Multi-core Embedded Wireless Sensor Networks: Architecture and Applications

Arslan Munir, *Member, IEEE,* Ann Gordon-Ross, *Member, IEEE,* and Sanjay Ranka, *Fellow, IEEE, AAAS*

Abstract—Technological advancements in the silicon industry, as predicted by Moore's law, have enabled integration of billions of transistors on a single chip. To exploit this high transistor density for high performance, embedded systems are undergoing a transition from single-core to multi-core. Although a majority of embedded wireless sensor networks (EWSNs) consist of single-core embedded sensor nodes, multi-core embedded sensor nodes are envisioned to burgeon in selected application domains that require complex in-network processing of the sensed data. In this paper, we propose an architecture for heterogeneous hierarchical multi-core embedded wireless sensor networks (MCEWSNs) as well as an architecture for multi-core embedded sensor nodes used in MCEWSNs. We elaborate several compute-intensive tasks performed by sensor networks and application domains that would especially benefit from multi-core embedded sensor nodes. This paper also investigates the feasibility of two multi-core architectural paradigms—symmetric multiprocessors (SMPs) and tiled many-core architectures (TMAs)—for MCEWSNs. We compare and analyze the performance of an SMP (an Intel-based SMP) and a TMA (Tilera's TILEPro64) based on a parallelized information fusion application for various performance metrics (e.g., runtime, speedup, efficiency, cost, and performance per watt). Results reveal that TMAs exploit data locality effectively and are more suitable for MCEWSN applications that require integer manipulation of sensor data, such as information fusion, and have little or no communication between the parallelized tasks. To demonstrate the practical relevance of MCEWSNs, this paper also discusses several state-of-the-art multi-core embedded sensor node prototypes developed in academia and industry. We further discuss research challenges and future research directions for MCEWSNs.

Index Terms—wireless sensor networks, multi-core, embedded systems, symmetric multiprocessors, tiled many-core architecture, near-threshold computing, heterogeneous, compressive sensing

1 Introduction and Motivation

Embedded wireless sensor networks (EWSNs) consist of sensor nodes with embedded sensors to sense data about a phenomenon and these sensor nodes communicate with neighboring sensor nodes over wireless links. Many emerging EWSN applications (e.g., surveillance, volcano monitoring) require a plethora of sensors (e.g., acoustic, seismic, temperature, and, more recently, image sensors and/or smart cameras) embedded in the sensor nodes. Although traditional EWSNs equipped with scalar sensors (e.g., temperature, humidity) transmit most of the sensed information to a sink node (base station node), this *sense-transmit* paradigm is becoming infeasible for information-hungry applications equipped with a plethora of sensors, including image sensors and/or smart cameras.

Processing and transmission of the large amount of sensed data in emerging applications exceeds the capabilities of traditional EWSNs. For example, consider a military EWSN deployed in a battlefield, which requires various sensors, such as imaging, acoustic, and electromagnetic sensors. This application presents various challenges for existing EWSNs since transmission of high-resolution images and video streams over bandwidth-limited wireless links from sensor nodes to the sink node is infeasible. Furthermore, meaningful processing of multimedia data (acoustic, image, and video in this example) in real-time exceeds the capabilities of traditional EWSNs consisting of single-core embedded sensor nodes [1][2], and requires more powerful embedded sensor nodes to realize this application.

Since single-core EWSNs will soon be unable to meet the increasing requirements of information-rich applications (e.g., video sensor networks), next generation sensor nodes must possess enhanced computation and communication capabilities. For example, the transmission rate for the first generation Mica motes was 38.4 kbps whereas the second generation Mica motes (MicaZ motes) can communicate at 250 kbps using IEEE 802.15.4 (Zigbee) [3]. Despite these advances in communication, limited wireless bandwidth from sensor nodes to the sink node makes timely transmission of multimedia data to the sink node infeasible. In traditional EWSNs, the communication energy dominates the computation energy. For example, an embedded sensor node produced by Rockwell Automation [4] expends 2000x more energy for transmitting a bit than that of executing a single instruction [5]. Similarly, transmitting a 15 frames per second (FPS) digital video stream over a wireless Bluetooth link takes 400 mW [6].

Fortunately, there exists a tradeoff between transmission and computation in an EWSN, which is

- *This paper has supplementary material available at http://ieeexplore.ieee.org, provided by the authors. The supplementary material contains additional discussions, details, figures, etc., for multi-core embedded wireless sensor networks proposed in this paper. Authors' e-mails: {arslan@rice.edu, ann@ece.ufl.edu, ranka@cise.ufl.edu}* well-suited for in-network processing for information-rich applications and allows transmission of only event descriptions (e.g., detection of a target of interest) to the sink node to conserve energy. Technological advancements in multi-core architectures have made multi-core processors a viable and cost-effective choice for increasing the computational ability of embedded sensor nodes. Multi-core embedded sensor nodes can extract the desired information from the sensed data and communicate only this processed information, which reduces the data transmission volume to the sink node. By replacing a large percentage of communication with in-network computation, multi-core embedded sensor nodes could realize large energy savings that would increase the sensor network's overall lifetime.

Multi-core embedded sensor nodes enable energy savings over traditional single-core embedded sensor nodes in two ways. First, reducing the energy expended in communication by performing in-situ computation of sensed data and transmitting only processed information. Second, a multi-core embedded sensor node allows the computations to be split across multiple cores while running each core at a lower processor voltage and frequency, as compared to a single-core system, which results in energy savings. Utilizing a single-core embedded sensor node for information processing in information-rich applications requires the sensor node to run at a high processor voltage and frequency to meet the application's delay requirements, which increases the power dissipation of the processor. A multi-core embedded sensor node reduces the number of memory accesses, clock speed, and instruction decoding, thereby enabling higher arithmetic performance at a lower power consumption as compared to a single-core processor [6].[1]

This paper investigates the feasibility of two multi-core architectures that can be used in processing units of embedded sensor nodes for multi-core embedded wireless sensor networks (MCEWSNs): symmetric multiprocessors (SMPs) and tiled many-core architectures (TMAs).[2] We consider SMPs because SMPs are ubiquitous and pervasive, which provides a standard/fair basis for comparing with other novel architectures (e.g., TMAs) [7]. We consider Tilera's TILEPro64 for TMAs because of Tilera's innovative architectural features (e.g., three-way issue superscalar tiles, on-chip mesh interconnect, and dynamic distributed cache (DDC) technology). Despite the diversity of application domains for MCEWSNs (e.g., military, health, satellites), many application domains have information fusion as one of the most critical applications, and hence we parallelize the information fusion application both for SMPs and TMAs. We compare and analyze the performance of an SMP (an Intel-based SMP) and a TMA (Tilera's TILEPro64) for performance evaluation.

The choice of a multi-core architecture dictates the high-level parallel languages since some multi-core architectures support proprietary parallel languages whose benchmarks are not available open source (e.g., Tilera's TILEPro64). Tilera provides a multi-core development environment (MDE) ilib API [8] whereas many SMPs (e.g., the Intel-based SMP) support OpenMP (Open Multi-processing), hence the *cross-architectural evaluation* results may be affected by the parallel language's efficiency. However, our analysis provides insights into the attainable performance per watt from these two multi-core architectures for MCEWSNs. To the best of our knowledge, this paper is the first to highlight the feasibility and application of multi-core technology in EWSNs. Although few initiatives study the feasibility of multi-core technology for EWSNs [9][10], no prior work proposes an MCEWSN architecture based on multi-core embedded sensor nodes. Furthermore, motives and application domains for MCEWSNs have not yet been characterized. Our main contributions are as follows:

- Proposal of a heterogeneous hierarchical MCEWSN and associated multi-core embedded sensor node architecture.
- Elaboration on several computation-intensive tasks performed by sensor networks that would especially benefit from multi-core embedded sensor nodes.
- Characterization and discussion of various application domains for MCEWSNs.
- Discussion of several state-of-the-art multi-core embedded sensor node prototypes developed in academia and industry.[3]
- Parallelization of an information fusion application for two multi-core architectures (SMPs and TMAs) that can be used in embedded sensor nodes' processing units.
- Comparison and analysis of the performance and performance per watt of SMPs and TMAs based on our parallelized information fusion application. This analysis demonstrates performance and performance per watt advantages attained by multi-core embedded sensor nodes as compared to single-core embedded sensor nodes.

The remainder of this paper is organized as follows. Section 2 proposes an MCEWSN architecture. Potential application domains amenable to MCEWSNs are discussed in Section 3. Results are presented in Section 4. Section 5 discusses the research challenges and future research directions for MCEWNS and Section 6 concludes this paper.

[1] The supplementary material document posted online discusses prior work related to multi-core embedded sensor nodes.

[2] The discussion of sensor nodes' multi-core architectures and parallel computing metrics that we use to evaluate these architectures is presented in Section 3 of the supplementary material document.

[3] Detailed in Section 5 of the supplementary material document.

2 Multi-core Embedded Wireless Sensor Network Architecture

Fig. 1 depicts our proposed heterogeneous hierarchical MCEWSN architecture, which satisfies the increasing in-network computational requirements of emerging EWSN applications. The heterogeneity in the architecture subsumes the integration of numerous single-core embedded sensor nodes and several multi-core embedded sensor nodes. We note that homogeneous hierarchical single-core EWSNs have been discussed in literature for large EWSNs (EWSNs consisting of a large number of sensor nodes) [11][12]. Our proposed architecture is hierarchical since the architecture comprises of various clusters (a group of embedded sensor nodes in communication range with each other) and a sink node. A hierarchical network is well suited for large EWSNs since small EWSNs, which consist of only a few sensor nodes, can send the sensed data directly to the base station or sink node.

Each cluster consists of several leaf sensor nodes and a cluster head. Leaf sensor nodes contain a single-core processor and are responsible for sensing, pre-processing sensed data, and transmitting sensed data to the cluster head nodes. Since leaf sensor nodes are not intended to perform complex processing of sensed data in our proposed architecture, a single-core processor sufficiently meets the computational requirements of leaf sensor nodes. Cluster head nodes consist of a multi-core processor and are responsible for coalescing/fusing the data received from leaf sensor nodes for transmission to the sink node in an energy- and bandwidth-efficient manner. Our proposed architecture with multi-core cluster heads is based on practical reasons since sending all the collected data from the cluster heads to the sink node is not feasible for bandwidth limited EWSNs, which warrants complex processing and information fusion to be carried out at cluster head nodes and only the concise processed information is transmitted to the sink node.[4]

The sink node contains a multi-core processor and is responsible for transforming high-level user queries from the control and analysis center (CAC) to network-specific directives, querying the MCEWSN for the desired information, and returning the requested information to the user/CAC. The sink node's multi-core processor facilitates post-processing of the information received from multiple cluster heads. The post-processing at the sink node includes information fusion and event detection based on aggregated data from all of the sensor nodes in the network. The CAC further analyzes the information received from the sink node and issues control commands and queries to the sink node.

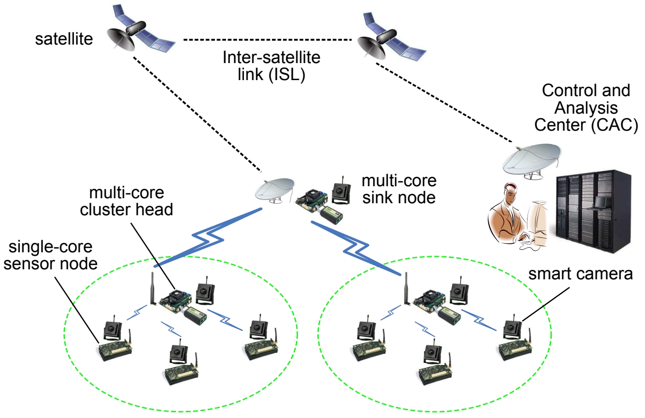


Fig. 1: A heterogeneous multi-core embedded wireless sensor network (MCEWSN) architecture.

MCEWSNs can be coupled with a satellite backbone network that provides long-haul communication from the sink node to the CAC since MCEWSNs are often deployed in remote areas with no wireless infrastructure, such as a cellular network infrastructure. The satellites in the satellite backbone network communicate with each other via inter-satellite links (ISLs). Since a satellite's uplink and downlink bandwidth is limited, a multi-core processor in the sink node is required to process, compress, and/or encrypt the information sent to the satellite backbone network.

Even though this paper focuses on heterogeneous MCEWSNs, homogenous MCEWSN architectures are an extension of our proposed architecture (Fig. 1) where leaf sensor nodes also contain a multi-core processor. In a homogeneous MCEWSN equipped with multiple sensors, each processor core in a multi-core embedded sensor node can be assigned to process one sensing task (e.g., one processor core handles sensed temperature data and another processor core handles sensed humidity data and so on) as opposed to single-core embedded sensor nodes where the single processor core is responsible for processing all of the sensed data from all of the sensors. We focus on heterogeneous MCEWSNs as we believe that heterogeneous MCEWSNs would serve as a first step towards integration of multi-core and sensor networking technology because of the following reason. Due to the dominance of single-core embedded sensor nodes in existing EWSNs, replacing all of the single-core embedded sensor nodes with multi-core embedded sensor nodes may not be feasible and cost-effective given that only a few multi-core embedded sensor nodes operating as cluster heads could meet an application's in-network computation requirements. Hence, our proposed heterogeneous MCEWSN would enable a smooth transition from single-core to multi-core EWSNs.[5]

3 MCEWSN Application Domains

MCEWSNs are suitable for sensor networking application domains that require complex in-network

[4] Section 4 of the supplementary material document elaborates on several compute-intensive tasks that motivated the emergence of MCEWSNs.

[5] Section 2 of the supplementary material document depicts the architecture of a multi-core embedded sensor node in our MCEWSN.

information processing such as wireless video sensor networks, wireless multimedia sensor networks, satellite-based wireless sensor networks, space shuttle sensor networks, aerial-terrestrial hybrid sensor networks, and fault-tolerant sensor networks. In this section, we discuss these application domains for MCEWSNs.[6]

3.1 Wireless Video Sensor Networks (WVSNs)

Wireless video sensor networks (WVSNs) are WSNs in which smart cameras and/or image sensors are embedded in the sensor nodes. WVSNs emulate the compound eye found in certain arthropods. Although WVSNs are a subset of wireless multimedia sensor networks (WMSNs), we discuss WVSNs separately to emphasize the WVSNs' stand-alone existence. WVSNs are suitable for applications in areas such as homeland security, battlefield monitoring, and mining. For example, video sensors deployed at airports, borders, and harbors provide a level of continuous and accurate monitoring and protection that is otherwise unattainable. We discuss the application of multi-core embedded sensor nodes both for image- and video-centric WVSNs.

In image-centric WVSNs, multiple image/camera sensors observe a scene from multiple directions and are able to describe objects in their true three-dimensional appearance by overcoming occlusion problems. Low-cost imaging sensors are readily available, such as CCD and CMOS imaging sensors from Kodak, and the Cyclops camera from the University of California at Los Angeles (UCLA) designed as an add-on for Mica sensor nodes [6]. Image pre-processing involves convolutions and data-dependent operations using a limited neighborhood of pixels. The signal processing algorithms for image processing in WVSNs typically exhibit a high degree of parallelism and are dominated by a few regular kernels (e.g., FFT) that are responsible for a large fraction of the execution time and energy consumption. Accelerating these kernels on multi-core embedded sensor nodes would achieve significant speedup in execution time and reduction in energy consumption, and would help achieve real-time computational requirements for many applications in energy-constrained domains.

Video-centric WVSNs rely on multiple video streams from multiple embedded sensor nodes. Since sensor nodes can only serve low-resolution video streams given the sensor nodes' resource limitations, a single video stream alone does not contain enough information for vision analysis such as event detection and tracking, however, multiple sensor nodes can capture video streams from different angles and distances together providing enormous visual data [3]. Video encoders rely on intraframe compression techniques that reduce redundancy within one frame and interframe compression techniques (e.g., predictive coding) that exploit redundancy among subsequent frames [1]. Video coding techniques require complex algorithms that exceed the computing power of single-core embedded sensor nodes. The visual data from numerous sensor nodes can be combined to give high-resolution video streams, however, this processing requires multi-core embedded sensor nodes and/or cluster heads.

3.2 Wireless Multimedia Sensor Networks (WMSNs)

A wireless multimedia sensor network (WMSN) consists of wirelessly connected embedded sensor nodes that can retrieve multimedia content such as video and audio streams, still images, and scalar sensor data of the observed phenomenon. WMSNs target a large variety of distributed, wireless, streaming multimedia networking applications ranging from home surveillance to military and space applications. A multimedia sensor captures audio and image/video streams using an embedded microphone and a micro-camera.

Various sensors in a WMSN coordinate closely to achieve application goals. For example, in a military application for target detection and tracking, acoustic and electromagnetic sensors can enable early detection of a target but may not provide adequate information about the target. Additional target details, such as type of vehicle, equipped armaments, and onboard personnel, are often required and gathering these details requires image sensors. Although the sensing ability in most sensors is isotropic and attenuates with distance, a distinct characteristic of video/image sensors is these sensors' directional sensing ranges. Recently, omnicameras have become available, which can provide complete coverage of the scene around a sensor node, however, applications are limited to close range scenarios to guarantee sufficient image resolution for moving objects [3]. To ensure full coverage of the sensor field, a set of directional cameras is required to capture enough information for activity detection. The image and video sensors high sensing cost limits these sensors continuous activation given constrained embedded sensor node resources. Hence, the image and video sensors in a WMSN require sophisticated control such that the image and video sensors are triggered only after a target is detected based on sensed data from other lower cost sensors, such as acoustic and electromagnetic.

Desirable WMSN characteristics include the ability to store, process in real-time, correlate, and fuse multimedia data originating from heterogeneous sources [1]. Multimedia contents, especially video streams, require data rates that are orders of magnitude higher than those supported by traditional single-core embedded sensor nodes. To process multimedia data in real-time and to reduce the wireless bandwidth demand, multi-core embedded sensor nodes in the network are required. Multi-core embedded sensor nodes facilitate

[6] Section 5 of the supplementary material document describes several state-of-the-art multi-core embedded sensor node prototypes.

in-situ processing of voluminous information from various sensors, notifying the CAC only once an event is detected (e.g., target detection).

3.3 Satellite-based Wireless Sensor Networks (SBWSN)

A satellite-based wireless sensor network (SBWSN) is a wireless communication sensing network composed of many satellites, each equipped with multi-functional sensors, long-range wireless communication modules, thrusters for attitude adjustment, and a computational unit (potentially multi-core) to carry out processing of the sensed data. Traditional satellite missions are extremely expensive to design, build, launch, and operate, thereby motivating the aerospace industry to focus on distributed space missions, which would consist of multiple small, inexpensive, and distributed satellites coordinating to attain mission goals. SBWSNs would enable robust space missions by tolerating the failure of a single or a few satellites as compared to a large single satellite, where a single failure could compromise the success of a mission. SBWSNs can be used for a variety of missions, such as space weather monitoring, studying the impact of solar storms on Earth's magnetosphere and ionosphere, environmental monitoring (e.g., pollution, land, and ocean surface monitoring), and hazard prediction (e.g., flood and earthquake prediction).

Each SBWSN mission requires specific orbits and constellations to meet mission requirements and GPS provides an essential tool for orbit determination and navigation. Typical constellations include string-of-pearls, flower constellation, and satellite cluster. In particular, the flower constellation provides stable orbit configurations, which are suitable for *micro-satellite* (mass < 100 kg), *nano-satellite* (mass < 10 kg), and *pico-satellite* (mass < 1 kg) missions. Important orbital factors to consider in SBWSN design are relative range (distance) and speed between satellites, the inter-satellite link (ISL) access opportunity, and the ground-link access opportunity. The access time is the time for two satellites to communicate with each other and depends on the distance between the satellites (range). Satellites in an SBWSN can be used as an interferometer, which correlates different images acquired from slightly different angles/view points in order to get better resolution and more meaningful insights.

All of the satellites in an SBWSN collaborate to sense the desired phenomenon, communicate over long distances through beam-forming over an ISL, and maintain the network topology through self-organized mobility [13]. Studies indicate that IEEE 802.11b (Wi-Fi) and IEEE 802.16 (WiMax) can be used for inter-satellite communications (communication between satellites) and IEEE 802.15.4 (Zigbee) can be used for intra-satellite (communication between sensor nodes within a satellite) communications [14]. We point out that the IEEE 802.11b protocol requires modifications for use in an ISL where the distance between satellites is more than one kilometer since the IEEE 802.11b standard normally supports a communication range within 300 meters. The feasibility of wireless protocols for inter-satellite communication depends on the range, power requirements, medium access control (MAC) features, and support for mobility. The intra-satellite protocols are mainly selected based on power since the range is small within a satellite. A low duty cycle and the ability to put the radio to sleep are desirable features for intra-satellite communication protocols. For example, the MICA2DOT mote, which requires 24 mW of active power and 3 μW of standby power, supplied by a 3 V 750 mAh battery cell can last for 27,780 hours $\approx$ three years and two months, while operating at a duty cycle of 0.1% (supported by Zigbee) [15].

Since an individual satellite within an SBWSN may not have sufficient power to communicate with a ground station, a sink satellite in an SBWSN can communicate with a ground station, which is connected to the CAC. Ground communication in SBWSNs takes place in very-high frequency (VHF) (30 MHz – 300 MHz) and ultra-high frequency (UHF) (300 MHz – 3 GHz) bands. VHF frequencies pass through the ionosphere with effects, such as scintillation, fading, Faraday's rotation, and multi-path effects during intense solar cycles due to reflection of the VHF signals. UFH frequencies, in which both S- and L-bands lie, can suffer severe disruptions during a solar storm. For a formation of several SNAP-1 nano-satellites, the typical downlink data rate is 38.4 kbps or 76.8 kbps maximum [15], which necessitates multi-core embedded sensor nodes in SBWSNs to perform in-situ processing so that only event descriptions are sent to the CAC.

3.4 Space Shuttle Sensor Networks (3SN)

A space shuttle sensor network (3SN) corresponds to a network of sensors aimed to monitor a space shuttle during pre-flight, ascent, on-orbit, and re-entry phases. Battery-operated embedded wireless sensors can be easily bonded to the space shuttle's structure and enable real-time monitoring of temperature, triaxial vibration, strain, pressure, tilt, chemical, and ultrasound data. MCEWSNs would enable real-time monitoring of space vehicles not possible by ground-based sensing systems. For example, the Columbia space shuttle accident was caused by damage done when foam shielding dislodged from the external fuel tank during the shuttle's launch, which damaged the wing's leading edge panels [16]. The vehicle lacked on-board sensors that could have enabled ground personnel to determine the extent and location of the damage. Ground-based cameras captured images of the impact but were not able to reliably characterize the location and severity of the impact and resulting damage.

MCEWSNs for space shuttles, currently under development, would be used for space shuttle main engine (SSME) crack investigation, space shuttle environmental control life support system (ECLSS) oxygen and nitrogen flexhose analysis, and wing leading edge impact detection. Since the amount of data acquired during the 10-minute ascent period is nearly 100 MB, the time to download all data, even for a single event, via the radio frequency (RF) link is prohibitively long. Hence, information fusion algorithms are required in 3SNs to minimize the quantity and increase the quality of data being transmitted via the RF link. Furthermore, MCEWSNs would enable a 10x reduction in the installation costs for the shuttle as compared to the sensing systems based on traditional wired approaches [16].

3.5 Aerial-Terrestrial Hybrid Sensor Networks (ATHSNs)

Aerial-terrestrial hybrid sensor networks (ATHSNs), which consist of ground sensors and aerial sensors, integrate terrestrial sensor networks with aerial/space sensor networks. To connect remote terrestrial EWSNs to a CAC located far away in urban areas, ATHSNs can include a satellite backbone network. The satellite backbone network is widely available at remote locations and provides a reliable and broadband communication network [17][18]. Various satellite communication choices are possible, such as WildBlue, HughesNet, and NASA's geostationary operational environmental satellite (GOES) system. However, a satellite's uplink and downlink bandwidth is limited, and requires pre-processing as well as compression of sensed data, especially multimedia data such as image and video streams. Multi-core embedded sensor nodes are suitable for ATHSNs, and are capable of carrying out the processing and compression of high-quality image and video streams for transmission to and from a satellite backbone network.

Aerial networks in ATHSNs may consist of unmanned aerial vehicles (UAVs) and satellites. For example, consider an ATHSN in which UAVs contain embedded image and video sensors such that only the image scenes that are of significant interest from a military strategy perspective are sensed in greater detail. The working of ATHSNs consisting of UAVs and satellites can be described concisely in seven steps [17]: 1) Ground sensors detect the presence of a hostile target in the monitored field and store events in memory; 2) The satellite periodically contacts multi-core cluster heads in the terrestrial EWSN to download updates about the target's presence; 3) Satellites contact UAVs to acquire image data about the scene where the intrusion is detected; 4) UAVs gather image data through the embedded image sensors; 5) The embedded multi-core sensors in UAVs process and compress the image data for transmission to the satellite backbone network in a bandwidth-efficient manner; 6) The satellite backbone network relays the processed information received from the UAVs to the CAC; 7) The satellite backbone network relays the commands (e.g., launching the UAVs' arsenals) from the CAC to the UAVs.

Ye et al. [18] have implemented an ATHSN prototype for an ecological study using temperature, humidity, photosynthetically active radiation (PAR), wind speed, and precipitation sensors. The prototype consists of a small satellite dish and a communication modem for integrating a terrestrial EWSN with the WildBlue satellite backbone network, which provides commercial service. The prototype uses Intel's Stargate processor as the sink node, which provides access control and manages the use of the satellite link.

The transformational satellite (TSAT) system is a future generation satellite system that is designed for military applications by National Aeronautics and Space Administration (NASA), the U.S. Department of Defense (DoD), and the Intelligence Community (IC) [17]. The TSAT system is a constellation of five satellites, placed in geostationary orbit, that constitute a high-bandwidth satellite backbone network, which allows terrestrial units to access optical and radar imagery from UAVs and satellites in real-time. TSAT provides broadband, reliable, worldwide, and secure transmission of data. TSAT supports RF communication links with data rates up to 45 Mbps and laser communication links with data rates up to 10-100 Gbps [17].

3.6 Fault-Tolerant (FT) Sensor Networks

The sensor nodes in an EWSN are typically deployed in harsh and unattended environments, which makes fault-tolerance (FT) an important consideration in EWSN design, particularly for space-based WSNs. For example, the temperature of aerospace vehicles varies from cryogenic to extremely high temperature, and pressure from vacuum to very high pressure. Additionally, shock and vibration levels during launch can cause component failures. Furthermore, high levels of ionizing radiation requires electronics to be FT if not radiation-hardened (rad-hard). Multi-core embedded sensors can provide hardware-based (e.g., triple modular redundancy (TMR) or self-checking pairs (SCP)) as well as software-based (e.g., algorithm-based fault tolerance (ABFT)) FT mechanisms for applications requiring high reliability. Computations, such as pre-processing and data fusion, can be replicated on multiple cores so that if radiation corrupts processing on one core, processing on other cores would still enable reliable computation of results.

4 Results

In this section, we present performance and performance per watt results for the two multi-core architectures (SMPs and TMAs) that can be used in MCEWSNs. For the SMP architecture, we evaluate an eight-core Intel-based SMP consisting of two Intel Xeon E5430 quad-core processors fabricated at 45nm CMOS lithography [19] with a maximum clock frequency of 2.66 GHz

TABLE 1: Performance results for the information fusion application for SMP$^{\text{2xQuadXeon}}$ when M = 40.

Problem Size	# of Cores	Execution Time (s)	Speedup	Efficiency	Cost	Perf.	Perf. per watt
N	p	T_p	$S = T_s/T_p$	$E = S/p$	$C = T_p \cdot p$	(MOPS)	(MOPS/W)
3000,000	1	12.02	1	1	12.02	1073.2	22.36
3000,000	2	7.87	1.53	0.76	15.74	1639.14	25.61
3000,000	4	4.03	2.98	0.74	16.12	3201	33.34
3000,000	6	2.89	4.2	0.7	17.34	4463.67	34.87
3000,000	8	2.48	4.85	0.61	19.84	5201.6	32.51

TABLE 2: Performance results for the information fusion application for the TILEPro64 when M = 40.

Problem Size	# of Tiles	Execution Time (s)	Speedup	Efficiency	Cost	Perf.	Perf. per watt
N	p	T_p	$S = T_s/T_p$	$E = S/p$	$C = T_p \cdot p$	(MOPS)	(MOPS/W)
3000,000	1	70.65	1	1	70.65	182.6	34.07
3000,000	2	35.05	2	1	70.1	368	64.33
3000,000	4	17.18	4.1	1.02	68.72	750.87	116.6
3000,000	6	11.48	6.2	1.03	68.9	1123.69	156.94
3000,000	8	8.9	7.94	0.99	71.2	1449.44	183.94
3000,000	10	6.79	10.4	1.04	67.9	1899.85	221.17
3000,000	50	1.46	48.4	0.97	73	8835.62	384.66

[7]. For conciseness, we will refer to this SMP as SMP$^{\text{2xQuadXeon}}$ in the remainder of this paper. Results in this paper focus only on parallelization to demonstrate the performance and performance per watt advantages that can be attained by leveraging multi-core embedded sensor nodes. Implementation of a complete MCEWSN architecture (Fig. 1) for real-world applications, such as video surveillance, is a focus of our future research work. Considering the significance of information fusion for EWSNs, we parallelize an information fusion application both for SMPs and TMAs to investigate the suitability of the two architectures for MCEWSNs. We analyze an information fusion application as an example to demonstrate the performance and performance per watt advantages of multi-core embedded sensor nodes as compared to single-core embedded sensor nodes, although other sensor applications can be parallelized to demonstrate similar advantages.[7]

We parallelize the information fusion application for SMPs and TMAs using OpenMP and Tilera's MDE ilib API. The purpose of this comparison between SMPs and TMAs is to investigate the feasibility of SMPs and TMAs as multi-core processor architectures for cluster heads and sink nodes in MCEWSNs. This comparison also reveals the advantages of using a multi-core processor over a single-core processor in embedded sensor nodes in terms of performance and performance per watt.

We obtain the power consumption values of the SMPs and TMAs from the devices' respective datasheets and use these values in our power model[8] [7]. For example, the TILEPro64 has maximum active and idle mode power consumptions of 28 W and 5 W, respectively [20][21]. Intel's Xeon E5430 has a maximum power consumption of 80 W and a minimum power consumption of 16 W in an extended HALT state [19][22].

SMP$^{\text{2xQuadXeon}}$'s performance results for the information fusion application are depicted in Table 1, where N = 3,000,000 event-triggered samples and the moving average filter window size is M = 40. T_s and T_p denote the serial and parallel run times, respectively. MOPS denotes Mega operations per second and MOPS/W denotes MOPS per watt. In order to optimize the application to the architecture as much as possible, we used compiler optimization level -O3. As an example, SMP$^{\text{2xQuadXeon}}$ (an eight-core processor) reveals a 4.85x speedup in MOPS as compared to a single-core processor. Additionally, the performance per watt results reveal the multi-core system's power efficiency. As an example, a four-core (p = 4) SMP-based processor attains 49% better performance per watt as compared to a single-core processor. These results verify that SMP-based sensor nodes are more performance- and power-efficient as compared to single-core sensor nodes.

Table 2 depicts the performance results for the information fusion application, obtained with the compiler optimization level -O3, for the TMA-based multi-core processor (TILEPro64) when N = 3,000,000 and M = 40. Results reveal that the TMA-based multi-core processor speeds up the execution time proportionally to the number of tiles p (i.e., ideal speed up) as compared to a comparable single-core processor (i.e., executing the application on a single TMA tile). The efficiency remains close to one and the cost remains nearly constant as the number of tiles increases indicating ideal scalability of the TMA-based multi-core

[7] Section 6 of the supplementary material document presents further details on our experimental setup.

[8] Eq. (1) in the supplementary material document.

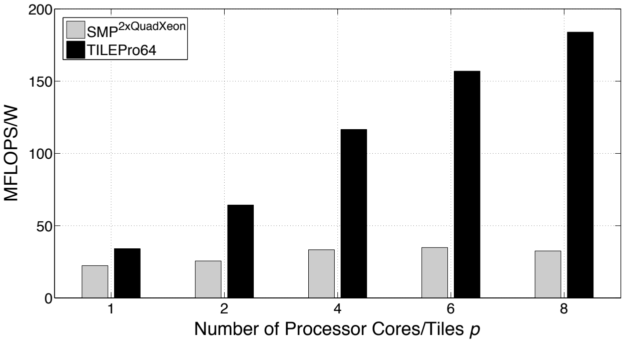


Fig. 2: Performance per watt (MOPS/W) comparison between SMP^{2xQuadXeon} and the TILEPro64 for the information fusion application when N = 3,000,000.

processor for the information fusion application. For example, the TMA-based multi-core processor increases MOPS and MOPS/W by 48.4x and 11.3x, respectively, for p = 50 as compared to a single TMA tile.

These results verify that TMAs provide better performance per watt as compared to a comparable single processor-core architecture. Hence, an embedded sensor node using TMAs as processing units is more performance- and power-efficient as compared to an embedded sensor node using a single-core processing unit.

Fig. 2 compares the SMP^{2xQuadXeon} and the TILEPro64 with respect to performance per watt for a varying number of cores/tiles for the information fusion application. As an example, for an eight-core/tile system, the TILEPro64's performance per watt is 466% higher than the SMP's performance per watt. In summary, results show that the TILEPro64 provides improved performance per watt as compared to the SMP^{2xQuadXeon} mainly due to the fact that the information fusion application operates on private data that can be parallelized using the ilib API. This parallelization exploits high data locality when operating on the sensed data, which enables fast access to private data and results in higher internal memory bandwidth, and thus increased MOPS and MOPS/W.

There are two main reasons why the SMP^{2xQuadXeon} attains lower performance than the TILEPro64 for information fusion. First, shared memory applications are more suited to SMP architectures, which can exploit data locality more effectively. Second, the OpenMP-based parallel programming constructs sections and parallel forces operating threads to share data even if the data can be independently processed by each thread. When we parallelized the information fusion application for the SMP^{2xQuadXeon}, we first tried using independent copies of the data for each thread, similarly to the TILEPro64, however, this introduced large memory requirements and subsequently segmentation faults. Therefore, we were forced to store the data in shared memory since OpenMP currently does not support specifying private data for individual threads, even though private data can be indicated for all the parallel computation threads. Consequently, inherent OpenMP limitations that preclude the declaration of thread-specific private data partially accounts for the SMP's lower performance. On the contrary, Tilera's ilib API permits ideal data distribution for the information fusion application (i.e., data that is received from the first source is only private to the first thread, and the other threads have no information on this data, data that is received from the second source is only private to the second thread, and so forth).

5 Research Challenges and Future Research Directions

Despite few initiatives towards MCEWSNs, the domain is still in its infancy and requires addressing some challenges to facilitate ubiquitous deployment of MCEWSNs. In this section, we discuss several research challenges and future research directions for MCEWSNs.

Application Parallelization: Parallelization of existing serial applications and algorithms can be challenging considering the limited number of parallel programmers as compared to serial programmers. Parallel applications with limited scalability present challenges for efficient utilization of multi-core and future many-core embedded sensor nodes. Furthermore, synchronization between different cores by the use of barriers and locks limit the attainable speedup from parallel applications. A poor speedup due to limited scalability as the number of cores increases can diminish the energy and performance benefits attained by parallelization of sensor applications. To minimize potential performance degradation for parallel applications with limited scalability, designers can restrict these applications to a limited number of cores while turning off remaining cores to save power or utilizing other cores by multiprogramming other sensor applications on those cores. Consequently, existing operating systems for embedded sensor nodes (e.g., TinyOS [23], MANTIS [24]) would require updating their schedulers for efficient scheduling of multi-programmed workloads and would also require some middleware support (e.g., OpenMP) to support multi-threading of parallel applications.

Signal Processing & Computer-Vision: Advances in sensor technology have led to a dramatic increase in the amount of data sensed, which is fueled by both the reduced cost of sensors and increased deployment over a large class of applications. This sensed data deluge problem exacerbates for MCEWSNs and places immense stress on our ability to process, store, and obtain meaningful information from the data. The fundamental reason behind the data deluge problem comes from sensor designs that are based on the Nyquist sampling theorem [25], which has been the dogma in traditional signal processing. However, as we build sensors and sensing platforms with increasing capabilities (e.g., MCEWSNs involving hyper-spectral imaging), designs based on Nyquist sampling are prohibitively costly because of high-resolution sensors and extremely fast data processing requirements. The failure of Nyquist sampling lies in its inability to exploit redundant structures in signals. This redundancy and compressibility in signals forms the basis of Fourier and wavelet transforms. Research in sensing and processing systems that exploit the redundant structures in signals include sparse models, union-of-subspace models, and low-dimensional manifold models. The data deluge problem in MCEWSNs can be addressed in three fundamental ways: 1) parsimonious signal representations that facilitate efficient processing of visual signals; 2) novel compressive and computational imaging systems for sensing of data; and 3) scalable algorithms for large scale machine learning systems. These novel techniques to address the data deluge problem in MCEWSNs requires further research.

Another related research avenue for MCEWSNs is compressive sensing for high-dimensional visual signals, which requires sensors with capabilities that go beyond sensing two-dimensional (2D) images. Examples of these novel sensors include the Lytro camera for sensing light fields [26], the Kinect system that provides scene depth [27], and flexible camera-arrays that provide unique tradeoffs in the spatial, temporal, and angular resolutions of the incident light. Design of novel models, sensors, and technologies is imperative to better characterize objects with complex visual properties.

Furthermore, distilling information from a large number of low-resolution video streams obtained from multiple video sensors requires novel algorithms since current computer-vision and signal processing algorithms can only analyze a few high-resolution images.

Reconfigurability: Reconfigurability in MCEWSNs is an important research avenue that would allow the network to adapt to new requirements by integrating code upgrades (e.g., a more efficient algorithm for video compression may be discovered after deployment). Mobility and self-adaptability of embedded sensor nodes requires further research to obtain the desired view of the sensor field (e.g., an image sensor facing downward towards the earth may not be desirable).

Energy Harvesting: Considering that the battery energy is the most critical resource constraint for sensor nodes in MCEWSNs, research and development in energy-efficient batteries and energy-harvesting systems would be beneficial for MCEWSNs.

Near-Threshold Computing (NTC): NTC refers to using a supply voltage (V_{DD}) that is close to a single transistor's threshold voltage V_t (generally V_{DD} is slightly above V_t in near-threshold operation whereas V_{DD} is below V_t for sub-threshold operation). Lowering the supply voltage reduces power consumption and increases energy efficiency by lowering the energy consumed per operation. With the advent of MCEWSNs leveraging many-core chips, sub- or near-threshold designs become a natural fit for these highly parallel architectures. Considering the stringent power constraints of the many-core chips leveraged in MCEWSNs, sub- or near-threshold designs may be the only practical way to power up all of the cores in these chips [28]. Hence, NTC provides a promising solution for the *dark silicon* problem (transistor under-utilization) in many-core architectures. However, widespread adoption of NTC in MCEWSNs for reduced power consumption requires addressing NTC challenges such as increased process, voltage, and temperature variations, subthreshold leakage power, and soft error rates.

Heterogeneous Architectures: MCEWSNs would benefit from parallel computer architecture research. Specifically, a heterogeneous many-core architecture that could leverage both super- and near-threshold computing to meet performance and energy requirements of sensing applications might provide a promising solution for MCEWSNs. The heterogeneous architecture can integrate super-threshold (nominal voltage) SMP cores and near-threshold single instruction multiple data (SIMD) cores [29]. Research indicates that a combination of NTC and parallel SIMD computations achieves excellent energy efficiency for easy-to-parallelize applications [30]. With this heterogeneous architecture, sensing applications' tasks with less parallelism can be scheduled to high-power SMP cores whereas tasks with abundant parallelism will benefit from scheduling on low-power near-threshold SIMD cores. Hence, research in heterogeneous architectures would enable a single architecture to serve a broad range of sensing applications with varying degrees of parallelism.

Transistor Technology: With ongoing technology scaling, conventional planar CMOS devices suffer from increasing susceptibility to numerous variations, such as circuit performance, short channel effects, delay, or leakage. Research in novel transistor technologies that improve the energy efficiency, provide better resistance to process variation, and are amenable for nano-scale fabrication would benefit sensor nodes in MCEWSNs. One of the promising transistor technologies for future process nodes (22nm and below) is FinFET, in which the channel is a slab (fin) of undoped silicon perpendicular to the substrate [31]. The increased electrostatic control of the FinFET gate over the channel enables high on-current to off-current ratio, which improves carrier mobility, and is promising for near-threshold low-power designs. Other advantages of FinFET over planar CMOS include reduced random dopant fluctuations, lower parasitic junction capacitance, suppression of short channel effects, leakage currents, and parametric variations. However, the widespread transition to FinFET requires further research in prediction models for performance, energy, and process variation for this transistor technology as well as a complete overhaul of the current fabrication process.

6 Conclusions

In this paper, we proposed an architecture for heterogeneous hierarchical multi-core embedded wireless sensor networks (MCEWSNs). Compute-intensive tasks such as information fusion, encryption, network coding, and software defined radio, will benefit in particular from the increased computational power offered by multi-core embedded sensor nodes. Many wireless sensor networking application domains, such as wireless video sensor networks, wireless multimedia sensor networks, satellite-based sensor networks, space shuttle sensor networks, aerial-terrestrial hybrid sensor networks, and fault-tolerant sensor networks, can benefit from MCEWSNs. Perceiving the potential benefits of MCEWSNs, several initiatives have been undertaken in both academia and industry to develop multi-core embedded sensor nodes, such as InstraNode, satellite-based sensor nodes, and smart camera motes.

This paper evaluated two multi-core architectures, symmetric multiprocessors (SMPs) and tiled many-core architectures (TMAs), for multi-core embedded sensor nodes in an MCEWSN based on a parallelized information fusion application. Results revealed that the TILEPro64 exhibited better scalability and attained better performance per watt than the SMPs for the information fusion application. We further highlighted the research challenges and future research avenues for MCEWSNs. Specifically, MCEWSNs would benefit from advancements in application parallelization, signal processing, computer-vision, reconfigurability, energy harvesting, near-threshold computing, heterogeneous architectures, and transistor technology.

References

[1] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, “Wireless Multimedia Sensor Networks: Applications and Testbeds,” *Proceedings of the IEEE*, vol. 96, no. 10, pp. 1588–1605, October 2008.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless Sensor Networks: A Survey,” *Elsevier Computer Networks*, vol. 38, no. 4, pp. 393–422, March 2002.

[3] Y. Liu and S. K. Das, “Information-Intensive Wireless Sensor Networks: Potential and Challenges,” *IEEE Communications Magazine*, vol. 44, no. 11, pp. 142–147, November 2006.

[4] Rockwell, “Rockwell Automation,” October 2011. [Online]. Available: www.rockwellautomation.com

[5] T. T.-O. Kwok and Y.-K. Kwok, “Computation and Energy Efficient Image Processing in Wireless Sensor Networks Based on Reconfigurable Computing,” in *Proc. of the International Conference on Parallel Processing Workshops (ICPPW)*, Columbus, Ohio, August 2006.

[6] R. Kleihorst, B. Schueler, A. Danilin, and M. Heijligers, “Smart Camera Mote with High Performance Vision System,” in *Proc. of the Workshop on Distributed Smart Cameras (DSC)*, Boulder, Colorado, October 2006.

[7] A. Munir, A. Gordon-Ross, and S. Ranka, “Parallelized benchmark-driven performance evaluation of SMPs and tiled multi-core architectures for embedded systems,” in *Proc. of IEEE International Performance Computing and Communications Conference (IPCCC)*, Austin, Texas, December 2012.

[8] TILERA, “Tilera Multicore Development Environment: iLib API Reference Manual,” in *Tilera Official Documentation*, April 2009.

[9] A. Y. Dogan, D. Atienza, A. Burg, I. Loi, and L. Benini, “Power/Performance Exploration of Single-Core and Multi-core Processor Approaches for Biomedical Signal Processing,” in *Proc. of the Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, Madrid, Spain, September 2011.

[10] M. Coates, “Distributed Particle Filters for Sensor Networks,” in *Proc. of the IEEE/ACM 3rd International Symposium on Information Processing in Sensor Networks(IPSN)*, Berkeley, California, April 2004.

[11] R. Rajagopalan and P. Varshney, “Data-Aggregation Techniques in Sensor Networks: A Survey,” *IEEE Communications Surveys & Tutorials*, vol. 8, no. 4, pp. 48–63, 2006.

[12] R. Kulkarni, A. Forster, and G. Venayagamoorthy, “Computational Intelligence in Wireless Sensor Networks: A Survey,” *IEEE Communications Surveys & Tutorials*, vol. 13, no. 1, pp. 68–96, 2011.

[13] W. Li, T. Arslan, J. Han, A. T. Erdogan, A. El-Rayis, N. Haridas, and E. Yang, “Energy Efficiency Enhancement in Satellite Based WSN through Collaboratoin and Self-Organized Mobility,” in *Proc. of the IEEE Aerospace Conference*, Big Sky, Montana, March 2009.

[14] T. Vladimirova, C. Bridges, G. Prassinos, X. Wu, K. Sidibeh, D. Barnhart, A.-H. Jallad, J. Paul, V. Lappas, A. Baker, K. Maynard, and R. Magness, “Characterizing Wireless Sensor Motes for Space Applications,” in *Proc. of the NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, Edinburgh, United Kingdom, August 2007.

[15] V. Lappas, G. Prassinos, A. Baker, and R. Magness, “Wireless Sensor Motes for Small Satellite Applications,” *IEEE Antennas and Propagation Magazine*, vol. 48, no. 5, pp. 175–179, October 2006.

[16] K. Champaigne, “Wireless Sensor Systems for Near-term Space Shuttle Missions,” in *Proc. of the Conference & Exposition on Structural Dynamics (IMAC)*, 2005. [Online]. Available: http://sem.org/Proceedings/ConferencePapers-Paper.cfm?ConfPapersPaperID=23262

[17] M. Hamdi, N. Boudriga, and M. Obaidat, “Bandwidth-Effecitve Design of a Satellite-Based Hybrid Wireless Sensor Network for Mobile Target Detection and Tracking,” *IEEE Systems Journal*, vol. 2, no. 1, pp. 74–82, March 2008.

[18] W. Ye, F. Silva, A. DeSchon, and S. Bhatt, “Architecture of a Satellite-Based Sensor Network for Environmental Observation,” in *Proc. of the Earth Science Technology Conference (ESTC)*, Adelphi, Maryland, June 2008.

[19] Intel, “Intel Xeon Processor E5430,” April 2011. [Online]. Available: http://ark.intel.com/Product.aspx?id=33081

[20] J. Williams, C. Massie, A. George, J. Richardson, K. Gosrani, and H. Lam, “Characterization of Fixed and Reconfigurable Multi-Core Devices for Application Acceleration,” *ACM Trans. on Reconfigurable Technology and Systems*, vol. 3, no. 4, November 2010.

[21] TILERA, “TILEmPower Appliance User’s Guide,” in *Tilera Official Documentation*, January 2010.

[22] Intel, “Quad-Core Intel Xeon Processor 5400 Series Datasheet,” August 2008. [Online]. Available: http://www.intel.com/assets/PDF/datasheet/318589.pdf

[23] TinyOS, “TinyOS,” June 2013. [Online]. Available: http://www.tinyos.net/

[24] MANTIS, “MANTIS — MultimodAl NeTworks of In-situ Sensors,” June 2013. [Online]. Available: http://mantisos.org/index/tiki-index.php.html

[25] A. C. Sankaranarayanan, C. Studer, and R. G. Baraniuk, “CS-MUVI: Video Compressive Sensing for Spatial-Multiplexing Cameras,” in *Proc. of the IEEE International Conference on Computational Photography (ICCP)*, Seattle, Washington, April 2012.

[26] LYTRO, “LYTRO — Light Field Camera,” August 2013. [Online]. Available: http://www.lytro.com/camera/

[27] Kinect, “Kinect,” August 2013. [Online]. Available: http://en.wikipedia.org/wiki/Kinect

[28] B. H. Calhoun and D. Brooks, “Can Subthreshold and Near-Threshold Circuits Go Mainstream?” *IEEE Micro*, vol. 30, no. 4, pp. 80–85, August 2010.

[29] S. Seo, R. G. Dreslinski, M. Woh, C. Chakrabarti, S. Mahlke, and T. Mudge, “Diet SODA: A Power-Efficient Processor for Digital Cameras,” in *Proc. of International Symposium on Low Power Electronics and Design (ISLPED)*, Austin, Texas, August 2010.

[30] S. Seo, R. G. Dreslinski, M. Woh, Y. Park, C. Chakrabarti, S. Mahlke, D. Blaauw, and T. Mudge, “Process Variation in Near-Threshold Wide SIMD Architectures,” in *Proc. of Design Automation Conference (DAC)*, San Francisco, California, June 2012.

[31] W. Zhang, J. G. Fossum, L. Mathew, and Y. Du, “Physical Insights Regarding Design and Performance of Independent-Gate FinFETs,” *IEEE Transactions on Electronic Devices (T-ED)*, vol. 52, no. 10, pp. 2198–2206, October 2005.

Arslan Munir received his B.S. in Electrical Engineering from the University of Engineering and Technology (UET), Lahore, Pakistan, in 2004, and his M.A.Sc. degree in Electrical and Computer Engineering (ECE) from the University of British Columbia (UBC), Vancouver, Canada, in 2007. He received his Ph.D. degree in ECE from the University of Florida (UF), Gainesville, Florida, USA, in 2012. He is currently a postdoctoral research associate in the ECE department at Rice University, Houston, Texas, USA. From 2007 to 2008, he worked as a software development engineer at Mentor Graphics in the Embedded Systems Division. He was the recipient of many academic awards including the Gold Medals for the best performance in Electrical Engineering, academic Roll of Honor, and doctoral fellowship from Natural Sciences and Engineering Research Council of Canada (NSERC). He received a Best Paper award at the IARIA International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM) in 2010. His current research interests include embedded systems, cyber-physical/transporation systems, low-power design, computer architecture, multi-core platforms, parallel computing, dynamic optimizations, fault-tolerance, and computer networks.

Ann Gordon-Ross received her B.S and Ph.D. degrees in Computer Science and Engineering from the University of California, Riverside (USA) in 2000 and 2007, respectively. She is currently an Associate Professor of Electrical and Computer Engineering at the University of Florida (USA) and is a member of the NSF Center for High Performance Reconfigurable Computing (CHREC) at the University of Florida. She is also the faculty advisor for the Women in Electrical and Computer Engineering (WECE) and the Phi Sigma Rho National Society for Women in Engineering and Engineering Technology. She received her CAREER award from the National Science Foundation in 2010 and Best Paper awards at the Great Lakes Symposium on VLSI (GLSVLSI) in 2010 and the IARIA International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM) in 2010. Her research interests include embedded systems, computer architecture, low-power design, reconfigurable computing, dynamic optimizations, hardware design, real-time systems, and multi-core platforms.

Sanjay Ranka is a Professor in the Department of Computer Information Science and Engineering at University of Florida. His current research interests are energy efficient computing, high performance computing, data mining and informatics. Most recently he was the Chief Technology Officer at Paramark where he developed real-time optimization software for optimizing marketing campaigns. Sanjay has also held positions as a tenured faculty positions at Syracuse University and as a researcher/visitor at IBM T.J. Watson Research Labs and Hitachi America Limited.

Sanjay earned his Ph.D. (Computer Science) from the University of Minnesota and a B. Tech. in Computer Science from IIT, Kanpur, India. He has coauthored two books: Elements of Neural Networks (MIT Press) and Hypercube Algorithms (Springer Verlag), 75 journal articles and 125 refereed conference articles. His recent work has received a student best paper award at ACM-BCB 2010, best paper runner up award at KDD-2009, a nomination for the Robbins Prize for the best paper in journal of Physics in Medicine and Biology for 2008, and a best paper award at ICN 2007.

He is a fellow of the IEEE and AAAS, and a member of IFIP Committee on System Modeling and Optimization. He is the associate Editor-in-Chief of the Journal of Parallel and Distributed Computing and an associate editor for IEEE Transactions on Parallel and Distributed Computing, IEEE Transactions on Computers, Sustainable Computing: Systems and Informatics, Knowledge and Information Systems, and International Journal of Computing.

Supplementary Material for “Multi-core Embedded Wireless Sensor Networks: Architecture and Applications”

Arslan Munir, *Member, IEEE,* Ann Gordon-Ross, *Member, IEEE,* and Sanjay Ranka, *Fellow, IEEE, AAAS*

Abstract—Technological advancements in the silicon industry, as predicted by Moore’s law, have enabled integration of billions of transistors on a single chip. To exploit this high transistor density for high performance, embedded systems are undergoing a transition from single-core to multi-core. Although a majority of embedded wireless sensor networks (EWSNs) consist of single-core embedded sensor nodes, multi-core embedded sensor nodes are envisioned to burgeon in selected application domains that require complex in-network processing of the sensed data. In this paper, we propose an architecture for heterogeneous hierarchical multi-core embedded wireless sensor networks (MCEWSNs) as well as an architecture for multi-core embedded sensor nodes used in MCEWSNs. We elaborate several compute-intensive tasks performed by sensor networks and application domains that would especially benefit from multi-core embedded sensor nodes. This paper also investigates the feasibility of two multi-core architectural paradigms—symmetric multiprocessors (SMPs) and tiled many-core architectures (TMAs)—for MCEWSNs. We compare and analyze the performance of an SMP (an Intel-based SMP) and a TMA (Tilera’s TILEPro64) based on a parallelized information fusion application for various performance metrics (e.g., runtime, speedup, efficiency, cost, and performance per watt). Results reveal that TMAs exploit data locality effectively and are more suitable for MCEWSN applications that require integer manipulation of sensor data, such as information fusion, and have little or no communication between the parallelized tasks. To demonstrate the practical relevance of MCEWSNs, this paper also discusses several state-of-the-art multi-core embedded sensor node prototypes developed in academia and industry. We further discuss research challenges and future research directions for MCEWSNs.

Index Terms—wireless sensor networks, multi-core, embedded systems, symmetric multiprocessors, tiled many-core architecture

1 Introduction

This document presents additional details supplementing our IEEE Transactions on Parallel and Distributed (TPDS) paper with the title *“Multi-core Embedded Wireless Sensor Networks: Architecture and Applications”*.

Advancements in silicon technology, embedded systems, sensors, micro-electro-mechanical systems, and wireless communications have led to the emergence of embedded wireless sensor networks (EWSNs). EWSNs consist of sensor nodes with embedded sensors to sense data about a phenomenon and these sensor nodes communicate with neighboring sensor nodes over wireless links (we refer to wireless sensor networks (WSNs) as EWSNs since sensor nodes are embedded in the physical environment/system). EWSNs have applications in various domains, including surveillance, environment monitoring, traffic monitoring, volcano monitoring, and health care.

Processing and transmission of the large amount of sensed data in emerging applications exceeds the capabilities of traditional EWSNs. For example, consider a military EWSN deployed in a battlefield, which requires various sensors, such as imaging, acoustic, and electromagnetic sensors. In this application, images are appropriate for visually monitoring the battlefield, and electromagnetic and acoustic sensors enable efficient detection and tracking of targets of interest. Once a target is detected, high resolution images and/or video sequences may be required in real-time for detailed study of the target [1]. This application presents various challenges for existing EWSNs since transmission of high-resolution images and video streams over bandwidth-limited wireless links from sensor nodes to the sink node is infeasible. Furthermore, meaningful processing of multimedia data (acoustic, image, and video in this example) in real-time exceeds the capabilities of traditional EWSNs consisting of single-core embedded sensor nodes [2][3], and requires more powerful embedded sensor nodes to realize this application.

Technological advancements in multi-core architectures have made multi-core processors a viable and cost-effective choice for increasing the computational ability of embedded sensor nodes. Preliminary studies have demonstrated the energy-efficiency of multi-core embedded sensor nodes as compared to single-core embedded sensor nodes in an EWSN. For example, Dogan et al. [4] evaluated single-

- *Arslan Munir is with the Department of Electrical and Computer Engineering, Rice University, Houston, Texas, USA. Ann Gordon-Ross is with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA. Ann Gordon-Ross is also affiliated with the NSF Center for High-Performance Reconfigurable Computing (CHREC) at the University of Florida. Sanjay Ranka is with the Department of Computer and Information Science and Engineering at the University of Florida. e-mail: {arslan@rice.edu, ann@ece.ufl.edu, ranka@cise.ufl.edu}*

and multi-core architectures for biomedical signal processing in wireless body sensor networks (WBSNs) where both energy-efficiency and real-time processing are crucial design objectives. Results revealed that the multi-core architecture consumed 66% less power than the single-core architecture for high biosignal computation workloads (i.e., 50.1 Mega operations per seconds (MOPS)) whereas the multi-core architecture consumed 10.4% more power than that of the single-core architecture for relatively light computation workloads (i.e., 681 Kilo operations per second (KOPS)).

This supplementary material document is organized as follows. Section 2 proposes a multi-core embedded sensor node architecture for multi-core embedded wireless sensor networks (MCEWSNs). Section 3 discusses multi-core architectures for multi-core embedded sensor nodes and parallel computing metrics that we use to evaluate these architectures. Section 4 elaborates on several compute-intensive tasks that motivated the emergence of MCEWSNs. Section 5 discusses several prototypes of multi-core embedded sensor nodes. Experimental setup details for the information fusion application are presented in Section 6.

2 Multi-core Embedded Sensor Node Architecture

Fig. 1 depicts the architecture of a multi-core embedded sensor node in our MCEWSN. The multi-core embedded sensor node consists of a sensing unit, a processing unit, a storage unit, a communication unit, a power unit, an optional actuator unit, and an optional location finding unit (optional units are represented by dotted lines in Fig. 1) [2].

2.1 Sensing Unit

The sensing unit senses the phenomenon of interest and is composed of two subunits: sensors (e.g., camera/image, audio, and scalar sensors (e.g., temperature, pressure)) and analog-to-digital converters (ADCs). Image sensors can either leverage traditional charge-coupled device (CCD) technology or complementary metal-oxide-semiconductor (CMOS) imaging technology. The CCD sensor accumulates the incident light energy as the charge accumulated on a pixel, which is then converted into an analog voltage signal. In CMOS imaging technology, each pixel has its own charge-to-voltage conversion and other processing components, such as amplifiers, noise correction, and digitization circuits. The CMOS imaging technology enables integration of the lens, an image sensor, and image compression and processing technology on a single chip. ADCs convert the analog signals produced by sensors to digital signals, which serve as input to the processing unit.

2.2 Processing Unit

The processing unit consists of a multi-core processor and is responsible for controlling sensors, gathering and processing sensed data, executing the system software that coordinates sensing, communication tasks, and interfacing with the storage unit. The processing unit for traditional sensor nodes consists of a single-core processor for general-purpose applications, such as periodic sensing of scalar data (e.g., temperature, humidity). High-performance single-core processors would be infeasible to meet computational requirements since these single-core processors would require operation at high processor voltage and frequency. A processor operating at a high voltage and frequency consumes an enormous amount of power since power increases proportionally to the operating processor frequency and square of the operating processor voltage. Furthermore, even if these energy issues are ignored, a single high-performance processor core may not be able to meet the computational requirements of emerging applications, such as multimedia sensor networks, in real-time.

Multi-core processors distribute the computations across the available cores, which speeds up the computations as well as conserves energy by allowing each processor core to operate at a lower processor voltage and frequency. Multi-core processors are suitable for streaming and complex, event-based monitoring applications, such as in smart camera sensor networks, that require data to be processed and compressed as well as require extraction of key information features. For example, the IC3D/Xetal single-instruction multiple-data (SIMD) processor, which consists of a linear processor array (LPA) with 320 reduced instruction set computers (RISC)/processors, is being used in smart camera sensor networks [5].

2.3 Storage Unit

The storage unit consists of the memory subsystem, which can be classified as *user memory* and *program memory*, and a memory controller, which coordinates memory accesses between different processor cores. The user memory stores sensed data when immediate data transmission is not possible due to hardware failures, environmental conditions, physical layer jamming, limited energy reserves, or when the data requires processing. The program memory is used for programming the embedded sensor node and using flash memory for the program memory provides persistent storage of application code and text segments. Static random-access memory (SRAM), which does not need periodic refreshing but is expensive in terms of area and power consumption, is used as dedicated processor memory. Synchronous dynamic random-access memory (SDRAM) is typically used as user memory. For example, the Imote2 embedded sensor node, which contains a Marvell PXA271 XScale processor operating

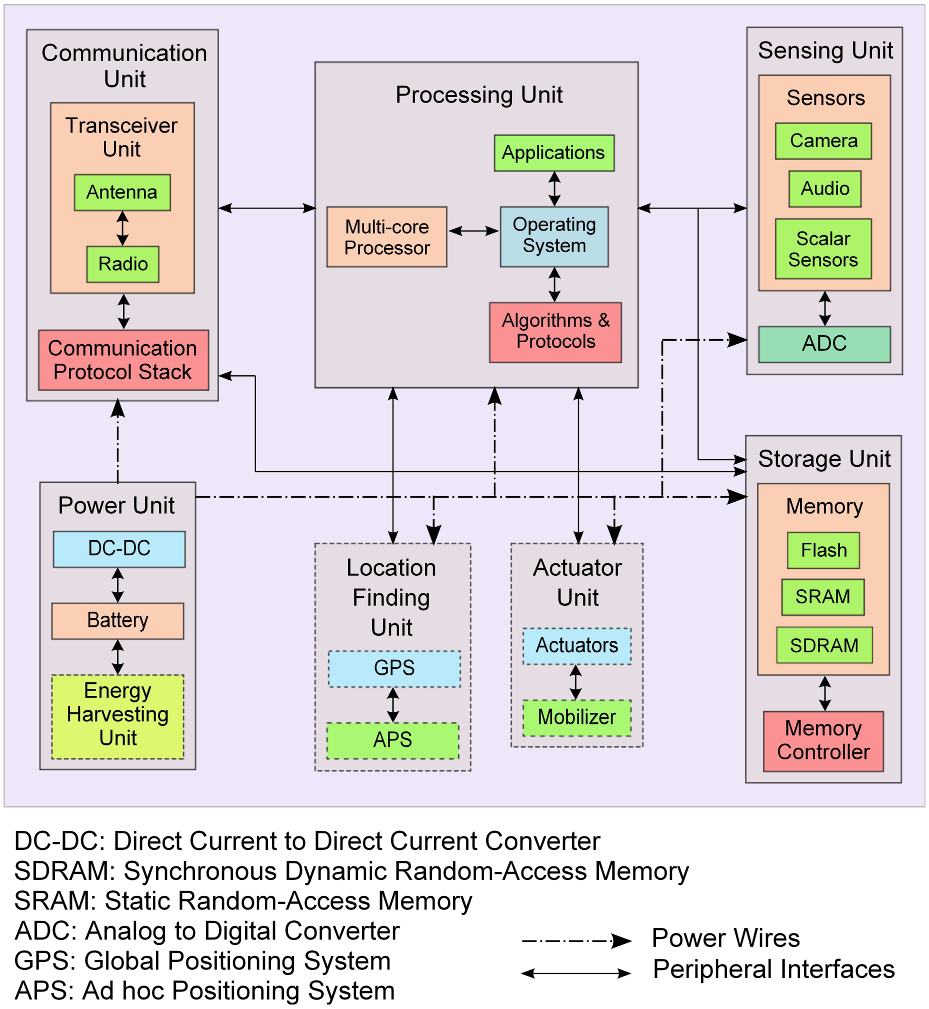


Fig. 1: Multi-core embedded sensor node architecture.

between 13 and 416 Mhz, has 256 KB SRAM, 32 MB Flash, and 32 MB SDRAM [6].

2.4 Communication Unit

The communication unit interfaces the embedded sensor node to the wireless network and consists of a transceiver unit (transceiver and antenna) and the communication unit software. The communication unit software mainly consists of the communication protocol stack, and the physical layer software in the case of software defined radio (SDR). The transceiver unit consists of either a wireless local area network (WLAN) card, such as an IEEE 802.11b compliant card, or an IEEE 802.15.4 compatible card, such as a Texas Instrument/Chipcon CC2420 chipset. The choice of a transceiver unit card depends on the application requirements such as desired range and allowable power. The maximum transmit power of IEEE 802.11b cards is higher as compared to IEEE 802.15.4 cards, which results in a higher communication range but consumes more power. For example, the Intel PRO/Wireless 2011 card has a data rate of 11 Mbps and a typical transmit power of 18 dBm, but draws 300 mA and 170 mA for sending and receiving, respectively. The CC2420 802.15.4 radio has a maximum data rate of 250 kbps and a transmit power of 0 dBm, but draws 17.4 mA and 19.7 mA for sending and receiving, respectively.

2.5 Power Unit

The power unit supplies power to various components/units on the embedded sensor node and dictates the sensor node's lifetime. The power unit consists of a battery and a DC-DC converter. The DC-DC converter provides a constant supply voltage to the sensor node. The power unit may be augmented by an optional energy-harvesting unit that derives energy from external sources, such as solar cells. Although multi-core embedded sensor nodes are more power efficient as compared to single-core embedded sensor nodes, energy-harvesting units in multi-core cluster heads and the sink node would prolong the MCEWSN's lifetime. Energy-harvesting units are more suitable for cluster heads and the sink node as these nodes perform more computations as compared to the single-core leaf sensor nodes. Furthermore, incorporating energy-harvesting units in only a few embedded sensor nodes (i.e., cluster heads and sink nodes) would not substantially increase the cost of EWSN deployment. Without an energy-harvesting unit, MCEWSNs would only be suitable for applications with relatively small lifetime requirements.

2.6 Actuator Unit

The optional actuator unit consists of actuators (e.g., motors, servos, linear actuators, air muscles, muscle wire, camera pan tilt, etc.) and an optional mobilizer unit for sensor node mobility. Actuators enhance the sensing task by opening/closing a switch/relay to control functions, such as a camera or antenna orientation and repositioning sensors. Actuators, in contrast to sensors that only sense a phenomenon, typically affect the operating environment by opening a valve, emitting sound, or physically moving the sensor node.

2.7 Location Finding Unit

The optional location finding unit determines a sensor node's location. Depending on the application requirements and available resources, the location finding unit can either be global positioning system (GPS)-based or ad hoc positioning system (APS)-based. Even though GPS is highly accurate, the GPS components are expensive and require direct line of sight between the sensor node and satellites. APS determines a sensor node's position with respect to defined *landmarks*, which may be other GPS-based sensor nodes [7]. A sensor node estimates the distance from itself to the landmark based on direct communication and the received communication signal strength. A sensor node that is two hops away from a landmark estimates its distance based on the distance estimate of a sensor node one hop away from a landmark via the message propagation. A sensor node with distance estimates to three or more landmarks can compute its own position via triangulation.

3 Multi-core Architectures and Parallel Computing Metrics

In this section, we describe the multi-core architectures that we evaluate in our study as well as parallel computing metrics that we leverage for this evaluation.

3.1 Multi-core Architectures

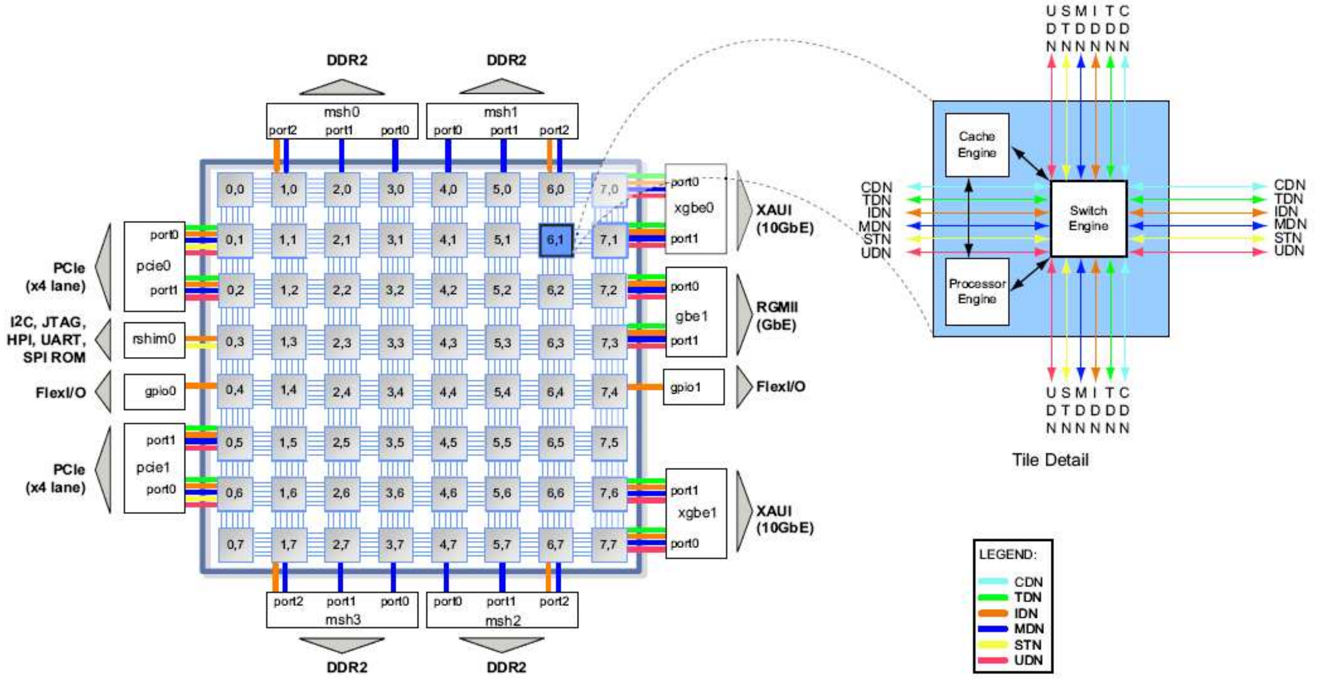
In this subsection, we give an overview of the two multi-core architectures that can be used as processing units in multi-core embedded sensor nodes (Fig. 1). We note that the operating frequency of the studied multi-core architectures is much higher than the ones that can be used for multi-core embedded sensor nodes. However, our purpose in this paper is to evaluate the architectural paradigms' feasibility for multi-core embedded sensor nodes and a lower operating frequency of the studied architectures in real multi-core embedded sensor nodes would only scale down the presented results without any significant changes to the performance trends. Hence, leveraging high computing power SMPs and TMAs will not affect the feasibility insights obtained from benchmark-driven cross-architectural evaluation, which is the intent of this work.

3.1.1 Symmetric Multiprocessors (SMPs)

In the parallel architecture domain, SMPs are the most pervasive and prevalent type, and are therefore an ideal processor candidate for MCEWSNs. SMPs offer a global physical address space, provide symmetric access to main memory, and have private caches. The processors and memory modules communicate over a shared interconnect, the most common being a shared bus [8]. We evaluate an eight-core Intel-based SMP consisting of two Intel Xeon E5430 quad-core processors fabricated at 45nm CMOS lithography [9] with a maximum clock frequency of 2.66 GHz. Each core contains 32 KB of level one instruction (L1-I) cache, 32 KB of level one data (L1-D) cache, and 12 MB of level two (L2) unified cache. Intel's enhanced front-side bus (FSB) running at 1333 MHz provides enhanced inter-core communication throughput [10]. For conciseness, we will refer to this SMP as $\mathrm{SMP}^{\mathrm{2xQuadXeon}}$ in the remainder of this paper.

3.1.2 Tiled Many-Core Architectures (TMAs)

TMAs are constructed using modular elements—*tiles*—which provides easy scalability to any arbitrary number of tiles. For intra-tile communication, each tile connects to a switch (communication router) within a high-performance interconnection network and each switch connects to a neighboring switch, which constrains the interconnection wire length to be no longer than the tile width. Examples of TMAs include the Intel's Tera-Scale research processor, the Raw processor, and Tilera's TILE*Pro*64 and TILE-Gx processor family [11][12]. Fig. 2 depicts our evaluated TMA, which is Tilera's TILEPro64 processor, fabricated at 90nm CMOS lithography and consists of 64 tiles (cores) in an 8 x 8 grid. Each tile has a three-way very long instruction word (VLIW) pipelined processor, which can execute up to three instructions per cycle (IPC). The switches are non-blocking, which provides a power-efficient on-chip interconnection mesh network operating at 31 Tbps. Each tile has 8 KB of L1-I cache, 8 KB of L1-D cache, and 64 KB of L2 cache, collectively providing 5 MB of on-chip cache with Tilera's *dynamic distributed cache (DDC)* technology. An operating system (OS) can be run independently on each tile or the tiles can be grouped to run a multi-processing OS (e.g., SMP Linux [13]). The TILEPro processor family is suitable for a variety of application domains, such as advanced networking, wireless infrastructure, telecom, digital multimedia, and cloud computing. Our prior work [14] provides further details on TMAs.



DDR2: Double Data Rate 2 Synchronous Dynamic Random-Access Memory (SDRAM)

PCIe: Peripheral Component Interconnect Express

XAUI: X (Ten) Attachment Unit Interface

JTAG: Joint Test Action Group

CDN: Coherence Dynamic Network

TDN: Tile Dynamic Network

IDN: I/O Dynamic Network

MDN: Memory Dynamic Network

STN: Static Network

UDN: User Dynamic Network

Fig. 2: Tilera's TILEPro64 processor (adapted from [15]).

3.2 Parallel Computing Device Metrics

In this section, we define the metrics used to quantitatively compare our investigated multi-core architectures.

Run Time: The *serial run time* T_s of a program is the time required to execute the program on a sequential computer. The *parallel run time* T_p is the time elapsed from the start of a program to the moment the last processor finishes execution.

Speedup: The speedup S measures the performance gain achieved via application parallelization as compared to the execution time of the best sequential implementation of the application. S is defined as the ratio of the *serial run time* T_s to the *parallel run time* T_p to solve the same problem (i.e., $S = T_s/T_p$).

Efficiency: The fraction of time a processor is actively executing an application is the system's efficiency E. E is computed as the ratio of the speedup S to the number of processors p (i.e., $E = S/p$).

Cost: The collective processor time required to execute an application in a parallel system is the system's cost C. C on a parallel system is computed as the product of the parallel run time T_p and the number of processors p (i.e., $C = T_p \cdot p$). A parallel system is *cost optimal* if the parallel system's cost is proportional to the execution time of the best known sequential algorithm on a single processor [16].

Scalability: A parallel system's scalability evaluates the efficiency of application parallelization as the number of processors increases, wherein an optimally-scalable parallel system maintains a speedup increase proportional to the increase in the number of processors and the problem size [16].

Power: A processor's total (system-level) power consumption comprises both the dynamic and static power consumptions. The dynamic power consumption depends on the supply voltage, clock frequency, capacitance, and the signal activity whereas the static power consumption mainly depends on the supply voltage, temperature, and capacitance [17]. Our system-level power model estimates a multi-core system's power consumption, and considers both the active and the idle mode power consumptions. Our power estimation model can be used to estimate the system's performance per watt. The power consumption of a system with N processor cores and p active processor cores is:

$$P^p = p \cdot \frac{P^{active}_{max}}{N} + (N - p) \cdot \frac{P^{idle}_{max}}{N} \quad (1)$$

where P^{active}_{max} and P^{idle}_{max} denote the system's maximum active and idle mode power consumptions, respectively. P_{max}^{active}/N and P_{max}^{idle}/N give the active and idle mode power consumptions per core (and the associated switching and interconnection network circuitry), respectively. We consider state-of-the-art power saving mechanisms, such as instructions to switch idle cores and associated circuitry (switches, clock, interconnection network) into a low-power idle state (e.g., Tilera's `NAP` instruction puts a tile into a low-power `IDLE` mode [18]).

Performance per Watt: Performance per watt evaluates a device's delivered/attainable performance while taking the device's power consumption into consideration. We report performance with respect to MOPS or Mega floating point operations per second (MFLOPS), and performance per watt with respect to MOPS per watt (MOPS/W) or MFLOPS per watt (MFLOPS/W).

4 Compute-Intensive Tasks Motivating the Emergence of MCEWSNs

Many applications require embedded sensor nodes to perform various compute-intensive tasks that often exceeds the computing capability of traditional single-core sensor nodes. These tasks include information fusion, encryption, network coding, software defined radio, etc., and motivate the emergence of MCEWSNs. In this section, we discuss these compute-intensive tasks requiring multi-core support in an embedded sensor node.

4.1 Information Fusion

A critical processing task in EWSNs is information fusion, which can benefit from a multi-core processor in an embedded sensor node. EWSNs produce a large amount of data that must be processed, delivered, and assessed according to application objectives. Since the transmission bandwidth is limited, information fusion condenses the sensed data and transmits only the selected fused information to the sink node. Additionally, the data received from neighboring sensor nodes is often redundant and highly correlated, which warrants fusing the sensed data. Formally, information fusion encompasses theory, techniques, and tools created and applied to exploit the synergy in the information acquired from multiple sources (sensors, databases, etc.) such that the resulting fused data/information is considered qualitatively or quantitatively better in terms of accuracy or robustness than the acquired data from any single data source [19]. Data aggregation is an instance of information fusion in which the data from various sources is aggregated using summarization functions (e.g., minimum, maximum, and average) that reduce the volume of data being manipulated. Information fusion can reduce the amount of data traffic, filter noisy measurements, and make predictions and inferences about a monitored entity.

Information fusion can be computationally expensive, especially for video sensing applications. Unlike scalar data, which can be combined using relatively simple mathematical manipulations such as average and summation, video data is vectorial and requires complex computations to fuse (e.g., edge detection, histogram formation, compression, filtering, etc.). Reducing transmission overhead via information fusion in video sensor networks requires a substantial increase in intermediate processing, which warrants the use of multi-core cluster heads in MCEWSNs. Multi-core cluster heads fuse data received from multiple sensor nodes to eliminate redundant transmission and provide fused information to the sink node with minimum data latency. Data latency is the sum of the delay involved in data transmission, routing, and information fusion/data aggregation [20]. Data latency is important in many applications, especially real-time applications, where freshness of data is an important factor. Multi-core cluster heads can fuse data much faster than single-core sensor nodes, which justifies the use of multi-core cluster heads in MCEWSNs with complex real-time computing requirements.

Omnibus Model for Information Fusion: The Omnibus model [21] guides information fusion for sensor-based devices. Fig. 3 illustrates the Omnibus model with respect to our MCEWSN architecture and we exemplify the model's usage by considering a surveillance application performing target tracking based on acoustic sensors [19]. The *Observe* stage, which can be carried out at single-core sensor nodes and/or multi-core cluster heads, uses a filter (e.g., moving average filter) to reduce noise (*Signal Processing*) from acoustic sensor data provided by the embedded sensor nodes (*Sensing*). The *Orientate* stage, which is carried out at multi-core cluster heads, uses the filtered acoustic data for range estimation (*Feature Extraction*) and estimates the target's location and trajectory (*Pattern Processing*). The *Decide* stage, which is carried out at multi-core cluster heads and/or multi-core sink nodes, classifies the sensed target (*Context Processing*) and determines whether the target represents a threat (*Decision Making*). If the target is a threat, the *Act* stage, which is carried out at the control and analysis center (CAC), intercepts the target (*Control*) (e.g., with a missile) and activates available armaments (*Resource Tasking*).

4.2 Encryption

Security is an important issue in many sensor networking applications since sensors are deployed in open environments and are susceptible to malicious attacks. The sensed and/or aggregated data must be encrypted for secure transmission to the sink node. The two main practical issues involved in encryption are the size of the encrypted message and the encryption execution time. Privacy homomorphisms (PHs) are encryption functions suitable for MCEWSNs that allow

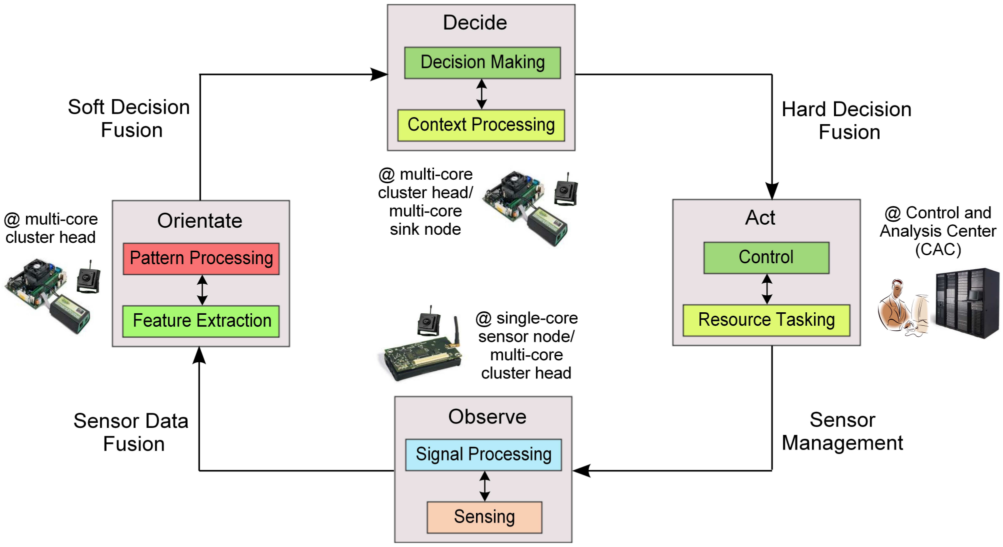


Fig. 3: Omnibus sensor information fusion model for an MCEWSN architecture.

a set of operations to be performed on encrypted data without knowing the decryption functions [20]. PHs use a positive integer $d \geq 2$ for computing the secret key for encryption such that the size of the encrypted data increases by a factor of d as compared to the original data. The security of the encrypted data increases with d as well as the execution time for encryption. For example, the execution time for encryption of one byte of data is 3,481 clock cycles on a MICA2 mote when $d = 2$ and increases to 4,277 clock cycles when $d = 4$. MICA2 motes cannot handle the computations for $d \geq 4$ [20], hence, applications requiring greater security require multi-core sensor nodes and/or cluster heads to perform these computations.

4.3 Network Coding

Network coding is a coding technique to enhance network throughput in multi-nodal environments, such as EWSNs. Despite the effectiveness of network coding for EWSNs, excessive decoding cost associated with network coding hinders the technique's adoption in traditional EWSNs with constrained computing power [22]. Future MCEWSNs will enable adoption of sophisticated coding techniques, such as network coding to increase network throughput.

4.4 Software Defined Radio (SDR)

SDR is a radio in which some or all of the physical layer functions execute as software. The radio in existing EWSNs is hardware-based, which results in higher production costs and minimal flexibility in supporting multiple waveform standards [23]. MCEWSNs can realize SDR-based radio by enabling fast, parallel computation of signal processing operations needed in SDR (e.g., fast Fourier transform (FFT)). SDR-based MCEWSNs would enable multi-mode, multi-band, and multi-functional radios that can be enhanced using software upgrades.

5 Multi-core Embedded Sensor Nodes

Several initiatives towards multi-core embedded sensor nodes have been undertaken by academia and industry for various real-time applications. In this section, we describe several state-of-the-art multi-core embedded sensor node prototypes.

5.1 InstraNode

InstraNode is a dual-core sensor node for real-time health monitoring of civil structures, such as highway bridges and skyscrapers. InstraNode is equipped with a 4000 mAh lithium-ion battery, three accelerometers, a gyroscope, and an IEEE 802.11b (Wi-Fi) card for communication with other nodes. One low-power processor core in InstraNode runs at 3 V and 4 MHz and is dedicated to sampling data from sensors whereas the other faster, high-power processor core runs at 4.3 V and 40 MHz and is responsible for networking tasks, such as transmission/reception of data and execution of a routing algorithm. Furthermore, InstraNode possesses multi-modal operation capabilities such as wired/wireless and battery-powered/AC-adaptor powered options. Experiments indicate that the InstraNode outperforms single-core sensor nodes in terms of power-efficiency and network performance [24].

5.2 Mars Rover Prototype Mote

Etchison et al. [25] have proposed a high-performance EWSN for the Mars Rover, which consists of dual-core mobile sensor nodes and a wireless cluster consisting of multiple processors to process image data gathered from the sensor nodes and to make decisions based on gathered information. The prototype mote consists of a Micro ATX motherboard with Intel's dual-core Atom processor, 2 GB of RAM, and is powered by a 12 V/5 A DC power supply for lab testing. Each mote performs data acquisition, processing, and transmission.

5.3 Satellite-Based Sensor Node (SBSN)

Vladimirova et al. [26] have developed a system-on-chip (SoC) satellite-based sensor node (SBSN). The SBSN prototype contains a SPARC V8 LEON3 soft processor core, which allows configuration in an SMP architecture [27]. The LEON3 processor core runs software applications and interfaces with the upper layers of the communication stack using the IEEE 802.11 protocol. The SBSN prototype uses a number of intellectual property (IP) cores, such as a hardware accelerated Wi-Fi MAC, a transceiver core, and a Java co-processor. The Java co-processor enables distributed computing and Internet protocol (IP)-based networking functions in SBWSNs. The inter-satellite communication module (ISCM) in the SBSN prototype adheres to IEEE 802.11 and CubeSat design specifications. The ISCM supports ground communication links and inter-satellite links (ISLs) at variable data rates and configurable waveforms to adapt to channel conditions. The ISCM incorporates S-band (2.4 GHz) and a 434/144 MHz radio frontend interfaced to a single reconfigurable modem. The ISCM uses a high-end AD9861 ADC/digital-to-analog converter (DAC) for the 2.4 GHz radio frontend for a Maxim 2830 radio and a low-end AD7731 for the 434/144 MHz frontend for an Alinco DJC-7E radio. Additionally, ISCM incorporates current and temperature sensors and a 16-bit microcontroller for housekeeping purposes.

5.4 Multi-CPU-based Sensor Node Prototype

Ohara et al. [28] have developed a prototype for an embedded sensor node using three PIC18 central processing units (CPUs). The prototype is supplied by a configurable voltage stabilized power supply, but the same voltage is supplied to all CPUs. The prototype allowed each CPU's frequency to be statically changed by changing a corresponding ceramic resonator. Experiments revealed that the multi-CPU sensor node prototype consumed 76% less power as compared to a single-core sensor node for benchmarks that involved sampling, root mean square calculation, and pre-processing samples for transmission.

5.5 Smart Camera Mote

Kleihorst et al. [29] developed a smart camera mote, which consists of four basic components: color image sensors, an IC3D SIMD processor (a member of the Philips' Xetal family of SIMD processors) for low-level image processing, a general purpose processor for intermediate and high-level processing and control, and a communication module. Both of the processors are coupled with a dual-port random-access memory (RAM) that enables these processors to work in a shared workspace. The IC3D SIMD processor consists of a linear array of 320 RISC processors. The peak pixel performance of the IC3D processor is approximately 50 Giga operations per second (GOPS). Despite high pixel performance, the IC3D processor is an inherently low-power processor, which makes the processor suitable for multi-core embedded sensor nodes. The power consumption of the IC3D processor for typical applications, such as feature finding or face detection, is below 100 mW in active processing modes.

6 Results

In this section, we describe the information fusion application experimental setup details.[1]

We consider a hierarchical MCEWSN for information fusion such that each cluster head receives sensing measurements from ten single-core sensor nodes equipped with temperature, pressure, humidity, acoustic, magnetometer, accelerometer, gyroscope, proximity, and orientation sensors [30]. To reduce the random white noise from sensor measurements, a moving average filter, which computes the arithmetic mean of a number of input measurements to produce each output measurement, is executed on the cluster head. Given an input sensor measurement vector $\boldsymbol{x} = (x(1), x(2), \ldots)$, the moving average filter estimates the true sensor measurement vector after noise removal $\boldsymbol{y} = (\hat{y}(1), \hat{y}(2), \ldots)$ as:

$$\hat{y}(k) = \frac{1}{M} \sum_{i=0}^{M-1} x(k-i), \ \ \forall k \geq M \tag{2}$$

where the filter window size M indicates the number of fused input sensor measurements. Given sensor measurements with random white noise, the moving average filter reduces the noise variance by a factor of $\sqrt{M}$. M should be chosen such that M is the smallest value that reduces the noise in accordance with the application's requirements. After calculating the cluster's nodes' filtered sensor measurements (i.e., after applying moving average filter) for each of the sensor node in the cluster, the cluster head determines the sensed measurements minimum, maximum, and average values. This information fusion requires $100 \cdot N(3+M)$ operations with a runtime complexity of $\mathcal{O}(NM)$ where N is the number of sensor measurements. Our results evaluate a parallelized information fusion application using our parallel performance metrics (Section 3.2) to illustrate the advantages for leveraging multi-core as compared to single core architectures for cluster heads.

Acknowledgments

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the National Science Foundation (NSF) (CNS-0953447 and CNS-0905308). Any opinions, findings, and conclusions or recommendations expressed in this

[1] Detailed results for the information fusion application are presented in Section 4 of the main paper.

material are those of the author(s) and do not necessarily reflect the views of the NSERC and the NSF.

References

[1] M. Hamdi, N. Boudriga, and M. Obaidat, "Bandwidth-Effecitve Design of a Satellite-Based Hybrid Wireless Sensor Network for Mobile Target Detection and Tracking," *IEEE Systems Journal*, vol. 2, no. 1, pp. 74–82, March 2008.

[2] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "Wireless Multimedia Sensor Networks: Applications and Testbeds," *Proceedings of the IEEE*, vol. 96, no. 10, pp. 1588–1605, October 2008.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," *Elsevier Computer Networks*, vol. 38, no. 4, pp. 393–422, March 2002.

[4] A. Y. Dogan, D. Atienza, A. Burg, I. Loi, and L. Benini, "Power/Performance Exploration of Single-Core and Multi-core Processor Approaches for Biomedical Signal Processing," in *Proc. of the Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, Madrid, Spain, September 2011.

[5] A. Kak, "Parallel Histogram-based Particle Filter for Object Tracking on SIMD-based Smart Cameras," in *Purdue Robot Vision Lab*, Purdue University, West Lafayette, Indiana, October 2011. [Online]. Available: https://engineering.purdue.edu/RVL/Research/SIMD_PF/index.html

[6] MEMSIC, "Imote2 Hardware Bundle for Wireless Sensor Networks," October 2011. [Online]. Available: www.memsic.com

[7] A. Munir and A. Gordon-Ross, "Optimization approaches in wireless sensor networks," in *Sustainable Wireless Sensor Networks*, W. Seah and Y. K. Tan, Eds. InTech, 2010. [Online]. Available: http://www.intechopen.com/articles/show/title/optimization-approaches-in-wireless-sensor-networks

[8] D. Culler, J. Singh, and A. Gupta, *Parallel Computer Architecture: A Hardware/Software Approach*. Morgan Kaufmann Publishers, Inc., 1999.

[9] Intel, "Intel Xeon Processor E5430," April 2011. [Online]. Available: http://ark.intel.com/Product.aspx?id=33081

[10] ——, "Quad-Core Intel Xeon Processor 5400 Series Datasheet," August 2008. [Online]. Available: http://www.intel.com/assets/PDF/datasheet/318589.pdf

[11] S. Keckler, K. Olukotun, and H. Hofstee, *Multicore Processors and Systems*. Springer, 2009.

[12] TILERA, "TILEPro Processor Family," March 2013. [Online]. Available: http://www.tilera.com/products/processors/TILEPro_Family

[13] IBM, "Linux and Symmetric Multiprocessing," May 2011. [Online]. Available: http://www.ibm.com/developerworks/library/l-linux-smp/

[14] A. Munir, F. Koushanfar, A. Gordon-Ross, and S. Ranka, "High-Performance Optimizations on Tiled Many-Core Embedded Systems: A Matrix Multiplication Case Study," *The Journal of Supercomputing*, 2013.

[15] TILERA, "Tile Processor Architecture Overview for the TILEPro Series," in *Tilera Official Documentation*, November 2009.

[16] V. Kumar, A. Grama, A. Gupta, and G. Karypis, *Introduction to Parallel Computing*. The Benjamin/Cummings Publishing Company, Inc., 1994.

[17] E. Musoll, "A Cost-Effective Load-Balancing Policy for Tile-Based, Massive Multi-Core Packet Processors," *ACM Transactions on Embedded Computing Systems*, vol. 9, no. 3, February 2010.

[18] TILERA, "Tile Processor Architecture Overview for the TILEPro Series," in *Tilera Official Documentation*, November 2009.

[19] E. F. Nakamura, A. A. Loureiro, and A. C. Frery, "Information Fusion for Wireless Sensor Networks: Methods, Models, and Classifications," *ACM Computing Surveys*, vol. 39, no. 3, August 2007.

[20] R. Rajagopalan and P. Varshney, "Data-Aggregation Techniques in Sensor Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 8, no. 4, pp. 48–63, 2006.

[21] M. Bedworith and J. O'Brien, "The Omnibus Model: A New Model for Data Fusion?" *IEEE Aerospace and Electronic Systems Magazine*, vol. 15, no. 4, pp. 30–36, April 2000.

[22] D. Kim, K. Park, and W. Ro, "Network Coding on Heterogeneous Multi-Core Processors for Wireless Sensor Networks," *Sensors*, vol. 11, no. 8, pp. 7908–7933, 2011.

[23] G. R. Murthy, "Control, Communication and Computing Units: Converged Architectures," *International Journal of Computer Applications*, vol. 1, no. 4, pp. 49–54, 2010.

[24] C. Park, Q. Xie, and P. Chou, "InstraNode: Dual-Microcontroller Based Sensor Node for Real-Time Structural Health Monitoring," in *Proc. of the IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, Santa Clara, California, September 2005.

[25] J. Etchison, G. Skelton, Q. Pang, and T. Hulitt, "Mobile Intelligent Sensor Network Used for Data Processing," Jackson Sate University, Jackson, Mississippi, 2010. [Online]. Available: http://www.iiis.org/CDs2010/CD2010SCI/SCI_2010/PapersPdf/SA874PZ.pdf

[26] T. Vladimirova, C. Bridges, J. Paul, S. Malik, and M. Sweeting, "Space-based Wireless Sensor Networks: Design Issues," in *Proc. of the IEEE Aerospace Conference*, Big Sky, Montana, March 2009.

[27] Aeroflex, "Leon3 Processor," October 2011. [Online]. Available: http://www.gaisler.com/cms/index.php?option=com_content&task=view&id=13&Itemid=53

[28] S. Ohara, M. Suzuki, S. Saruwatari, and H. Morikawa, "A Prototype of a Multi-Core Wireless Sensor Node for Reducing Power Consumption," in *Proc. of the International Symposium on Applications and the Internet (SAINT)*, Turku, Finland, July 2008.

[29] R. Kleihorst, B. Schueler, A. Danilin, and M. Heijligers, "Smart Camera Mote with High Performance Vision System," in *Proc. of the Workshop on Distributed Smart Cameras (DSC)*, Boulder, Colorado, October 2006.

[30] Android, "SensorEvent," November 2011. [Online]. Available: http://developer.android.com/reference/android/hardware/SensorEvent.html