GreenWeb: Hosting High-Load Websites Using Low-Power Servers

Brad Everman and Ziliang Zong
Computer Science Department, Texas State University, San Marcos, Texas
brad@txstate.edu, ziliang@txstate.edu

***Abstract*—Today, there are millions of web servers hosting billions of websites. To ensure quality of service (QoS), it has become conventional wisdom that websites, especially high-load websites, must be deployed on high-end servers. In fact, most of these costly servers are energy-hungry and vastly underutilized, thereby wasting significant amounts of energy and dollars. This paper explores the viability of using low-power commodity servers to host high-load websites while still maintaining comparable QoS. We demonstrate that, with certain software optimization (e.g. caching and content delivery network - CDN) enabled, low-power servers are able to host high-load websites without degrading the quality of web services.**

***Index Terms*—Energy Efficiency, High-Load Websites, Low-Power Servers**

I. Introduction

In 2018, Netcraft reports that there are over 1.7 billion websites worldwide, which are hosted on more than 7 million web servers [1]. To guarantee the response time of web services, most enterprise websites with high- or medium- load are deployed on powerful and energy-hungry servers. It has become a common practice and conventional wisdom that high-load websites must be deployed on high-end servers, despite the fact these web servers are largely underutilized during their daily operation. According to the McKinsey report [2], typical enterprise server utilization is merely 5% - 15%. This low server utilization is also supported by a report from HostGator, a popular web-hosting platform [3].

With such a low utilization, hosting websites on high-end servers may not be a cost-effective solution. Suppose we have two server options to host a website. The high-end server has two Intel E5-2630 processors with a Passmark score of 16,123 [4] and a total thermal design power (TDP) of 160W. Passmark is a widely used benchmarking utility that rates the performance of a server. The low-power server has a quad-core Intel Celeron J3455 processor with a TDP of only 10W, and a much lower Passmark score of 2,138 [5]. Comparing the Passmark score and TDP of both processors, the low-end Celeron processor is capable of performing twice as much work per watt at full load. This sheds light that using low-power servers to host websites could be a viable and more effective solution, provided that the QoS can meet user expectations.

This work answers the following three basic questions:

1) Is deploying high-load websites on low-power servers a viable alternative solution overall?
2) If not a completely viable replacement, what level of service is a low-power server capable of maintaining?
3) What software optimization can help improve the QoS of high-load websites running on low-power servers?

To answer these questions with convincing quantitative evaluation, it is crucial to select a website that can represent real-world workloads of high-load websites powered by recent and popular web technologies. We choose the FEST website, which provides information for a large annual festival in Texas (the URL of the website is undisclosed for privacy concerns), for a number of reasons:

1) Availability - Direct administrative access is available.
2) Analytics - Web traffic statistics are accessible.
3) Applicability - Several hundred thousand visitors use the website, thus providing a legitimate source of real-world workload (average 60,000 requests/day during the festival and the peak traffic is about 85,000 daily views).
4) WordPress - FEST is powered by the WordPress Content Management System (CMS) - one of the most popular web technologies today. WordPress is deployed on approximately 28.9% of all websites [6] such as CNN, CBS, BBC etc. Therefore, measuring a web host's ability to serve a WordPress site is an excellent indicator of real-world relevance.
5) Portability - FEST uses a plugin called All-In-One WordPress Migration [7] - which creates an archived version of FEST in its current state. The plugin provides the capability to import a 100% accurate copy of the WordPress site, which ensures that FEST would run identically on all low-power servers being evaluated.

The experimental results of five severs (one high-end server and four low-power servers) show that low-end servers have great potential to meet the QoS requirement of high- or medium- loads websites with much lower power consumption.

The remainder of this paper is organized as follows. Section II describes the system configuration and testing methodology. Section III analyzes the experimental results in terms of both QoS and energy efficiency. Section IV discusses related work. Section V concludes the paper and points out the weakness of our current work for future research.

II. System Configuration and Test Methodology

A. Web Servers Hardware Specification

Five servers are evaluated in our experiments (Please refer to Table I for detailed hardware specifications).

978-1-5386-7466-6/18/$31.00 ©2018 IEEE

TABLE I
THE SPECIFICATION OF EVALUATED SERVERS

Name	CPU	Memory	Disk	CPU TDP
Synology DS718+ [8]	Celeron J3455	10GB DDR3	2x120GB SSDs in RAID1	10W
Synology DS1815+ [9]	Atom C2538	6GB DDR3	4x6TB HDD in SHR [10]	15W
Apple Mac Mini [11]	i5-3210M	16GB DDR3	1TB Fusion Drive [12]	35W
Dell PowerEdge 410	2x Xeon E5504	16GB UDIMM	2x500GB SAS HDD in RAID1	2x80W
Raspberry Pi 3 Model B [13]	ARM Cortex-A53	1GB	32GB Micro SD card	$<$1W

B. Web Servers Software Configuration

Ubuntu 16.04.4 LTS [14] is selected as the host operating system to deploy FEST because it is easy to install and compatible with all servers except the Raspberry Pi. Debian Raspbian [15] is chosen as the operating system for Raspberry Pi instead. The two Synology servers run a proprietary operating system called DSM [16], which provides a Virtual Machine (VM) manager and allows a VM to be deployed as needed. For both Synology devices, only one VM is created utilizing all available processor cores, 20 GB of disk space, and 2 GB of memory. As peak memory usage never exceeds 1 GB during testing, limiting the memory has no performance impact and allows ease of VM migration between devices.

After installing a barebone Ubuntu/Raspbian on each system or VM (including an ssh daemon for remote access), Virtualmin [17], a LAMP [18] administration interface, is installed immediately. Virtualmin downloads, installs, and configures the entire web hosting environment, and provides a means to easily deploy a WordPress instance automatically via Perl scripts. This greatly accelerates the deployment process while maintaining the desired homogeneous environment. The LAMP stack is configured with Apache 2.4.18(Ubuntu), PHP 7.0.28-0ubuntu0.16.04.1, and MySQL 14.14 Distrib 5.7.21, the most current versions in the Ubuntu 16.04 LTS repository. As Raspbian is based on Debian Stretch, the Pi is configured with Apache 2.4.25-3+deb9u4, PHP 7.0.27-0+deb9u1, and MariaDB 10.1.26-0+deb9u1.

Using the Virtualmin scripting utility, WordPress 4.9.5 is installed in the base directory of the virtual host. The FEST website is imported with the migration plugin, and then tested to ensure all content is loaded and all links work properly.

C. User Behavior Analysis

Historical user behavior analysis indicates that users typically load the main page first, then select a menu item (usually schedule or buy-tickets). Users usually spend between 1 - 6 seconds reading the current page. It is also observed that the main page, buy-tickets page, and schedule page constitute over 70% of the total website traffic. Inspired by these observations, a tool is created using Locust [19], to scale simultaneous connections and stress test each server. A Locust script loads the main page, the buy-tickets page, and the schedule page. Since the frequency of visiting these pages is different, each page load is weighted according to the frequency of hits, with the main page weighted as three, the buy-tickets page as two, and the schedule page as one. The weights represent how many times a particular page is visited compared to other pages, i.e. in this case the main page would be loaded three times for every single loading of the schedule page. The minimum and maximum wait times are set to 1 and 6 seconds, respectively. The wait periods determine how long a simulated user will wait to move from one page to another page [20]. These values are selected to reflect the user behavior on the FEST website. A separate server (with i3-3240 CPU, 16GB DDR3 memory and 1TB SSD) runs the load stress testing script to simulate user behaviors at different intensities.

D. Power Measurement

A Watts Up? .Net power meter is used to measure the power consumed by each server. A Python script written by Yongpil Yoon [21] is used to record the power data in one second intervals as the system is running the load testing script.

E. Uptime

For each system, another local script is started before the test begins, which records the uptime of the run. This aims to monitor the load averages of each system during the test and match those results with the power data profiled by the Watts Up? Python script. Both systems are forced to synchronize their clocks using the Network Time Protocol (NTP [22]) before each experiment to ensure data consistency. Clock synchronization brings an added benefit of being able to easily collate all the data and analyze it at the per-second granularity.

F. Testing Methodology

After each server is set up and configured, we conduct the following identical steps to conduct each experiment:

1) Plug in Watts Up? meter and boot the tested server.
2) Synchronize hardware clocks of each system using NTP.
3) Start uptime recording on the target server.
4) Run the Python script to log power usage to a CSV file.
5) Run the Locust script and record timestamp.
6) Match timestamps of the Python and Locust scripts and compare system load, power usage, and web requests at the per-second level.

The Locust script is executed with varying numbers of simultaneous users: 25, 50, and 100. It is worth noting that the simulated workloads in our experiments can represent real-world high-traffic websites. For example, 25 simultaneous users is roughly equivalent to 15 million hits/month, or similar traffic to geico.com [23], 50 users is equivalent to 30 million hits/month, comparable to cisco.com [23], and 100 users is equivalent to 53 million hits/month, about the same traffic as expedia.com [23]. We calculate the hits/month by considering requests per second as indicated in Figures 1 - 5. Specifically, 25 cached users is equivalent to 6.89 r/s for the DS718+. 6.89*3600 seconds per hour*24 hours*30 days = 17.8 million total hits in a month. The calculations for 50 and 100 simultaneous users can be extrapolated similarly.

Each experiment has five separate runs, which are considered independently and combined to normalize the final results, under the following three different scenarios:

1) Base installation of Wordpress
2) Caching enabled
3) Caching and Content Delivery Network (CDN) enabled

G. Caching and CDN

1) Caching: WordPress performs several tasks to display a web page. When a user requests a web page, the page content is retrieved from a database, then PHP scripts apply styles (e.g. CSS) to the content and produce a valid HTML document that can be displayed on the end-user's web browser. Without caching, these expensive operations must be performed each time a user views a web page. With caching, the web server can create a static HTML page once, and send it to as many users that request it as possible. This greatly reduces the response time and can serve more users with less CPU load.

2) Content Delivery Network (CDN): A CDN is a geographically distributed network of proxy servers that store popular content (e.g. videos and images) that have the greatest demand from websites. Enabling CDN could reduce the workload by offloading some of the work from the FEST server to the CDN server. It can also mitigate slow disk performance of low-power servers by eliminating reads of large images from its local disk drives. Meanwhile, it is possible to increase latency of a website load if the CDN server itself is experiencing problems. As setup of a CDN for testing purposes is not straightforward, the built-in CDN service from the WordPress Jetpack plugin is utilized in our experiments.

III. Results

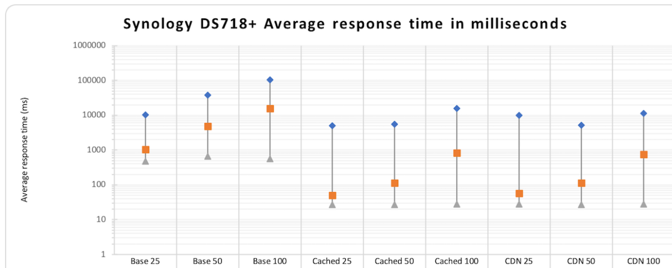
A. Response Time

Per DoubleClick, 53% of all mobile website visits are abandoned if the load time is over three seconds [24]. Anything longer than two seconds is what a normal user calls "annoying". Therefore, we consider a response time of under 2,000 ms as "reasonable", and define anything over 5,000 ms in the "infuriating" category. Obviously, this is a completely subjective measurement, dependent upon the user in question. Nonetheless, for fair comparisons, we create the following latency-based Quality of Service (QoS) evaluation scale metric to explicitly evaluate the performance of each server:

1) Superb QoS <50 ms
2) Great QoS <500 ms
3) Good QoS <1,000 ms
4) Reasonable QoS <2,000 ms
5) Mediocre QoS <3,500 ms
6) Infuriating QoS <5,000 ms
7) Abysmal QoS <8,000 ms
8) Absurd QoS >15,000 ms

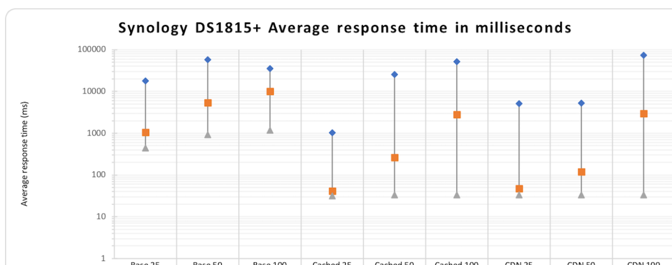
Figures 1 - 5 plot the average response time of all five evaluated servers, which are discussed in detail below. For all figures, CDN refers to enabling **both** CDN and caching.

1) Base Case: The base case refers to the non-cached/non-CDN case. In this case, the two Synology devices are capable of maintaining a *Reasonable* level of service with 25 simultaneous users, with minimal failures. The Pi is closer to *Reasonable*, but still quite usable. The maximum response time for all three servers is *Abysmal*, but these appear to be outliers and are better thought of as failures. The Mac Mini and PowerEdge are both *Great* performers with 25 users, and max response times still being *Reasonable*. With 50 and 100 users, the Pi is manually terminated (no data generated) due to being completely unresponsive. However, the Mac Mini still manages *Good* QoS for 100 users.



	Base 25	Base 50	Base 100	Cached 25	Cached 50	Cached 100	CDN 25	CDN 50	CDN 100
Max	10401	37757	105371	5073	5593	15893	10077	5198	11457
Average	1054	4851	15693	51	114	819	57	113	744
Min	476	651	569	27	27	28	28	27	28
Requests per second	5.51	6.01	4.61	6.89	13.67	22.97	6.92	13.40	22.73
Failures	2	74	896	1	0	22	0	5	33

Fig. 1. DS718+ Response Time



	Base 25	Base 50	Base 100	Cached 25	Cached 50	Cached 100	CDN 25	CDN 50	CDN 100
Max	17697	57713	34867	1027	25588	52003	5071	5273	73781
Average	1055	5373.2	10068	42	266	2838	47	120	2954
Min	442	932	1189	32	33	33	33	33	33
Requests per second	5.51	5.57	1.84	7.00	13.06	15.18	6.90	13.48	14.96
Failures	0	124	970	3	1	29	1	2	34

Fig. 2. DS1815+ Response Time

2) Cache Enabled: Enabling caching dramatically improves response times for all systems. The DS718+ provides *Great* QoS with 50 users and *Good* QoS at 100 users. The DS1815+ performs *Great* with 50 users, but begins to drift into the *Mediocre* range at the level of 100 users. Both the PowerEdge and Mac Mini are *Superb* for all numbers of users.

3) Cache and CDN Enabled: There is no significant difference in the response times between the cached and cached+CDN scenarios, and in a number of cases the response time increases slightly on average when CDN is enabled.

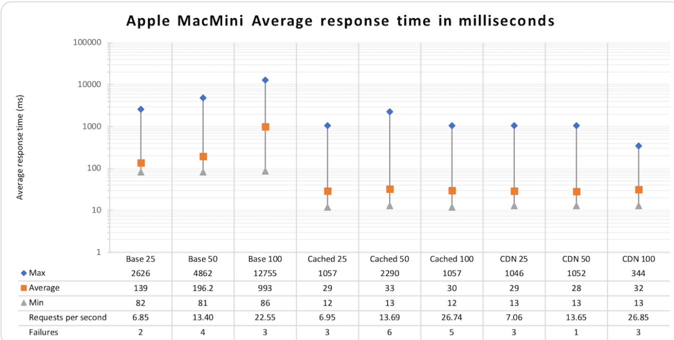


Fig. 3. Mac Mini Response Time

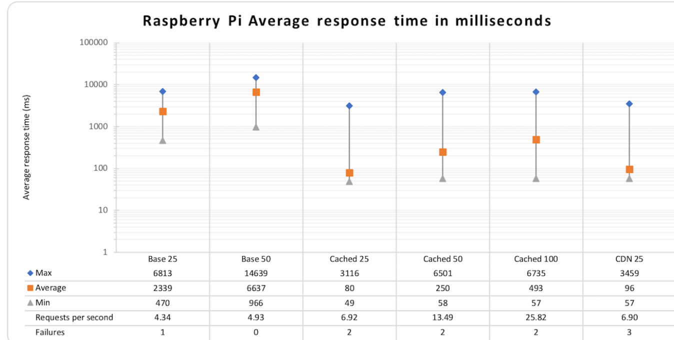


Fig. 5. Pi Response Time

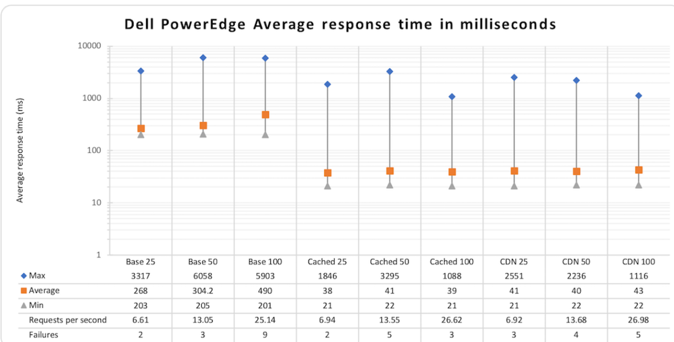


Fig. 4. PowerEdge Response Time

Mac Mini is impressive, which achieves *Great* QoS and only consumes less than 10W (with caching and CDN enabled). The two Synology devices are expected to perform similarly, but the DS1815+ uses more power than the DS718+ probably because it uses HDDs while the DS718+ uses SSDs.

A surprising finding is that the PowerEdge server still consumes noticeable power (13.4W) when being shut down but remaining plugged into the power source. This has been verified multiple times to eliminate the possibility of equipment malfunction as we initially thought the Watts Up? meter was experiencing problems. In fact, 13.4 W is even more than the active power of DS718+, Pi, or Mac Mini while they are running the cached and CDN tests.

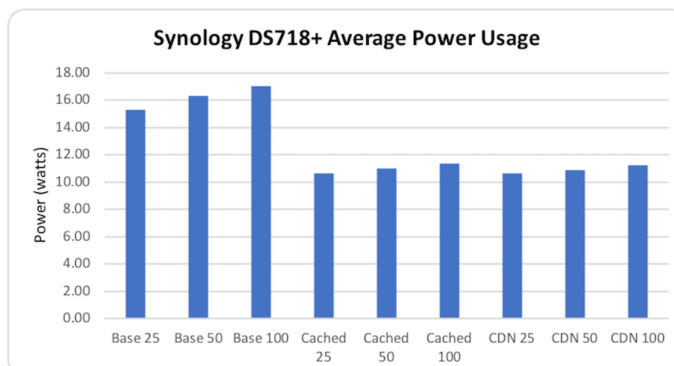


Fig. 6. DS718+ Power Usage

B. Linux System Load Average

The average Linux system load of all five evaluated servers is recorded and calculated by averaging the complete set of 1 minute uptime readings taken per-second during each experiment. In general, the load averages are directly correlated with the response times, as would be expected.

1) Base Case: With 100 simultaneous users, the load is so high for the two Synology devices as to be a completely unfeasible solution. The Pi is unable to handle 25 users.

2) Cache Enabled: With caching, all systems are able to perform below their reasonable maximums.

3) Cache and CDN Enabled: In all cases, enabling the CDN reduces the load averages, which possibly benefits from the decreased I/O requests.

C. Power

Figures 6 - 10 demonstrate the power results of all five servers. As expected, the Dell PowerEdge is the most power-hungry server, and the Pi uses the least amount of power.

In every system, for any number of users, the power usage of the base case is generally much higher than the cached/CDN enabled cases. For example, the power usage of PowerEdge, DS718+, and Mac Mini reduces by approximately 20%, 35%, and 65% respectively for 100 users when caching and CDN are both enabled. The DS1815+ saves roughly 15% of power when comparing the base 50 and cached 50 configurations. With 25 simultaneous users and caching enabled, the Pi does not even show any power usage. The power efficiency of the

D. Summary

To summarize, 1) the response time, system load average, and power usage are directly related. When the number of requests for a web page increases, the system must work harder to maintain a reasonable level of service. This in turn taxes the CPU and I/O, which results in more energy consumption. 2) The Pi is not truly a viable platform for dynamic, high-traffic web sites, but with caching and CDN, it can definitely be used to deploy WordPress instances that have medium workloads and serve a reasonable number of users. 3) The Dell PowerEdge achieves the best performance but consumes vastly more energy than other low power servers. 4) The similarities of the two Synology devices in their CPUs yet

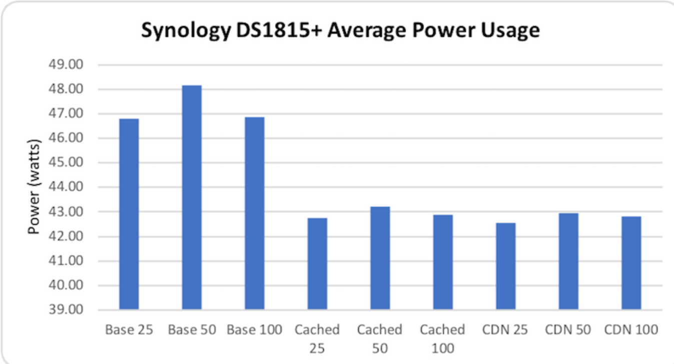


Fig. 7. DS1815+ Power Usage

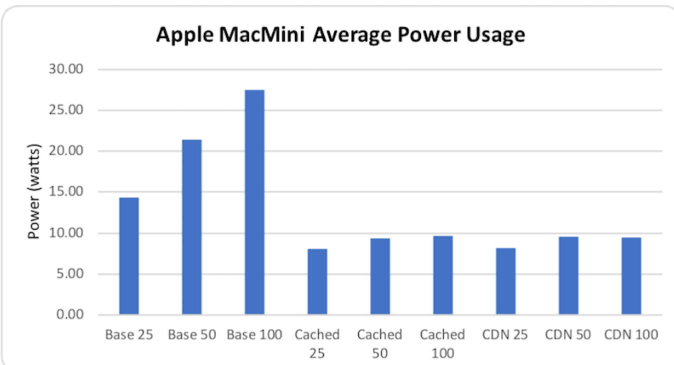


Fig. 8. Mac Mini Power Usage

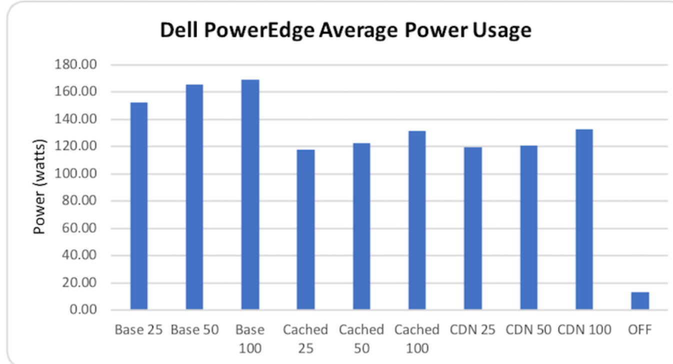


Fig. 9. PowerEdge Power Usage

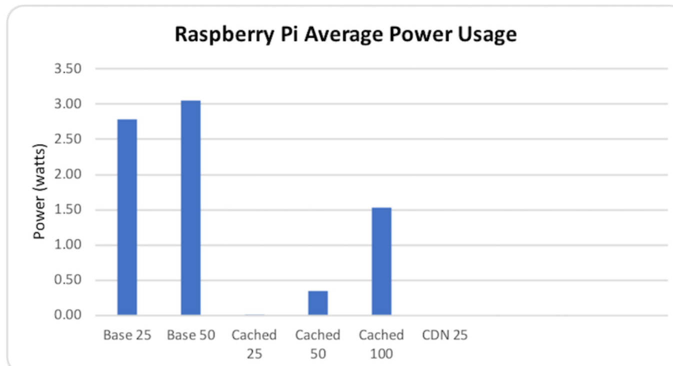


Fig. 10. Pi Power Usage

divergent results, indicate that the utilization of SSDs can significantly improve the performance and energy efficiency of web servers. 5) The superior laptop-based hardware and architecture design make the Mac Mini a high performer (on-par with PowerEdge) but much more energy efficient option for web servers. A more in-depth analysis for each server is discussed as follows:

1) Raspberry Pi: The Pi performs adequately for medium workloads when caching is employed. It is able to handle 6.92 r/s (requests per second), which is equivalent to nearly 600,000 requests per 24-hour period, well over the 85,000 total user requests required by the FEST website on its busiest day. The Pi's extremely low power makes it a reasonable option for a web server when power saving is a top priority.
2) Synology DS718+: The DS718+ with SSDs is an excellent performer, which can handle several million hits per-day when caching is enabled (22.97 r/s for 100 users).
3) Synology DS1815+: The DS1815+ is similar to the DS718+, but the 2X-4X power usage and conventional HDDs make it less effective than DS718+ overall. For a website with reasonable traffic (i.e. less than or equivalent to the base 25 users case), the DS1815+ may be an alternative solution.
4) Mac Mini: The performance and energy efficiency of Mac Mini is highly impressive. When running the 100 users case, the Mini is able to maintain a *Good* QoS with few failures while staying under 30W. With caching and the CDN enabled, 100 users could be handled more quickly than the PowerEdge utilizing merely 1/12th of the energy. The power-to-performance ratio is the best of all the systems tested.
5) Dell PowerEdge: The PowerEdge is fast but consumes the highest power of all servers. Such servers are ubiquitous in hosting commercial websites specifically because they are capable and dependable. The built-in redundancy such as dual NICs and power supplies ensure the server will have little to no downtime.

IV. Related Work

The majority of previous literature focused on using Dynamic Voltage and Frequency Scaling (DVFS) or demand based workload scheduling algorithms to reduce the energy and operational cost of web servers. For example, Abbasi et al. proposed a solution of serving websites from different data centers based upon Dynamic Application Hosting Management (DAHM) [25]. DAHM leveraged physical servers from different geographic regions and reduced latency and consequently power use by communicating less with clients. Similarly, Deng et al. introduced distributed web hosting, which chose cloud-based hosts based on their ability to meet strict service-level agreement (SLA) requirements while simultaneously employing a low carbon footprint [26]. Al-Qudah et al. proposed a dynamic hosting algorithm, which allocated resources to websites based on demand and using DVFS to reduce energy consumption of servers whenever the demand is low [27]. Chen et al. proposed a hybrid mechanism by combining the state queuing analysis and the feedback control theory to reduce energy consumption without sacrificing SLAs [28]. IBM researchers Elnozahy et al. built a simulator to evaluate the impact of different energy conservation policies on web servers [29]. The two key techniques to control different policies on web servers were DVFS and request batching, which saved 17% - 42% of energy for workloads across a broad range of intensities. Bohrer et al. conducted a case study using the 1998 Winter Olympics website [30]. They were able to reduce 23% - 36% of CPU energy by using DVFS while keeping server responsiveness within reasonable limits. Svanfeldt-Winter et al. conducted a cost and energy evaluation of ARM-based web servers, compared with Intel Xeon-Based servers [31]. Varghese et al. proposed a cluster of Raspberry Pi systems to supplant conventional datacenter servers, demonstrating the viability of low-powered systems as a possible web server platform [32]. This is further demonstrated in the work done by Pahleval et al. which indicates that extremely low voltage systems ARM-based systems can provide reliable QoS [33]. Another solution, NapSAC, is proposed by Krioukov et al. [34]. By leveraging low-powered servers in a heterogeneous environment, NapSAC seeks to lower power usage while maintaining QoS via a scheduling algorithm.

The scope of most existing literature is limited to improving the energy efficiency of high-end servers through various optimization, or leveraging low-powered systems in a traditional environment. Our work is distinguished from aforementioned studies by focusing on low-power servers that are traditionally considered incapable of hosting high-load websites and can be utilized "off-the-shelf", without any special configuration. Moreover, this study goes beyond traditional DVFS algorithms and explores the impact of caching and Content Delivery Network (DNS) on QoS and energy efficiency.

V. Conclusions and Future Work

With the booming of the Internet in past decades, the number of websites and web servers has increased tremendously. Conventionally, high-load websites tend to be deployed on high-end servers for the sake of performance and reliability. However, the nature of web applications and website traffic determines that most web servers cannot be fully utilized, which leads to high cost of ownership and huge energy waste. In this paper, we conduct a quantitative study to verify that low-power servers could be a viable solution to replace high-cost servers for hosting high-load websites, provided that the key functionality (e.g. system reliability and virtual machine management) and QoS can be preserved.

Numerous research issues have not been addressed in our current work but are worth exploring in the future. For example, what are the key challenges of running websites on low-power servers? What if some websites have a large amount of web content that cannot fit in the relatively small cache of low-power servers? The current scale of test is on a single web server. How can we address the architecture issues when a cluster of low-power web servers are necessary to host a high-load website? How can we address the inherent weaknesses of low-power servers (e.g. usability and reliability)? Can these low-power servers operate at full capacity 24/7 without failures?

References

[1] https://news.netcraft.com/archives/2018/03/27/march-2018-web-server-survey.html.

[2] https://www.forbes.com/sites/benkepes/2015/06/03/30-of-servers-are-sitting-comatose-according-to-research/.

[3] https://support.hostgator.com/articles/cpu-resource-usage.

[4] https://www.cpubenchmark.net/multi_cpu.html.

[5] https://www.cpubenchmark.net/cpu.php?cpu=Intel+Celeron+J3455+\%40+1.50GHz&id=2875.

[6] https://www.codeinwp.com/blog/wordpress-statistics/.

[7] https://wordpress.org/plugins/all-in-one-wp-migration/.

[8] https://www.synology.com/en-us/products/DS718+.

[9] https://www.cnet.com/products/synology-disk-station-ds1815-plus-nas-server-0-gb/specs/.

[10] https://www.synology.com/en-us/knowledgebase/DSM/tutorial/Storage/What_is_Synology_Hybrid_RAID_SHR.

[11] https://everymac.com/systems/apple/mac_mini/specs/mac-mini-core-i5-2.5-late-2012-specs.html.

[12] Apple Fusion Drive. https://support.apple.com/en-us/HT202574.

[13] https://www.raspberrypi.org/products/raspberry-pi-3-model-b/.

[14] Scale out with Ubuntu Server. https://www.ubuntu.com/server.

[15] https://www.raspberrypi.org/downloads/raspbian/.

[16] https://www.synology.com/en-us/dsm/6.1.

[17] Virtualmin https://virtualmin.com/.

[18] Wikipedia LAMP (software bundle). https://virtualmin.com/.

[19] Locust.io - an open source load testing tool. https://locust.io/.

[20] https://docs.locust.io/en/stable/writing-a-locustfile.html.

[21] https://github.com/yyongpil/wattsup.

[22] https://en.wikipedia.org/wiki/Network_Time_Protocol.

[23] SimilarWeb traffic analytics. https://www.similarweb.com/.

[24] https://www.doubleclickbygoogle.com/articles/mobile-speed-matters/.

[25] Abbasi, Z., Mukherjee, T., Varsamopoulos, G., and Gupta, S.K.S. DAHM: A green and dynamic web application hosting manager across geographically distributed data centers. *ACM Journal on Emerging Technologies in Computing Systems (JETC) 8(4), 2012.*

[26] Deng, N., Stewart, C., Gmach, D., Arlitt, M., and Kelley, J. Adaptive green hosting. ACM ICAC '12, pp. 135–144.

[27] Al-Qudah, Z., Alzoubi, H., Allman, M., Rabinovich, M., and Liberatore, V. Efficient application placement in a dynamic hosting platform. ACM WWW '09, pp. 281–290.

[28] Chen, Y., Das, A., Qin, W., Sivasubramaniam, A., Wang, Q., and Gautam, N. Managing server energy and operational costs in hosting centers. *ACM SIGMETRICS Performance Evaluation*, 2005.

[29] Elnozahy, M., Kistler, M., Rajamony, R. Energy Conservation Policies for Web Servers. *USENIX Symposium on Internet Technologies and Systems*, 2003.

[30] Bohrer, P., Elnozahy, N., Keller, T., Kistler, M., Charles Lefurgy, C., McDowell, C., Rajamony, R. The Case for Power Management in Web Servers. *Power Aware Computing*, 2002.

[31] Svanfeldt-Winter, O., Lafond, S., Lilius, J. Cost and Energy Reduction Evaluation for ARM Based Web Servers. *IEEE International Conference on Dependable, Autonomic and Secure Computing*, 2011.

[32] Varghese, B., Carlsson, N., Jourjon, G., Mahanti, A., Shenoy, P. Greening Web Servers: A Case for Ultra Low-power Web Servers. *Green Computing Conference (IGCC)*, 2014.

[33] Pahlevan, A., Picorel, J., Zarandi, A., Rossi, D., Zapater, M., Bartolini, A., Del Valle, P., Atienza, D., Benini, L., Falsafi, B. Towards Near-Threshold Server Processors. *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2016.

[34] Krioukov, A., Mohan, P., Alspaugh, S., Keys, L., Culler, D., Katz, R. NapSAC: Design and Implementation of a Power-proportional Web Cluster. *Proceedings of the First ACM SIGCOMM Workshop on Green Networking*, 2010.