Morgan Chase for hedgefund portfolio evaluation [12] and by almost all Wall Street firms for high frequency trading [13]. In these cases, companies were able to ascertain that there was sufficient scale for the targeted application that the upfront development and capital costs would be amortized by a lower total cost of ownership (TCO) and better computational properties. Already, we have seen early examples of customization, with Intel providing custom SKUs for cloud providers [14].

At a single node level, we know that ASICs can offer order-magnitude improvements in energy-efficiency and cost-performance over CPU, GPU, and FPGA. In this paper, we extend this trend and consider the possibility of ***ASIC Clouds***. ASIC Clouds are purpose-built datacenters comprised of large arrays of ASIC accelerators, whose purpose is to optimize the TCO of large, high-volume chronic computations that are emerging in datacenters today. ASIC Clouds are not ASIC supercomputers that scale up problem sizes for a single tightly-coupled computation; rather, ASIC Clouds target workloads consisting of many independent but similar jobs (eg., the same function, but for many users, or many datasets), for which standalone accelerators have been shown to attain improvements for individual jobs.

As more and more services are built around the Cloud model, we see the emergence of planet-scale workloads. For example, Facebook's face recognition algorithms are used on 2 billion uploaded photos a day, each requiring several seconds on a GPU [15], Siri answers speech queries, genomics will be applied to personalize medicine, and YouTube transcodes all user-uploaded videos to Google's VP9 format. As computations of this scale become increasingly frequent, the TCO improvements derived from the reduced marginal hardware and energy costs of ASICs will make it an easy and routine business decision to create ASIC Clouds.

ASIC Clouds Exist Today. This paper starts by examining the first large-scale ASIC Clouds, Bitcoin cryptocurrency mining clouds, as real-world case studies to understand the key issues in ASIC Cloud design. Bitcoin clouds implement the consensus algorithms in Bitcoin cryptocurrency systems. Although much is secretive in the Bitcoin mining industry, today there are 20 megawatt facilities in existence, and 40 megawatt facilities are under construction [16], and the global power budget dedicated to ASIC Clouds, large and small, is estimated by experts to be in the range of 300-500 megawatts. After Bitcoin, the paper then examines other applications including YouTube-style video transcoding, Litecoin mining and Convolutional Neural Networks.

Specializing the ASICs. At the heart of every ASIC Cloud is an ASIC design, which typically aggregates a number of accelerators into a single chip. ASICs achieve large reductions in silicon area and energy consumption versus CPUs,

The first two authors contributed equally. To Appear in ISCA 2016.

GPUs, and FPGAs because they are able to exactly provision the required resources needed for the computation. They can replace area-intensive, energy-wasteful instruction interpreters with area-efficient, energy-efficient parallel circuits. ASIC designers can dial in exactly the optimal voltage and thermal profile for the computation. They can customize the I/O resources, instantiating precisely the right number of DRAM, PCI-e, HyperTransport and Gig-E controllers, and employ optimized packages with optimal pin allocation.

Bitcoin ASIC specialization efforts have been prolific: over 27 unique Bitcoin mining ASICs have been successfully implemented in the last three years [17]. The first three ASICs were developed in 130 nm, 110 nm, and 65 nm, respectively; 55 nm and 28 nm versions followed quickly afterwards. Today, you can find chips manufactured in the state-of-the art FinFET technologies: Intel 22 nm and TSMC 16 nm. Although many original designs employed standard-cell design, competitive designs are full-custom, have custom packages, and as of 2016, operate at near-threshold voltages.

Specializing the ASIC Server. In addition to exploiting specialization at the ASIC design level, ASIC Clouds can specialize the server itself. A typical datacenter server is encrusted with a plethora of x86/PC support chips, multi-phase voltage regulators supporting DVFS, connectors, DRAMs, and I/O devices, many of which can be stripped away for a particular application. Moreover, typical Xeon servers embody a CPU-centric design, where computation (and profit!) is concentrated in a very small area of the PCB, creating extreme hotspots. This results in heavy-weight local cooling solutions that obstruct delivery of cool air across the system, resulting in sub-optimal system-level thermal properties. ASIC Servers in Bitcoin ASIC Clouds integrate arrays of ASICs organized evenly across parallel shotgun-style airducts that use wide arrays of low-cost heatsinks to efficiently transfer heat out of the system and provide uniform thermal profiles. ASIC Cloud servers use a customized printed circuit board, specialized cooling systems and specialized power delivery systems, and can customize the DRAM type (e.g., LP-DDR3, DDR4, GDDR5, HBM...) and DRAM count for the application at hand, as well as the minimal necessary I/O devices and connectors required. Further, they employ custom voltages in order to tune TCO.

Specializing the ASIC Datacenter. ASIC Clouds can also exploit specialization at the datacenter level, optimizing rack-level and datacenter-level thermals and power delivery that exploit the uniformity of the system. More importantly, cloud-level parameters (e.g., energy provisioning cost and availability, depreciation and ... taxes) are pushed down into the server and ASIC design to influence cost- and energy- efficiency of computation, producing the TCO-optimal design.

Analyzing Four Kinds of ASIC Clouds. In this paper we begin by analyzing Bitcoin mining ASIC Clouds in depth, and distill both their unique characteristics and characteristics that are likely to apply across other ASIC Clouds. We develop the tools for designing and analyzing Pareto- and TCO- optimal ASIC Clouds. By considering ASIC Cloud chip design, server design, and finally data center design in a bottom-up way, we reveal how the designers of these novel systems can optimize the TCO in real-world ASIC Clouds.

From there, we examine other ASIC Clouds designs, extending the tools for three exciting emerging cloud workloads: YouTube-style Video Transcoding, Litecoin mining and Convolutional Neural Networks.

When To Go ASIC Cloud. Finally, we examine when it makes sense to design and deploy an ASIC Cloud, considering NRE. Since inherently ASICs and ASIC Clouds gain their benefits from specialization, each ASIC Cloud will be specialized using its own combination of techniques. Our experience suggests that, as with much of computer architecture, many techniques are reused and re-combined in different ways to create the best solution for each ASIC Cloud.

2. BITCOIN: AN EARLY ASIC CLOUD

In this section, we overview the underlying concepts in the Bitcoin cryptocurrency system embodied by Bitcoin ASIC Clouds. An overview of the Bitcoin cryptocurrency system and an early history of Bitcoin mining can be found in [18].

Cryptocurrency systems like Bitcoin provide a mechanism by which parties can semi-anonymously and securely transfer money between each other over the Internet. Unlike closed systems like Paypal or the VISA credit card system, these systems are open source and run in a distributed fashion across a network of untrusted machines situated all over the world. The primary mechanism that these machines implement is a global, public ledger of transactions, called the *blockchain*. This blockchain is replicated many times across the world. Periodically, every ten minutes or so, a block of new transactions is aggregated and posted to the ledger. All transactions since the beginning can be inspected[1].

Mining. A distributed consensus technique called *Byzantine Fault Tolerance* determines whose transactions are added to the blockchain, in the following way. Machines on the network request work to do from a third-party *pool server*. This work consists of performing an operation called *mining*, which is a computationally intense operation which involves brute force partial inversion of a cryptographically hard hash function like SHA256 or scrypt. The only known way to perform these operations is to repeatedly try a new inputs, and run the input through the cryptographic function and see if the output has the requisite number of starting zeros. Each such attempt is a called a *hash*, and the number of hashes that a machine or group of machines performs is called its *hashrate*, which is typically quote in terms of billions of hashes per second, or *gigahash per second* (*GH/s*). When a machine succeeds, it will broadcast that it has added a block to the ledger, and the input value is the *proof of work* that it has played by the rules. The other machines on the network will examine the new block, determine if the transaction is legitimate (i.e. did somebody try to create currency, or transfer more money than was available from a particular account, or is the proof-of-work invalid), and if it is, they will use this new updated chain and attempt to post their transactions to the end of the new chain. In the infrequent case where two machines on the network have found a winning hash and broadcasted new blocks in parallel, and the chain has "forked", the long version has priority.

The first two ASIC Clouds analyzed in this paper target mining for the two most dominant distributed cryptocurrencies: Bitcoin and Litecoin. People are incentivized to perform mining for three reasons. First, there is an ideological reason: the more machines that mine, the more secure the cryptocurrency network is from attacks. Second, every time a machine succeeds in posting a transaction to the blockchain, it receives a *blockchain reward* by including a payment transaction to its own account. In the case of Bitcoin, this reward is substantial: 25 bitcoins (or BTC), valued

[1] See `http://blockchain.info` to see real-time ledger updates.

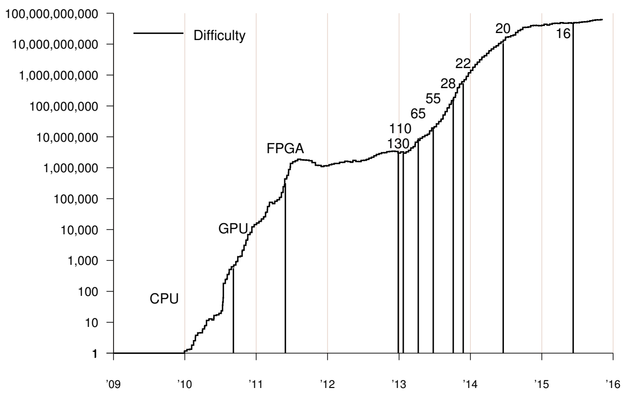

Figure 1: **Rising Global Bitcoin ASIC Computation and Corresponding Increase in Bitcoin ASIC Cloud Specialization.** Numbers are ASIC nodes, in nm. Difficulty is the ratio of the current world Bitcoin hash throughput relative to the initial mining network throughput, 7.15 MH/s. In the six-year period preceding Nov 2015, throughput has increased by 50 billion times, corresponding to a world hash rate of approximately 575 million GH/s. The first release date of a miner on each ASIC node is annotated.

at $10,725 on the BTC-USD exchanges in April 2016. Since approximately 144 blocks are mined per day, the total value per day of mining is around $1.5M USD. Mining is the only way that currency is created in the Bitcoin system. Third, the machine also receives optional tips attached to the transaction; these tips comprise only a few percent of revenue.

In order to control the rate at which new Bitcoin are created, approximately every 2016 blocks (or two weeks), the difficulty of mining is adjusted by increasing the number of leading zeros required, according to how fast the last group of 2016 blocks was solved. Thus, with slight hysteresis, the fraction of the 3600 bitcoins distributed daily that a miner receives is approximately proportional to the ratio of their hashrate to the world-wide *network hashrate.*

Economic Value of Bitcoins. Bitcoins have become increasingly valuable over time as demand increases. The value started at around $0.07, and increased by over 15,000× to over $1,000 in late 2013. Since then, the price has stabilized, and as of late April 2016, is around $429, and over $6.5 billion USD worth of BTC are in circulation today. Famously, early in the Bitcoin days, a pizza was purchased for 10,000 BTC, worth $4.3 million USD today. The value of a BTC multiplied by yearly number of BTC mined determines in turn the yearly revenue of the entire Bitcoin mining industry, which is currently at $563M USD per year.

3. RAMPING THE TECHNOLOGY CURVE TO ASIC CLOUD

As BTC value exponentially increased, the global amount of mining has increased greatly, and the effort and capital expended in optimizing machines to reduce TCO has also increased. This effort in turn increases the capabilities and quantity of machines that are mining today. Bitcoin ASIC Clouds have rapidly evolved through the full spectrum of specialization, from CPU to GPU, from GPU to FPGA, from FPGA to older ASIC nodes, and finally to the latest ASIC nodes. ASIC Clouds in general will follow this same evolution: rising TCO of a particular computation justifies increasingly higher expenditure of NRE's and development costs, leading to greater specialization.

Figure 1 shows the corresponding rise in total global network hashrate over time, normalized to the difficulty running on a few CPUs. The difficulty and hashrate have increased by an incredible factor of 50 billion since 2009, reaching approximately 575 million GH/s as of November 2015.

By scavenging data from company press releases, blogs, bitcointalk.org, and by interviewing chip designers at these companies, we have reconstructed the progression of technology in the Bitcoin mining industry, which we annotate on Figure 1, and describe in this section.

Gen 1-3. The first generation of Bitcoin miners were CPU's, the second generation were GPU's and the third generation were FPGAs. See [18] for more details.

Gen 4. The fourth generation of Bitcoin miners started with the first ASIC (ASICMiner, standard cell, 130-nm) that was received from fab in late December 2012. Two other ASICs (Avalon, standard cell, 110-nm and Butterfly Labs, full custom, 65-nm) were concurrently developed by other teams with the first ASIC and released shortly afterwards. These first ASICs, built on older, cheaper technology nodes with low NREs, served to confirm the existence of a market for specialized Bitcoin mining hardware.

These first three ASICs had different mechanisms of deployment. ASICMiner sold shares in their firm on an online bitcoin-denominated stock exchange, and then built their own mining datacenter in China. *Thus, the first ASICs developed for Bitcoin were used to create an ASIC Cloud system.* The bitcoins mined were paid out to the investors as dividends. Because ASICMiner did not have to ship units to customers, they were the first to be able to mine and thus captured a large fraction of the total network hash rate. Avalon and Butterfly Labs used a Kickstarter-style pre-order sales model, where revenue from the sales funded the NRE of the ASIC development. As the machines become available, they were shipped sequentially by customer order date.

Gen 5. The fifth generation of Bitcoin miners started when, upon seeing the success of the first group of ASICs, a second group of firms with greater capitalization developed and released the second wave of ASICs which used better process technology. Bitfury was the first to reach 55-nm in mid 2013 with a best-of-class full custom implementation, then Hashfast reached 28-nm in Oct. 2013, and there is evidence that 21, Inc hit the Intel 22-nm node around Dec 2013.

Gen 6. The current generation, the fifth generation of mining ASICs, is by companies that survived the second wave, and targets bleeding edge nodes as they came out (e.g. TSMC 20-nm and TSMC 16-nm). So far, these advanced nodes have only been utilized by ASIC manufacturers whose intent is to populate and run their own ASIC Clouds.

Moving to Cloud Model. Most companies that build Bitcoin mining ASICs, such as Swedish firm KnCminer, have moved away from selling hardware to end users, and instead now maintain their own private clouds [19], which are located in areas that have low-cost energy and cooling. For example KnCminer has a facility in Iceland, because there is geothermal and hydroelectric energy available at extremely low cost, and because cool air is readily available. Bitfury created a 20 MW mining facility in the Republic of Georgia, where electricity is also cheap. Their datacenter was constructed in less than a month, and they have raised funds for a 100 MW data center in the future.

Optimizing TCO. Merged datacenter operation and ASIC development have become the industry norm for several reasons. First, the datacenter, enclosing server and the ASIC

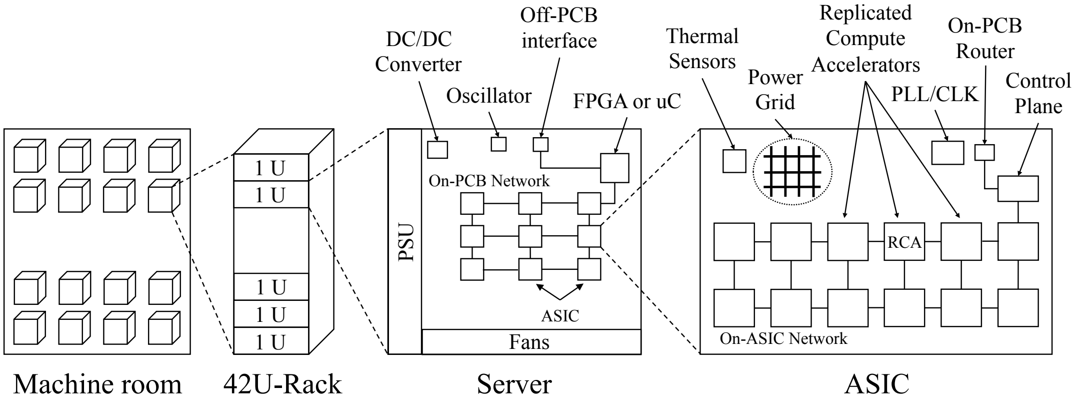

Figure 2: **High-Level Abstract Architecture of an ASIC Cloud.**

can be co-designed with fewer unknowns, eliminating the need to accommodate varying customer environments (energy cost, temperature, customs and certifications, 220V/110V, setup guides, tech support...) and enabling new kinds of optimizations that trade off cost, energy efficiency and performance. Second, ASIC Cloud bringup time is greatly shortened if the product does not have to be packaged, troubleshooted and shipped to the customer, which means that the chips can be put into use earlier. Finally, meeting an exact target for an ASIC chip is a challenging process, and tuning the system until it meets the promised specifications exactly (energy efficiency, performance) before shipping to the customer delays the deployment of the ASICs and the time at which they can start reducing TCO of the computation at hand.

4. PARETO- AND TCO- OPTIMALITY

In ASIC Clouds, two key metrics define the design space: hardware cost per performance ($ per op/s, which for Bitcoin is $ per GH/s), and energy per operation (Watts per op/s, equivalent to Joules per op, which for Bitcoin is W per GH/s). Designs can be evaluated according to these metrics, and mapped into a Pareto space that trades cost and energy efficiency. **Joint knowledge and control over datacenter and hardware design allows for the ASIC designers to select the single TCO-optimal point by correctly weighting the importance of cost per performance and energy per op among the set of Pareto-optimal points.**

5. ARCHITECTURE OF AN ASIC CLOUD

We starting by examining the design decisions that apply generally across ASIC Clouds. Later, we design four example ASIC Cloud for Bitcoin, Litecoin, Video Transcoding, and Convolutional Neural Networks.

At the heart of any ASIC Cloud is an energy-efficient, high-performance, specialized *replicated compute accelerator, or RCA,* that is multiplied up by having multiple copies per ASICs, multiple ASICs per server, multiple servers per rack, and multiple racks per datacenter. Work requests from outside the datacenter will be distributed across these RCAs in a scale-out fashion. All system components can be customized for the application to minimize TCO.

Figure 2 shows the architecture of a basic ASIC Cloud. Starting from the left, we start with the data center's machine room, which contains a number of 42U-style racks. In this paper, we try to minimize our requirements for the machine room because in many cases, after an array of GPU or CPU-based machines has been replaced with a new kind of nascent ASIC Cloud, it may occupy only a tiny part of a datacenter[2], and thus have little flexibility in dictating the machine room's parameters. Accordingly, we employ a modified version of the standard warehouse scale computer model from Barroso et al [21]. We assume that the machine room is capable of providing inlet air to the racks at 30° C.

ASIC Cloud Servers. Each rack contains an array of servers. Each server contains a high-efficiency power supply (PSU), an array of inlet fans, and a customized printed circuit board (PCB) that contains an array of specialized ASICs, and a control processor (typically an FPGA or microcontroller, but also potentially a CPU) that schedules computation across the ASICs via a customized on-PCB multidrop or point-to-point interconnection network. The control processor also routes data from the off-PCB interfaces to the on-PCB network to feed the ASICs. Depending on the required bandwidth, the on-PCB network could be as simple as a 4-pin SPI interface, or it could be high-bandwidth HyperTransport, RapidIO or QPI links. Candidate off-PCB interfaces include PCI-e (like in Convey HC1 and HC2), commodity 1/10/40 GigE interfaces, and high speed point-to-point 10-20 gbps serial links like Microsoft Catapult's inter-system SL3 links. All these interfaces enable communication between neighboring 1U modules in a 42U rack, and in many cases, across a rack and even between neighboring racks.

Since the PSU outputs 12V DC, our baseline ASIC server contains a number of DC/DC converters which serve to step current down to the 0.4-1.5 V ASIC core voltage. Finally, flip-chip designs have heat sinks on each chip, and wire-bonded QFNs have heat sinks on the PCB backside.

ASICs. Each customized ASIC contains an array of RCA's connected by an on-ASIC interconnection network, a router for the on-PCB (but off-ASIC) network, a control plane that interprets incoming packets from the on-PCB network and schedules computation and data onto the RCA's, thermal sensors, and one or more PLL or CLK generation circuits. In Figure 2, we show the Power Grid explicitly, because for high power density or low-voltage ASICs, it will have to be engineered explicitly for low IR drop and high current. Depending on the application, for example, our Convolutional Neural Network ASIC Cloud, the ASIC may use the

[2] In the case of Bitcoin, the scale of computation has been increased so greatly that the machine rooms are filled with only Bitcoin hardware and as a result are heavily customized for Bitcoin to reduce TCO, including the use of immersion cooling [20].

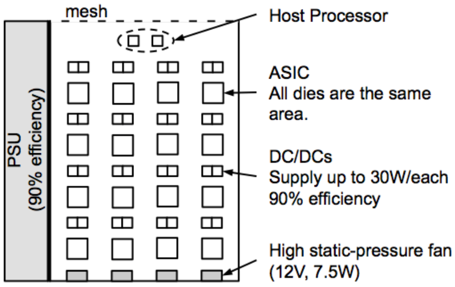

Figure 3: **The ASIC Cloud server model.**

on-ASIC network for high-bandwidth interfaces between the replicated compute accelerators, and the on-PCB network between chips at the 1U server level. If the RCA requires DRAM, then the ASIC contains a number of shared DRAM controllers connected to ASIC-local DRAMs. The on-PCB network is used by the PCB control processor to route data from the off-PCB interfaces to the ASICs to the DRAMs.

This paper examines a spectrum of ASIC Clouds with diverse needs. Bitcoin ASIC Clouds required no inter-chip or inter-RCA bandwidth, but have ultra-high power density, because they have little on-chip SRAM. Litecoin ASIC Clouds are SRAM-intensive, and have lower power density. Video Transcoding ASIC Clouds require DRAMs next to each ASIC, and high off-PCB bandwidth. Finally, our DaDianNao-style [22] Convolutional Neural Network ASIC Clouds make use of on-ASIC eDRAM and HyperTransport links between ASICs to scale to large multichip CNN accelerators.

Voltage. In addition to specialization, voltage optimization is a key factor that determines ASIC Cloud energy efficiency and performance. We will show how the TCO-optimal voltage can be selected across ASIC Clouds.

6. DESIGN OF AN ASIC SERVER

In this section, we examine the general principals in ASIC Cloud Server design to find the Pareto frontier across *$ per op/s* and *W per op/s*. Using area, performance and power density metrics of an RCA we show how to optimize the ASIC Server by tuning the number of RCAs placed on each chip; the number of chips placed on the PCB; their organization on the PCB; the way the power is delivered to the ASICs; how the server is cooled; and finally the choice of voltage. Subsequent sections apply these principles to our four prototypical ASIC Clouds.

6.1 ASIC Server Overview

Figure 3 shows the overview of our baseline ASIC Cloud server. In our study, we focus on 1U 19-inch rackmount servers. The choice of a standardized server form factor maximizes the compatibility of the design with existing machine room infrastructures, and also allows the design to minimize energy and components cost by making use of standardized high-volume commodity server components. The same analysis in our paper could be applied to 2U systems as well. Notably, almost all latest-generation Bitcoin mining ASIC Cloud servers have higher maximum power density than can be sustained in a fully populated rack; so racks are generally not fully populated. Having this high density makes it easier to allocate the number of servers to a rack according to the data center's per-rack power and cooling targets without worrying about space constraints.

The servers employ forced-air cooling system for heat removal, taking cold air at 30° C from the front using a number of 1U-high fans, and exhausting the hot air from the rear. The power supply unit (PSU) is located on the leftmost side of the server, and a thin wall separates the PSU from the PCB housing. Because of this separation and its capability of cooling itself, the PSU is ignored in terms of thermal analysis in the remaining part of this section. Figure 3 provides the basic parameters of our thermal model.

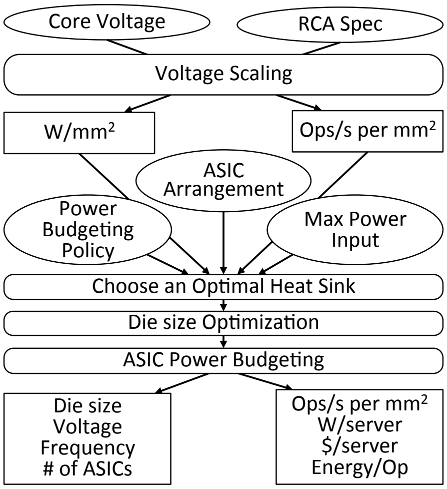

Figure 4: **ASIC Server Evaluation Flow.** The server cost, per server hash rate, and energy efficiency are evaluated using RCA properties, and a flow that optimizes server heat sinks, die size, voltage and power density.

6.2 ASIC Server Model

In order to explore the design space of ASIC Cloud Servers, we have built a comprehensive evaluation flow, shown in Figure 4, that takes in application parameters and a set of specifications and optimizes a system with those specs. We repeatedly run the evaluation flow across the design space in order to determine Pareto optimal points that trade off *$ per op/s* and *W per op/s*.

Given an implementation and architecture for the target RCA, VLSI tools are used to map it to the target process (in our case, fully placed and routed designs in UMC 28-nm using Synopsys IC compiler), and analysis tools (e.g. PrimeTime) provide information on frequency, performance, area and power usage, which comprise the *RCA Spec*. This information and a *core voltage* is then applied to a voltage scaling model that provides a spectrum of Pareto points connecting W per mm^2 and op/s per mm^2. From there, we compute the optimal ASIC die size, ASIC count, and heat sink configuration for the server, while ensuring that the transistors on each die stay within maximum junction temperature limits. Then, the tool outputs the optimized configuration and also the performance, energy, cost, and power metrics. Table 1 shows the input parameters.

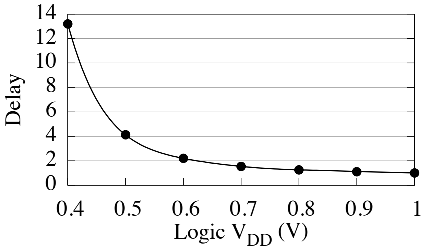

Figure 5: **The Delay-Voltage curve for 28-nm logic.**

Parameters	Description
Replicated Compute Accelerator	Power density (W/mm^2), Perf. density (Ops/s/mm^2), Critical paths @ nom. voltage, 1 V
Core Voltage	0.4 V − 1.5 V
DC/DC Efficiency & Cost PSU 208->12V Efficiency & Cost	90 %, \$0.33 per Amp 90 %, \$0.13 per Watt

Table 1: **Inputs of the server model.**

To assess the effect of voltage scaling throughout this paper, we applied a delay-voltage curve shown in Figure 5 to the logic part of the critical path, computing the effect on both power density (W/mm^2) and performance (ops/s per mm^2). This curve is inferred from I-V characteristics taken from [23, 24] and normalized at 1.0 V. For SRAMs, we assume the SRAM is on a separate power rail that supports a higher minimum supply voltage than for logic, because of the challenges in downward scaling SRAM voltages. The dynamic power is evaluated by the new frequency and voltage while leakage is affected only by the voltage.

Iterative trials find the best heat sink configuration, optimizing heat sink dimensions, material and fin topology.

6.3 Thermally-Aware ASIC Server Design

In this subsection, we optimize the plumbing for ASIC Cloud Servers that are comprised of arrays of ASICs. We start by describing the power delivery system and packaging. Then we optimize the heat sinks and fans for the servers, and optimally arrange the ASICs on the PCB.

Our results are computed by building physical models of each component in the server and simulating each configuration using the ANSYS Icepak version 16.1 Computational Fluid Dynamic (CFD) package. Based on these configurations, we built a validated Python model. After completing our Pareto study, we resimulated the Pareto-optimal configurations to confirm calibration with Icepak.

6.3.1 Power Delivery & ASIC Packaging

Power is delivered to the ASICs via a combination of the power supply, which goes from 208V to 12V, and an array of DC/DC converters, which go from 12V to chip voltage. One DC/DC converter is required for every 30A used by the system. The ASICs and onboard DC/DC converters are the major heat sources in our ASIC server model (Figure 3). Host processor heat is negligible in the ASIC Servers we consider.

We employ Flip-Chip Ball Grid Array (FC-BGA) with direct heat sink attach for ASIC packaging because of its superior thermal conductivity and power delivery performance relative to alternative wire-bond based technologies.

6.3.2 Optimizing the Heat Sink and Fan

Thermal considerations have a great impact on a packaged ASIC's power budget and the server's overall performance. The heat sink's cooling performance depends not only on its dimensions and materials but also on how those ASICs and heat sinks are arranged on the PCB because of the airflow. This section looks closely into the cooling capability of a heat sink, and airflow is considered in the next section.

Parameters	Value
Width	≤ 85 mm (Limited by ASIC density)
Height	35 mm (Limited to 1U height) A heat spreader of 3 mm thick is included
Depth	≤100 mm (Limited by the PCB depth)
Fin thickness	0.5 mm
# of fins	≥ 1 mm between two fins
Materials	Al (200 W/mK) for fins Al or Copper (400 W/mK) for heat spreader
Air Volume	Determined by static pressure and fan curve

Table 2: **Essential parameters of an ASIC heat sink.**

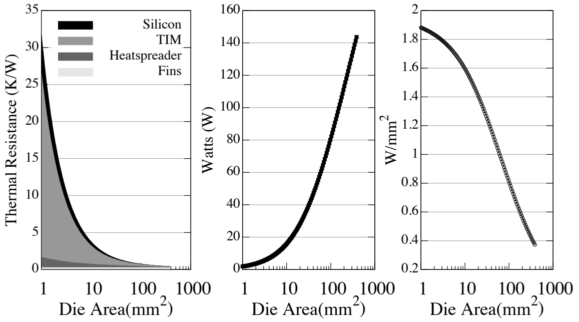

Figure 6: **Heat sink performance versus die area.** The thermal resistance for small dies is dominated by TIM. Bigger dies make the most out of the heat sink. However, acceptable power density decreases as the die area increases, resulting in smaller area for a higher power density design.

The ASIC's forced-air cooling system comprises a heat spreader glued to a silicon die using a thermal interface material (TIM). Fans blow cool air over fins that are welded to the heat spreader. The fins run parallel to air flow to maximize heat transfer. Table 2 shows the key parameters.

Heat conducts from a die to a larger surface through a TIM and a heat spreader. Increasing heat spreader size increases the surface area for better cooling, and provides more area for fins to improve the total heat resistance. Larger silicon dies can dissipate more heat since the thermal resistance induced by TIM is the dominant bottleneck because of its poor thermal conductivity and inverse proportionality to die area (Figure 6). However, increasing fin count and depth enlarges the pressure drop induced by the heat sink, resulting in less airflow from the cooling fans.

Commercial fans are characterized by a *fan curve* that describes how much air it can supply under a certain pressure drop due to static pressure. Our model takes as input a fan curve and ASIC count per row and evaluates the heat sink dimensions that achieve maximum power dissipation as a whole for a certain die area. According to the evaluation, the number of ASICs in a row affects the depth of each heat sink. As the number of ASICs increases, the heat sinks becomes less deep to reduce pressure drop and keep the airflow rate up. Generally, the densest packed fins are preferable.

6.3.3 How should we arrange ASICs on the PCB?

In this section, we analyze the impact of ASIC placement on the PCB for our ASIC server. The PCB layout matters when the server houses multiple high-power components with large heat sinks. Those components make the

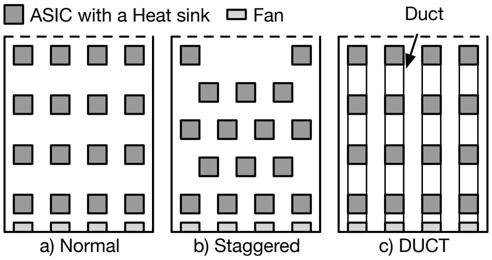

Figure 7: **PCB Layout Candidates.** (a) Grid pattern (b) Staggered pattern to reduce bypass airflow (c) Deploying ducts that surrounds ASICs in a column to further reduce bypass airflow (DUCT layout)

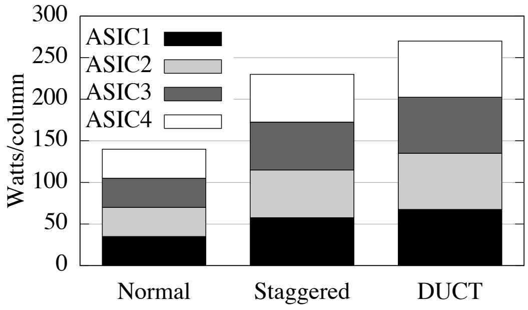

Figure 8: **Power comparison of three PCB layouts.** The staggered layout improves by 64 % in removable heat over the normal layout while DUCT gains an additional 15 %. DUCT is superior because it results in less bypass airflow between two ASICs. In each layout, the optimal heat sinks are attached on top of the dies. All layouts use the same fans.

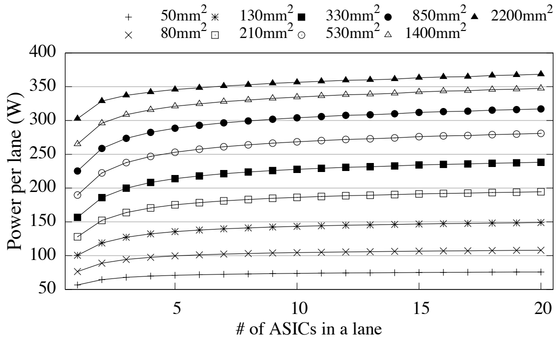

Figure 9: **Max power per lane for different number of ASICs in a lane.** Increasing the number of ASIC dies while keeping the total silicon area fixed increases the total allowable power, because heat generation is more evenly spread across the lane. Greater total area also increases the allowable power since there is more TIM.

server design more challenging not only due to heat flux but also because they behave as obstacles that disturb smooth airflow coming from the fans. The result is inefficient airflow that induces a lot of bypass airflow venting out of the server without contributing to cooling. In other words, ineffective PCB layout requires more fans and fan power to remove the same amount of heat.

We examine three different PCB layouts for our ASIC server, shown in Figure 7. For this experiment, we assume 1.5 kW high-density servers with 16 ASICs on a PCB, employing the same number and type of fans. The ASIC is assumed to be 100 mm^2 with an optimal heat sink. All the ASICs consume the same power.

In *Normal* and *DUCT* layout, ASICs are aligned in each column, while in the *Staggered* layout, ASICs of odd and even rows have been staggered to maximally spread hot air flows. The DUCT layout has four enclosures ("ducts") that constrain all the airflow supplied by a corresponding fan to ASICs in the same column. The fan is directly abutted to the front of the enclosure, and hot air is exhausted out the rear.

Employing identical cooling fans, iterative simulations gradually increase the ASICs' power until at least some part of one die reaches the maximum junction temperature of the process, 90 °C. Figure 8 shows the results.

By moving from Normal layout to Staggered layout, *65 % more heat per chip* can be removed with no additional cost. The DUCT layout is even better, allowing *15 % more heat per chip* than the Staggered layout, because almost all airflow from the fans contributes to cooling. Although Staggered layout is more efficient than the Normal layout, wide temperature variation is observed in the Staggered because of uneven airflow to each ASIC and the ASIC receiving the poorest airflow would constraint the power per chip. In DUCT layout, inexpensive enclosures gain 15 % improvement in consumable power for the same cooling. Accordingly, we employ the DUCT layout in our subsequent analysis.

6.3.4 More chips versus fewer chips

In the design of an ASIC Server, one choice we have is, how many chips should we place on the PCB, and how large, in *mm*^{2} of silicon, should each chip be? The size of each chip determines how many RCAs will be on each chip.

In a uniformly distributed power layout in a lane, each chip receives almost the same amount of airflow, while the chip in downstream receives hotter air due to the heat removed from upstream ASICs. So, typically the thermally bottlenecking ASIC is the one in the back.

Intuitively, breaking heat sources into more smaller ones spreads heat sources apart, so having many small ASIC should be easier to cool than a few larger ASICs. This intuition is verified in Figure 9. We hold the total silicon area used in each row to be fixed, and evaluate the influence of spreading the silicon by using more chips. In our analysis, we reduce the depths of the heat sinks as chip count increases in order to keep airflow up and maximize cooling. Additionally, as we increase the total amount of silicon area, our capacity to dissipate heat also increases. This is because the thermal interface glue between die and heat spreader is a major path of resistance to the outside air, so increasing die size reduces this resistance. Thus, placing a small total die area in a lane is ineffective, because limited surface area between die and spreader becomes the dominant bottleneck.

From a TCO point of view, increasing the number of chips increases the cost of packaging, but not by much. Using Flip Chip, the packaging cost is a function of die size because of yield effects. Pin cost is based on the number of pins, which is set by power delivery requirements to the silicon. Our package cost model, based on input from industry veterans, suggests the per-chip assembly cost runs about $1.

6.4 Linking Power Density & ASIC Server Cost

Using the server thermal optimization techniques described in the previous subsection, we can now make a critical connection between an RCA's properties and the number of RCA's that we should place in an ASIC, and how many of those ASICs we should place in an ASIC Server.

After designing an RCA using VLSI tools, we can compute its power density (W/*mm*^{2}) using simulation tools like Primetime. Then, using our server thermal design scripts, we can compute the ASIC Server configuration that minimizes $ per op/s; in particular how many total RCA's should be placed in an ASIC Server lane, and also how many chips the RCAs should be divided into.

Figure 10 shows the results, which are representative across typical ASIC Cloud designs. In this graph, Watts (W) is a proxy for performance (ops/s); given the same RCA, maximizing Watts maximizes server performance. If power density is high, then very little silicon can be placed in a lane within the temperature limits, and it must be divided into many smaller dies. As power density decreases, then more silicon can fit per lane, and fewer chips can be used. Sensibly, moving left, silicon area cost dominates total server cost; moving right, PCB, cooling and packaging costs dominate.

In the next section, we will continue by connecting volt-

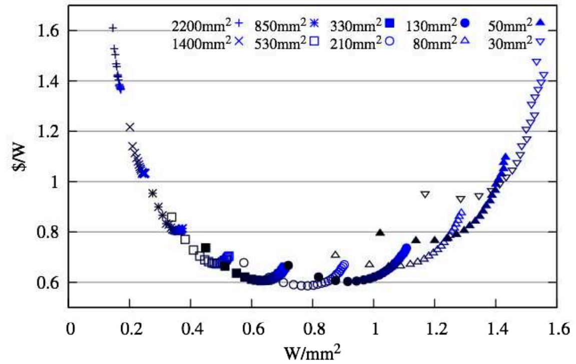

Figure 10: **Minimizing cost by optimizing the number of RCA's per ASIC Server lane (measured in mm^2), and the number of chips it should be divided into.** Point series with equal silicon area start from the right with the maximum number of chips, 20, and decrease until individual chip sizes hits the limit of 600 mm^2. Low power density RCA's are dominated by silicon costs, while high power density RCA's are dominate by cooling, packaging and server component overheads.

age scaling of the RCA's to our model. By adjusting voltage, we can change the power density of the RCA's to traverse the X axis in Figure 10, creating a spectrum of tradeoffs between the two key metrics for ASIC Servers: $ per op/s and W per op/s. The next section will also incorporate DC/DC converter costs, which are application dependent.

7. BITCOIN ASIC CLOUD DESIGN

In this section, we specialize our generic ASIC Cloud architecture to the architecture of a Bitcoin ASIC Cloud. Architecturally, the Bitcoin RCA repeatedly executes a Bitcoin hash operation. The hash operation uses an input 512 bit block that is reused across billions of hashes; and then repeatedly does the following: It mutates the block and performs a SHA256 hash on it. The output of this hash is then fed into another SHA256 hash, and then a leading zero count is performed on the result and compared to the target leading zero count to determine if it has succeeded in finding a valid hash. The two SHA256 hashes comprise 99% of the computation. Each SHA256 consists of 64 rounds.

There are two primary styles for implementing the Bitcoin RCA. The most prevalent style is the pipelined implementation, which unrolls all 64 rounds of SHA256 and inserts pipeline registers, allowing a new hash to be performed every cycle by inserting a new mutated block each cycle [25, 18]. The less prevalent style, used by Bitfury, performs the hash in place, and has been termed a rolled core.

We created an industrial quality pipelined Bitcoin RCA in synthesizeable RTL. We searched the best design parameters such as the number of pipeline stages and cycles per stage to realize the best energy efficiency, Watts per GH/s, by iteratively running the designs through the CAD tools and performing power analysis with different parameter values.

The Bitcoin implementation is fully-pipelined and consists of 128 one-clock stages, one per SHA256 round.

The RTL was synthesized, placed, and routed with the 1.0V UMC 28nm standard cell library and Synopsys Design Compiler and IC Compiler. We performed parasitic extraction and evaluated dynamic power using PrimeTime. The resulting delay and capacitance information are used to evaluate the clock latency and power consumption of the design.

Our final implementation occupies 0.66 mm^2 of silicon in the UMC 28-nm process. At the UMC process's nominal voltage 1.0V, it runs at 830 MHz, and attains a staggering power density of 2W per mm^2. Because SHA256 is comprised of combinational logic and flip-flops, and contains no RAMs, the energy density is extremely high. Moreover, data in cryptographic circuits is essentially random and has extremely high transition activity factors: 50% or higher for combinational logic, and 100% for flip flops.

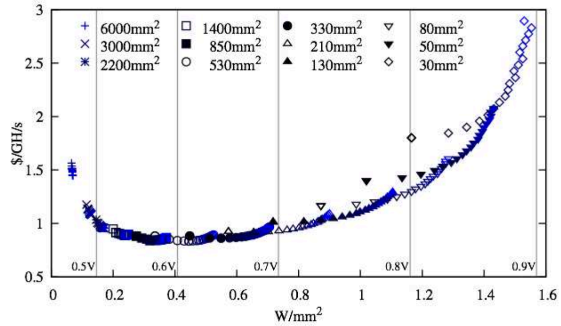

Figure 11: **Bitcoin Voltage versus Cost Performance.** Each series represents a fixed total amount of silicon per lane, while going from right to left decreases chip count from 20. Due to Bitcoin's high power density, voltage over 0.6 V is suboptimal for $/GH/s because of high cooling overheads and small silicon per server.

7.1 Building a Bitcoin ASIC Cloud

Though Bitcoin performance scales out easily with more RCA's, it is a near-worst case for dark silicon, and high power density prevents full-frequency implementations.

Using the infrastructure we developed in Section 6, we ran simulations exploring the design space of servers for a Bitcoin ASIC Cloud Server. Figure 11 shows an inital set of results, comparing $ per GH/s to power density, W/mm^2, as the amount of silicon is varied and the number of chips decreases from 20 going from right to left. The corresponding core voltages that with frequency adjustment tune the RCAs to that particular power density are given. These voltages represent the maximum voltage that that server can sustain without exceeding junction temperatures. In non-thermally limited designs, we would expect higher voltage systems to be lower cost per performance; but we can see that Bitcoin ASICs running at higher voltages are dominated designs, because the power density is excessive, and server overheads dominate. All Pareto-optimal designs are below 0.6 V.

Pareto-Optimal Servers. Although power density (W/mm^2) and voltage are important ASIC-level properties, at the ASIC Cloud Server level, what we really want to see are the tradeoffs between energy-efficiency (W per op/s) and cost-perf, ($ per op/s). To attain this, we ran simulations that exhaustively evaluated all server configurations spanning a range of settings for total silicon area per lane, total chips per lane, and all operating voltages from 0.4 up in increments of 0.01V, and pruning those combinations that violate system requirements. In Figure 12, we show a subset of this data, which shows for servers with 10 chips per lane, the relationship between silicon area per lane and voltage. Points in a series start on the top at 0.4V and increase down the page. Note that these points represent voltage-customized machines rather than the metrics of a single system under DVFS.

Generally, as more silicon is added to the server, the optimal voltages go lower and lower, and the server energy efficiency improves. Initially, $ per op/s decreases with increasing silicon as server overheads are amortized, but eventually silicon costs start to dominate and cost starts to rise. Cost-efficiency of the smaller machines declines with lower

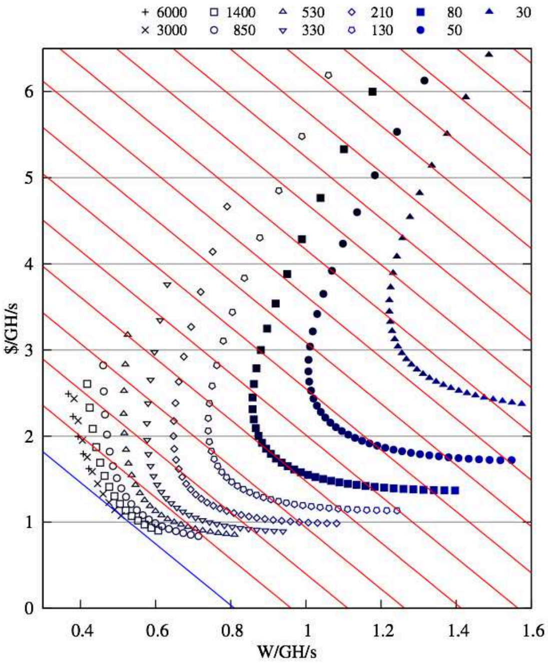

Figure 12: **Bitcoin Cost versus Energy Efficiency Pareto.** Servers with 10 chips per lane. Each point series shows total silicon per lane and voltage increases from top to bottom points. Diagonal lines represent equal TCO per GH/s, with min. TCO at lower left.

voltage because performance is decreasing rapidly and many PCB- and server-level cost overheads remain constant.

Table 3 shows the Pareto-optimal points that represent energy-optimal and cost-optimal designs across all simulation points. These are the end points of the Pareto frontier.

In the most energy-efficient design, the server runs at ultra-low near-threshold voltages, 0.40V and 70 MHz, and occupies 6,000 *mm*^{2} of silicon per lane spread across 10 chips of maximum die size, 600 *mm*^{2}. Since it employs almost a full 12" wafer per server, the cost is highest, at $2.49 per GH/s, but energy efficiency is excellent: 0.368 W per GH/s.

In the most cost-efficient design, the server is run at a much higher voltage, 0.62V and 465 MHz, and occupies less silicon per lane, 530 *mm*^{2}, spread across 5 chips of 106 *mm*^{2}. $ per GH/s is minimized, at $0.833, but energy efficiency is not as good: 0.788 W per GH/s.

Figure 13 shows the cost breakdown for these two Pareto-optimal servers. In the energy-optimal server, silicon costs dominate all other costs, since high energy-efficiency minimizes cooling and power delivery overheads. In the cost-optimal server, large savings in silicon cost gained by higher voltage operation are partially offset by much higher DC/DC converter expenses, which are the dominate cost of the system. PSU and fan overheads also increase with voltage.

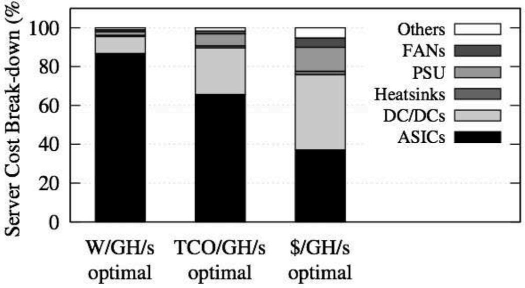

Figure 13: **Bitcoin servers cost breakdown.**

Table 3: **Bitcoin ASIC Cloud Optimization Results.**

	W/GH/s Optimal	**TCO/GH/s Optimal**	**$/GH/s Optimal**
# of ASICs per lane	10	10	5
# of Lanes	8	8	8
Logic Voltage (V)	0.40	0.49	0.62
Clock Frequency (MHz)	70	202	465
Die Size (mm^2)	600	300	106
Silicon/Lane (mm^2)	6,000	3,000	530
Total Silicon (mm^2)	48,000	24,000	4,240
GH/s/server	5,094	7,341	2,983
W/server	1,872	3,731	2,351
$/server	12,686	7,901	2,484
W/GH/s	0.368	0.508	0.788
$/GH/s	2.490	1.076	0.833
TCO/GH/s	4.235	3.218	4.057
Server Amort./GH/s	2.615	1.130	0.874
Amort. Interest/GH/s	0.161	0.069	0.054
DC CAPEX/GH/s	0.884	1.222	1.895
Electricity/GH/s	0.319	0.441	0.684
DC Interest/GH/s	0.257	0.355	0.550

TCO-Optimal Servers. A classic conundrum since the beginning of energy-efficiency research in computer architecture has been how to weight energy efficiency and performance against each other. In the absence of whole system analysis, this gave birth to such approximations as Energy-Delay Product and Energy-Delay^2. A more satisfying intermediate solution is the Pareto Frontier analysis shown earlier in this section. But, which of the many optimal points is most optimal? This dilemma is not solely academic. In Bitcoin Server sales, the primary statistics that are quoted for mining products are in fact the exact ones given in this paper: $ per GH/s and W per GH/s. Many users blindly chose the extremes of these two metrics, either over-optimizing for energy-efficiency or for cost-efficiency.

Fortunately, in the space of ASIC Clouds, we have an easy solution to this problem: TCO analysis. TCO analysis incorporates the datacenter-level constraints including the cost of power delivery inside the datacenter, land, depreciation, interest, and the cost of energy itself. With these tools, we can correctly weight these two metrics and find the over-all optimal point (TCO-optimal) for the ASIC Cloud.

In this paper, we employ a refined version of the TCO model by Barroso et al [21]. Electricity is $0.06 per KWh. In Figure 12, we annotate lines of equal TCO according to this model. The lowest TCO is found on the bottom left. As we can see, TCO is most optimized for large silicon running at relatively low, but not minimal, voltages.

The TCO-optimal design is given in Table 3. The server runs at moderate near-threshold voltages, 0.49V and 202 MHz, and occupies 3,000 *mm*^{2} of silicon per lane spread across 10 chips of moderate die size, 300 *mm*^{2}. Cost is between the two Pareto-optimal extremes, at $1.076 per GH/s, and energy efficiency is excellent but not minimal: 0.508 W per GH/s. The TCO per GH/s is $3.218, which is significantly lower than the TCO with the energy-optimal server, $4.235, and the TCO with the cost-optimal server, $4.057. The portion of TCO attributable to ASIC Server cost is 35%; to Data Center capital expense is 38%, to electricity, 13.7%, and to interest, about 13%. Finally, in Figure 13, we can see the breakdown of Server components; silicon dominates, but

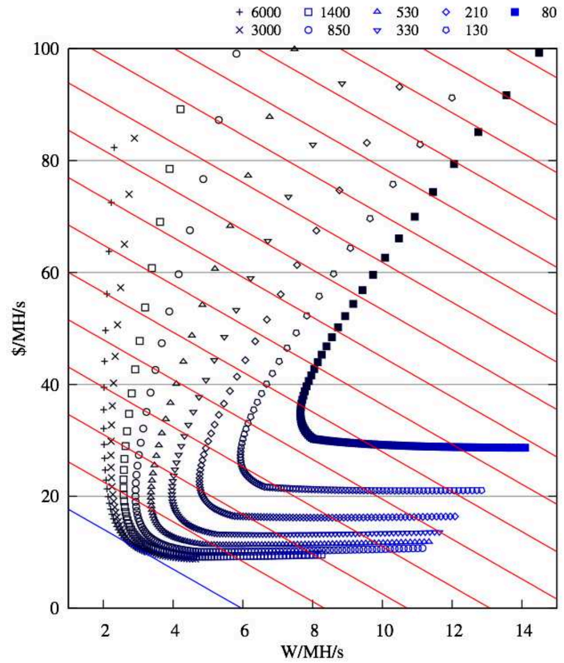

Figure 14: **Litecoin Cost versus Energy Efficiency Pareto.** Servers with 12 chips per lane. Each point series shows total silicon per lane and voltage increases from top to bottom points. Diagonal lines represent equal TCO per MH/s, with min. TCO at lower left.

DC/DC is not insignificant.

Voltage Stacking. Because DC/DC power is significant, some Bitcoin mining ASIC Clouds employ voltage stacking, where chips are serially chained so that their supplies sum to 12V, eliminating DC/DC converters. We modified our tools to model voltage stacking. The TCO-optimal voltage-stacked design runs at 0.48V and 183 MHz, employs the same chips, and achieves 0.444 W per GH/s, $ 0.887 per GH/s, and a TCO per GH/s of $2.75, a significant savings.

8. LITECOIN ASIC CLOUD DESIGN

The Litecoin cryptocurrency is similar to Bitcoin, but employs the Scrypt cryptographic hash instead of the Bitcoin hash, and is intended to be dominated by accesses to large SRAMs. We implemented a Scrypt-based Litecoin replicated compute accelerator in RTL and pushed it through a full synthesis, place and route flow using Synopsys Design Compiler and IC Compiler. We applied our ASIC Cloud design exploration toolkit to explore optimal Litecoin server designs. Our results show that because Litecoin consists of repeated sequential accesses to 128KB memories, the power density per mm^2 is much lower, which leads to larger chips at higher voltages versus Bitcoin. Because Litecoin is so much more memory intensive, performance is typically measured in megahash per second (MH/s). SRAM V_{min} is set to 0.9V.

Pareto and TCO analysis is in Figure 14 and stats for the final designs are given in Table 4.

Litecoin results are remarkably different from Bitcoin. In the most energy-efficient design, the server runs at moderate near-threshold voltages, 0.47V and 169 MHz, and occupies 6,000 mm^2 of silicon per lane spread across 10 chips of maximum die size, 600 mm^2. Since it employs almost a full 12" wafer per server, the cost is highest, at $36.67 per MH/s, but energy efficiency is excellent: 2.011 W per MH/s.

In the most cost-efficient design, the server is run at a much higher voltage, 0.91V and 849 MHz, and occupies less silicon per lane, 300 mm^2, spread across 10 chips of 300 mm^2. $ per MH/s is minimized, at $8.75, but energy efficiency is not as good: 4.475 W per MH/s.

Table 4: **Litecoin ASIC Server Optimization Results.**

	W/MH/s Optimal	TCO/MH/s Optimal	$/MH/s Optimal
# of ASICs per lane	10	12	10
# of Lanes	8	8	8
Logic Voltage (V)	0.47	0.70	0.91
Clock Frequency (MHz)	169	615	849
Die Size (mm^2)	600	500	300
Silicon/Lane (mm^2)	6,000	6,000	3,000
Total Silicon (mm^2)	48,000	48,000	24,000
MH/s/server	319	1,164	803
W/server	641	3,401	3,594
$/server	11,689	12,620	7,027
W/MH/s	2.011	2.922	4.475
$/MH/s	36.674	10.842	8.750
TCO/MH/s	48.860	23.686	27.523
Server Amort./MH/s	38.508	11.384	9.188
Amort. Interest/MH/s	2.366	0.700	0.565
DC CAPEX/MH/s	4.835	7.024	10.759
Electricity/MH/s	1.746	2.537	2.886
DC Interest/MH/s	1.405	2.041	3.126

The TCO-optimal server operates at moderate super-threshold voltage, 0.70V and 615 MHz, and also occupies 6,000 mm^2 of silicon per lane spread across 12 chips of large die size, 500 mm^2. Cost is between the two Pareto-optimal extremes, at $10.842 per MH/s, and energy efficiency is excellent but not minimal: 2.922 W per MH/s. The TCO per MH/s is $23.686, which is significantly lower than the TCO with the energy-optimal server, $48.86, and the TCO with the cost-optimal server, $27.532. The portion of TCO attributable to ASIC Server cost is 48.1%; to Data Center capital expense is 29.7%, to electricity, 10.7%, and to interest, about 11.5%.

Qualitatively, these varying results are a direct consequence of the SRAM-dominated nature of Litecoin, and thus the results are likely to be more representative of accelerators with a high percentage of memory. The current world-wide Litecoin mining capacity is 1,452,000 MH/s, so 1,248 servers would be sufficient to meet world-wide capacity.

9. VIDEO TRANSCODING ASIC CLOUDS

Video transcoding is an emerging planet-scale computation, as more users record more of their lives in the cloud, and as more governments surveil their citizens. Typical video services like YouTube receive uploaded videos from the user, and then distribute frames across the cloud in parallel to re-encode in their proprietary format. We model an ASIC Cloud, *XCode*, that transcodes to H.265 (or HEVC), and model the RCA based on parameters in [26]. For this design, the RCAs on an ASIC share a customized memory system: ASIC-local LPDDR3 DRAMs to store the pre- and post- transcoded video frames. *Thus, this RCA is most representative of accelerators that require external DRAM.*

In our PCB layout, we model the space occupied by these DRAMs, which are placed in rows of 3 on either side of the ASIC they connect to, perpendicular to airflow, and limit the number of ASICs placed in a lane given finite server depth. We also model the more expensive PCBs required by DRAM, with more layers and better signal/power integrity. We employ two 10-GigE ports as the off-PCB interface, and

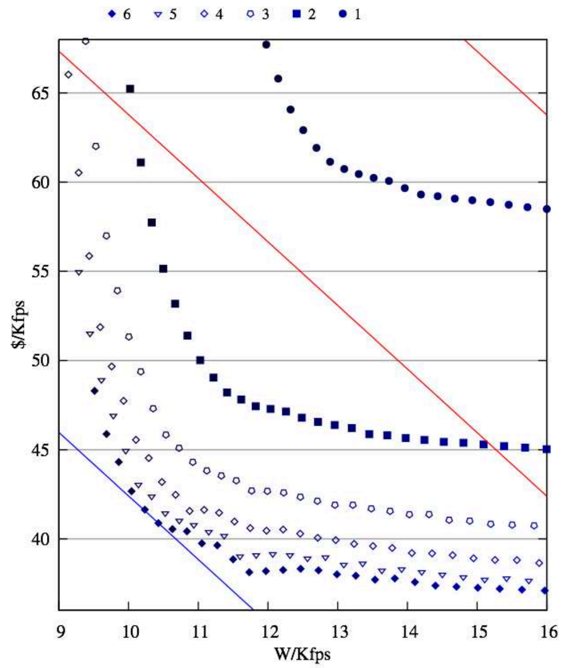

Figure 15: **Video Transcoding Pareto Curve.** Each point series corresponds to 5 ASICs per lane and a certain number of DRAMs per ASIC, and varies voltage; highest logic voltage is at lower right. Lower-left most diagonal line indicates lowest TCO per Kfps.

model the area and power of the memory controllers assuming that they do not voltage scale, and model pin constraints.

Our ASIC Cloud simulator explores the design space across number of DRAMs per ASIC, logic voltage, area per ASIC, and number of chips. DRAM cost and power overhead are not insignificant, and so the Pareto-optimal designs ensure DRAM bandwidth is saturated, which means that chip performance is set by DRAM count. As voltage and frequency is lowered, area increases to meet the performance requirement. One DRAM satisfies 22 RCA's at 0.9V.

Figure 15 shows the Pareto analysis, using point series corresponding to # of DRAMs per ASIC. Points to the upper left have lower voltage and higher area. To an extent, larger DRAM count leads to more Pareto-optimal solutions because they have greater performance per server and minimize overheads. The Pareto points are glitchy because of variations in constants and polynomial order for various server components as they vary with voltage.

Table 5 shows stats for the Pareto optimal designs. The cost-optimal server packs the maximum number of DRAMs per lane, 36, maximizing performance. However, increasing the number of DRAMs per ASIC requires higher logic voltage (1.4V!) to stay within the max die area constraint, resulting in less energy efficient designs. Hence, the energy-optimal design has fewer DRAMs per ASIC and per lane (24), while gaining back some performance by increasing ASICs per lane which is possible due to lower power density at 0.53V. The TCO-optimal design increases DRAMs per lane, 30, to improve performance, but is still close to the optimal energy efficiency at 0.8V, resulting in a die size and frequency between the other two optimal points.

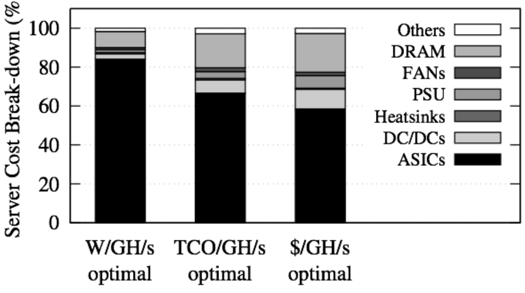

Figure 16: **Video Transcoding Server cost breakdown.**

Table 5: **Video Transcoding ASIC Cloud Optimization Results.**

	W/Kfps Optimal	**TCO/Kfps Optimal**	**$/Kfps Optimal**
# of DRAMs per ASIC	3	6	9
# of ASICs per lane	8	5	4
# of Lanes	8	8	8
Logic Voltage (V)	0.53	0.80	1.40
Clock Frequency (MHz)	163	439	562
Die Size (mm^2)	595	456	542
Silicon/Lane (mm^2)	4,760	2,280	2,168
Total Silicon (mm^2)	38,080	18,240	17,344
Kfps/server	127	159	190
W/server	1,109	1,654	3,216
$/server	10,779	6,482	6,827
W/Kfps	8.741	10.428	16.904
$/Kfps	84.975	40.881	35.880
TCO/Kfps	129.416	86.971	107.111
Server Amort./Kfps	89.224	42.925	37.674
Amort. Interest/Kfps	5.483	2.638	2.315
DC CAPEX/Kfps	21.015	25.07	40.639
Electricity/Kfps	7.590	9.055	14.678
DC Interest/Kfps	6.105	7.283	11.806

Figure 16 shows the cost breakdown. Silicon always dominates, but DRAMs and DC/DC occupy a greater percentage as performance and voltage are scaled up, respectively.

10. CONVOLUTIONAL NEURAL NET ASIC CLOUD DESIGN

We chose our last application to be Convolutional Neural Networks (CNN), a deep learning algorithm widely used in data centers. We based our RCA on DaDianNao [22] (DDN), which describes a 28-nm eDRAM-based accelerator chip for Convolutional and Deep Neural Networks. In their chip design, they have HyperTransport links on each side allowing the system to gluelessly scale to a 64-chip system in an 8-by-8 mesh. We evaluate ASIC Cloud servers that implement full 8-by-8 DDN systems, presuming that the maximum CNN size is also the most useful. Our RCA is iden-

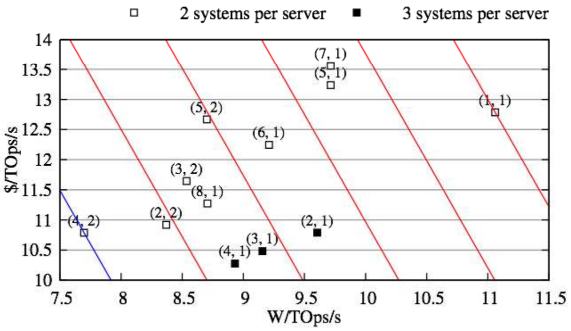

Figure 17: **Convolutional Neural Net Pareto Curve.** Twelve possible ASIC Cloud designs for 8x8 DaDianNao. The parenthesized numbers show number of RCAs per chip in each direction. Lower left diagonal line represents min. TCO per TOps/s.

Table 6: **Convolutional Neural Network ASIC Cloud Results.**

	W/TOps/s Optimal	TCO/TOps/s Optimal	$/TOps/s Optimal
Chip type	4x2	4x2	4x1
# of ASICs per lane	2	2	6
# of Lanes	8	8	8
Logic Voltage (V)	0.90	0.90	0.90
Clock Frequency (MHz)	606	606	606
Die Size (mm^2)	454	454	245
Silicon/Lane (mm^2)	908	908	1,470
Total Silicon (mm^2)	7,264	7,264	11,760
TOps/s/server	235	235	353
W/server	1,811	1,811	3,152
$/server	2,538	2,538	3,626
W/TOps/s	7.697	7.697	8.932
$/TOps/s	10.788	10.788	10.276
TCO/TOps/s	42.589	42.589	46.92
Server Amort./TOps/s	11.327	11.327	10.790
Amort. Interest/TOps/s	0.696	0.696	0.663
DC CAPEX/TOps/s	18.506	18.506	21.474
Electricity/TOps/s	6.684	6.684	7.756
DC Interest/TOps/s	5.376	5.376	6.238

tical to one DDN chip, except that we sensibly replace the HyperTransport links between RCAs with on-chip network links, if the RCAs are co-located on the same ASIC. HyperTransport interfaces are still used between chips. The more RCAs that are integrated into a chip, the fewer total HyperTransport links are necessary, saving cost, area and power. In this scenario, we assume that we do not have control over the DDN micro-architecture, and thus that voltage scaling is not possible. RCAs are arranged in 8x8 arrays that are partitioned across an equal or smaller array of ASICs. ASICs connect over HyperTransport within a lane and also nearest-neighbor through the PCB across lanes. For example, a 4x2 ASIC has 4 nodes in the lane direction and 2 nodes in the across-lane direction. 2 ASICs per lane and 4 lanes would be required to make a complete 8x8 system. We then pack as many 8x8 systems as thermally and spatially possible, and provision the machine with multiple 10-GigE off-PCB links. Up to 3 full 64-node DDN systems fit in a server.

TCO analysis in Figure 17 shows the results for the 12 different configurations, and Table 6 shows the Pareto and TCO optimal designs. Similar to the XCode ASIC Cloud, performance is only dependent on the number of 8x8 DDN systems, making the system more cost efficient by putting the most possible number of systems on a server. We allow partial chip usage, e.g. arrays that have excess RCA's that are turned off, but these points were not Pareto Optimal.

The cost-optimal system used ASICs with fewer RCAs, so 3 systems to be squeezed in. The energy- and TCO- optimal system only fit two 8x8 systems per server, but had a larger, squarish array of RCAs (4x2), removing many HyperTransport links, and thus minimizing energy consumption.

11. CLOUD DEATHMATCH

In Table 7, we step back and compare the performance of CPU Clouds versus GPU Clouds versus ASIC Clouds for the four applications that we presented. ASIC Clouds outperform CPU Cloud TCO per op/s by 6,270x; 704x; and 8,695x for Bitcoin, Litecoin, and Video Transcode respectively. ASIC Clouds outperform GPU Cloud TCO per op/s by 1057x, 155x, and 199x, for Bitcoin, Litecoin, and Convolutional Neural Nets, respectively.

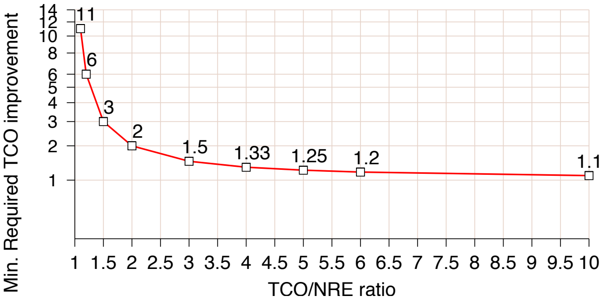

Figure 18: **Breakeven point for ASIC Clouds.**

12. WHEN DO WE GO ASIC CLOUD?

Given these extraordinary improvements in TCO, what determines when ASIC Clouds should be built? In this paper, we have shown some clear examples of planet-scale applications that could merit ASIC Clouds. The key barrier is the cost of developing the ASIC Server, which includes both the mask costs (about $ 1.5M for the 28 nm node we consider here), and the ASIC development costs, which collectively, we term the non recurring engineering expense (NRE).

We propose the *two-for-two rule*. If the cost per year (i.e. the TCO) for running the computation on an existing cloud exceeds the NRE by 2X, and you can get at least a 2X TCO per op/s improvement, then going ASIC Cloud is likely to save money. Figure 18 shows a wider range of breakeven points. Essentially, as the TCO exceeds the NRE by more and more, the required speedup to breakeven declines. As a result, almost any accelerator proposed in the literature, no matter how modest the speedup, is a candidate for ASIC Cloud, depending on the scale of the computation.

The promise of TCO reduction via ASIC Clouds suggests that both Cloud providers and silicon foundries would benefit by investing in technologies that reduce the NRE of ASIC design, including open source IP such as RISC-V, in new labor-saving development methodologies for hardware and also in open source backend CAD tools. With time, mask costs fall by themselves, and in fact older nodes such as 40 nm are likely to provide suitable TCO per op/s reduction, with half the mask cost and only a small difference in performance and energy efficiency from 28 nm.

Governments should also be excited about ASIC Clouds because they have the potential to reduce the exponentially growing environmental impact of datacenters across the world. Foundries should be excited because ASIC Cloud low-voltage operation leads to greater silicon wafer consumption than CPUs within environmental energy limits.

13. RELATED WORK

GF11 [27] was perhaps one of the first ASIC Clouds, accelerating physics simulation with custom asics. Catapult [11] pioneered the creation of FPGA Clouds.

ASIC Cloud-worthy accelerators with planet-scale applicability are numerous, including those targeting graph processing [28], database servers [29], Web Search RankBoost [30], Machine Learning [31, 32], gzip/gunzip [33] and Big Data Analytics [34]. Tandon et al [35] designed accelerators for similarity measurement in natural language processing. Many efforts [36, 37, 38, 39] have examined hardware acceleration in the context of databases, key value stores and memcached.

Other research has examine datacenter-level power and thermal optimization. Skach et al [40] proposed thermal time shifting for cooling in datacenters. Facebook optimized

Table 7: **CPU Cloud vs. GPU Cloud vs. ASIC Cloud Deathmatch.**

Application	Perf. metric	Cloud HW	Perf.	Power (W)	Cost ($)	lifetime (years)	Power/ op/s.	Cost/ op/s.	TCO/ op/s.
Bitcoin	GH/s	C-i7 3930K(2x)	0.13	310	1,272	3	2,385	9,785	20,192
Bitcoin	GH/s	AMD 7970 GPU	0.68	285	400	3	419	588	3,404
Bitcoin	GH/s	28nm ASIC	7,341	3,731	7,901	1.5	0.51	1.08	3.22
Litecoin	MH/s	C-i7 3930K(2x)	0.2	400	1,272	3	2,000	6,360	16,698
Litecoin	MH/s	AMD 7970 GPU	0.63	285	400	3	452	635	3,674
Litecoin	MH/s	28nm ASIC	1,164	3,401	12,620	1.5	2.92	10.8	23.7
Video Transcode	Kfps	Core-i7 4790K	0.0018	155	725	3	88,571	414,286	756,489
Video Transcode	Kfps	28nm ASIC	159	1,654	6,482	1.5	10.4	40.9	87.0
Conv Neural Net	TOps/s	NVIDIA Tesla K20X	0.26	225	3,300	3	865	12,692	8,499
Conv Neural Net	TOps/s	28nm ASIC	235	1,811	2,538	1.5	7.70	10.8	42.6

power and thermals in the context of general-purpose datacenters [41]. Pakbaznia et al [42] examined ILP techniques for minimizing datacenter power. Lim et al [43] examined wimpy cores, cooling, thermals and TCO.

14. SUMMARY

We propose ASIC Clouds, a new class of cloud for planet-scale computation architected around pervasive specialization from the ASIC to the ASIC Cloud Server to the ASIC datacenter. We examined the architectural tradeoffs in ASIC Clouds, and applied the results to four types of ASIC Clouds. We believe that the analysis and optimizations in this paper will speed the development of new classes of ASIC Clouds.

Acknowledgments. This work was partially supported by NSF Award 1228992, Nokia and AMD Gifts, and by the Center for Future Architectures Research (C-FAR), one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA.

15. REFERENCES

[1] G. Venkatesh, J. Sampson, N. Goulding, S. Garcia, V. Bryksin, J. Lugo-Martinez, S. Swanson, and M. Taylor, "Conservation cores: reducing the energy of mature computations," in *ASPLOS*, 2010.

[2] N. Goulding *et al.*, "GreenDroid: A Mobile Application Processor for a Future of Dark Silicon," in *HOTCHIPS*, 2010.

[3] N. Hardavellas, M. Ferdman, B. Falsafi, and A. Ailamaki, "Toward dark silicon in servers," *IEEE Micro*, 2011.

[4] H. Esmaeilzadeh, E. Blem, R. S. Amant, K. Sankaralingam, and D. Burger, "Power limitations and dark silicon are challenging the future of multicore," *TOCS*, 2012.

[5] M. B. Taylor, "Is dark silicon useful? harnessing the four horesemen of the coming dark silicon apocalypse," in *DAC*, 2012.

[6] M. Taylor, "A landscape of the new dark silicon design regime," *Micro, IEEE*, Sept-Oct. 2013.

[7] A. Pedram, S. Richardson, S. Galal, S. Kvatinsky, and M. Horowitz, "Dark memory and accelerator-rich system optimization in the dark silicon era," *IEEE Design and Test*, 2016.

[8] J. Sampson, G. Venkatesh, N. Goulding-Hotta, S. Garcia, S. Swanson, and M. B. Taylor, "Efficient Complex Operators for Irregular Codes," in *HPCA*, 2011.

[9] Venkatesh et al, "Qscores: Configurable co-processors to trade dark silicon for energy efficiency in a scalable manner," in *MICRO*, 2011.

[10] M. Shafique, S. Garg, J. Henkel, and D. Marculescu, "The EDA Challenges in the Dark Silicon Era: Temperature, Reliability, and Variability Perspectives," in *DAC*, 2014.

[11] Putnam *et al.*, "A reconfigurable fabric for accelerating large-scale datacenter services," in *ISCA*, 2014.

[12] Stephen Weston, "FPGA Accelerators at JP Morgan Chase," 2011. Stanford Computer Systems Colloquium, `https://www.youtube.com/watch?v=9NqX1ETADn0`.

[13] John Lockwood, "Low-Latency Library in FPGA Hardware for High-Frequency Trading," 2012. Hot Interconnects, `https://www.youtube.com/watch?v=nXFcM1pGOIE`.

[14] Q. Hardy, "Intel betting on (customized) commodity chips for cloud computing," in *NY Times*, Dec 2014.

[15] Y. LeCun, "Deep learning and convolutional neural networks," in *HOTCHIPS*, 2015.

[16] G. Caffyn, "Bitfury announces 'record' immersion cooling project," in *Coindesk*, Oct 2015.

[17] "List of Bitcoin Mining ASICs," Retrieved 2016. `https://en.bitcoin.it/wiki/List_of_Bitcoin_mining_ASICs`.

[18] M. Taylor, "Bitcoin and the age of bespoke silicon," in *CASES*, 2013.

[19] L. E. Nelson and R. Fichera, "Vendor landscape: Private cloud overview," in *Forrester Research Report*, Oct 2015.

[20] A. Kampl, "Bitcoin 2-phase immersion cooling and the implications for high performance computing," in *Electronics Cooling Magazine*, February 2014.

[21] L. A. Barroso, J. Clidaras, and U. Hölzle, "The datacenter as a computer: An introduction to the design of warehouse-scale machines," *Synthesis lectures on computer architecture*, vol. 8, 2013.

[22] Y. Chen, T. Luo, S. Liu, S. Zhang, L. He, J. Wang, L. Li, T. Chen, Z. Xu, N. Sun, and O. Temam, "Dadiannao: A machine-learning supercomputer," in *MICRO*, 2014.

[23] Packan *et al.*, "High performance 32nm logic technology featuring 2nd generation high-k + metal gate transistors," in *IEDM*, Dec 2009.

[24] Yeh *et al.*, "A low operating power finfet transistor module featuring scaled gate stack and strain engineering for 32/28nm soc technology," in *IEDM*, Dec 2010.

[25] J. Barkatullah and T. Hanke, "Goldstrike 1: Cointerra's first-generation cryptocurrency mining processor for bitcoin," *Micro, IEEE*, vol. 35, Mar 2015.

[26] Ju *et al.*, "18.6 a 0.5 nj/pixel 4k h. 265/hevc codec lsi for multi-format smartphone applications," in *ISSCC*, 2015.

[27] J. Beetem, M. Denneau, and D. Weingarten, "The gf11 supercomputer," in *ISCA*, 1985.

[28] J. Ahn, S. Hong, S. Yoo, O. Mutlu, and K. Choi, "A scalable processing-in-memory accelerator for parallel graph processing," in *ISCA*, 2015.

[29] L. Wu, A. Lottarini, T. Paine, M. Kim, and K. Ross, "Q100: The architecture and design of a database processing unit," in *ASPLOS*, 2014.

[30] N. Xu, X. Cai, R. Gao, L. Zhang, and F. Hsu, "Fpga acceleration of rankboost in web search engines," *TRETS*, vol. 1, Jan. 2009.

[31] Liu et al, "Pudiannao: A polyvalent machine learning accelerator," in *ASPLOS*, 2015.

[32] T. Chen, Z. Du, N. Sun, J. Wang, C. Wu, Y. Chen, and O. Temam, "Diannao: A small-footprint high-throughput accelerator for ubiquitous machine-learning," in *ASPLOS*, 2014.

[33] M. S. Abdelfattah, A. Hagiescu, and D. Singh, "Gzip on a chip: High performance lossless data compression on fpgas using opencl," in *IWOCL*, 2014.

[34] S. Jun, M. Liu, S. Lee, Hicks, Ankcorn, M. King, S. Xu, and Arvind, "Bluedbm: An appliance for big data analytics," in *ISCA 2015*, 2015.

[35] P. Tandon, J. Chang, R. G. Dreslinski, V. Qazvinian, P. Ranganathan, and T. F. Wenisch, "Hardware acceleration for similarity measurement in natural language processing," in *ISLPED*, 2013.

[36] A. Gutierrez, M. Cieslak, B. Giridhar, R. G. Dreslinski, L. Ceze, and T. Mudge, "Integrated 3d-stacked server designs for increasing physical density of key-value stores," in *ASPLOS*, 2014.

[37] O. Kocberber, B. Grot, J. Picorel, B. Falsafi, K. Lim, and P. Ranganathan, "Meet the walkers: Accelerating index traversals for in-memory databases," in *MICRO*, 2013.

[38] K. Lim, D. Meisner, A. G. Saidi, P. Ranganathan, and T. F. Wenisch, "Thin servers with smart pipes: Designing soc accelerators for memcached," in *ISCA*, 2013.

[39] Li *et al.*, "Architecting to achieve a billion requests per second throughput on a single key-value store server platform," in *ISCA*, 2015.

[40] M. Skach, M. Arora, C. Hsu, Q. Li, D. Tullsen, L. Tang, and J. Mars, "Thermal time shifting: Leveraging phase change materials to reduce cooling costs in warehouse-scale computers," in *ISCA*, 2015.

[41] E. Frachtenberg, A. Heydari, H. Li, A. Michael, J. Na, A. Nisbet, and P. Sarti, "High-efficiency server design," in *SC*, 2011.

[42] E. Pakbaznia and M. Pedram, "Minimizing data center cooling and server power costs," in *ISLPED*, 2009.

[43] K. Lim, P. Ranganathan, J. Chang, C. Patel, T. Mudge, and S. Reinhardt, "Understanding and designing new server architectures for emerging warehouse-computing environments," in *ISCA*, 2008.