Harnessing Single Board Computers for Military Data Analytics

Suzanne J. Matthews, Ph.D.

Department of Electrical Engineering & Computer Science

United States Military Academy, West Point NY 10996

Executive summary: *This chapter covers the use of Single Board Computers (SBCs) to expedite on-site data analytics for a variety of military applications. On-site data summarization and analytics is increasingly critical for command, control and intelligence (C^2I) operations, as excessive power consumption and communication latency can restrict the efficacy of down-range operations. SBCs offer power-efficient, inexpensive data processing capabilities while maintaining a small form factor. We discuss the use of SBCs in a variety of domains, including wireless sensor networks, unmanned vehicles, and cluster computing. We conclude with a discussion of existing challenges and opportunities for future use.*

Introduction

Recent advances in computer architecture and processor design have given rise to *single board computers* (or SBCs), where the entirety of the computer is printed on a single circuit board. As the ecosystem of SBCs continues to evolve in compute capabilities and efficiency, the devices become increasingly attractive for military applications, especially for localized data pre-processing and analysis.

Typically, such analysis is conducted by systems such as laptops, portable desktop computers, or via satellite communication with remote high performance computing (HPC) clusters. In a battlefield environment, the power consumption and cooling requirements of larger systems can be imposing, especially in harsh climates. Dust, high temperatures, and fluctuations in power are all confounding elements. Furthermore, the latency and security requirements of satellite communications can delay the analysis of the information needed for effective command, control and intelligence (C^2I).

Single board computers offer several advantages in the military domain. First, their small form factor and relative inexpensiveness enable high versatility. Second, flash storage enables fast access to data without the latency or power consumption of spinning disk storage (Cox, et al., 2013). Lastly their System-on-a-Chip (SoC) processors enable data storage capacities and processing capabilities that far outstrip microcontrollers and Field Programmable Gate Arrays (FPGAs). SBCs are easily reprogrammable, and have a wide array of ports that enable them to be used as standalone computers or mounted on other devices for a variety of applications.

We predict SBCs will play a vital role in future military operations. The expected growth of battlefield data requires new technology that can efficiently summarize data collected from sensors and other devices into a format that is readable for human operators involved in C^2I activities. We predict the power-efficient, yet inexpensive SBC will play a critical role in future missions as a “middle man”, operating in a hierarchy of devices that work in tandem with microcontrollers and sensors to locally analyze data, consequently expediting the time required for C^2I capabilities.

In this chapter, we present examples of efforts that utilize SBCs for a variety of data analytics activities that have direct application to the military, and discuss challenges and opportunities for the future use of SBCs. This chapter is not meant to be an exhaustive survey. Instead, we seek to highlight certain key classes of applications for SBCs in the military domain. Notably, we do not cover the use of SBCs for breaking into networks or covertly collecting information (packet sniffing, port scanning, skimming, spoofing, siphoning data, etc.). Instead, our goal is to discuss applications in which SBCs can expedite the analysis of information on-site.

Single Board Computers, FPGAs, and Microcontrollers

Before we continue, it is necessary to discuss how SBCs differ from other categories of low-energy chipsets such as microcontrollers and FPGAs. A *microcontroller* contains only a subset of the functionality of an SBC and is designed to run a single program upon booting. Microcontrollers do not have operating systems and are extremely resource constrained. For example, the Arduino 101 has 24 KB of memory and 196 KB of Flash storage. The Arduino costs approximately $30.00, and has a processor speed of 32 MHz. In contrast, modern SBCs support up to 8 GB of memory, and support high-capacity microSD flash storage.

Field Programmable Gate Arrays (FPGAs) offer a greater level of flexibility than microcontrollers through their circuit reprogrammability, but require knowledge of a hardware descriptor language such as VHDL. Programming FPGAs also requires a significant learning curve, even for experienced programmers. In contrast, the Arduino's programming language is very C-like, and is more accessible for novice programmers. SBCs feature the greatest level of language flexibility of all, with many SBCs programmable in C/C++, Python, and other common programming languages.

Unlike microcontrollers and FPGAs, SBCs are fully functioning computers. They feature operating systems, Random Access Memory (RAM), and power-efficient System-on-a-Chip (SoC) processors that are commonly found in smartphones. While microcontrollers and FPGAs can be used in some contexts, we argue that their diminished on-board resources limit their usefulness for many data processing applications required for C^2I.

While smartphones and other mobile devices have similar processors to SBCs, their limited number of ports and lack of general purpose input/output (GPIO) interfaces makes them difficult to integrate into systems that combine hardware and software, such as unmanned vehicles. Common SBCs typically have a wide array of ports. For example, the Raspberry Pi 3 has 4 USB ports, an Ethernet port, a full-sized HDMI port, a microUSB power connector and a 4-pole A/V header. In contrast, most smartphones only have a single port for charging.

The rest of the chapter discusses common SBCs and how researchers have begun to explore their use for data summarization and analysis applications relevant to the military.

Name	**Price**	**Processor Type (Number of Cores)**	**Memory**	**Dimensions (inches)**	**Power (W)**	**Weight (g)**
Raspberry Pi Zero	$5.00	ARM 11 (1 core)	512 MB	1.18 x 2.56	0.7	9
Raspberry Pi 3	$35.00	ARM A53 (4 cores)	1 GB	3.4 x 2.2	5	42
Adapteva Parallella	$99.00	ARM A9 (2 cores) 16 Epiphany cores	1 GB	2.1 x 3.5	5	42.5
NVidia Jetson TK1	$192.00	ARM A15 (4 cores) 192 Cuda cores	2 GB	5 x 5	58	120

Table 1: Popular Single Board Computers.

A Snapshot of the Current Ecosystem of Single Board Computers

Table 1 contains some popular SBCs, ordered according to price. Most of the mentioned SBCs were initially released in the last five years. We stress that the listed SBCs represent just a fraction of the ecosystem of available devices. The cheapest of the listed SBCs is just $5.00. The most expensive is $192.00.

Perhaps the most popular SBC in use today is the Raspberry Pi, a credit-card sized computer retailing at $35.00. Initially released in 2012 with a 700 MHz processor and 256 MB of RAM, the Raspberry Pi has enjoyed annual memory and CPU upgrades, while maintaining its $35.00 price-point and form factor. Released in 2016, the Raspberry Pi 3 has 1.2 GHz quad-core

ARM processor, 1 GB of RAM, integrated wireless and Bluetooth, and removable microSD flash storage with capacities that can exceed 64 GB.

In January 2014, Intel announced the Intel Edison, a *computer on module* (COM), which is a subtype of SBC. Unlike a regular SBC, a COM must be mounted on a base-board in order to make use of its I/O interface. In November 2015, the Raspberry Pi Foundation released the Raspberry Pi Zero, a $5.00 SBC designed to compete with the Intel Edison in the *Internet of Things* (IoT) and wearables market. In February 2017, the $10.00 Raspberry Pi Zero W was released. It is a variant of the Raspberry Pi Zero, with integrated wireless and Bluetooth. Intel discontinued the Edison project in early 2017 (Intel 2017).

The low cost and energy consumption of the SBCs mentioned above are largely due to their relatively “weak” CPUs. In addition to an ARM CPU, the Parallella and NVidia Jetson SBCs each feature additional chipsets capable of handling more compute-intensive operations. Released to the general public in 2014, the Parallella is a credit-card sized SBC that boasts a 16-core Epiphany co-processor, while maintaining a similar power profile to the Raspberry Pi 3. NVidia released the Jetson TK1 in May 2014, which contains a 192-core Tegra K1 GPU.

Application 1: Expediting Information Flow in Wireless Sensor Networks

A *wireless sensor network* (WSN) consists of a collection of sensors that gather information about the surrounding environment, and work in tandem to communicate that information back to one or more central resources. Prior surveys (Winkler, Tuchs, Hughes, & Barclay, 2008) (Đurišić, Tafa, Dimić, & Milutinović, 2012) have extensively discussed the military applications of WSNs. In the battlefield arena, WSNs are often employed for force protection (e.g. perimeter security/infiltration detection) and monitoring militant activity. For example, a large number of wireless sensors dropped in enemy territory can form a wireless sensor network that can indicate enemy presence in remote areas (Malladi & Agrawal, 2002). This increases C^2I capabilities while minimizing risk to ground troops.

Single board computers can expedite the information flow in a wireless sensor network. A natural location for SBCs in a wireless sensor network is at the so-called “gateway” nodes, which act as a central point for information gathering and processing. SBCs can also be used in tandem with microcontrollers to create various workflows. (Bell, 2013) dedicates an entire book discussing how the Raspberry Pi SBC can be used in conjunction with Arduino microcontrollers for wireless sensor network applications. For example, wireless sensors can connect to an Arduino, which can in turn store data on a Raspberry Pi running a MySQL database.

Due to their larger data capacities and more powerful processors, SBCs also enable a greater level of at-node data pre-processing, reducing the amount of raw data that must be transferred across the wireless sensor network. (Winkler, Tuchs, Hughes, & Barclay, 2008) note that at-node data pre-processing is critical for energy efficient wireless sensor networks, as the energy required to transfer data often exceeds the amount required to process the data at source.

(Vujović & Maksimović, 2014) explore the feasability of the Raspberry Pi as a sensor node in a wireless sensor network. In their paper, Vujović *et al.* compare the Raspberry Pi to five commerical wireless sensors, and gathered benchmarks on form factor, power consumption, cost, and memory. While the Raspberry Pi is physically larger than the surveyed commerical wireless sensor nodes, it costs approximately 3 to 8 times less, and has 4,000 times the memory. However, the authors note that the Raspberry Pi consumes more power than the commercial wireless sensor nodes, are difficult to power via battery, and lack Bluetooth or integrated wireless capability. That said, it is important to note that this paper was published prior to the release of the Raspberry Pi 3 and Raspberry Pi Zero W, both of which have integrated wireless and Bluetooth. We also note that the Intel Edison (with the Arduino breakout board) has integrated WIFI, Bluetooth, and Ethernet connectivity.

We note the power consumption of both the Intel Edison and Raspberry Pi Zero is around 1 Watt. While microcontrollers are more power-efficient, the additional processing power available at-node may reduce the overall power consumption and latency required to communicate data across the network. PorcupineLabs' PiSense (Porcupine Labs, 2015) is a recent mobile WSN effort that uses Raspberry Pis to collect sensor data. Students at Stanford (Hong, Raymond, & Shackelford, 2014) also developed EdiSense, a Edison-based WSN data store designed to help prevent data loss in Bluetooth Low Energy (BLE) networks.

Application 2: On-Board Data Analysis and Summarization for Unmanned Vehicles

An *unmanned vehicle* is a machine that moves through and responds to its environment in an unsupervised or semi-supervised manner. Examples include unmanned aerial vehicles (UAVs) and autonomous ground vehicles (AGVs). The military applications of such systems are extensive; UAVs and AGVs enable tactical insight into areas that are dangerous to military personnel, facilitating surveillance and scouting missions and the transport of goods or payload without risk to operator. UAVs and AGVs are commonly used for applications such as terrain mapping, transport (Yamauchi, 2004), and surveillance (Samad, Bay, & Godbole, 2007).

To enable high mobility, the on-board computer used to gather and summarize sensor data must necessarily be lightweight and power-efficient. At the same time, non-trivial computational power is needed to process acquired sensor and image data, which are fed continuously to the on-board computer via mounted cameras. When controlled by a remote operator, the captured data is streamed to a remote location via satellite. We note that SBCs with a single ARM SoC may lack the processing power for such applications. However, SBCs with

Graphics Processing Units (GPUs) can accomplish the task while maintaining the small form factor and power-efficiency needed for integration with UAVs and AGVs.

The Raspberry Pi has been used extensively in hobbyist drone projects. For example, Hardware Attached on Top (HAT) such as the Navio 2 (Emlid, 2016) can be used in conjunction with the Raspberry Pi and open-sourced UAV platforms such as ArduPilot Mega (APM) (ArduPilot Mega Project) or PixHawk (Meier, Honegger, & Pollefeys, 2015) to create drones that are capable of operating in various flight modes and transferring video to a remote operator or network. Traditionally, compute-intensive image processing applications such as feature extraction, video summarizing, object detection, and tracking are delegated to a remote compute system with greater processing power.

Researchers have recently begun to explore the extent to which SBCs can perform on-board image processing and summarization. (Rebouças, Eller, Habermann, & Hideiti Shiguemori, 2013) explored how well a Raspberry Pi can analyze captured UAV flight image data. (Choi, Geeves, Alsalam, & Gonzalez, 2016) used an on-board Raspberry Pi and PixHawk to capture the location of a stationary target. (Da Silva, Brito, & Nogueira de Moura, 2015) studied the suitability of the Raspberry Pi and Intel Edison for on-board aerial image processing to the DE2i-150 FPGA development kit. While the DE2i-150 was over 700 ms faster on average, its excess weight (800g) and higher energy consumption caused the researchers to conclude that the Pi and Edison were better choices for UAV design. In a separate effort, (Vega, et al., 2015) compared the efficacy of the Raspberry Pi 2 and the quad-core ARM processor of the NVidia Jetson as an on-board computer for summarizing video data prior to transfer to a remote UAV operator. In this application, the goal is for the on-board computer's frame rate to be equivalent to the observed frame rate by the UAV operator. The researchers demonstrated that by switching to the NVidia Jetson from a Raspberry Pi, they were able to achieve acceptable frame rate results.

A key strength of the NVidia Jetson TK1 is the presence of the on-board Tegra GPU, which has 192 Cuda cores. This makes the board a prime candidate for more intensive on-board image processing applications for autonomous vehicles. (Meng, Wang, & Leong, 2015) used the NVidia Jetson as an on-board computer for quadcopters deploying their SkyStitch software. For this application, the Cuda cores were used for feature extraction and outlier removal of captured image data. The researchers note that delegating feature extraction to the on-board computer enabled SkyStitch to perform very efficiently. (Kachris, Stamelos, & Soudris, 2016) discuss how video encoding applications can be conducted with a high energy efficiency using NVidia Jetson TK1 and a "race to sleep" approach for controlling CPU utilization.

The NVidia Jetson has also been used for computer vision applications for autonomous vehicles. Two recent efforts discuss how the NVidia Jetson can be used for real-time lane detection applications. (Lee & Kim, 2016) tested a custom lane-detection approach on the NVidia Jetson TK1. When benchmarking their results with a CalTech public dataset, they were able to achieve up to 96% accuracy at a frame rate of 44 FPS. (Kim, Beak, & Park, 2016) demonstrated how Hough space image transformation can be used for lane detection using the Nvidia Jetson TK1. Sense and Avoidance algorithms are also a critical part of collision prevention and enemy avoidance in autonomous vehicles. (Zsedrovits, et al., 2016) and (Zsedrovits, et al., 2015) implemented sense and avoidance algorithms on the NVidia Jeston, which yielded targeted results.

Application 3: Portable and Localized Cluster Computing for Battlefield Environments

Cluster computing involves a collection of computers networked together to accomplish a common goal. The classic example is the Beowulf cluster, where the individual computers are composed of identical units of low-cost commodity hardware, commonly referred to as commercial off-the-shelf or “COTS”. The assembled clusters are loaded with libraries and programs that enable communication and data sharing between nodes. Modern day clusters are used to create military data centers and HPC research facilities.

While soldiers deployed down-range do not usually need access to HPC systems, compute-intensive applications can be offloaded to a remote HPC center via satellite. Issues with power, security, latency and wireless network bandwidth can prevent soldiers from maintaining a connection to a remote system or transfer large amounts of data for analysis, especially in a battlefield environment. Due to their high cost, power consumption, and cooling requirements, it is infeasible to deploy traditional HPC systems down-range.

A collection of SBCs can be configured into a Beowulf cluster to create a portable, power-efficient compute cluster that can be deployed in a battlefield arena. We stress that SBCs clusters are not designed to compete with traditional HPC systems in terms of raw computing power. However, they can serve as an alternative to multi-core desktop computers down-range. The ARM architecture is also expected to play a heavy role in the design of future HPC architectures. Software designed for ARM-based SBC clusters can arguably be ported to future ARM HPC systems as they become available.

A key advantage of using a cluster of SBCs in a battlefield environment is that it removes single points of failure (Cox, et al., 2013), which can be disastrous for military applications. Furthermore, the use of an SBC cluster can expedite the processing and analysis of data on-site, enabling the resulting analysis to be accessed immediately. Furthermore, any amount of local data processing and analysis reduces the latency of transfer over satellite communication channels, expediting the speed at which intelligence can be communicated to command centers.

Several researchers have explored the feasibility of SBC clusters for high-throughput data processing applications, specifically in the context of Hadoop MapReduce and Apache Spark. IridisPi (Cox, et al., 2013) studied the efficacy of Hadoop's filesystem on a 64-node Raspberry Pi cluster. They note that while their cluster is capable of computing map and reduce tasks, the Hadoop I/O results are very slow. (Anwar, Krish, & Butt, 2014) studied the performance of various ARM architectures for various Hadoop applications, and noted that benchmarking on the 10-node Raspberry Pi cluster was orders of magnitude slower than other architectures. (Kaewkasi & Srisuruk, 2014) built an Apache Spark cluster out of 22 CubieBoard SBC nodes, which had more powerful ARM Cortex A8 nodes (compared to the older ARMv6 technology implemented in the original Raspbery Pi). Despite the improved CPU, the researchers still concluded that Hadoop I/O issues reduced the efficacy of using SBC clusters for this purpose. More recently, (Kachris, Stamelos, & Soudris, 2016) studied the power efficiency of Spark applications on various ARM-based architectures, including the Raspberry Pi 3. While they did not build an SBC cluster, they noted that the power efficiency of the ARM architectures makes them attractive for future exploration with Apache Spark.

SBC clusters have shown considerable promise for compute-intensive applications. The goal for such applications is to distribute the total computations over multiple cores and nodes, in an effort to reduce application execution time. Distributed computation is enabled through the Message Passing Interface (MPI), an industry standard that is widely used on HPC systems.

Several researchers have used SBC clusters in conjunction with MPI to speed up various computations. The 64-node cluster designed by (Cox, et al., 2013) used MPI for compute-intensive jobs. (Kiepert, 2013) built a 32-node Raspberry Pi cluster and benchmarked the performance of a Monte Carlo estimation of Pi (PMCPI) over multiple nodes, compared to a 64-bit Intel Xeon processor. While execution on a single Raspberry Pi node was significantly slower than the Intel processor, PMCPI's run time on 32 nodes was roughly equivalent to its single-thread execution on the Intel Xeon. (Matthews, Blaine, & Brantly, 2016) compared the performance of two SBC clusters (Raspberry Pi 2 and Parallella respectively) against a high-end laptop on the application of password cracking using John the Ripper and MPI. The authors also theorized how the clusters could be applied to other applications in the cyber domain, such as counter-RPA and intrusion detection.

SBC clusters can also be used to run or simulate web servers. (Varghese, Carlsson, Jourjon, Mahanti, & Shenoy, 2014) proposed Raspberry Pi webservers as a green alternative to traditional enterprise solutions at small to medium-sized institutions. The authors found that their Raspberry Pi cluster was able to serve on average 17 to 23 times more requests per Watt than traditional servers. For static web content, their cluster was able to service up to 200 requests per second, which is sufficient for small websites. The authors note that for dynamic content, the cluster was suboptimal, achieving only up to 20 requests per second. Overall however, the authors argue that an SBC cluster is a green alternative to traditional webservers, especially for smaller institutions.

(Tso, White, Jouet, Singer, & Pezaros, 2013) created a 56-node Raspberry Pi "mini cloud" data center test-bed with the use of Linux containers. (Abrahamsson, et al., 2013) created a 300-node Raspberry Pi cluster test-bed for experimenting with cloud services. In a more recent study, (Pahl, Helmer, Miori, Sanin, & Lee, 2016) explored the use of SBC clusters in conjunction with containers to serve as "edge-clouds" between IoT devices and larger data centers. Similar to the use of single SBCs in wireless sensor networks, the authors propose that

SBC clusters can be used for intermediate data processing for data gathered from IoT devices, reducing the amount of data required to be transferred to a larger data center. (Spillner, Beck, Schill, & Bohnert, 2015) implemented a “stealth database” on a cluster of 8 Raspberry Pi nodes. (Djanali, Arunanto, Pratomo, Studiawan, & Nugraha, 2014) explored the use of a 10-node Raspberry Pi cluster as a honeypot server for SQL Injection attacks.

Challenges and Opportunities for SBCs in Future Military Operations

The analysis of large amounts of data will play a critical role in future warfare operations. In their article *CyberWar is Coming!* Arquilla and Ronfeldt paint a new picture of war, where "light, highly mobile forces" with decentralized information systems provide commanders with "unparalleled intelligence" (Arquilla & Ronfeldt, 1993). More recently, Kott *et al.* discuss *The Internet of Battle Things*, in which large numbers of decentralized "intelligent" devices will be gathering and communicating intelligence at a scale previously unseen in modern warfare. A key challenge the article mentioned is the reduction of the vast amount of information produced by devices in a battlefield arena to manageable levels to a summarized format that is meaningful and readable to human actors (Kott, Swami, & West, 2016).

Single Board Computers can be an inexpensive way to achieve these data summarization goals. With the expected deluge of data, we cannot always rely on satellite communication due to security or latency concerns. Setting up local HPC systems for data processing is not an option, due to the high cost required to build, power and cool such systems, which becomes nigh impossible in environments with high temperatures and scarce water resources. Furthermore, local HPC centers are difficult to move, and represent a high-value target for the enemy.

Power-efficient, decentralized data analytics increases the security and reduces latency of acquiring C^2I data. SBCs have a distinct advantage over microcontrollers and FPGAs due to their ease of reprogrammability and increased processing speeds and memory capacities. The lack of moving parts in an SBC makes it more ideal for use in harsh climates than standard computers. It also removes a single points of failure, since each SBC can easily be replaced.

In the future, it is possible that each soldier has their own personal SBC and a set of microSD cards as part of their standard equipment. Unlike many microcontrollers and FPGAs, most of the mentioned SBCs do not have integrated flash memory. From a security standpoint, this can be advantage, especially when the goal is to minimize the enemy's ability to capture mission-critical data. Soldiers can swap microSD cards into the device based on needed applications. In the case that data needs to be quickly removed due to the enemy's approach, microSD cards can be easily removed from the SBC, leaving the device behind. Even if equipment needs to be destroyed, it can be done so at reduced expense, given the relative cheapness of microSD cards and SBCs. This is an example of the "disposable security" discussed by (Kott, Swami, & West, 2016).

In visualizing the 2050 battle arena, (Kott, Alberts, & Wang, 2015) state that intelligent warfare will be prevelant, with compact and mobile variants of current systems such as UAVs and "fire and forget" missiles. We have already discussed extensively in this chapter how SBCs can assist with local data processing needs of UAVs by acting as lightweight, yet powerful on-board computers. The use of SBCs for "fire and forget" missiles has also been previously explored by (Ramirez, Blaine, & Matthews, 2015). In their paper, they discuss the use of Raspberry Pis in conjunction with the Message Passing Interface (MPI) to create "smart rounds" that can receive independent configuration instructions from a magazine server. In the design, Raspberry Pis are mounted on mortar rounds, and wirelessly receive information from the magazine server. We note the paper was published prior to the release of the Raspberry Pi Zero, which can be used to implement the design at even lower cost.

Real technical challenges do exist that limit SBC’s current use in tactical environments. For example, some SBCs such as the Raspberry Pi do not have a built-in Real-Time Clock (RTC) module, to facilitate low cost. However, a RTC module can easily be purchased separately and wired to the Pi. Other more expensive SBCs such as the Intel Edison already have built-in RTC modules.

Perhaps the greatest technical challenge facing SBCs in the military domain is their current inability to be sustained on battery power for extended periods of time. However, there is significant evidence that this hurdle will soon be overcome. In the short term, USB power packs and lithium-ion polymer batteries are being designed with SBCs in mind. In the long term, we predict that SBCs will be powered by newly-invented solid state batteries. Led by John Goodenough (the inventor of the lithium-ion battery), the new solid state batteries are inexpensive, provide three times the energy as their lithium-ion counterparts, and can operate under sub-zero temperatures (Zaragoza, 2017). It is expected that this new technology will lead to longer lasting rechargeable batteries for hand-held devices. We predict that the use of this technology will be crucial to the success of SBCs for future military data processing applications.

Lastly, we note that the SBC ecosystem is quickly evolving, partially to respond to the demand of the internet of things (IoT) market. ARM chipsets like those found in SBCs continue to have lower fabrication costs, ensuring that more powerful SoC will appear on future SBCs. The increasing ubiquity of chipsets with multiple cores and greater Random Access Memory means that programmers can create parallel applications that decrease application run-time and increase data processing capabilities. For example, NVidia recently released the Jetson TX2, a

128-bit SBC with 256 CUDA cores, a quad-core ARM processor and 8 GB of RAM (NVidia, 2017). It is advertised as a "supercomputer on module" and an "embedded platform for autonomous everything".

Intel has also been working hard to create a viable SBC with an on-board Intel processor. While Intel released a follow-up to its Edison COM called the Intel Joule in 2016 (a 64-bit SBC with a quad-core Intel Atom CPU, and 3 GB of RAM), the project was discontinued less than a year later. Since then, Intel has begun to promote the UP^2, the most advanced models of which have an Intel Pentium Quad-Core processor, 8 GB of RAM, and retails at $319.00 (UP Squared, 2017). More impressively, the UP^2 can operate in temperatures from 32-140 degrees Fahrenheit, making it suitable for deployment in areas with high temperature. One attraction of an Intel processor on an SBC is portability; data processing and summarizing techniques that are created on an Intel laptop should run "as-is" on an Intel SBC, reducing the time between development and deployment.

Summary

Single board computers are energy efficient platforms that are potentially useful for a wide variety of data analytics operations, especially in the military realm. Portability, security, and the ability to operate in potentially power-unstable environments is extremely important for military data analytics applications occurring down-range. SBCs can be incorporated with traditional hardware to create a heterogeneous ecosystem of devices that are capable of performing data summarization and preliminary analysis at every step. SBCs are also lightweight and have a low power profile, enabling them to be incorporated into unmanned aerial vehicles and autonomous ground vehicles. Further data processing capabilities are possible by networking SBCs into clusters.

We are not arguing that individual SBCs or SBC clusters should compete in the same arena as large data centers or high performance computing resources. When considering performance per Watt or raw compute numbers, larger traditional systems easily beat individual SBCs or SBC clusters (Cloutier, Paradis, & Weaver, 2014). However, SBCs are portable and consume less total energy that traditional systems.

There is also an argument to be made about energy proportional computing. (Barroso & Hölzle, 2007) observe that for Google servers, peak energy efficiency occurs at peak utilization. They argue that system designers should focus on developing machines that consume energy in proportion to the work performed (Barroso & Hölzle, 2007). While the researchers were arguing for a fundamental change in server design, (Da Costa, 2013) suggests an alternative for achieving greater energy efficiency: combining more powerful servers with low power processors such as the Intel Atom and the Raspberry Pi. His experiments show that the incorporation of SBCs in a data center alongside more powerful Intel i7s results in a more energy efficient system that a typical center containing more homogeneous architecture.

Lastly, we strongly believe that we are witnessing only the beginning of an “arms race” in the development of single-board computers. The ecosystem is evolving at considerable speed; the specific SBC models discussed in this chapter may rapidly become obsolete in the coming years. However, the trend toward the future is obvious: SBCs have the power to transform the way data is transferred and summarized in a battlefield environment. SBCs can be used to support a strategy of localized data processing, reducing the total latency in a network of communicating battlefield devices and increasing the speed at which data is summarized and analyzed for use by human operators and smart devices. For these reasons, we predict SBCs will play a critical role in future warfare operations.

Disclaimer

The views expressed in this work are solely those of the author, and do not necessarily reflect those of the U.S. Military Academy, the U.S. Army, or the Department of Defense.

References

Abrahamsson, P., Helmer, S., Phaphoom, N., Nicolodi, L., Preda, N., Miori, L., . . . Bugoloni, S. (2013). Affordable and energy-efficient cloud computing clusters: the bolzano raspberry pi cloud cluster experiment. *IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom),* (pp. 170-175). Bristol: IEEE.

Anwar, A., Krish, K. R., & Butt, A. R. (2014). On the use of microservers in supporting hadoop applications. *2014 IEEE International Conference on Cluster Computing (CLUSTER)* (pp. 66-74). Madrid: IEEE.

ArduPilot Mega Project. (last accessed 2016, December 14). Retrieved from ArduPilot Mega, the Open Source Autopilot: http://www.ardupilot.co.uk/

Arquilla, J., & Ronfeldt, D. (1993). Cyberwar is coming! *Comparative Strategy 12(2),* pp. 141-165.

Barroso, L. A., & Hölzle, U. (2007). The case for energy-proportional computing. *IEEE Computer, vol 40*, 33-37.

Bell, C. (2013). *Beginning Sensor Networks with Arduino and Raspberry Pi (Technology in Action).* Apress.

Choi, H., Geeves, M., Alsalam, B., & Gonzalez, L. F. (2016). Open source computer-vision based guidance system for UAVs on-board decision making. *2016 IEEE Aerospace Conference* (pp. 1-5). IEEE.

Cloutier, M. F., Paradis, C., & Weaver, V. M. (2014). Design and analysis of a 32-bit embedded high-performance cluster optimized for energy and performance. *Hardware-Software Co-Design for High Performance Computing (Co-HPC)* (pp. 1-8). IEEE.

Cox, S. J., Cox, J. T., Boardman, R. P., Johnston, S. J., Scott, M., & O’Brien, N. S. (2013).

Iridis-pi: a low-cost, compact demonstration cluster. *Cluster Computing, 17(2)*, 349-358.

Da Costa, G. (2013). Heterogeneity: The Key to Achieve Power-Proportional Computing. *13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*, (pp. 656-662). Delft.

da Silva, J. F., Brito, A. V., & Nogueira de Moura, H. (2015). An Embedded System for Aerial Image Processing from Unmanned Aerial Vehicles. *2015 Brazilian Symposium on Computing Systems Engineering (SBESC)* (pp. 154-157). IEEE.

Djanali, S., Arunanto, F. X., Pratomo, B. A., Studiawan, H., & Nugraha, S. G. (2014). SQL injection detection and prevention system with raspberry Pi honeypot cluster for trapping attacker. *International Symposium on Technology Management and Emerging Technologies (ISTMET)* (pp. 163-166). Bandung: IEEE.

Đurišić, M. P., Tafa, Z., Dimić, G., & Milutinović, V. (2012). A survey of military applications of wireless sensor networks. *2012 Mediterranean conference on embedded computing (MECO)* (pp. 196-199). IEEE.

Emlid. (2016). *Navio2: Linux autopilot on Raspberry Pi*. Retrieved from Emlid, Last accessed Dec. 6, 2016: https://emlid.com/

Hong, J., Raymond, J., & Shackelford, J. (2014). *EdiSense: A Replicated Datastore for IoT Data.* Stanford University.

Intel. (2016, August). *Discover the Intel® Joule™ Compute Module*. Retrieved from Intel Software Developer Zone, last accessed March 2017: https://software.intel.com/en-us/iot/hardware/joule

Intel. (2017, November). *Discontinued Maker & Innovation Products.* Retrieved from Intel Software Developer Zone, last accessed November 2017. https://software.intel.com/en-us/iot/hardware/discontinued

Kachris, C., Stamelos, I., & Soudris, D. (2016). Performance and Energy evaluation of Spark applications on low-power SoCs. *International Conference on Embedded Computer Systems: Architectures, Modelling, Simulation (SAMOS).* IEEE.

Kaewkasi, C., & Srisuruk, W. (2014). A study of big data processing constraints on a low-power hadoop cluster. *2014 International Computer Science and Engineering Conference (ICSEC)* (pp. 267-272). IEEE.

Kiepert, J. (2013). *Creating a raspberry pi-based beowulf cluster.* Boise: Boise State Univeristy.

Kim, H.-S., Beak, S.-H., & Park, S.-Y. (2016). Parallel Hough Space Image Generation Method for Real-Time Lane Detection. *17th International Conference on Advanced Concepts for Intelligent Vision Systems* (pp. 81-91). Lecce, Italy: Springer.

Kott, A., Alberts, D. S., & Wang, C. (2015, December). Will Cybersecurity Dictate the Outcome of Future Wars? *Computer 48(12),* pp. 98-101.

Kott, A., Swami, A., & West, B. J. (2016, December). The Internet of Battle Things. *Computer 46 (12),* pp. 70-75.

Lee, Y., & Kim, H. (2016). Real-time lane detection and departure warning system on embedded platform. *International Conference on Consumer Electronics* (pp. 1--4). Berlin: IEEE.

Malladi, R., & Agrawal, D. P. (2002). Current and Future Applications of Mobile and Wireless Networks. *Communications of the ACM, 45(10),* 144-146.

Matthews, S. J., Blaine, R. W., & Brantly, A. (2016). Evaluating Single Board Computer Clusters for Cyber Operations. *The 1st International Conference on Cyber Conflict in the United States (CyconUS).* Washington DC: IEEE.

Meier, L., Honegger, D., & Pollefeys, M. (2015). PX4: A Node-Based Multithreaded Open

Source Robotics Framework for Deeply Embedded Platforms. *2015 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 6235-6240). Seattle: IEEE.

Meng, X., Wang, W., & Leong, B. (2015). SkyStitch: A Cooperative Multi-UAV-based Real-time Video Surveillance System with Stitching. *Proceedings of the 23rd ACM International Conference on Multimedia* (pp. 261--270). Brisbane: ACM.

NVidia. (2017, March). *The NVidia Jetson: the embedded platform for autonomous everything*. Retrieved from NVidia Embedded Systems, last accessed March 2017: http://www.nvidia.com/object/embedded-systems-dev-kits-modules.html

Pahl, C., Helmer, S., Miori, L., Sanin, J., & Lee, B. (2016). A Container-Based Edge Cloud PaaS Architecture Based on Raspberry Pi Clusters. *IEEE International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)* (pp. 117-124). Vienna: IEEE.

Porcupine Labs. (2015). *PiSense*. http://www.porcupinelabs.com/pisense/.

Ramirez, Z. J., Blaine, R. W., & Matthews, S. J. (2015). Augmenting the Remotely Operated Automated Mortar System with Message Passing. *CrossTalk*, 12-15.

Rebouças, R. A., Eller, Q. d., Habermann, M., & Hideiti Shiguemori, E. (2013). Embedded System for Visual Odometry and Localization of Moving Objects in Images Acquired by Unmanned Aerial Vehicles. *013 III Brazilian Symposium on Computing Systems Engineering* (pp. 35-40). IEEE.

Samad, T., Bay, J. S., & Godbole, D. (2007). Network-centric systems for military operations in urban terrain: the role of UAVs. *Proceedings of the IEEE 95(1)*, 92-107.

Spillner, J., Beck, M., Schill, A., & Bohnert, T. M. (2015). Stealth Databases: Ensuring User-Controlled Queries in Untrusted Cloud Environments. *2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC)*, (pp. 261-270). Limassol.

Stokke, K. R., Stensland, H. K., Griwodz, C., & Halvorsen, P. (2015). Energy efficient video encoding using the tegra K1 mobile processor. *Proceedings of the 6th ACM Multimedia Systems Conference* (pp. 81-84). ACM.

Tso, F. P., White, D. R., Jouet, S., Singer, J., & Pezaros, D. P. (2013). The glasgow raspberry pi cloud: A scale model for cloud computing infrastructures. *IEEE 33rd International Conference on Distributed Computing Systems Workshops* (pp. 108-112). Philadelphia: IEEE.

UP Squared (2017). *It's time to get squared: UP Squared*. Retrieved from Up Board website, last accessed November 2017: http://www.up-board.org/upsquared/

Varghese, B., Carlsson, N., Jourjon, G., Mahanti, A., & Shenoy, P. (2014). Greening web servers: A case for ultra low-power web servers. *2014 International Green Computing Conference (IGCC)* (pp. 1-8). Dallas: IEEE.

Vega, A., Lin, C. C., Swaminathan, K., Buyuktosunoglu, A., Pankanti, S., & Bose, P. (2015). Resilient, UAV-embedded real-time computing. *2015 33rd IEEE International Conference on Computer Design (ICCD)* (pp. 736-739). New York: IEEE.

Vujović, V., & Maksimović, M. (2014). Raspberry Pi as a wireless sensor node: performances and constraints. *37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (pp. 1013-1018). IEEE.

Winkler, M., Tuchs, K.-D., Hughes, K., & Barclay, G. (2008). Theoretical and practical aspects of military wireless sensor networks. *Journal of Telecommunications and Information Technology*, 37-45.

Yamauchi, B. M. (2004). PackBot: a versatile platform for military robotics. *Defense and Security* (pp. 228-237). International Society for Optics and Photonics.

Zaragoza, S. (2017, February). Lithium-Ion Battery Inventor Introduces New Technology for Fast-Charging, Noncombustible Batteries. *UTNews: University of Texas at Austin Press Releases*, p. 1.

Zsedrovits, T., Peter, P., Bauer, P., Pencz, B. J., Hiba, A., I. Gozse, I., . . . Bokor, J. (2016, September). Onboard visual sense and avoid system for small aircraft. *IEEE Aerospace and Electronic Systems Magazine, vol. 31, no. 9*, pp. 18-27.

Zsedrovits, T., Zarándy, Á., Pencz, B., Hiba, A., Nameth, M., & Vanek, B. (2015). Distant aircraft detection in sense-and-avoid on kilo-processor architectures. *2015 European Conference on Circuit Theory and Design (ECCTD)* (pp. 1-4). Trondheim: IEEE.