Evaluating Single Board Computer Clusters for Cyber Operations

Suzanne J. Matthews, Raymond W. Blaine, and Aaron F. Brantly

***Abstract*—The emergence of single board computers (SBCs) has enabled individuals cheap and portable access to multi-core architectures. In this paper, we discuss the use of SBC clusters to assist in cyberspace operations. The small form-factor of SBCs make them highly portable, allowing soldiers to easily transport individual units and clusters. While each individual SBC is not very powerful, a cluster of SBCs can greatly increase the computational power available for cyberspace applications down range for relatively low cost. We discuss common SBC architectures and present a case study in which two clusters of SBCs are used to crack canonically "weak" passwords encoded with bcrypt. Our results show that an 8-node Parallella SBC cluster can crack password files up to 5.95 times faster than a high end laptop, at roughly half the cost. We also present several novel applications for offensive and defensive cyberspace operations using SBCs and SBC clusters. We believe that our work can be used to develop novel parallel military applications incorporating SBCs, and is useful for educating soldiers and end-users about the potentials (and dangers) of parallel processing.**

***Index Terms*—Cyber security, single board computer, SBC, Raspberry Pi, Parallella, cluster, military.**

I. Introduction

The growing cheapness and shrinking nature of computer hardware has led to the emergence of single board computers (SBCs). Highly portable, affordable, and extremely power efficient, SBCs are very popular in the hobbyist and maker communities and have been adapted for a variety of projects. However, the low cost and power efficiency of SBCs are largely due to their relatively weak processors. Laptop computers, while larger and more expensive, provide performance that far outstrip single SBCs while maintaining high portability.

In this paper, we examine the utility of SBC clusters for offensive and defensive cyberspace operations. Offensive Cyberspace Operations (OCO) are intended to project power by the application of force in and through cyberspace. Defensive Cyberspace Operations (DCO) protect national interests and infrastructure against external threats. While OCO and DCO do not often require high performance computational capabilities, there are instances in which it is necessary for operational objectives. Traditional high-performance computing (HPC) clusters are extremely expensive to implement and maintain in a tactical environment. While electronic warfare officers may communicate remotely with HPC systems, this is extremely time-consuming, especially when the war-fighter is limited by access to a low-bandwidth wireless network. In contrast, SBC clusters can be easily deployed in a tactical environment, and provide immediate high computational performance at relatively low cost. The elimination of satellite communication in the workflow can reduce the time needed to gain a tactical advantage by several hours to minutes.

To illustrate the performance of SBC clusters, we conduct a case study on password cracking in which we compare the clusters to a high-end laptop. We focus on two popular multi-core SBCs, the Raspberry Pi 2 and the Parallella, and build a 128-core SBC cluster of each. Next, we encode $5,000$ commonly used "weak" passwords with bcrypt, and compare the cracking speeds of our SBC clusters with the aforementioned laptop. Our experimental benchmarking with John the Ripper (JtR) indicates that the Parallella and Raspberry Pi 2 clusters can crack our password files up to 5.95 and 3.63 times faster (respectively) than the laptop computer at roughly half the cost. We also discuss future use-cases where such systems might be useful for OCO and DCO applications.

The rest of the paper is organized as follows. Section II discusses our SBCs and SBC clusters under study and compares their form factor and cost to the laptop in our study. Section III discusses our JtR case study and its results. Section IV presents theoretical cyberspace applications for SBC clusters. We conclude our paper in Section V and make some suggestions for future avenues of exploration.

II. Overview of Systems

While there are many types of SBCs, we focus on the Raspberry Pi 2 and Parallella. The Raspberry Pi is arguably the most popular and widely-known SBC, with the Raspberry Pi 2 being equipped with a quad-core ARM processor. The Parallella, while not as universally known, is arguably the most powerful credit-card sized SBC, with a 16-core Epiphany co-processor. For comparison purposes, we build a 128-core Raspberry Pi 2 and a separate 128-core Parallella cluster. We discuss the form factor, cost, and portability of each of these clusters below.

A. The Raspberry Pi Computer

The Raspberry Pi is a low-power, credit-card sized SBC initially released in 2012. The Raspberry Pi 2 (released in February 2015, Figure 1) is equipped with 1 GB of RAM, a 900 Mhz quad-core ARM Cortex A7 CPU, 10/100 Ethernet,

SJ Matthews is with the Department of Electrical Engineering and Computer Science, United States Military Academy, West Point, NY, 10996 USA. e-mail: suzanne.matthews@usma.edu

RW Blaine is with the U.S. Army Cyber Protection Brigade, Fort Gordon GA USA. e-mail: raymond.w.blaine.mil@mail.mil

AF Brantly is with the Army Cyber Institute, United States Military Academy, West Point, NY 10996 USA. e-mail: aaron.brantly@usma.edu

Manuscript received July 1, 2016; accepted August 15, 2016.

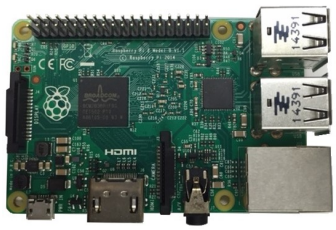

Fig. 1. Raspberry Pi 2 4-core SBC.

Fig. 2. Parallella 16-core SBC.

and retails for $35.00. It supports both the Linux and Windows operating systems. A single 5-Volt DC 2-Amp power supply is sufficient to power the device, and the unit consumes up to 4 watts of power. The Raspberry Pi is very popular, having been adapted for a variety of applications, including wireless sensor networks [1], robotics [2], and UAVs [3].

Several researchers have explored how to harness the Raspberry Pi for high-performance applications. Notable projects include IridisPi [4], PiCloud [5], Boise State [6], and FUB [7]. In all of these efforts, the low performance of individual Raspberry Pi nodes is offset by networking the devices together to work in tandem. The Raspberry Pi 2's low power requirements allow for the creation of cheap, energy-efficient clusters that can serve as a test-bed for several parallel applications.

The Raspberry Pi's small form-factor and high portability also makes it attractive to cyber security professionals. Muniz and Lakhani recently published a book [8] that focuses on the Raspberry Pi and Kali Linux for penetration testing. Muniz notes that the device would be relatively easy to hide at a target's location and is useful as a remote penetration-testing unit [9]. Other efforts such as the Rogue Pi [10] and Glastopf Pi [11] explore using the Raspberry Pi for packet sniffing and as honey pot servers respectively.

B. The Parallella Computer

The Parallella (Figure 2) is another credit-card sized SBC with huge potential for high performance and security applications. First made available to the general public in 2014, the Parallella has a dual-core ARM A9 CPU, a 16-core Epiphany co-processor, and 1 GB of RAM. The Parallella can be powered by a single 5-Volt DC 2.5-Amp power supply, and the unit consumes up to 5 watts of power. However, the Parallella's increased computational capabilities require it to have a much higher price-point than the Raspberry Pi 2. The microserver edition of the Parallella retails at $99.00, while the desktop edition retails at $149.00.

While the Parallella is a much newer system than the Raspberry Pi, several researchers have explored the efficacy

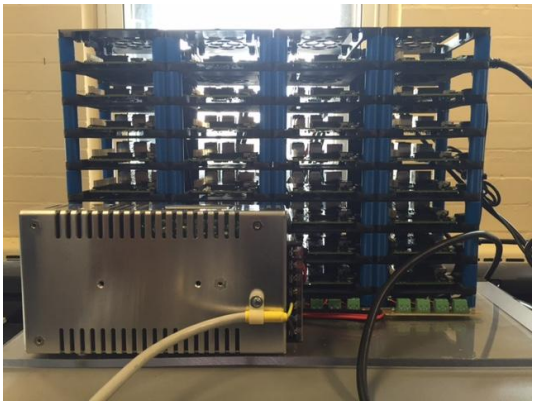

Fig. 3. 32-node Raspberry Pi 2 Cluster.

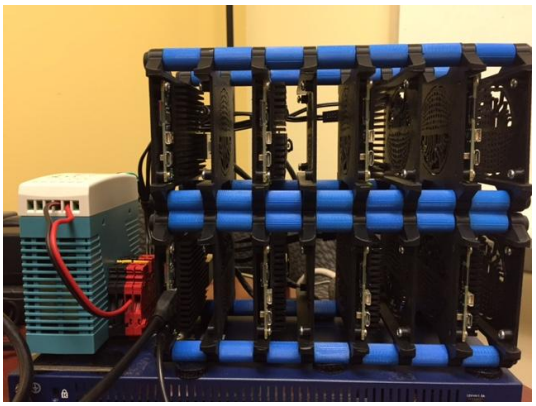

Fig. 4. 8-node Parallella Cluster.

of high-performance applications on the Parallella, developing new programming models and languages. A few parallel efforts [12], [13], [14] explore image processing on the Parallella, and measure the board's performance in both floating point operations and power consumption. While current GPU technology outstrips the Parallella's processing power, the Parallella is small, self-contained, and extremely energy efficient. Recently, researchers have begun developing compilers that enable people to use popular parallel libraries such as OpenMP [15] and MPI [16] on the Parallella. It is expected that in the coming years, the language and library support for the board will only increase.

C. Overview of SBC Clusters

We construct a 128-core Beowulf cluster using each of our SBCs under study. The Raspberry Pi 2 cluster (Figure 3) consists of thirty-two Raspberry Pi SBCs organized in a 4×8 configuration, for a total of 128 ARM compute cores. This cluster includes two 16-port 10/100 switches, two 5-Volt DC 20-Amp power supplies, and a 500 GB USB mounted hard drive that acts as a Network File System (NFS) drive for the cluster. The Parallella cluster (Figure 4) consists of eight Parallella SBCs organized in a 4×2 grid, for a total of 128 Epiphany cores. The cluster includes a 16-port Gigabit Ethernet switch, a dedicated 5-Volt DC 10-Amp power supply, and a 500 GB USB hard-drive that serves as a NFS.

Component	No. of Units	Cost p/Unit	Total
Raspberry Pi Nodes	32	$35.00	$1120.00
5V (20A) power supply	2	$39.95	$79.90
8GB MicroSD card	32	$5.98	$191.36
Case Components	32	$5.00	$160.00
Ethernet Cables	32	$2.00	$64.00
Router	1	$20.00	$20.00
16-port 10/100 switch	2	$48.47	$96.94
Powered USB hub	1	$22.93	$22.93
Cooling fan	2	$15.00	$30.00
500 GB USB HD	1	$50.00	$50.00
TOTAL	**137**		**$1,835.13**

TABLE I
RASPBERRY PI 2 CLUSTER COMPONENT COSTS.

Component	No. of Units	Cost p/Unit	Total
Parallella Nodes	8	$149.00	$1192.00
5V (10A) power supply	1	$19.95	$19.95
8GB MicroSD card	8	$5.98	$47.84
Case Components	8	$5.00	$40.00
Ethernet Cables	8	$2.00	$16.00
Router	1	$20.00	$20.00
16-port Gigabit switch	1	$71.99	$71.99
Powered USB hub	1	$22.93	$22.93
Cooling fan	1	$15.00	$15.00
500 GB USB HD	1	$50.00	$50.00
TOTAL	**38**		**$1,457.78**

TABLE II
PARALLELLA CLUSTER COMPONENT COSTS.

Tables I and II illustrate cost breakdowns for our two clusters. Note that we do not include the cost of peripherals such as keyboard, monitor, and mouse. While the individual Parallella units are more expensive than Raspberry Pi units, it is $\$377.35$ cheaper to create our Parallella cluster. The latter cluster also enjoys a smaller form factor: it measures $11.25"L \times 4"W \times 9"H$, weighs approximately 5 pounds, and draws approximately 74 Watts of power. In contrast, the Raspberry Pi cluster measures approximately $17.5"L \times 10"W \times 9.75"H$, weighs approximately 15 pounds, and draws approximately 174 watts of power. Laptops typically use 50 to 100 watts of power, while desktop computers use around 200 watts. The Parallella and Raspberry Pi 2 clusters have 38 and 137 components respectively. Based on size, weight, and cost, the Parallella cluster is cheaper and more portable than the Raspberry Pi cluster, and is comparable in weight and power consumption to a laptop. We note that GPUs, while enabling high performance, can independently consume over 300 watts of power [17]. Since GPUs require a CPU to act as a supervisory host, systems using GPUs can consume over 600 watts of power [17].

III. CASE STUDY: JOHN THE RIPPER

To demonstrate the power of our clusters, we concentrate on the application of password cracking. Despite their weaknesses, passwords remain the most widely used form of user authentication [18]. Passwords are usually stored on an authentication server in hashed form. When a user attempts to authenticate his or her credentials, the server simply hashes the user's inputted password and compares it to the stored hashed credentials on the server. Unsurprisingly, these hashed passwords serve as high-value targets for malicious parties.

Freely available password crackers such as John the Ripper (JtR) [19] and HashCat [20] enable hackers to crack password hashes and steal user authentication credentials. One of the most popular forms of attacks is the dictionary attack, in which a known list of passwords is used to create a rainbow table of hashed passwords. To crack a password hash, it suffices to simply check the rainbow table for the corresponding hash. Lists of common user passwords are freely available on the web, and are frequently used to seed dictionary attacks.

To protect against rainbow tables and dictionary attacks, system administrators typically "salt" user passwords. Salts are strings of random characters that are combined with the user password prior to hashing. In an ideal scenario, each hashed password will have its own unique salt. This technique forces an attacker to generate a separate rainbow table for each password, greatly reducing the speed at which passwords can be cracked. Choosing a good hash function is also important for ensuring password security. In the case of user authentication passwords, "good" hash functions are slow enough to impede cracking attempts, but fast enough to enable the user to authenticate in a timely manner.

For this case study, we perform dictionary attacks on common user passwords using John the Ripper (JtR) [19], a popular open-source password cracking suite. We choose JtR for its huge popularity, and its support of a variety of architectures and operating systems. JtR supports multi-threading, enabling it to leverage multiple cores. The passwords we attempt to crack are encrypted using the bcrypt [21] hashing algorithm. Designed in the late 90s, bcrypt is a slow hash function with a strong reputation. It is based on the Blowfish block cipher [22] designed by Bruce Schneier. Furthermore, Malvoni *et. al.* [23] recently ported the JtR application to run on the Epiphany co-processor, making it ideal for the Parallella SBC. To the best of our knowledge, no other password cracking suite currently supports the Epiphany architecture. We suspect this is due to the fact that Epiphany is still a relatively new architecture, and expect this to change with time.

A. Methods

For our experiments, we download a collection of the $10,000$ most common user passwords, compiled by Mark Burnett, and published in June 2011. The passwords were the most commonly used out of approximately 6.5 million users, and represents 99.8% of all user passwords [24]. The top $1,000$ passwords in the list are used by approximately 91% of all examined users in 2011 [24]. From the collection of $10,000$ common passwords, we randomly select $5,000$ passwords (with replacement) and hash them using the bcrypt Python library. Each password is salted with a random 128-bit salt of costs 5, 8 and 10. As a result, each file contains $5,000$ unique password hashes.

We run experiments on the Parallella and Raspberry Pi clusters, varying the number of cores from 16 to 128, in increments of 16. In addition, we compare our run-time performance to JtR $1.8.0$ running on a Linux partition of a 2015 MacBook Pro.

System	salt=5	salt=8	salt=10
MacBook Pro Laptop	689.0	88.63	21.98
Raspberry Pi 2	85.44	10.85	2.719
Parallella	1205	154.4	38.71

TABLE III
MEASURED CRACKS PER SECOND (C/S) VARYING SALT COST AND ARCHITECTURE (SINGLE-CORE COMPARISON).

This high-end system contains a quad-core Intel i7, 16 GB of RAM, and costs approximately $\$4,000.00$. We also measure the cracked passwords per second (c/s) outputted from JtR on these separate machines (see Table III). Both the Raspberry Pi cluster and the MacBook Pro laptop use JtR's `--fork` option to enable execution on all system cores. The JtR `pot` file was removed between successive executions.

The Parallella and Raspberry Pi clusters use the Message Passing Interface (MPI) [25] to use multiple SBCs in addition to multiple cores. Our master password file is split into N sub-files (where N is the number of SBCs), with each SBC node running JtR locally on all its cores to crack the password sub-file assigned to it. The systems spend the majority of their run-time cracking passwords; the NFS server's network communication overhead has been experimentally shown to be negligible. While this strategy of parallelization may not be appropriate for all scenarios, it is a good choice for our passwords, as each hash in our collection is unique.

B. Results

Table III shows the cracks per second (c/s) statistics obtained when running JtR on each of our different architectures. Please note that the c/s is for one node, and (in the case of the Raspberry Pi and laptop) one core. Unsurprisingly, the c/s decreases as the salt cost goes up. Higher salt costs yield slower hashes, reducing the speed at which a particular hash can be cracked. We note that our Parallella numbers are very consistent with the results published by Malvoni *et. al.* [23]. Across architectures, the Parallella clocks the highest c/s at 1205 on a salt cost of 5. In contrast, the laptop achieves a c/s of 689.0, while the Raspberry Pi averages a c/s of just 85.44 on the same file. This is unsurprising, given the Raspberry Pi's 900 MHz CPU. From these results, one may expect the Raspberry Pi cluster to perform the worst at cracking our files. Our next set of experiments show that this is not the case.

Figures 5 and 6 depict our experimental results. The x-axis is the total number of cores utilized, and ranges from 16 to 128. The raw running time (in seconds) is shown on the log-scaled y-axis. For a salt cost of 5, the MacBook Pro is able to crack $5,000$ passwords in approximately 4.44 hours. In contrast, a single 16-core Parallella requires 5.83 hours to crack the same file, and 4 Raspberry Pis (16 cores) require nearly 9.13 hours. As we increase the number of cores, the story quickly begins to change. Two Parallellas (32 cores) can crack the password file in 2.96 hours, a speedup of 40%. 32 Raspberry Pi cores (8 nodes) can crack the file in 4.63 hours, just 11 minutes slower than our MacBook Pro. When we use 128 Parallella cores, we can crack the file in 45 minutes, a speedup of 5.84 over the MacBook Pro. The Raspberry Pi cluster can crack the same file in 71.2 minutes using 128 cores, a speedup up 3.76 over the MacBook Pro.

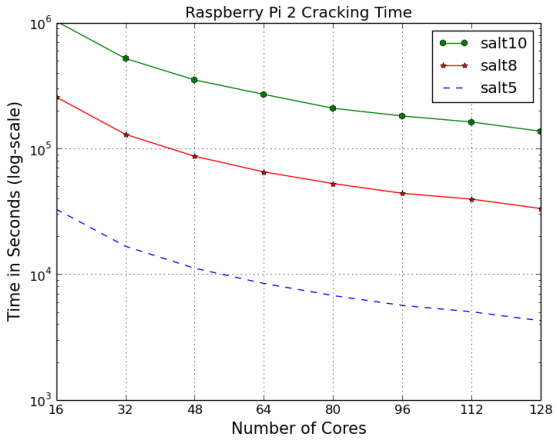

Fig. 5. Time required for Raspberry Pi 2 cluster to crack 5,000 passwords.

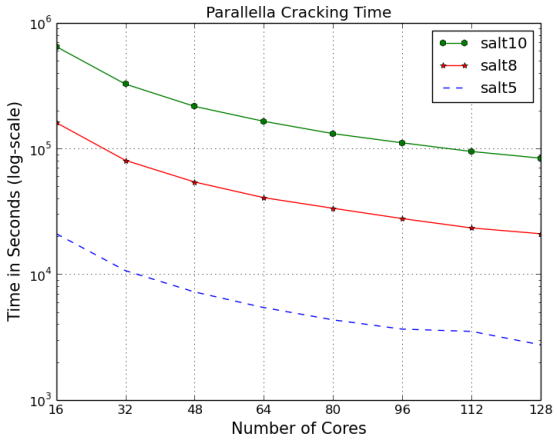

Fig. 6. Time required for Parallella cluster to crack 5,000 passwords.

The performance of our clusters become more pronounced on our password files with higher salt cost. The MacBook Pro was able to crack the $5,000$ password file with salt cost 8 in approximately 34.3 hours. A single Parallella and Raspberry Pi 2 takes significantly longer to crack the file, requiring 44.6 hours and 71.39 hours, respectively. However, two Parallellas (32 cores) can crack the file in 22.34 hours, a speedup of 1.53 over the MacBook Pro. Eight Raspberry Pis (32 cores) require 35.98 hours to crack the passwords, 1.67 times longer than the MacBook Pro. When using 128 cores, the Parallella cluster requires only 5.82 hours to crack the same file, a speedup of 5.89 over the MacBook Pro. 128 Raspberry Pi 2 cores require only 9.25 hours, a speedup of 3.71 over the MacBook Pro. On our $5,000$ password file encoded with salts with cost of 10, the MacBook Pro requires 5.77 days to crack the file. In contrast, our Parallella cluster requires only 23.25 hours when utilizing 128 cores, a speedup of 5.95. The Raspberry Pi 2 cluster can crack the same file in 1.58 days using 128 cores, a speedup of 3.63 over the MacBook Pro.

IV. Discussion

Our results clearly indicate that at least for password cracking, the combination of many small SBCs into a single cluster can outperform a high end laptop. The clusters are highly portable, and can be built for less than half of the laptop. The modular case for both SBCs was designed and open-sourced [26], [27] by Matthews and Blackmon in 2015. Any size cluster can be created from the case design, and the components can be cheaply 3-D printed. Since the units in the cluster are modular, the cluster can be taken apart and transported individually by soldiers in a platoon. We note that the most data is stored on an external USB hard drive; if data needs to be removed quickly, it is sufficient to remove the NFS drive. As an extra precaution, the microSD cards can easily be removed.

We foresee a variety of future capabilities enabled by the use of large volumes of SBCs working in parallel on complex computational tasks. From a military perspective, SBCs fit well within the concept of Cyber Support for Corps and Below [28], [29], but the application is not limited to military functions and can extend across a variety of needs cases including law-enforcement, intelligence, science, and importantly security. We propose the use of parallel SBCs for novel offensive and defensive cyberspace operation applications, including ingress into internet enabled devices, counter-RPA, distributed IDS/IPS, and ICS/SCADA system protection. We note that these applications are theoretical and are still in need of demonstration.

Lastly, we note that all of our applications could be implemented on a remote HPC system. However, when cyber war-fighters are operating a.) in areas unable to leverage remote HPC systems, b.) toward extremely time-sensitive objectives, and/or c.) in situations where proximate control and security of hardware is of importance, SBC clusters can be a real asset. We also note that SBC clusters consume less power than desktop systems with GPUs. In tactical situations overseas, power reliability and cooling requirements are major issues, especially in climates with high temperatures. We argue that the low cost and power consumption of SBCs enable them to be deployed down-range, enabling soldiers to run computationally-intensive tasks and get immediate results.

A. OCO Application: Ingress into Internet Enabled Devices

Internet enabled devices are a mainstay of modern society. CISCO predicts that more than 50 billion devices will be connected to the internet by 2020 [30]. Most of these devices are either unprotected or protected by a Personal Identification Number (PIN). In the aftermath of the 2015 San Bernardino attacks, the FBI sought to access the phone of one of the two perpetrators to examine its contents for potential links to other terrorist plots [31]. While the FBI was ultimately able to access the device [32], longer PINs can take a long time to brute force.

An SBC cluster can quickly and cheaply supply the necessary computational power for cracking PINs, which requires a much lower number of permutations than standard alphanumeric passwords. The high portability and low power requirements of SBC clusters allow them to be deployed on-site for cracking purposes, or deployed over a wireless network. While brute force protection schema such as those implemented by Apple to protect the iPhone [33] are possible on some devices, the rapidly growing ecosystem of Internet enabled devices suggest that large numbers of these devices will remain susceptible to brute force style attacks.

B. OCO Application: Counter RPA Strategies

The increased use of remotely piloted aircraft (RPAs), also referred to as drones, poses a significant challenge to military and civilian operations across a broad range of environments. The military and intelligence applications of drones are well documented and debated [34], [35]. Well known penetrations [36] leveraging vulnerabilities of drones in combat settings further challenge the current use of RPAs in conflict zones. As individuals and states increasingly develop and leverage drone technology, the military and civilian aviation industries must develop counter-RPA tools to ensure offensive and defensive tactical advantage. At present, most commercial drones operate without encrypted command and control features; yet this is unlikely to remain true indefinitely. However, the inclusion of encryption will likely be slow and expensive [36].

The Army Cyber Institute recently demonstrated the Cyber Rifle [37] (comprised of a directional (Yagi) antenna and a single Raspberry Pi) to exploit a known vulnerability within commercial Parrot Drones. This exploitation operates independently of the controller unit managing the drone's flight operations and sends a remote shut-down signal to the drone. The integration of tools such as Skygrabber [38] or broadcast antennas with a parallel machine offers opportunities for the rapid and efficient breaching of drones in real-time. For example, the "Gorgon Stare" [36] simultaneous video-feed capability advertised by the United States Air Force could theoretically be breached by multiple SBCs running Skygrabber in tandem.

A SBC cluster can aid in counter-RPA strategies by expediting the process of fuzz testing, or "fuzzing". Fuzz testing is a black-box software testing strategy in which random input is continuously fed into a software system until it crashes or enters a non-standard state. A SBC cluster, targeting parallel copies of a drone's operating system, can quickly help identify new RPA vulnerabilities. A more targeted strategy would be to test only a set of well-known RPA vulnerabilities against an unknown drone's operating system. Consider the situation where cyber war-fighters down-range encounter and capture an enemy drone of an unknown operating system. With a cluster of N SBC nodes, and dictionary of V known vulnerabilities, the soldiers can theoretically find common vulnerabilities in the enemy drone's operating system in V/N time. Another strategy would have subsets of SBC nodes target separate enemy RPA operating systems, enabling independent and simultaneous fuzzing. Both these strategies expedite the development of targeted anti-RPA weaponry down-range.

C. DCO Application: Distributed IDS/IPS

The use of intrusion detection/prevention systems (IDS/IPS) is a standard practice for DCO. The primary difference between the two systems is the ability to alert or drop anomalous packets for the IDS and IPS respectively. A potential shortfall of many of these systems is their inability to deal with high throughput networks. This is typically attributed to either the packet capture process or the CPU usage required for packet inspection. If the packet capture process is overwhelmed or the CPU usage is exhausted, packets are either dropped or not inspected, therefore defeating the purpose of the IDS/IPS. For example, Snort [39], a popular open-source IDS, is inherently serial in nature, supporting single-threaded processing.

SBCs are a potential solution for parallel packet inspection. Snort's multi-instance feature [39] may be adapted for use in a cluster, but there may be better options. Suricata [40] and Bro [41] are two very popular IDS open-source projects that are a more ideal fit for a SBC cluster. Suricata has native multi-threading support [40]. Previous research shows that a 624-core system running Suricata dropped only 7% of packets with a 20 Gbps throughput, compared to a 53% drop rate with Snort [42]. While Bro does not have built-in multi-threading, it can support a distributed architecture [41], recommending a node for every 80 Mbps of traffic [43]. Prior research [44] shows that a cluster of Bro nodes can achieve significant performance gains while minimizing the potential latency as a result of computationally intensive analysis. These systems can easily be implemented on SBC clusters. While commercial parallel IDS/IPS systems exist, a SBC cluster is cheaper and more portable than a standard COTS cluster, and enables cyber developers to create application specific IPS/IDS tools for use in tactical situations.

D. DCO Application: Monitoring ICS/SCADA systems for alarm events

Protecting Supervisory Control and Data Acquisition (SCADA) systems and Industrial Control Systems (ICS) at large is a top priority for the United States [45]. SCADA systems include national energy infrastructure such as electric, nuclear, and natural gas. In a SCADA system, a computer acting in a supervisory role is responsible for gathering sensor data from remote terminal units (RTUs). This sensor data includes metered readings of various aspects of the system. For example, a power grid's collected data can include timestamped current and voltage readings. For the security of SCADA systems, rapid anomaly (or "alarm event") detection is of critical importance. The fast detection of an alarm event enables operators to be quickly alerted to potential problems, curtailing potential attacks. United States SCADA infrastructure are increasingly vulnerable; 50,000 attacks against the Nation's SCADA infrastructure were detected in 2014 alone [46].

SBC clusters can help facilitate the rapid detection of alarm events. Their low cost, size, and power consumption enables easy integration into existing SCADA systems. The supervisory system (communicating to the head SBC node through an approved IP address) will send digitized sensor data to the SBC cluster in set time intervals. Let N be the number of nodes in the SBC cluster and M be the number of data measurements. The head SBC node splits the data into N subsets, and has each node scans its assigned subset of data for anomalies. If anomalies are found, the nodes communicate the information back to the head node, which in turn triggers an alarm event in the supervisory system. Thus, an SBC cluster with N nodes can theoretically reduce amount of time required to detect anomalies to M/N time, greatly increasing operator responsiveness to an external attack.

V. Conclusions

Single Board Computers (SBCs) are a relatively new technology that are rapidly improving in cost, performance, and energy consumption. While a standalone SBC has relatively weaker performance than a standard laptop computer, a cluster of SBCs can outperform laptops at several tasks, while maintaining high portability and low cost. To demonstrate this claim, we built two 128-core SBC clusters that cost less than $\$2,000.00$. Using the John the Ripper (JtR) password cracking, we use the two clusters to crack $5,000$ common passwords encoded with bcrypt. We compare the cracking speed of our clusters to a high-end, $\$4,000.00$ laptop. Despite costing a fraction of the laptop, our clusters crack passwords up to 5.95 times faster than the laptop computer, while maintaining relatively low power consumption and high portability. We also discuss how SBC clusters could be used in other cyber-related applications, such as counter-RPA, distributed IDS/IPS and defense of ICS/SCADA systems.

We strongly believe that SBCs and SBC clusters will play a critical part in future offensive and defensive cyberspace operations. SBC clusters are low-cost, low-power, high-performance, and extremely modular and can be scaled up and down for a variety of use cases. In particular, it should be noted SBC clusters are of value in environments where it is either infeasible or extremely costly to use remote HPC architectures, especially due to limits in data usage, network bandwidth, or risk to data security.

Given our experimental results, we believe the Parallella SBC merits further exploration. In addition, the Raspberry Pi 3 [47] was recently released. Other recently released SBCs that are valuable for study include the 8-core Odroid XU4 [48] and the NVidia Jetson TX1 [49], which has 256 Cuda cores. Future work will concentrate on exploring additional SBCs, extending our JtR case study to include more trials and salts, and building proof-of-concept SBC systems for the applications discussed in this paper. Other areas of future research include benchmarking the performance of SMT solvers and popular fuzzers such as American Fuzzy Lop (AFL) [50] on our SBC clusters, and developing materials to educate soldiers about parallel processing using SBCs.

Acknowledgments

We are extremely grateful to Benjamin Klimkowski for his feedback on an earlier draft of this paper. Special thanks to Mr. Robert McKay of the Electronic Support Group (ESG) in the Department of Electrical Engineering & Computer

Science for building our clusters custom power supplies, and for his assistance in troubleshooting power issues related to the cluster. Special thanks also to Mr. Bradley Dick and Mr. Michael Soffos of ESG for facilitating the 3-D printing of our Raspberry Pi cluster's cases. The opinions expressed in this work are solely of the authors, and do not necessarily reflect the U.S. Military Academy, the U.S. Army, or the Department of Defense.

REFERENCES

[1] V. Vujović and M. Maksimović, "Raspberry pi as a wireless sensor node: performances and constraints," in *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on*. IEEE, 2014, pp. 1013–1018.

[2] Raspberry Pi, "Robot archives," Internet Website, Last accessed, 5 2016, https://www.raspberrypi.org/blog/tag/robots/.

[3] A. Baker, "Quadcopter," *MagPi*, vol. 19, pp. 4–7, 2013.

[4] S. J. Cox, J. T. Cox, R. P. Boardman, S. J. Johnston, M. Scott, and N. S. Obrien, "Iridis-pi: a low-cost, compact demonstration cluster," *Cluster Computing*, vol. 17, no. 2, pp. 349–358, 2014.

[5] F. P. Tso, D. R. White, S. Jouet, J. Singer, and D. P. Pezaros, "The glasgow raspberry pi cloud: A scale model for cloud computing infrastructures," in *2013 IEEE 33rd International Conference on Distributed Computing Systems Workshops*. IEEE, 2013, pp. 108–112.

[6] J. Kiepert, "Creating a raspberry pi-based beowulf cluster," Boise State University, Idaho, Tech. Rep., 2013, 1-17.

[7] P. Abrahamsson, S. Helmer, N. Phaphoom, L. Nicolodi, N. Preda, L. Miori, M. Angriman, J. Rikkilä, X. Wang, K. Hamily *et al.*, "Affordable and energy-efficient cloud computing clusters: the bolzano raspberry pi cloud cluster experiment," in *Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on*, vol. 2. IEEE, 2013, pp. 170–175.

[8] J. Muniz and A. Lakhani, *Penetration Testing with Raspberry Pi*. Packt Publishing Ltd, 2015.

[9] J. Muniz, "Raspberry pi as a hacking arsenal," *The Security Blogger*, 10 2014, www.thesecurityblogger.com/raspberry-pi-as-a-hacking-arsenal/.

[10] K. Wessel, "Developing the rogue pi," *Crush Beer Code*, 3 2013, http://crushbeercrushcode.org/2013/03/developing-the-rogue-pi/.

[11] J. Turla, "Glastopf pi: A simple yet cool honeypot for your raspberry pi," *INFOSEC Institute*, 6 2013, http://resources.infosecinstitute.com/glastopf-pi-a-simple-yet-cool-web-honeypot-for-your-raspberry-pi/.

[12] J. A. Ross, D. A. Richie, and P. S. J., "Implementing image processing algorithms for the epiphany many-core coprocessor with threaded mpi," in *Proceedings of the 2015 IEEE High Performance Extreme Computing Conference (HPECC15)*. IEEE, 2015.

[13] R. Jurevičius and V. Marcinkevičius, "Energy efficient platform for sobel filter implementation in energy and size constrained systems," in *Information, Electronic and Electrical Engineering (AIEEE), 2015 IEEE 3rd Workshop on Advances in*. IEEE, 2015, pp. 1–5.

[14] A. Varghese, B. Edwards, G. Mitra, and A. P. Rendell, "Programming the adapteva epiphany 64-core network-on-chip coprocessor," *International Journal of High Performance Computing Applications*, p. 1094342015599238, 2015.

[15] A. Papadogiannakis, S. N. Agathos, and V. V. Dimakopoulos, "Openmp 4.0 device support in the ompi compiler," in *International Workshop on OpenMP*. Springer, 2015, pp. 202–216.

[16] D. Richie, J. Ross, S. Park, and D. Shires, "Threaded mpi programming model for the epiphany risc array processor," *Journal of Computational Science*, vol. 9, pp. 94–100, 2015.

[17] T. Kreiss, "How much power does your graphics card need?" *Tom's Hardware: The Authority on Tech*, 1 2009, http://www.tomshardware.com/reviews/geforce-radeon-power,2122.html.

[18] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, "Passwords and the evolution of imperfect authentication," *Communications of the ACM*, vol. 58, no. 7, pp. 78–87, 2015.

[19] A. Peslyak, "John the ripper," Internet Website, Last accessed, 5 2016, http://www.openwall.com/john.

[20] J. Steube, "Hashcat advanced password recovery," Internet Website, Last accessed, 5 2016, https://github.com/hashcat/hashcat.

[21] N. Provos and D. Mazieres, "A future-adaptable password scheme." in *USENIX Annual Technical Conference, FREENIX Track*, 1999, pp. 81–91.

[22] B. Schneier, "Description of a new variable-length key, 64-bit block cipher (blowfish)," in *International Workshop on Fast Software Encryption*. Springer, 1993, pp. 191–204.

[23] K. Malvoni and J. Knezovic, "Are your passwords safe: Energy-efficient bcrypt cracking with low-cost parallel hardware," in *8th USENIX Workshop on Offensive Technologies (WOOT 14)*, 2014.

[24] B. Sterling, "Web semantics: The ten thousand worst passwords," *Wired Magazine*, 12 2013, https://www.wired.com/2013/12/web-semantics-the-ten-thousand-worst-passwords.

[25] W. Gropp, E. Lusk, N. Doss, and A. Skjellum, "A high-performance, portable implementation of the mpi message passing interface standard," *Parallel computing*, vol. 22, no. 6, pp. 789–828, 1996.

[26] S. J. Matthews and W. Blackmon, "Parallella case and cluster files," Internet Website, 6 2015, http://www.thingiverse.com/thing:892684.

[27] ——, "Raspberry pi case and cluster files," Internet Website, 6 2015, http://www.thingiverse.com/thing:892959.

[28] A. Brantly, "Strategic cyber maneuver," Internet Website, 10 2015, http://smallwarsjournal.com/jrnl/art/strategic-cyber-maneuver.

[29] E. Waage, "Phreaker, maker, hacker, ranger: One vision for cyber support to corps and below in 2025," *Small Wars Journal*, 8 2015, http://smallwarsjournal.com/jrnl/art/phreaker-maker-hacker-ranger-one-vision-for-cyber-support-to-corps-and-below-in-2025.

[30] D. Evans, "The internet of things: How the next evolution of the internet is changing everything," Cisco Systems White Paper, Tech. Rep., 4 2011.

[31] "United states v. apple," Internet Website, last accessed, 5 2016, https://assets.documentcloud.org/documents/2714001/SB-Shooter-Order-Compelling-Apple-Asst-iPhone.pdf.

[32] M. Zapotosky, "FBI has accessed san bernardino shooter's phone without Apple's help," *Washington Post*, 3 2016.

[33] "iOS security: iOS 9.3 or later," Internet Website, 5 2016, https://www.apple.com/business/docs/iOS_Security_Guide.pdf.

[34] A. Plaw, M. S. Fricker, and C. Colon, *The Drone Debate: A Primer on the US Use of Unmanned Aircraft Outside Conventional Battlefields*. Rowman & Littlefield, 2015.

[35] C. Bolkcom, "Homeland security: Unmanned aerial vehicles and border surveillance." DTIC Document, 2004.

[36] S. Gorman, Y. J. Dreazen, and A. Cole, "Insurgents hack u.s. drones," *Wall Street Journal*, 12 2009, http://www.wsj.com/articles/SB126102247889095011.

[37] J. Clark, "This cyber rifle is dirt cheap and easy to make. happy drone hunting," *Task and Purpose*, 3 2016, http://taskandpurpose.com/cyber-rifle-dirt-cheap-easy-make-happy-drone-hunting/.

[38] SkySoftware, "Skygrabber," Internet Website, 3 2010, http://www.skygrabber.com/en/index.php.

[39] "Snort 2.9.8.2 user manual," Internet Website, 2015, http://manual-snort-org.s3-website-us-east-1.amazonaws.com.

[40] "Suricata all features," Internet Website, June 2011, http://suricata-ids.org/features/all-features/.

[41] "Bro 2.3.1 documentation," Internet Website, last accessed, June 2016, https://www.bro.org/sphinx/intro/index.html.

[42] E. Albin, "A comparative analysis of the snort and suricata intrusion-detection systems," Ph.D. dissertation, Monterey, California. Naval Postgraduate School, 2011.

[43] "Bro cluster architecture," Internet Website, last accessed, June 2016, https://www.bro.org/sphinx/cluster/index.html.

[44] N. Weaver and R. Sommer, "Stress testing cluster bro." in *DETER*, 2007.

[45] Department of Defense, "The DOD cyber strategy," Internet Website, April 2015, http://www.defense.gov/News/Special-Reports/0415_Cyber-Strategy.

[46] Dell Computer Inc., "2015 Dell security annual threat report," Internet Website, 2015, https://software.dell.com/docs/2015-dell-security-annual-threat-report-white-paper-15657.pdf.

[47] Raspberry Pi Foundation, "Raspberry Pi 3 model B," Internet Website, 2016, https://www.raspberrypi.org/products/raspberry-pi-3-model-b/.

[48] Odroid, "Odroid xu4," Internet Website, 2016, http://www.hardkernel.com/main/products/prdt_info.php?g_code=G143452239825&tab_idx=2.

[49] Nvidia, "Nvidia Jetson TX1: the embedded platform for autonomous everything," Internet Website, 2016, http://www.nvidia.com/object/jetson-tx1-module.html.

[50] M. Zalewski, "American fuzzy lop - coredump.cx," Internet Website, last accessed, 8 2016, http://lcamtuf.coredump.cx/afl/.

Suzanne J. Matthews is an Assistant Professor of Computer Science in the Department of Electrical Engineering & Computer Science, a Research Fellow of the Network Science Center, and an Affiliate of the Cyber Research Center at the United States Military Academy, West Point. She received her B.S. and M.S. degrees in Computer Science from Rensselaer Polytechnic Institute, and her Ph.D. in Computer Science from Texas A&M University. Her research interests include parallel computing, data mining, and computational biology.

Raymond W. Blaine was commissioned a Signal Officer before becoming a Cyber Officer in 2015. His assignments include a variety of duty positions at Fort Bragg, N. C. He also has served two tours in OIF and one tour in OEF, as a Platoon Leader, Aide-de-Camp to the Chief of Staff MNC-I, and as S6 for 2-508 PIR respectively. He served as an Assistant Professor at USMA from 2012-2015. MAJ Blaine is currently a Team Leader in the 1st Battalion of the U.S. Army Cyber Protection Brigade.

Aaron F. Brantly is Assistant Professor of International Relations and Cyber in the Department of Social Sciences, Cyber Policy Fellow at the Army Cyber Institute and Cyber Fellow at the Combating Terrorism Center. He holds a Ph.D. in Political Science from the University of Georgia and a Master's of Public Policy from American University. His research focuses on national security policy issues in cyberspace including big data, terrorism, intelligence, decision-making and human rights.