Cost-Efficient Overclocking in Immersion-Cooled Datacenters

Majid Jalili*§, Ioannis Manousakis†, Íñigo Goiri*, Pulkit A. Misra*, Ashish Raniwala†, Husam Alissa‡, Bharath Ramakrishnan‡, Phillip Tuma§, Christian Belady‡, Marcus Fontoura†, and Ricardo Bianchini*

*Microsoft Research †Microsoft Azure ‡Microsoft CO+I §3M

***Abstract*—Cloud providers typically use air-based solutions for cooling servers in datacenters. However, increasing transistor counts and the end of Dennard scaling will result in chips with thermal design power that exceeds the capabilities of air cooling in the near future. Consequently, providers have started to explore liquid cooling solutions (*e.g.*, cold plates, immersion cooling) for the most power-hungry workloads. By keeping the servers cooler, these new solutions enable providers to operate server components beyond the normal frequency range (*i.e.*, overclocking them) all the time. Still, providers must tradeoff the increase in performance via overclocking with its higher power draw and any component reliability implications.**

In this paper, we argue that two-phase immersion cooling (2PIC) is the most promising technology, and build three prototype 2PIC tanks. Given the benefits of 2PIC, we characterize the impact of overclocking on performance, power, and reliability. Moreover, we propose several new scenarios for taking advantage of overclocking in cloud platforms, including oversubscribing servers and virtual machine (VM) auto-scaling. For the auto-scaling scenario, we build a system that leverages overclocking for either hiding the latency of VM creation or postponing the VM creations in the hopes of not needing them. Using realistic cloud workloads running on a tank prototype, we show that overclocking can improve performance by 20%, increase VM packing density by 20%, and improve tail latency in auto-scaling scenarios by 54%. The combination of 2PIC and overclocking can reduce platform cost by up to 13% compared to air cooling.

***Index Terms*—Datacenter cooling, server overclocking, workload performance, power management.**

I. Introduction

Motivation. Cloud providers typically use air-based solutions (*e.g.*, chillers, outside air) for cooling servers, as the wide availability of expertise and equipment make it easy to install, operate, and maintain such solutions.

Unfortunately, air cooling has many downsides. Its heat dissipation efficiency is low, which requires large heat sinks and fans that increase costs. Operating at higher component junction temperatures results in higher leakage power [65], which in turn negatively impacts energy efficiency [55]. It may also degrade performance, due to hitting thermal limits. When components approach those limits, performance is throttled by reducing clock frequency. Most importantly going forward, the trend of increasing transistor counts [66], coupled with the end of Dennard scaling, will result in *chips with thermal design power (TDP) that is beyond the capabilities of air cooling* in the near future [23], [66]. For example, manufacturers expect to produce CPUs and GPUs capable of drawing more than 500W in just a few years [23], [31], [66].

For these reasons, providers have started to explore liquid cooling solutions (*e.g.*, cold plates, liquid immersion) for their most power-hungry workloads [3], [20], [68], [74]. These technologies keep chip temperatures at a lower and narrower range than air cooling, reducing leakage power, eliminating the need for fans, and reducing datacenter Power Usage Effectiveness (PUE), *i.e.* the ratio of total power to IT (*e.g.*, server, networking) power. For example, Google cools its Tensor Processing Units (TPUs) with cold plates [25], [52]. Alibaba introduced single-phase immersion cooling (1PIC) tanks in their datacenters and showed that it reduces the total power consumption by 36% and achieves a PUE of 1.07 [74]. The BitFury Group operates a 40+ MW facility that comprises 160 tanks and achieves a PUE of 1.02 [3] with two-phase immersion cooling (2PIC).

Providers are still exploring the tradeoffs between these technologies. However, we argue that immersion, and 2PIC in particular, have significant advantages. Immersion is substantially easier to engineer and evolve over time than cold plates. In 2PIC, IT equipment is immersed into a tank filled with a dielectric liquid. The power dissipated by the equipment makes the dielectric material boil and change from liquid to gas. This phase change removes the heat from the chips, and later the vapor is converted back to liquid using a condenser.

Moreover, we argue that, because immersion cooling offers high thermal dissipation and low junction temperatures, it is possible to operate server parts at higher frequencies (*i.e.*, overclocking) for longer periods of time than ever possible before. In fact, the capability to overclock opens up many new directions to enhance system performance at scale.

However, overclocking does not come for free. Overclocking increases power consumption. Worse, it can impact component reliability (*i.e.*, stability and lifetime). Furthermore, overclocking might not improve the performance for all workloads. For example, overclocking the CPU running a memory-bound workload will not result in much improvement in performance. The problem of which component to overclock and when is harder for cloud providers because they usually manage Virtual Machines (VMs) and have little or no knowledge of the workloads running on the VMs. For these reasons, providers need to carefully manage overclocking to provide performance

§Majid Jalili is affiliated with the University of Texas at Austin, but was a Microsoft intern during this work.

benefits, while managing the associated risks and costs.

Our work. In this paper, we explore immersion cooling technologies and the ability to overclock components while managing its implications. We first compare air cooling, cold plates, and immersion cooling for datacenters, and show that 2PIC is the most promising technology. We also describe our immersion cooling tank prototypes (one of which we recently started using in a production environment [47]).

Given the benefits of 2PIC, we then explore many aspects of overclocking, starting with its power, component lifetime and stability, and total cost of ownership (TCO) implications. We then show how representative cloud workloads benefit from operating servers at higher frequencies using one of our tank prototypes. For example, we show how much workload tail latency or execution time can be shortened when increasing the frequency of the processor, the GPU, and the memory. This helps to understand when overclocking can be beneficial and when it is wasteful.

With this exploration in mind, we propose several scenarios in which cloud providers can take advantage of overclocking, each striking a different balance between performance, power, and component lifetime. We explore three of them in more detail: (1) offering high-performance VMs, (2) using overclocking to oversubscribe servers, and (3) using overclocking to improve VM auto-scaling.

For the auto-scaling scenario, we build a system that leverages overclocking to (i) hide the latency of creating new VMs, or (ii) postpone the creation of those VMs in the hopes of not needing them (we call this latter approach "scale up and then out"). To know whether and which component to overclock, we leverage hardware counters and bottleneck analysis.

Our evaluation uses realistic cloud workloads running in one of our 2PIC tank prototypes. The results show that overclocking can improve workload performance by 20% and increase VM packing density by 20% when combined with CPU oversubscription. Our TCO analysis shows that increasing density by just 10% would reduce the cost per virtual core for Microsoft Azure by 13% in comparison to today's air-cooled scenario. For the auto-scaling scenario, our results show that overclocking can improve the tail latency of a latency-sensitive workload by 54% compared to a traditional auto-scaling system.

Related work. We are not aware of prior studies of 2PIC using realistic cloud workloads. Like us, the Computational Sprinting work [21], [30], [59] considered overclocking when thermal concerns are alleviated. However, those papers relied on phase-change materials that could not sustain overclocking for long periods. In contrast, from the thermal perspective, 2PIC enables overclocking all the time, opening up many new avenues for how to use it. In fact, we are not aware of any prior proposals to use overclocking extensively, much less in the context of oversubscription or auto-scaling.

Summary. Our main contributions are:

- We explore the tradeoffs in liquid cooling and demonstrate three 2PIC tank prototypes with different immersed hardware.

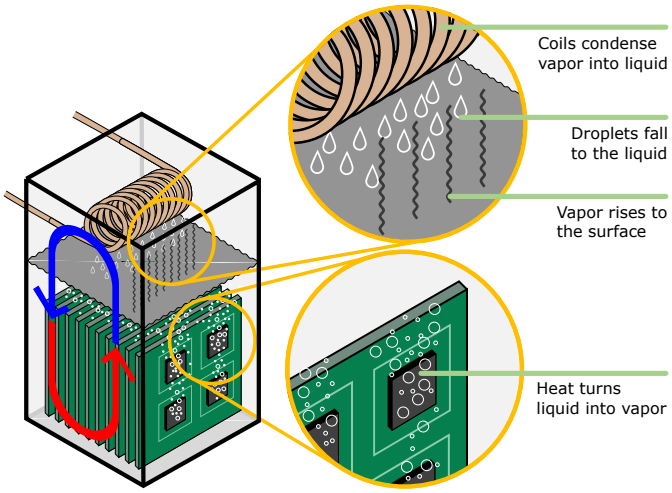

Fig. 1. Two-phase immersion cooling. IT is submerged into a dielectric liquid that changes phase (boils). Vapor rises to the top where it rejects heat and condenses back to liquid form. This process requires no additional energy. [1]

- Overclocking has several side-effects that must be traded-off carefully against its potential performance benefits. To this end, we quantify the impact of overclocking in terms of performance, power, lifetime, stability, and TCO. We show overclocking performance and power results for multiple realistic cloud workloads run in the tank prototypes.
- We propose several scenarios for providers to leverage overclocking in datacenters to reduce costs and/or enhance customer experience. We build an overclocking-enhanced VM auto-scaling system, as a detailed case study.
- We present extensive overclocking and auto-scaling results from running realistic workloads in our tank prototypes.

II. Immersion Cooling in Datacenters

Datacenter cooling today. Datacenters have been using air-based solutions to remove heat, including mainly (1) chiller-based, (2) water-side economized, and (3) direct evaporative (free) cooling. Chiller-based cooling uses a closed-loop system with a water chiller. Water-side economizing adds a cooling tower to lower water temperatures via evaporation. Free cooling uses outside air and, when necessary, spraying the air with water to reduce temperatures.

Liquid cooling. Recently, cloud providers introduced liquid cooling to tackle rising chip temperatures. These initial efforts have typically placed cold plates on the most power-hungry components. Fluid flows through the plates and piping to remove the heat produced by those components. Although efficient, each cold plate needs to be specifically designed and manufactured for each new component, which increases engineering complexity and time to market. Moreover, cold plates remove localized heat from power-hungry components but typically still require air cooling for other components.

Immersion cooling. An alternative to cold plates is immersion cooling, where entire servers are submerged in a tank and

[1] Figure rights belong to Allied Control Limited. Permission to edit and modify has been granted to the authors of this work.

TABLE I
COMPARISON OF THE MAIN DATACENTER COOLING TECHNOLOGIES.

	Average PUE	Peak PUE	Server fan overhead	Max server cooling
Chillers [12]	1.70	2.00	5%	700 W
Water-side [41]	1.19	1.25	6%	700 W
Direct evaporative [41]	1.12	1.20	6%	700 W
CPU cold plates [15]	1.08	1.13	3%	2 kW
1PIC [5]	1.05	1.07	0%	2 kW
2PIC [2]	1.02	1.03	0%	>4kW

TABLE II
MAIN PROPERTIES FOR TWO COMMONLY USED DIELECTRIC FLUIDS.

Liquid property	3M FC-3284	3M HFE-7000
Boiling point	50°C	34°C
Dielectric constant	1.86	7.4
Latent heat of vaporization	105 J/g	142 J/g
Useful life	>30 years	>30 years

the heat is dissipated by direct contact with a dielectric liquid. There are no recurring engineering overheads. The heat removal can happen in a single- or two-phase manner. In 1PIC, the tank liquid absorbs the heat and circulates using pumps, whereas in 2PIC a phase-change process from liquid to vapor (via boiling) carries the heat away. As Figure 1 shows, the vapor naturally rises to the top of the tank where a colder coil condenses it back to liquid. No liquid is lost and the heat transfers to the coil condenser secondary loop. Ultimately, the heat carried in the coil is finally rejected with a dry cooler (not shown). 2PIC can dissipate large amounts of heat.

Comparison. The technologies above have different efficiency and power trade-offs, which we summarize in Table I. The table lists publicly disclosed PUEs, server fan overheads from Open Compute Platform Olympus servers [53], and data about our own 2PIC prototypes. Overall, chiller-based systems supply servers with low and constant temperature and humidity, but suffer from high PUE. Evaporative cooling lowers PUE, but exposes servers to more aggressive environmental conditions and higher server fan overheads. Cold plates lower PUE further, but with high engineering complexity. 2PIC achieves even lower PUE, due to its more efficient heat transfer, without the need for repeated engineering. The overheads of air cooling only increase with higher server power, whereas they remain fairly stable with liquid cooling.

In the rest of the paper, we focus exclusively on 2PIC, but most of our findings apply to 1PIC and cold plates as well.

Liquids for immersion. Electronic fluids are specifically engineered to effectively transfer heat from electronics without compromising their lifetime and performance. Fluorinated fluids have been extensively used in supercomputers and Bitcoin mining with 1PIC or 2PIC. Examples include 3M's Fluorinert family (FC-3284) and Novec 7000 (HFE-7000). They are designed to boil at specific temperatures. They are non-conductive, non-toxic, repel moisture, and do not mix with oxygen or air contaminants. They are also extremely inert due to their chemical composition, which does not promote interactions with the materials used in electronics. Table II summarizes their main properties.

Surfaces with heat flux greater than $10W/cm^2$ require boiling enhancing coating (BEC) [1], as it improves heat transfer performance. Commonly applied areas are copper boilers attached to the CPU integral heat spreader or directly on the integral heat spreader, and there is no need for a heat sink in liquid. In this paper, we use 3M's L-20227 BEC for the CPU, which improves boiling performance by 2× compared to un-coated smooth surfaces.

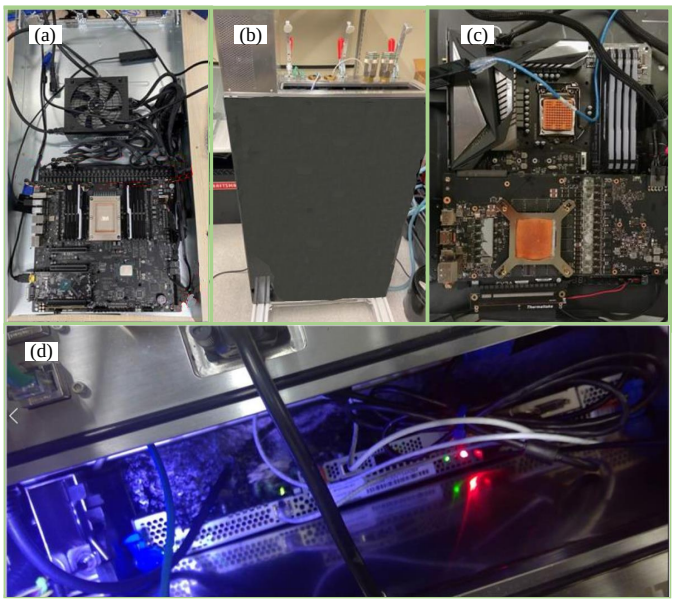

Fig. 2. Small 2PIC tank prototypes.

III. OUR 2PIC TANK PROTOTYPES

To study immersion cooling and aggressive component overclocking, we build three prototypes: (1) two small tanks able to host 2 servers each and (2) a large tank hosting 36 servers. We use the small tanks to evaluate overclocking and the large tank to perform thermal and reliability experiments.

Small tanks. Figures 2(b) and 2(d) show our two small tanks. Figure 2(d) shows a view from above with the dielectric liquid boiling. In small tank #1, we test a 28-core overclockable Xeon Skylake W-3175X, shown in Figure 2(a), cooled with the 3M Novec HFE-7000 liquid. We use data from this tank to extrapolate to datacenter servers. In small tank #2, we test an 8-core Intel i9900k with an overclockable Nvidia RTX 2080ti GPU, shown in Figure 2(c), cooled with the 3M FC-3284 liquid. In Figures 2(a) and 2(c), one can see the power-hungry components without their heat sinks and covered with BEC. In both tanks, we removed or disabled all the fans and removed any stickers that can contaminate the liquids.

Large tank. In the large tank, we test 36 Open Compute 2-socket server-class blades. Figure 3 shows three views of this prototype: (a) outside the tank; (b) a server being removed; and (c) all servers in place. Half of the blades are equipped with 24-core Intel Skylake 8168 server (TDP 205W), and half are equipped with 28-core Skylake 8180 server (TDP 205W). The 36 blades are non-overclockable, and they are used to test immersion thermal performance at a larger scale with

Fig. 3. Large 2PIC tank prototype.

FC-3284. These processors are locked versions of the Xeon W-3175X we use for overclocking. Each server consumes a maximum of 700W: 410W for the processor (205W for each socket), 120W for the memory (5W for each of the 24 DDR4 DIMMs), 26W for the motherboard, 30W for the FPGA, 72W for storage (12W for each flash drive), and 42W for the fans. We recently deployed this tank in a production environment [47].

Air-cooled baseline. The same Open Compute server configured for air cooling (*e.g.*, adding and enabling the fans) serves as our baseline throughout the paper. We conducted all the air-cooled experiments in a thermal chamber that supplied the server with 110 cubic feet of air per minute at 35°C.

IV. Overclocking in Immersion Cooling

Overclocking. Today, manufacturers allow CPUs and GPUs to operate beyond their base (or nominal) frequency within a controlled range [10], [33]. For example, Intel offers Turbo Boost v2.0 [33], which opportunistically increases core speed depending on the number of active cores and type of instructions executed. Figure 4 shows the allowable operating frequency ranges for server-class processors today. Most times, the processors operate within the guaranteed range between the minimum and the base frequency. Only when the thermal and power budgets permit, they can opportunistically operate at turbo frequency to improve their performance.

Our analysis of Azure's production telemetry reveals opportunities to operate processors at even higher frequencies (overclocking domain) still with air cooling, depending on the number of active cores and their utilizations. In this domain, we can opportunistically operate components beyond their pre-defined voltage, thermal and power design limits to further improve performance for short periods. However, such opportunities will diminish in future component generations with higher TDP values, as air cooling will reach its limits. In contrast, 2PIC has very high cooling capability and thereby provides guaranteed overclocking, irrespective of utilization and without a significant impact on operating temperatures.

Importantly, overclocking does not come for free and may significantly impact (1) power consumption, (2) component

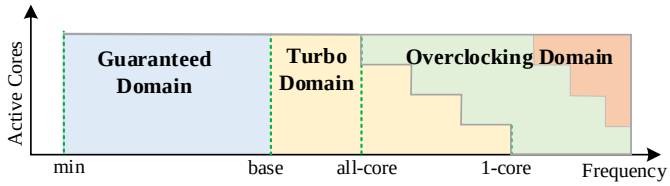

Fig. 4. Operating domains: guaranteed (blue), turbo (yellow) and overclocking (green). The non-operating domain boundary is marked with red.

TABLE III
Maximum attained frequency and power consumption for the two processors with air and FC-3284.

Platform	Skylake 8168 (24-core)		Skylake 8180 (28-core)	
Cooling	Air	2PIC	Air	2PIC
Observed T_{jmax}	92°C	75°C	90°C	68°C
Measured power	204.4W	204.5W	204.5W	204.4 W
Max turbo	3.1GHz	3.2 GHz	2.6GHz	2.7 GHz
BEC location	N/A	Copper plate	N/A	CPU IHS
Thermal resistance	0.22°C/W	0.12°C/W	0.21°C/W	0.08°C/W

lifetime, (3) computational stability, (4) environmentals, and (5) TCO. In this section, we evaluate the impact of enabling overclocking in datacenters through a series of experiments on servers in our tank prototypes. We then compare our observations to the air-cooled Open Compute server, which acts as the baseline. Although the processors in our large tank are locked and restricted to their TDP, we will, for the rest of this section, assume that they can be overclocked to 305W (*i.e.*, 100W higher than their original TDP); according to our lifetime model, which we describe below, overclocking by 100W in 2PIC provides the same processor lifetime as the air-cooled baseline with no overclocking. This overclocking translates to an additional 200W of power for a dual-socket server. We then leverage the overclockable Xeon W-3175X in small tank #1 to extrapolate on the expected implications of overclocking at scale.

Power consumption. Power is an important datacenter consideration as it increases infrastructure cost. In both air and immersion cooling, power draw is expected to increase substantially with higher operating frequencies. However, the power efficiency improvements that are enabled by 2PIC can partially offset this increase.

First, the static power of the processor reduces because it operates at lower temperatures. To quantify this reduction, we measured the temperature, power and performance for the two processors in our large tank and compared them to their air-cooled counterparts. Table III shows an improvement of one frequency bin (3%, 100MHz) when we reduced the temperature by 17-22°C. Keeping the same performance, we can save 11W of static power per socket. Second, immersion eliminates the need for fans and their power. In our Open Compute server, this represents 42W (6%) (Table I). Third, the peak PUE is reduced from 1.20 in evaporative-cooled datacenters to 1.03 in 2PIC. This is a reduction of 14% in total datacenter power. For our 700W servers, this represents a saving of 118W (= 700 × 1.20 × 0.14) per server. Putting all these saving together, we can reduce around 182W (= 2 × 11W for static, 42W for the fans, and 118W in PUE) per server. These gains can alleviate a substantial portion of the

TABLE IV
CHANGE IN RELIABILITY AS A FUNCTION OF OPERATIONAL PARAMETERS.

Failure Mode	Dependency T	Dependency ΔT	Dependency V	Description
Gate Oxide breakdown [34]	✔	×	✔	A low impedance source to drain path
Electro-migration [57]	✔	×	×	Material diffuses compromising gate structure
Thermal cycling [67]	×	✔	×	Micro-cracks due to expansion-contraction

TABLE V
PROJECTED LIFETIME COMPARISON FOR RUNNING A XEON PROCESSOR IN AIR AND 2PIC AT NOMINAL AND OVERCLOCKING CONDITIONS.

Cooling	OC	Voltage	Tj Max	DTj	Lifetime
Air cooling	×	0.90V	85°C	20°-85°C	5 years
Air cooling	✔	0.98V	101°C	20°-101°C	$<$ 1 year
FC-3284	×	0.90V	66°C	50°-65°C	$>$ 10 years
FC-3284	✔	0.98V	74°C	50°C-74°C	4 years
HFE-7000	×	0.90V	51°C	35°C-51°C	$>$10 years
HFE-7000	✔	0.98V	60°C	35°-60°C	5 years

increased power of overclocking.

Despite the power efficiency gains from 2PIC, we cannot overclock indiscriminately because doing so might result in hitting limits in the power delivery infrastructure of the datacenter. This problem is exacerbated by the power oversubscription initiatives undertaken by cloud providers for improving the power utilization of their datacenters [22], [38], [62], [70]. Overclocking in oversubscribed datacenters increases the chance of hitting limits and triggering power capping mechanisms. These mechanisms (*e.g.*, Intel RAPL [18]) rely on CPU frequency reduction and memory bandwidth throttling and therefore might offset any performance gains from overclocking. Hence, providers must perform overclocking carefully. For example, providers can overclock during periods of power underutilization in datacenters due to workload variability and diurnal patterns exhibited by long-running workloads. In addition, they can use workload-priority-based capping [38], [62], [70] to minimize the impact on critical/overclocked workloads when power limits are breached due to overclocking.

Takeaway 1: Overclocking increases power consumption substantially. However, immersion cooling provides power savings that partially offset the higher power requirements.

Lifetime. Increasing the operating frequency and consequently voltage can reduce the lifetime of electronics. However, immersion can compensate for the lifetime reduction with lower junction temperatures. To quantify the lifetime at different operating conditions, we obtained a 5nm composite processor model from a large fabrication company. Table IV summarizes the three lifetime degradation processes that are included in the model and govern lifetime: (1) gate oxide breakdown, (2) electro-migration, and (3) thermal cycling. These processes are time-dependent and accelerate the lifetime reduction. We also show their dependent parameters: junction temperature, temperature variance, and voltage. Overclocking and immersion affect these parameters substantially.

The model shows an exponential relationship between temperature, voltage, and lifetime, which is consistent with prior research [19], [42], [69], [72]. The company has validated the model through accelerated testing, which accounts for the impact of the three lifetime degradation processes as a function of workload, voltage, current, temperature, and thermal stress. This testing covers the useful lifetime ($\sim$5 years) of servers, before they are decommissioned. We use the model to calculate the temperature, power, and voltage at which electronics maintain the same predicted lifetime.

Table V shows the processor lifetime estimation for running a server in air and 2PIC, at nominal and overclocked frequencies. Our air-cooled baseline runs at nominal frequency, junction temperature of 85°C, and an expected lifetime of 5 years. The operating voltage and performance are based on the experimental voltage curve obtained from the overclockable Xeon W-3175X, which showed that to get from 205W to 305W, we would need to increase the voltage from 0.90V to 0.98V. With this power and voltage increase, we could get 23% higher frequency (compared to all-core turbo). By evaluating the model at these operating conditions, we estimate the expected processor lifetime for each configuration.

If we were to run the overclocked 305W processor in air, the junction temperature would increase to 101°C and the projected lifetime would be less than a year. In FC-3284, running at the nominal power (205W) results in a junction temperature of 66°C and more than 10 years expected lifetime. Running overclocked will increase temperatures to 74°C and lifetime would drop to approximately 4 years. By using HFE-7000, junction temperatures can be lowered to 51°C in the nominal case and 63°C when overclocking. Interestingly, when overclocking with this liquid, the lifetime matches the air-cooled baseline (5 years).

Finally, the model for predicting lifetime assumes worst-case utilization. Therefore, moderately-utilized servers will accumulate lifetime credit. Such servers can be overclocked beyond the 23% frequency boost for added performance, but the extent and duration of this additional overclocking has to be balanced against the impact on lifetime. To this end, we are working with component manufacturers to provide wear-out counters with their parts that can be used to trade-off between overclocking and lifetime.

Takeaway 2: Immersion cooling can compensate for the lifetime degradation due to overclocking, and thus paves the way for new server operating conditions and trade-offs.

Computational stability. Excessive overclocking may induce bitflips due to aggressive circuit timing and sudden voltage drops. Bitflips can cause applications to crash or produce erroneous results (silent errors). Fortunately, processors already implement error correction to protect against high-energy particle strikes. All processor results and cache accesses are verified, and bit errors are corrected whenever possible [32].

Through a period of 6 months of very aggressive overclocking, we logged the number of correctable errors for the two overclocking platforms and we saw no errors in small tank #1 and 56 CPU cache errors in small tank #2. Whenever possible, we also verified the computation results to detect any

silent errors and we did not observe any. The server would ungracefully crash though whenever we excessively pushed the voltage and frequency. It was unclear if the crashes were caused by voltage control or timing issues.

In contrast, our experience indicates that overclocking frequencies 23% higher than all-core turbo was stable and did not run into the risk of correctable or uncorrectable errors. In general, overclocking has to be balanced against computational stability, and can be accomplished, for example, by monitoring the rate of change in correctable errors. To this end, we are working with component manufacturers to define maximum overclocking frequencies for their parts to avoid computational instability, and monitor the relevant counters for safety.

Takeaway 3: Computational stability is unaffected for moderate increases in frequency and voltage, but excessive overclocking can potentially affect stability and needs to be carefully managed.

Environmental impact. If overclocking does produce an overall increase in energy consumption, it could be a source of CO_2. However, datacenters are expected to be powered primarily from renewable sources [6], [26], [45].

In terms of water usage, we have simulated the amount of water and project that the Water Usage Effectiveness (WUE) will be at par with evaporative-cooled datacenters.

Our two liquids have a high global warming impact potential (other liquids have lower potential, but we have not yet experimented with them). To minimize vapor loss from the fluids, we seal the tanks. However, large load variations and server servicing can cause vapor to escape. To tackle this issue, we implement mechanical and chemical systems that trap vapor both at the tank level and at the facility level.

Takeaway 4: Although overclocking might consume more energy, renewables and careful vapor management can enable environmentally-friendly overclocking at scale.

TCO. We worked with several teams across Azure on a complete TCO analysis of a 2PIC datacenter. The analysis compares an air-cooled datacenter with a non-overclockable and an overclockable 2PIC datacenter. It includes all costs: IT, datacenter construction, energy, operations, and taxes and fees. These factors are the same as those from prior work [12], [17], [37]. We compare the TCO for the different options in terms of cost per physical core as a metric of sellable capacity for cloud providers, including Azure.

The air-cooled baseline is a direct-evaporative hyperscale datacenter with Azure's latest server generation. The 2PIC datacenter includes additional costs for servers (*e.g.*, related to the ability to overclock them), tanks and fluid for immersion, and the costs to design a 2PIC datacenter (including mechanical and electrical infrastructures). We also include the redundancy costs (*e.g.*, power and networking) for iso-availability with the air-cooled baseline, and account for the average energy costs from overclocking.

Table VI shows the TCO analysis. We report relative values (blank cells indicate no change) to the air-cooled baseline for confidentiality; the contribution of the different factors towards the baseline's TCO is similar to that from prior work [12], [17], [24], [37]. Although using 2PIC adds costs for tanks and liquid, they are easily offset by the savings. In terms of cost per physical core, non-overclockable 2PIC datacenters are 7% cheaper than air-cooled ones. These savings primarily come from 2PIC lowering the datacenter PUE by 14% (Table I), which enables using the reclaimed power towards adding more servers and thereby amortizing costs (*e.g.*, construction, operations, energy) across more cores. In addition, there are server cost savings from eliminating fans and other server materials (*e.g.*, sheet metal). Network cost increases with 2PIC because of the additional servers.

TABLE VI
TCO ANALYSIS FOR 2PIC. THE NUMBERS ARE RELATIVE TO AN AIR-COOLED BASELINE.

	Non-overclockable 2PIC	Overclockable 2PIC
Servers	-1%	
Network	+1%	+1%
DC construction	-2%	-2%
Energy	-2%	
Operations	-2%	-2%
Design, taxes, fees	-2%	-2%
Immersion	+1%	+1%
Cost per physical core	-7%	-4%

Even overclockable 2PIC datacenters reduce TCO by 4%, when compared to the air-cooled baseline. In comparison to non-overclockable 2PIC, the capability to overclock increases the cost per physical core by 3% for two reasons. First, the power delivery infrastructure needs to be upgraded to support the higher server power requirements from overclocking. These upgrades negate the server cost savings obtained by switching from air-cooled to non-overclockable 2PIC. Second, overclocking increases energy cost. For our TCO analysis, we conservatively assume that overclocking always adds the maximum 200W of power to each server (100W per socket), and this translates to a $\sim$30% increase in server power and energy consumption over non-overclockable 2PIC. This increased energy consumption brings the energy cost of overclockable 2PIC back to that of the air-cooled baseline.

Finally, overclockable 2PIC provides cost savings when overclocking is used for oversubscription of servers. Section V describes the oversubscription use-case and the TCO impact of this use-case is presented in Section VI-C.

Takeaway 5: Immersion cooling enables using overclocking and can provide up to 7% reduction in cost per physical core in comparison to air-cooled datacenters.

Performance. Finally, the performance impact of overclocking depends on the workload-bounding resource. For example, overclocking the CPU running a memory-bound workload will not result in much improvement in performance. Similarly, overclocking the memory when the bottleneck is the CPU will be fruitless. The problem of which component to overclock and when is even harder for cloud providers because they have little or no knowledge of the workloads running inside the VMs. As we describe in Section V, counter-based models can

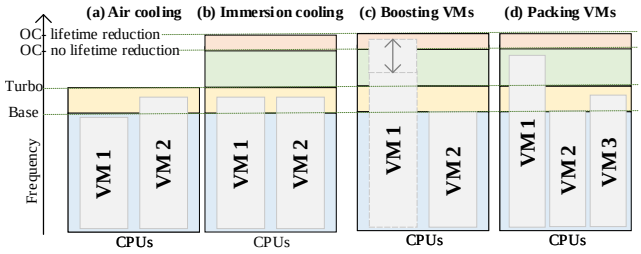

Fig. 5. Computation capability of processors running VMs in (a) air cooling and (b) immersion cooling environment. Cloud providers can use the extended capabilities in immersion to (c) offer high-performance VMs and (d) reduce costs through dense VM packing.

be used by providers to predict the impact of overclocking on a VM's workload. We also evaluate the impact of overclocking on different workloads in detail in Section VI-B.

Final takeaway: Despite the thermal benefits of immersion, providers must carefully use overclocking to provide performance benefits when they are achievable, while managing the associated risks and costs.

V. Overclocking use-cases in datacenters

This section describes use-cases for overclocking to reduce the cloud provider's costs and/or enhance customer experience. Section VI evaluates several of these use-cases.

2PIC allows operating servers at higher frequencies, compared to air-based cooling. Figure 5(a) and (b) illustrate this difference for CPUs. Figure 5(a) shows that Turbo is the maximum operating frequency of air-cooled CPUs; anything beyond it is restricted due to thermal constraints. In contrast, immersion cooling provides two higher frequency bands for operating CPUs as in Figure 5(b). There is no lifetime reduction when operating the CPUs in the green band. For our Skylake processor in HFE-7000, operating in this band means up to 23% higher frequency, with the same expected lifetime as the air-cooled counterpart. CPUs can also operate in the red band ($> 25\%$ frequency increase) with some lifetime impact.

By carefully managing the associated risks (Section IV), we can use the additional frequency bands to: (1) offer high-performance VMs to customers, (2) improve packing density of VMs on servers, (3) reduce capacity buffers in datacenters, (4) mitigate capacity crises in datacenters, and (5) augment VM auto-scaling solutions for enhanced customer experience. Below, we describe these use-cases taking CPU overclocking as an example. Overclocking other components also applies.

High-performance VMs. Cloud providers today offer VMs with Turbo Boost support [9], [49]. However, with the ability to overclock, a provider could offer new high-performance VM classes that run at even higher frequencies. For example, Figure 5(c) shows a high-performance VM1 running in the green band, while the regular VM2 runs at the base speed. Furthermore, VM1 may opportunistically run at frequencies in the red band. However, the decision of when to operate in the red band and for how long requires the provider to trade-off between performance and the risks described in Section IV.

Dense VM packing. Providers use multi-dimensional bin packing to place VMs on servers [28]. To cut costs, they

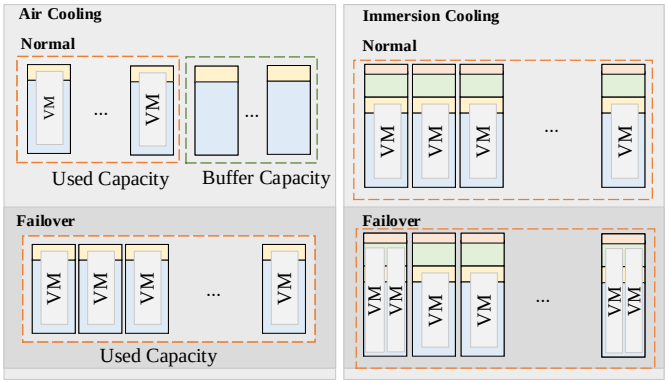

Fig. 6. Buffers with and without overclocking.

can increase VM packing density (VMs/server ratio) and thereby reduce the number of servers required. Even a single percentage point in tighter VM packing equates to hundreds of millions of dollars in savings for large cloud providers like Azure [28]. However, the potential for performance impact on VMs limits the provider's ability to increase VM packing density. The provider must place VMs so that they are unlikely to need the same resources at the same time.

When this rare scenario occurs, providers can use overclocking to mitigate it. Figure 5(d) shows an example where, through CPU oversubscription using overclocking, we can assign three VMs to the server, in contrast to Figure 5(a) where we could only assign two. Any performance loss that would be caused by oversubscription would then be compensated by overclocking. Importantly, this combination of techniques can only be used selectively, because (1) providers are only willing to oversubscribe third-party VMs when their customers are aware of it; (2) VMs often live long lifespans [16], so overclocking could be needed for long periods. To side-step the latter issue, overclocking could be used simply as a stop-gap solution to performance loss until live VM migration (which is a resource-hungry and lengthy operation) can eliminate the problem completely.

Buffer reduction. Providers typically reserve capacity (buffers) for events such as a service failover due to infrastructure failures. Upon such an event, any VMs affected by the failure get re-created on the reserve capacity. However, these events are rare and usually do not last long, so the buffers are unused or underutilized most of the time. The left side of Figure 6 illustrates the buffer in the air-cooled capacity being used only as a result of failures. For clarity, we illustrate the buffer as comprising full servers but it can also be spread across (partially filled) servers.

The provider can use overclocking to replace these static buffers with virtual ones. As the immersion cooling part of Figure 6 shows, during normal operation, it can use all the capacity to run VMs. Upon failure, it can re-create the affected VMs and overclock the servers that host them.

Capacity crisis mitigation. Providers build new datacenters based on demand forecasts. This capacity planning is often challenging because of supply-demand mismatches, such as construction delays, equipment shortages, incorrect forecasts.

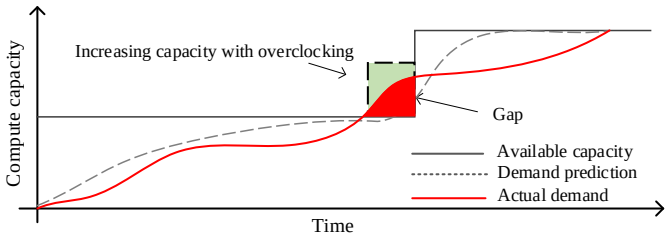

Fig. 7. Using overclocking to mitigate compute capacity gaps.

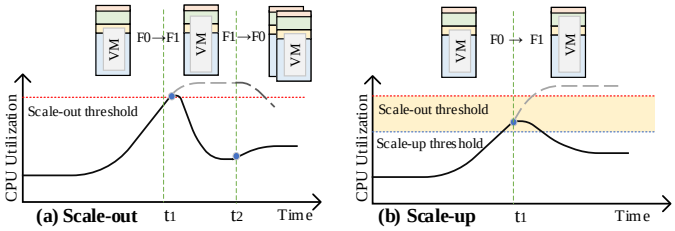

Fig. 8. Using overclocking to improve autoscaling.

A shortage in supply leads to customers being denied service until the additional servers are deployed. Figure 7 shows an example where the provider incorrectly predicted the demand and ended up with a capacity gap (red area). To bridge the gap, it can use overclocking to get more VMs deployed on to its existing infrastructure, assuming enough other resources (especially memory and storage space) are available.

Overclocking-enhanced auto-scaling. Providers offer solutions for auto-scaling the number of VMs in a deployment according to user-provided rules [7], [48]. For example, the user may specify that a new VM should be added (scale-out) if the average CPU utilization over the last 5 minutes exceeds 50%. Similarly, she may specify that, if average utilization is lower than 20% for the past 10 minutes, then one VM should be removed (scale-in). Although CPU utilization is the most common metric for auto-scaling, some users specify others like memory utilization, thread count, or queue length. In general, these metrics act as proxies for the application-level metrics that the user cares about (*e.g.*, tail latency).

However, scaling out is expensive today, as it may take tens of seconds to even minutes to deploy new VMs [4]. This scale-out overhead could impose a performance penalty on the application. Consequently, users typically set lower thresholds to start scaling out before the VMs are actually needed. Although providers have started predicting surges in load and scaling out proactively [8], the time required for scaling out can still impact application performance.

Providers can use VM deployment overclocking to mitigate the performance impact of scaling-out. Overclocking essentially *scales-up* the existing VM(s), while the new VM(s) are being deployed. Once the scale-out completes, all the VMs can be scaled-down to their regular frequency. Temporarily scaling-up and down is effective because changing frequencies only takes tens of μs [43], which is much faster than scaling out. Figure 8(a) illustrates the process of temporarily scaling-up a VM (from time t1 to t2) to hide the scale-out overhead; the VM is scaled-down after the scale-out completes.

As a variant of this use-case, overclocking can also postpone or even obviate the need for scaling out. Since changing clock frequencies typically impacts CPU utilization [51], the provider can scale-up VMs to keep the utilization below the user-specified scale-out threshold. The scale-up has to occur at a lower threshold than the scale-out. Figure 8(b) illustrates the process of scaling-up (at time t1) and preventing the scale-out altogether. For this approach to work in practice, the provider should model [51] the impact of frequency change on utilization. This step is crucial because overclocking VMs indiscriminately will increase the power consumption, could impact component lifetime, and might not even provide any reduction in utilization (Section IV). To track changing loads, the provider can periodically monitor the impact of overclocking on utilization and adjust the frequency (increase or decrease) accordingly. Finally, if overclocking is not enough, the scale-out rule will trigger when the CPU utilization crosses the higher threshold.

TABLE VII
EXPERIMENTAL FREQUENCY CONFIGURATIONS.

Config	(GHz)	Voltage offset (mV)	Turbo (GHz)	LLC (GHz)	Memory (GHz)
B1	3.1	0	✗	2.4	2.4
B2	3.4	0	✔	2.4	2.4
B3	3.4	0	✔	2.8	2.4
B4	3.4	0	✔	2.8	3.0
OC1	4.1	50	N/A	2.4	2.4
OC2	4.1	50	N/A	2.8	2.4
OC3	4.1	50	N/A	2.8	3.0

VI. EVALUATION

This section evaluates the use-cases from Section V. We start by describing our experimental setup and the representative cloud applications that we use for evaluation. Then, we study the benefits of component overclocking for applications running on high-performance VMs. Next, we evaluate using overclocking for enhancing the packing of VMs on servers and estimate the packing ratio at which the VM performance is not degraded. These results directly apply to the buffer reduction and capacity crisis mitigation use-cases. Finally, we present the design of our overclocking-enhanced auto-scaler and evaluate its benefits. Throughout this section, we also project on potential efficiencies at scale.

A. Experimental environment

Servers. We use the servers in small tanks #1 and #2 (Section II). Both servers have 128 GB DDR4 memory. Small tank #1 has the Xeon W-3175X processor with a 255W TDP. Small tank #2 has an Intel i9900k processor and a Nvidia 2080ti (250W TDP) GPU. Both servers run Windows Server 2019 Datacenter edition with Hyper-V enabled for virtualization. We perform all experiments with small tank #1, except for the GPU overclocking ones which we perform with tank #2.

Frequency configurations. We configure the two systems with different frequencies for every component. Table VII shows the seven configurations of tank #1 that overclock the core, uncore, and system memory. We include: (1) two production baseline configurations (B1 and B2) where we

TABLE VIII
GPU CONFIGURATIONS.

	Power (W)	Base (GHz)	Turbo (GHz)	Memory (GHz)	Voltage Offset (mV)
Base	250	1.35	1.950	6.8	0
OCG1	250	1.55	2.085	6.8	0
OCG2	300	1.55	2.085	8.1	100
OCG3	300	1.55	2.085	8.3	100

TABLE IX
APPLICATIONS AND THEIR METRIC OF INTEREST.

Application	#Cores	Description (In-house:Public)	Metric
SQL	4	BenchCraft standard OLTP (I)	P95 Lat
Training	4	TensorFlow model CPU training (I)	Seconds
Key-Value	8	Distributed key-value store (I)	P99 Lat
BI	4	Business intelligence (I)	Seconds
Client-Server	4	M/G/k queue application (I)	P95 Lat
Pmbench [71]	2	Paging performance (P)	Seconds
DiskSpeed [46]	2	Microsoft's Disk IO bench (P)	OPS/S
SPECJBB [64]	4	SpecJbb 2000 (P)	OPS/S
TeraSort [11]	4	Hadoop TeraSort (P)	Seconds
VGG [58]	16	CNN model GPU training (P)	Seconds
STREAM [44]	16	Memory bandwidth (P)	MB/S

do not overclock (we expect B2 to be the configuration of most datacenters today); (2) two configurations where we independently overclock the memory and uncore, including the last-level cache (B3 and B4); and (3) three configurations where we overclock combinations of all components (OC1-3).

Table VIII shows the configurations and the knobs for GPU overclocking. Here we include: (1) one baseline with turbo enabled; and (2) three configurations where we progressively overclock more GPU sub-components (OCG1-3).

Applications. Table IX describes the applications (5 in-house and 6 publicly-available) that we use, the number of cores that each application needs, and the metric of interest for the application. The top nine applications represent those that are commonly run in the cloud, whereas the bottom two help us evaluate the impact of GPU and memory overclocking.

B. High-performance VMs

This section evaluates the use-case where providers offer high-performance VMs by overclocking the processor, memory, and/or GPU.

Overclocking for cloud applications. Figure 9 shows the impact of overclocking on the metric of the interest and average and 99^{th}-percentile (P99) server power draw. We run only one instance of each application in isolation. For the six applications on the left of the figure, a lower metric of interest is better; for the two applications on the right, higher is better.

In all configurations, overclocking improves the metric of interest, enhancing performance from 10% to 25%. Core overclocking (OC1) provides the most benefit, with the exception of TeraSort and DiskSpeed. However, it also considerably increases the P99 power draw in a few cases. Cache overclocking (OC2) accelerates Pmbench and DiskSpeed, while incurring only marginal power overheads. Memory overclocking (OC3) improves performance slightly for four applications and significantly for memory-bound SQL. In all cases, memory overclocking substantially increases the power draw.

BI illustrates the importance of careful overclocking at scale. Although OC1 improves performance substantially with reasonable power penalty, overclocking other components increases the power draw without offering any performance gain. Training has a predictable access pattern, so the prefetcher can timely bring data blocks into the L1 cache. Thus, a faster cache or memory does not improve its performance significantly.

Memory overclocking for streaming applications. We further evaluate the impact of memory overclocking using STREAM [44] and the configurations from Table VII. We measure the sustainable memory bandwidth and the computation rate for four kernels: (1) copy, (2) scale, (3) add, and (4) triad. Figure 10 shows the bandwidths and power draws.

The highest performance improvement happens when the memory system is overclocked: B4 and OC3 achieve 17% and 24% improvements compared to B1, respectively. Increasing core and cache frequencies also has a positive impact on the peak memory bandwidth, as memory requests are served faster. As expected, the power draw increases with the aggressiveness of overclocking (10% average power increase).

GPU overclocking for machine learning training. We evaluate the impact of GPU overclocking using VGG [58] on PyTorch [54]. We train 6 CNN models and the input sizes fit entirely in the GPU's memory. Figure 11 shows the normalized execution time and absolute power for the 6 VGG models when overclocked using the configurations from Table VIII.

We observe that the execution time decreases by up to 15%, which is proportional to the frequency increase. We also see that increasing voltage, power, and frequency all help performance. However, for the most recent batch-optimized model VGG16B, overclocking is beneficial only to a certain extent; OCG2 (with GPU memory overclocking) offers marginal improvements over OCG1, and further GPU memory overclocking with OCG3 provides no additional improvement. Worse, the P99 power draw increases by 9.5% between OCG1 and OCG3, while offering little to no performance improvement. Overall, the P99 power draw during these runs was 231W, up from 193W for the baseline runs (+19%).

Summary. Our results indicate that applications can benefit from the overclocking of all components. However, providers must be careful to increase frequencies for only the bottleneck components, to avoid unnecessary power overheads.

C. Dense VM packing via oversubscription

In this section, we study using overclocking to mitigate the interference induced by core oversubscription in tank #1. We run experiments with latency-sensitive VMs only, and with a mix of latency-sensitive and batch workload VMs.

Latency-sensitive VMs only. We run 4 instances of SQL on 4 VMs, where each VM has 4 virtual cores (vcores for short). With no oversubscription, we would assign 16 physical cores (pcores) to run the 16 vcores across the 4 VMs. To study the impact of oversubscription, we vary the number of pcores assigned to run the VMs from 8 (50% oversubscription) to 16 (no oversubscription). To quantify the benefit of overclocking,

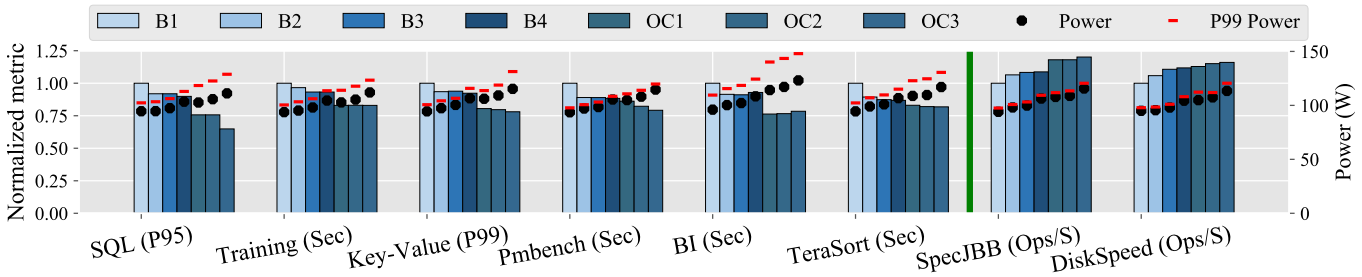

Fig. 9. Normalized metric, average and P99 server power draw for cloud workloads.

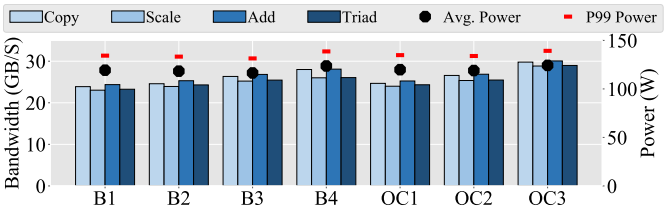

Fig. 10. Comparing bandwidth and power of STREAM.

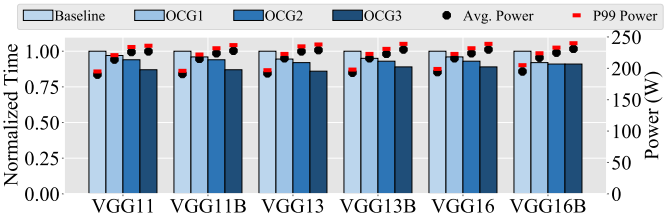

Fig. 11. Normalized execution time for the baseline and 3 overclocked GPUs.

we study server configurations B2 and OC3 from Table VII. Figure 12 shows the average P95 latencies.

As one would expect, the average P95 latency decreases as we increase the pcore count. Crucially, the latency for the best case of 16-pcore B2 is comparable (within 1%) to OC3 with 12 pcores. This means that we can save 4 pcores with overclocking, while providing the same level of performance as B2. These 4 pcores become available to run additional VMs.

In terms of power draw (with any inactive pcores in low-power idle state), the baseline with 12 and 16 pcores running SQL consumes on average 120W and 130W and at P99 126W and 140W, respectively. With 12 and 16 active pcores under OC3, the server consumes on average 160W and 173W, and at P99 169W and 180W, respectively. Given that the core and uncore frequencies both have increased by 20%, these increases (29%–33%) in power draw are expected.

Batch and latency-sensitive VMs. Table X shows three scenarios with two latency-sensitive (SQL and SPECJBB) and two batch applications (BI and TeraSort). Each scenario requires a total of 20 pcores, but we assign only 16 pcores (20% oversubscription). We run these three scenarios with the same two configurations (B2 and OC3). Figure 13 shows the improvement to the metric of interest for each application, compared to a baseline with the requisite number of pcores (20) under the B2 configuration.

In all the cases, oversubscribing the baseline by 20% (yellow bars) causes performance degradation. This shows that this

TABLE X
OVERSUBSCRIBING BATCH AND LATENCY-SENSITIVE VMS TOGETHER.

Scenarios	Workloads	vcores/pcores
Scenario 1	1×SQL, 1×BI, 1×SPECJBB, 2×TeraSort	20/16
Scenario 2	1×SQL, 1×BI, 2×SPECJBB, 1×TeraSort	20/16
Scenario 3	2×SQL, 1×BI, 1×SPECJBB, 1×TeraSort	20/16

oversubscription level under B2 has a negative performance impact. Latency-sensitive applications like SQL and SPECJBB suffer the highest degradation. In contrast, when we increase the core and uncore frequencies under OC3 (blue bars), all workloads show improvements of up to 17%. In fact, all of them improve by at least 6% compared to the baseline, except for TeraSort in Scenario 1.

TCO impact of denser VM packing. As we see above, overclocking can enable denser packing via oversubscription. To quantify the TCO impact of this, we assume using overclocking with 10% oversubscription of physical cores, which would leverage the stranded memory on Azure's servers.

For this analysis, we quantify the TCO per virtual core. We obtain the cost per virtual core with no oversubscription from the TCO analysis in Section IV, which assumes a 1:1 mapping between virtual and physical cores. Oversubscription of 10% in overclockable 2PIC would reduce the TCO per virtual core by 13% when compared to air-cooled ones. In fact, oversubscription also benefits non-overclockable 2PIC – it would reduce TCO by ∼10% since it amortizes overall costs across 10% more virtual cores.

Summary. Oversubscription using overclocking frees up cores that providers can use to run more VMs on the same hardware. Such oversubscription would reduce the cost per virtual core for Azure by 13%.

D. Overclocking-enhanced auto-scaling

We first describe a model to determine the impact of frequency change on CPU utilization. Our auto-scaler implementation uses this model to determine if (and by how much) scaling-up is beneficial. Next, we validate the model in the context of auto-scaling and then present the results for how overclocking improves auto-scaling.

Performance and utilization model. Our auto-scaler should carefully use overclocking for scaling-up because not all workloads (*e.g.*, memory-bound) will benefit from running at high CPU frequencies. We need a low-overhead mechanism for this because expensive models will prolong the impact on affected workloads. To this end, our auto-scaler uses a simple

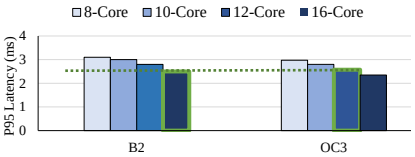

Fig. 12. Avg. P95 latency with 4 SQL VMs.

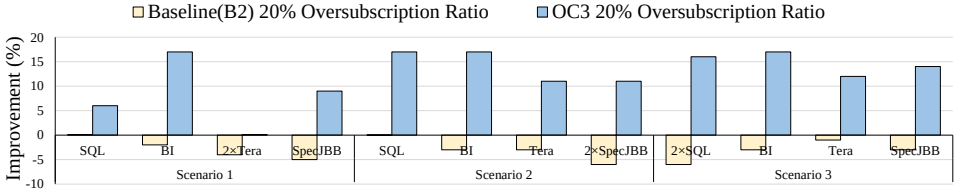

Fig. 13. Performance impact of overclocking while oversubscribing batch and latency-sensitive VMs.

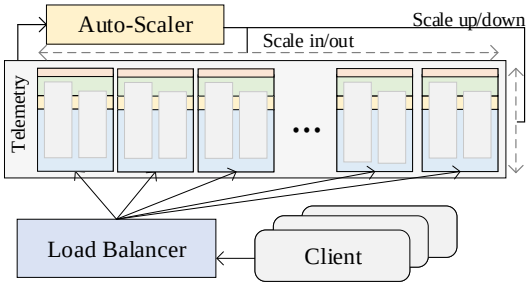

Fig. 14. Auto-scaling (ASC) architecture. ASC decides the number of VMs and the frequency of each core by initiating scale-out/in or scale-up/down.

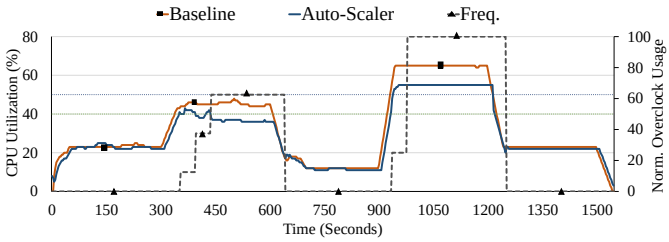

Fig. 15. Validating the model. When utilization increases, the model finds a frequency that lowers it. When utilization drops, the model lowers the frequency accordingly.

yet effective model proposed in [51]. This model uses two per-core architecture-independent counters called A_{perf} and P_{perf}. A_{perf} counts how many cycles the core is active and running. P_{perf} is similar to A_{perf}, except that it does not count cycles when the active core is stalled because of some dependency (*e.g.*, memory access). The model estimates the impact of frequency change from F_0 to F_1 on the current utilization $Util_t$ as follows:

$$Util_{t+1} = Util_t \times \left(\frac{\Delta P_{perf}}{\Delta A_{perf}} \times \frac{F_0}{F_1} + \left(1 - \frac{\Delta P_{perf}}{\Delta A_{perf}} \right) \right) \quad (1)$$

Overclocking-enhanced auto-scaler architecture and setup. Figure 14 shows the architecture of our overclocking-enabled auto-scaler. Clients send their requests to the load balancer and the server VMs respond to the requests. The auto-scaler decides when VMs must be added (scale-out) or removed (scale-in). To do so, it considers the average CPU utilization of the server VMs over the last 3 minutes (to avoid noise). It also decides when to change the frequency of server VMs (scale-up or down) using the average CPU utilization and the performance counters collected over the last 30 seconds. It makes these decisions every 3 seconds, based on the telemetry (*i.e.*, A_{perf}, P_{perf}, and $Util_t$) collected from the VMs. For scaling-up or down, the auto-scaler uses Equation 1 to find the minimum frequency, from the supported range of the processor, that keeps the average CPU utilization of VMs below or above the threshold, respectively.

We use the Client-Server application (Table IX) for our experiments. The client request arrivals are Markovian, the service times follow a General distribution, and there are k servers (*i.e.*, VMs). We run all the VMs in the Xeon server in tank #1. The threshold for scaling-out is 50% CPU utilization and 20% for scaling-in. To emulate the behavior of a real scale-out operation (*e.g.*, impact of network traffic), we make scaling-out in our system take 60 seconds. Scaling in or out is done by 1 VM at a time. We also set the scale-up and scale-down thresholds to 40% and 20%, respectively. The frequency range for scaling up or down is 3.4 GHz (B2) to 4.1 GHz (OC1), divided into 8 frequency bins; no other components are overclocked.

Model validation. To validate our model (Equation 1), we configure the auto-scaler to bring the utilization just below the scale-up threshold when this threshold is exceeded. We also configure it for scale-up/down only, *i.e.* no scale-out/in. We then launch three server VMs and adjust the client load every 5 minutes to 1000, 2000, 500, 3000, and 1000 queries per second (QPS). Figure 15 shows the CPU utilization over time with our auto-scaler and a baseline that does not change frequencies. The secondary y-axis shows the frequency as a percentage of the frequency range between B2 and OC1.

As the figure shows, once the CPU utilization exceeds 40%, the auto-scaler increases the frequency in steps until the utilization is under the threshold. As soon as the first peak ends (time 200 seconds), the auto-scaler scales down to the lowest frequency. For the second peak, the auto-scaler scales-up sharply to the maximum frequency to adjust to the client load. However, the utilization stays higher than the scale-out threshold, which would imply a scale-out invocation.

Note that every time we increase the frequency, the utilization decreases. This shows that our model predicts correctly; otherwise, we would see ineffective frequency changes or missed utilization targets. The reason it might take more than one frequency adjustment to bring the utilization below the threshold is that the utilization the auto-scaler sees is averaged over the last 30 seconds. In other words, the average might run behind the actual utilization during quick increases in load.

Overclocking-enhanced auto-scaler results. In the above experiments, we turned off scale-out/in in our auto-scaler. Now, we evaluate it with both scale-up/down and scale-out/in. We run three configurations: (1) baseline (B2 in Table VII); (2) OC-E, which scales up straight to OC1 frequency when the scale-out threshold is crossed, *i.e.* there are no scale-

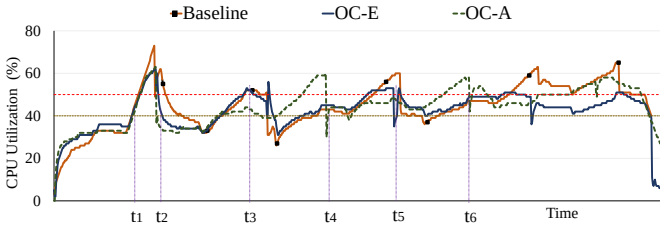

Fig. 16. Utilization of baseline, OC-E (overclock while scaling out), and OC-A (overclock before scaling out).

TABLE XI
RESULTS FOR THE FULL AUTO-SCALER EXPERIMENT.

Config	Norm P95 Lat	Norm Avg Lat	Max VMs	VM×hours
Baseline	1.00	1.00	6	2.20
OC-E	0.58	0.27	6	2.17
OC-A	0.46	0.23	5	1.95

up/down thresholds); and (3) OC-A, which tries to keep the utilization below the scale-up threshold to delay/avoid scale-out. We again start one server VM. For the client load, we start with 500 QPS and increase the load by steps of 500 every 5 minutes up to a maximum of 4000 QPS.

Figure 16 shows the CPU utilization with the three auto-scaler configurations over time. At time t_1 load increases, forcing all three systems to scale out one VM to reduce the load. However, comparing OC-E and OC-A to the baseline, we can see that (1) their utilization does not reach 70% as it does for the baseline, and (2) the utilization drops faster (time t_2) because of their higher frequency.

The advantage of OC-A over OC-E first appears at t_3, when OC-E and baseline must add one VM while OC-A manages the utilization increase by scaling up. The fact that OC-A is running with one fewer VM is the reason its utilization reaches a higher peak than the other systems at t_4. A similar pattern repeats between t_4 and t_6. During the rest of the execution, the load is so high that all systems are forced to scale out.

Table XI summarizes the results. The application's P95 latency improves by 42% and 54% for OC-E and OC-A, respectively, compared to the baseline. This is because when the client load increases, overclocking makes the existing server VMs run faster (as evident from the reduction in utilization in Figure 16) and thereby mitigates the performance impact. Additionally, the baseline and OC-E scale-out to 6 VMs, whereas scaling-up to prevent scale-out with OC-A helps reduce the number of VMs needed to 5. OC-A produces a savings of 0.25 VM×hours (11% compared to the baseline) for the user, and less capacity consumed for the cloud provider. For the provider, overclocking also results in a higher power draw by the server VMs. On average, the power draw of the server VMs increases by 7% with OC-E and 27% with OC-A over the baseline.

Summary. Overclocking-enhanced auto-scaling provides users with significant performance and cost improvements, by hiding the overhead of VM scale-out or reducing the number of VMs that need to be created.

VII. RELATED WORK

Our work is the first to explore the benefits of 2PIC in terms of overclocking (and its risks). We are also the first to propose and evaluate many overclocking scenarios for cloud providers. Nevertheless, prior works on advanced cooling, processor speed boosting, and power management are related.

Advanced cooling. Liquid cooling has been used to improve performance and power efficiency. For example, Google uses cold plates for its power-hungry TPUs [52], whereas Alibaba uses 1PIC to improve datacenter PUE [74]. Section II compares 2PIC to these technologies.

Other technologies, such as two-phase cold plates [61] and microfluidic cooling [63] have been proposed as well. The former combines 2PIC and cold plates: liquid circulates directly over hot components but boiling takes place within the cold plates. In the latter, sub-micrometer channels are carved between electrical paths and cold fluid circulates through them. We argue that 2PIC provides a better combination of effectiveness and design simplicity; it achieves similar or better cooling capabilities than its counterparts.

Increasing processor speed. Researchers have proposed computational sprinting, which uses phase-change materials, such as wax, to enhance the heat removal from processors [59], [60]. They studied a combination of running cores at higher frequencies and increasing the number of active cores for short periods (several minutes) until the material melts.

Other works have proposed similar approaches [13], [21], [30], [35], [36], [40], [50]. Autotune [40] enhances Turbo Boost [33] based on the observation that if any resource interference occurs, increasing frequency might be harmful and waste power. Paceline [27] uses an overclocked leader thread and an underclocked checker thread for execution. The leader aids execution through data prefetching and resolving branch outcomes, and the checker ensures correctness in the face of any errors introduced by overclocking. Cooperative boosting [56] studies the interaction of higher frequency, turbo, and temperature on system performance.

Esprint [14] and SprintCon [73] investigate computational sprinting under QoS constraints. Adrenaline [29] targets improving tail latency by boosting the processor speed for those tasks that slow down the system the most.

Overall, these works are generally limited to speeding up CPUs and only to turbo frequency. Given the limitations of their cooling systems, this boost can only last for short periods. Moreover, they have not studied the consequences and risks of operating in the overclocking domain. Being able to overclock for longer periods, while trading off against consequences and risks, enables many new opportunities (for performance, QoS, and capacity management) that we unearth in this paper.

Power capping and management. Power capping is used in datacenters for power safety [22], [38], [39], [62], [70]. These mechanisms typically rely on frequency reduction (opposite of overclocking) to manage power when it approaches circuit breaker limits. To reduce the impact of capping on critical workloads, prior works have proposed workload-priority-based

capping [38], [39], [62], [70]. These works are complementary and can be used to mitigate the performance impact of capping even on servers with overclocked components.

VIII. Conclusions

In this paper, we explored the use of liquid cooling and component overclocking by public cloud providers. We argued that 2PIC is the most promising technology and built three tanks to demonstrate it. We also proposed many scenarios for 2PIC-enabled overclocking, and discussed its benefits and risks. Our experimental evaluation studied overclocking in three scenarios, including as a performance-enhancing feature coupled with VM auto-scaling. We conclude that two-phase immersion and overclocking have enormous potential for next-generation cloud platforms.

Acknowledgements

We would like to thank the anonymous shepherd and reviewers for helping us improve this paper. We also thank Girish Bablani, Washington Kim, Nithish Mahalingam, Mark Shaw, Brijesh Warrier and Yanzhong Xu for their many helpful comments and suggestions.

References

[1] 3M, "3M™ Microporous Metallic Boiling Enhancement Coating (BEC) L-20227," https://multimedia.3m.com/mws/media/563566O/3mtm-microporous-metallic-boiling-enhancement-coating-l-20227.pdf?fn=L20227_6003603.pdf.

[2] ——, "Two-Phase Immersion Cooling," https://multimedia.3m.com/mws/media/1602994O/novec-immersion-cooling-article-english.pdf.

[3] ——, "Two-Phase Immersion Cooling A Revolution in Data Center Efficiency," Tech. Rep., 2015.

[4] S. I. Abrita, M. Sarker, F. Abrar, and M. A. Adnan, "Benchmarking vm startup time in the cloud," in *Benchmarking, Measuring, and Optimizing*, C. Zheng and J. Zhan, Eds., 2019.

[5] Alibaba Cloud, "Immersion Cooling for Green Computing," https://www.opencompute.org/files/Immersion-Cooling-for-Green-Computing-V1.0.pdf.

[6] Amazon, "AWS Sustainability," https://sustainability.aboutamazon.com/environment/the-cloud.

[7] Amazon EC2, "Amazon EC2 Auto Scaling," https://aws.amazon.com/ec2/autoscaling/.

[8] ——, "Predictive Scaling for EC2, Powered by Machine Learning," https://aws.amazon.com/blogs/aws/new-predictive-scaling-for-ec2-powered-by-machine-learning/.

[9] ——, "Processor state control for your EC2 instance," https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/processor_state_control.html.

[10] AMD, "Turbo Core Technology," https://www.amd.com/en/technologies/turbo-core.

[11] Apache Hadoop, "TeraSort," https://hadoop.apache.org/docs/current/api/org/apache/hadoop/examples/terasort/package-summary.html.

[12] 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, 2013.

[13] H. Cai, X. Zhou, Q. Cao, H. Jiang, F. Sheng, X. Qi, J. Yao, C. Xie, L. Xiao, and L. Gu, "GreenSprint: Effective Computational Sprinting in Green Data Centers," in *Proceedings of the International Parallel and Distributed Processing Symposium*, 2018.

[14] H. Cai, Q. Cao, F. Sheng, Y. Yang, C. Xie, and L. Xiao, "Esprint: Qos-aware management for effective computational sprinting in data centers," in *Proceedings of the International Symposium on Cluster, Cloud and Grid Computing*, 2019.

[15] T. J. Chainer, M. D. Schultz, P. R. Parida, and M. A. Gaynes, "Improving data center energy efficiency with advanced thermal management," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 7, no. 8, pp. 1228–1239, 2017.

[16] E. Cortez, A. Bonde, A. Muzio, M. Russinovich, M. Fontoura, and R. Bianchini, "Resource central: Understanding and predicting workloads for improved resource management in large cloud platforms," in *Proceedings of the Symposium on Operating Systems Principles*, 2017.

[17] Y. Cui, C. Ingalz, T. Gao, and A. Heydari, "Total cost of ownership model for data center technology evaluation," in *Proceedings of the Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, 2017.

[18] H. David, E. Gorbatov, U. R. Hanebutte, R. Khanna, and C. Le, "Rapl: Memory power estimation and capping," in *Proceedings of the International Symposium on Low-Power Electronics and Design*, 2010.

[19] D. DiMaria and J. Stathis, "Non-arrhenius temperature dependence of reliability in ultrathin silicon dioxide films," *Applied Physics Letters*, vol. 74, no. 12, 1999.

[20] T. Endo, A. Nukada, and S. Matsuoka, "Tsubame-kfc: A modern liquid submersion cooling prototype towards exascale becoming the greenest supercomputer in the world," in *Proceedings of the International Conference on Parallel and Distributed Systems*, 2014.

[21] S. Fan, S. M. Zahedi, and B. C. Lee, "The Computational Sprinting Game," in *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems*, 2016.

[22] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *Proceedings of the International Symposium on Computer Architecture*, 2007.

[23] Y. Fan, C. Winkel, D. Kulkarni, and W. Tian, "Analytical design methodology for liquid based cooling solution for high tdp cpus," in *Proceedings of the Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, 2018.

[24] I. Goiri, K. Le, J. Guitart, J. Torres, and R. Bianchini, "Intelligent placement of datacenters for internet services," in *Proceedings of the International Conference on Distributed Computing Systems*, 2011.

[25] Google, "Cloud Tensor Processor Unit," https://cloud.google.com/tpu.

[26] Google, "GCP Sustainability," https://cloud.google.com/sustainability.

[27] B. Greskamp and J. Torrellas, "Paceline: Improving Single-Thread Performance in Nanoscale CMPs through Core Overclocking," in *Proceedings of the International Conference on Parallel Architecture and Compilation Techniques*, 2007.

[28] O. Hadary, L. Marshall, I. Menache, A. Pan, E. E. Greeff, D. Dion, S. Dorminey, S. Joshi, Y. Chen, M. Russinovich, and T. Moscibroda, "Protean: VM Allocation Service at Scale," in *Proceedings of the Symposium on Operating Systems Design and Implementation*, 2020.

[29] C. Hsu, Y. Zhang, M. A. Laurenzano, D. Meisner, T. Wenisch, J. Mars, L. Tang, and R. G. Dreslinski, "Adrenaline: Pinpointing and reining in tail queries with quick voltage boosting," in *Proceedings of the International Symposium on High-Performance Computer Architecture*, 2015.

[30] Z. Huang, J. A. Joao, A. Rico, A. D. Hilton, and B. C. Lee, "DynaSprint: Microarchitectural Sprints with Dynamic Utility and Thermal Management," in *Proceedings of the International Symposium on Microarchitecture*, 2019.

[31] Intel, "New 2nd generation intel xeon scalable processor." [Online]. Available: https://www.intel.com/content/dam/www/public/us/en/documents/guides/2nd-gen-xeon-sp-transition-guide-final.pdf

[32] Intel, "New Reliability, Availability, and Serviceability (RAS) Features in the Intel® Xeon® Processor Family," https://software.intel.com/content/www/us/en/develop/articles/new-reliability-availability-and-serviceability-ras-features-in-the-intel-xeon-processor.html.

[33] ——, "Turbo Boost Technology 2.0," https://www.intel.com/content/www/us/en/architecture-and-technology/turbo-boost/turbo-boost-technology.html.

[34] B. Kaczer, R. Degraeve, M. Rasras, K. Van de Mieroop, P. J. Roussel, and G. Groeseneken, "Impact of mosfet gate oxide breakdown on digital circuit operation and reliability," *IEEE Transactions on Electron Devices*, 2002.

[35] F. Kaplan and A. K. Coskun, "Adaptive Sprinting: How to get the most out of Phase Change based passive cooling," in *Proceedings of the International Symposium on Low Power Electronics and Design*, 2015.

[36] S. Kondguli and M. Huang, "A Case for a More Effective, Power-Efficient Turbo Boosting," *Transactions on Architecture and Code Optimization*, vol. 15, no. 1, 2018.

[37] J. Koomey, K. Brill, P. Turner, J. Stanley, and B. Taylor, "A simple model for determining true total cost of ownership for data centers," *Uptime Institute White Paper*, 2007.

[38] A. Kumbhare, R. Azimi, I. Manousakis, A. Bonde, F. Frujeri, N. Mahalingam, P. Misra, S. A. Javadi, B. Schroeder, M. Fontoura, and R. Bianchini, "Prediction-based power oversubscription in cloud platforms," in *Proceedings of the USENIX Annual Technical Conference*, 2021, to appear.

[39] Y. Li, C. R. Lefurgy, K. Rajamani, M. S. Allen-Ware, G. J. Silva, D. D. Heimsoth, S. Ghose, and O. Mutlu, "A Scalable Priority-Aware Approach to Managing Data Center Server Power," in *Proceedings of the International Symposium on High-Performance Computer Architecture*, 2019.

[40] D. Lo and C. Kozyrakis, "Dynamic management of TurboMode in modern multi-core chips," in *Proceedings of the International Symposium on High-Performance Computer Architecture*, 2014.

[41] I. Manousakis, S. Sankar, G. McKnight, T. D. Nguyen, and R. Bianchini, "Environmental Conditions and Disk Reliability in Free-Cooled Datacenters," in *Proceedings of the International Conference on File and Storage Technologies*, 2016.

[42] D. Marcon, T. Kauerauf, F. Medjdoub, J. Das, M. Van Hove, P. Srivastava, K. Cheng, M. Leys, R. Mertens, S. Decoutere *et al.*, "A comprehensive reliability investigation of the voltage-, temperature-and device geometry-dependence of the gate degradation on state-of-the-art GaN-on-Si HEMTs," in *Proceedings of the International Electron Devices Meeting*, 2010.

[43] A. Mazouz, A. Laurent, B. Pradelle, and W. Jalby, "Evaluation of cpu frequency transition latency," *Computer Science-Research and Development*, vol. 29, no. 3–4, 2014.

[44] J. McCalpin, "Stream: Sustainable memory bandwidth in high performance computers," http://www.cs.virginia.edu/stream/.

[45] Microsoft, "Azure Sustainability," https://azure.microsoft.com/en-us/global-infrastructure/sustainability/.

[46] Microsoft, "Disk speed," https://github.com/Microsoft/diskspd.

[47] ——, "To cool datacenter servers, Microsoft turns to boiling liquid," https://news.microsoft.com/innovation-stories/datacenter-liquid-cooling/.

[48] Microsoft Azure, "Overview of autoscale in Microsoft Azure," https://docs.microsoft.com/en-us/azure/azure-monitor/platform/autoscale-overview.

[49] ——, "Virtual Machine series," https://azure.microsoft.com/en-us/pricing/details/virtual-machines/series/.

[50] N. Morris, C. Stewart, L. Chen, R. Birke, and J. Kelley, "Model-Driven Computational Sprinting," in *Proceedings of the European Conference on Computer Systems*, 2018.

[51] N. Mubeen, "Workload frequency scaling law: Derivation and verification," *Communications of the ACM*, vol. 61, no. 9, 2018.

[52] T. Norrie, N. Patil, D. H. Yoon, G. Kurian, S. Li, J. Laudon, C. Young, N. P. Jouppi, and D. Patterson, "Google's training chips revealed: TPUv2 and TPUv3," in *Proceedings of the Hot Chips Symposium*, 2020.

[53] Open Compute Project, "Server/ProjectOlympus," https://www.opencompute.org/wiki/Server/ProjectOlympus.

[54] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems 32*, 2019.

[55] M. K. Patterson, "The Effect of Data Center Temperature on Energy Efficiency," in *Proceedings of the Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, 2008.

[56] I. Paul, S. Manne, M. Arora, W. L. Bircher, and S. Yalamanchili, "Cooperative Boosting: Needy versus Greedy Power Management," in

[60] A. Raghavan, Y. Luo, A. Chandawalla, M. Papaefthymiou, K. P. Pipe, T. F. Wenisch, and M. M. Martin, "Computational Sprinting," in *Proceedings of the International Symposium on High-Performance Computer Architecture*, 2012.

Proceedings of the International Symposium on Computer Architecture, 2013.

[57] J. A. Prybyla, S. P. Riege, S. P. Grabowski, and A. W. Hunt, "Temperature dependence of electromigration dynamics in al interconnects by real-time microscopy," *Applied Physics Letters*, vol. 73, no. 8, pp. 1083–1085, 1998.

[58] PyTorch, "VGG-NETS," https://pytorch.org/hub/pytorch_vision_vgg/.

[59] A. Raghavan, L. Emurian, L. Shao, M. Papaefthymiou, K. P. Pipe, T. F. Wenisch, and M. M. Martin, "Computational sprinting on a hardware/software testbed," in *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems*, 2013.

[61] B. Ramakrishnan, Y. Hadad, S. Alkharabsheh, P. R. Chiarot, and B. Sammakia, "Thermal Analysis of Cold Plate for Direct Liquid Cooling of High Performance Servers," *Journal of Electronic Packaging*, vol. 141, no. 4, 2019.

[62] V. Sakalkar, V. Kontorinis, D. Landhuis, S. Li, D. De Ronde, T. Blooming, A. Ramesh, J. Kennedy, C. Malone, J. Clidaras, and P. Ranganathan, "Data Center Power Oversubscription with a Medium Voltage Power Plane and Priority-Aware Capping," in *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems*, 2020.

[63] T. E. Sarvey, Y. Zhang, C. Cheung, R. Gutala, A. Rahman, A. Dasu, and M. S. Bakir, "Monolithic Integration of a Micropin-Fin Heat Sink in a 28-nm FPGA," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 7, no. 10, 2017.

[64] Standard Performance Evaluation Corporation, "Spec jbb," https://www.spec.org/jbb2015/.

[65] H. Su, F. Liu, A. Devgan, E. Acar, and S. Nassif, "Full Chip Leakage Estimation Considering Power Supply and Temperature Variations," in *Proceedings of the International Symposium on Low-Power Electronics and Design*, 2003.

[66] Y. Sun, N. B. Agostini, S. Dong, and D. Kaeli, "Summarizing cpu and gpu design trends with product data," 2019, arXiv.

[67] M. Thouless, J. Gupta, and J. Harper, "Stress development and relaxation in copper films during thermal cycling," *Journal of Materials Research*, vol. 8, no. 8, p. 1845–1852, 1993.

[68] P. E. Tuma, "The merits of open bath immersion cooling of datacom equipment," in *Proceedings of the Semiconductor Thermal Measurement and Management Symposium*, 2010.

[69] E. Wu, J. Sune, W. Lai, E. Nowak, J. McKenna, A. Vayshenker, and D. Harmon, "Interplay of voltage and temperature acceleration of oxide breakdown for ultra-thin gate oxides," *Solid-State Electronics*, vol. 46, no. 11, 2002.

[70] Q. Wu, Q. Deng, L. Ganesh, C.-H. Hsu, Y. Jin, S. Kumar, B. Li, J. Meza, and Y. J. Song, "Dynamo: Facebook's data center-wide power management system," in *Proceedings of the International Symposium on Computer Architecture*, 2016.

[71] J. Yang and J. Seymour, "Pmbench: A micro-benchmark for profiling paging performance on a system with low-latency ssds," in *New Generations Information Technology*, S. Latifi, Ed., 2018.

[72] A. M. Yassine, H. Nariman, M. McBride, M. Uzer, and K. R. Olasupo, "Time dependent breakdown of ultrathin gate oxide," *IEEE Transactions on Electron Devices*, vol. 47, no. 7, 2000.

[73] W. Zheng, X. Wang, Y. Ma, C. Li, H. Lin, B. Yao, J. Zhang, and M. Guo, "SprintCon: Controllable and Efficient Computational Sprinting for Data Center Servers," in *Proceedings of the International Parallel and Distributed Processing Symposium*, 2019.

[74] Y. Zhong, "A Large Scale Deployment Experience Using Immersion Cooling in Datacenters." Alibaba Group: Open Compute Project Summit, 2019.