Waggle: An Open Sensor Platform for Edge Computing

Pete Beckman, Rajesh Sankaran, Charlie Catlett, Nicola Ferrier, Robert Jacob, Michael Papka
Argonne National Laboratory, USA
beckman@anl.gov

***Abstract*—Many advanced sensors are capable of producing extremely large and continuous data streams. Hyperspectral imagers, microphones, high-resolution cameras, and 3D scanning devices can easily generate gigabytes of data per day, making it impractical for many wireless sensor platforms to stream all collected data to the cloud for analysis. Furthermore, in some sensor deployments, privacy concerns may restrict the resolution or content of data leaving the devices and being routinely stored in the cloud. To address this, sensor platforms must reduce or transform the data in-situ, sending only the analyzed results to a central server. There are several challenges to designing an open sensor platform capable of leveraging advances in machine learning to support edge computing, including resilience, performance isolation, and data privacy. This paper describes the architecture of the *Waggle* platform developed at Argonne National Laboratory. As an open platform, Waggle supports a wide range of sensors, including experimental sensors to measure airborne pollutants such as hydrogen sulfide and ozone, as well as cameras intended to detect urban flooding and automobile traffic. The Waggle platform is used by the *Array of Things*, a National Science Foundation project to deploy 500 sensor platforms in the city of Chicago, beginning in 2016.**

Keywords—wireless sensor networks, machine learning, internet of things, urban pollution, low-power computing

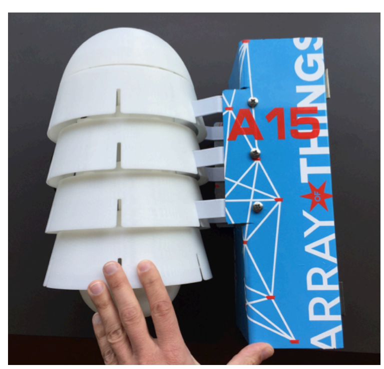
I. Introduction

Wireless sensor platforms span a wide range of designs, from lightweight wearable devices powered by battery or energy-harvesting technology to large unmanned remote monitoring stations used in climate research. Lightweight wireless sensors are often vertically integrated to achieve maximum power efficiency and a small, optimized form factor. However, for wireless sensor systems designed to use multiple high-bandwidth sensors, a programmable, general-purpose platform is desired. Many advanced sensors are capable of producing extremely large and continuous data streams. Hyperspectral imagers, such as those used to investigate ecosystem fluxes [1], microphones used to measure urban noise [2], and high-resolution cameras capable of counting pedestrians can easily generate multiple gigabytes of data per day, making it impractical to stream all collected data to the cloud. Furthermore, in some sensor deployments, privacy concerns may restrict the resolution of data leaving the devices and being routinely stored in the cloud. Ultimately, a new generation of in-situ sensing devices is needed where there is sufficient computational power to support machine learning across the complement of on-board sensors, moving from embedded devices that are "smart" (but inflexible) to those that can learn and discover. To address this, sensor platforms must use the computational power of multiple CPU/GPUs to reduce the data in-situ, and where sensors can impact privacy, sending only the analyzed results to a central server. An architecture to support edge computing capable of leveraging the latest advances in machine learning to analyze and classify data must address several challenges, including resilience, performance isolation, and data privacy. We describe here the architecture of *Waggle*, an open sensor platform for edge computing developed by Argonne National Laboratory.

II. Resilience

Advances in computer vision have made it practical to integrate inexpensive high-resolution image sensors into wireless sensor networks and process the data locally. For example, many computer vision techniques have been developed for counting pedestrians [3, typically assuming a centralized processing model that requires imagery to be transmitted and stored. This approach presents significant privacy challenges in urban environments, and moreover is cost-prohibitive in context of a network of hundreds or thousands of devices. Consequently the imagery must be processed in-situ, requiring a robust computing platform.

Wireless sensors are also often installed in locations that cannot be easily serviced. The *Array of Things (AoT)* [4] project in Chicago will deploy 500 Waggle-based sensor platforms beginning in 2016.



An Array of Things Sensor Node

The sensors are being installed on traffic signal light poles by the Chicago Department of Transportation. Installation is costly, with a team of three electricians installing and wiring the node 15 to 25 feet above the pavement using a bucket truck to lift the electrician to the required height. Traffic must often be diverted to install or service the sensor. Consequently, the devices must be extremely reliable and resilient to harsh weather conditions, as any physical access to the devices for maintenance or repair could readily be more expensive than the device itself. Ideally, the device should be

This work is supported by the U. S. Department of Energy under contract no. W-31-109-ENG-38

resilient to minor failures, operating at limited capacity rather than failing.

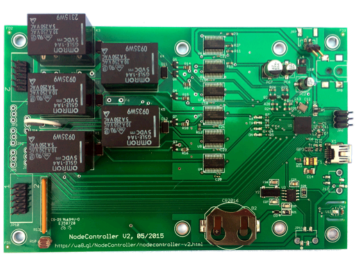
For traditional lightweight detect-and-transmit wireless sensors, rudimentary microcontrollers and extremely thin, simplified, often static, software stacks combined with a watchdog timer that can reboot the microprocessor provides sufficient reliability. However, the complex software stack required to support computer vision packages such as OpenCV [5], which leverages GPU processing via OpenCL or NVIDIA CUDA, commonly built on Linux, introduces reliability concerns. Simply put, feature-rich and sophisticated software environments are more prone to software as well as hardware faults. To mitigate against this, our architecture separates the computing and control system of Waggle node into three independent pieces:

Node Controller: A simple Amlogic quad-core ARM Cortex A5 single board computer (ODROID C1+) runs Linux and manages the sensor data cache, reads simple sensor values from attached devices, and manages the network stack (Ethernet, Wifi 802.11, or cellular LTE), including encryption and certification management.

Payload Processor: A Linux-based single board computer (ODROID XU4) based on the Samsung Exynos5422 CPU with four A15 cores, four A7 cores, and a Mali-T628 MP6 that supports OpenCL. Software packages such as OpenCV or Berkeley Caffe [6] provide edge-computing capability.

Waggle Manager: A custom-designed embedded microcontroller board Wagman manages power, monitors the operation of all system components, and resolves many classes of faults. It uses a reliable Atmel ATMEGA32U4-MU with industrial operational range (-40°C to 85°C). The microcontroller is commonly used in Arduino-class boards.

The Waggle Manager (WagMan) board is our approach to providing theWagman provides the resilience and simplicity of a hardened microcontroller, using classic mechanisms such as an embedded watchdog timer while also supporting Linux-based compute boards with extremely complex, and usually brittle, software stacks. In effect, the Linux boards become devices that can be managed. Instead of focusing on making the full-featured embedded OS resilient, Waggle borrows a design pattern from NASA deep space probes, and relies on the simplified and environmentally hardened WagMan to provide a safe mode that enables recovery and reprogramming of the compute boards.



WagMan: Resilience Module

WagMan employs internal sensor inputs that are out of band with respect to the Linux boards, including CPU temperature, power draw, and humidity of the enclosure. A simple digital signal between the Linux system and WagMan provides a software-driven heartbeat monitor. Sealed relays allow WagMan to completely manage electrical power to subsystems (including the Linux single board computers), permitting sequential power-up of subsystems to avoid power spikes or to electrically isolate and disconnect faulty boards.

The WagMan can also manage boot image selection on the ODROID single board computers. Each of the ODROIDs can boot from either a micro SD card (primary) or an eMMC card (secondary), which are both installed. The selection is done via a board-level logic line that is routed to the Wagman. By using the heartbeat line and boot selector, the WagMan becomes capable of sophisticated recovery schemes. For example, when the heartbeat is lost, the WagMan can try to reboot the node. After three failed attempts at rebooting the node, the WagMan can select the backup (secondary) boot image. The backup boot image can then determine how to rebuild or reinstall the image on the primary boot media. This allows for a "safe mode" that can reformat the media and reinstall a known configuration.

III. Performance Isolation

To isolate the in-situ analysis of streaming data from the core OS functionality of data caching, encryption, and WAN messaging to the cloud, software or hardware techniques can be used. The Linux kernel supports `cgroups`, a feature that isolates resource usage (CPU, memory, etc). This software feature is used by container systems such as Docker. Software techniques may be able to throttle and isolate the in-situ data analysis workload sufficiently to provide weak guarantees that the communication and management functionality of the Node Controller can run independently. However, computer vision software, deep learning frameworks, and GPU software drivers are all evolving rapidly and quite often, experimental. Bugs in GPU drivers can cause unrecoverable kernel panics if they write to an erroneous memory range. This underscores the importance of both platform resilience and separation of control and processing functions implanted in the Waggle architecture.

Waggle uses a straightforward and simple hardware approach to provide performance isolation –To this end a separate Payload Processor handles all high-bandwidth data streams that need edge computing. For AoT, this includes one high resolution camera looking up to the sky, one camera facing an intersection in the street, and a microphone. These three USB-connected devices can all be processed via the Payload Processor without introducing resilience concerns or interfering with core processes on the Node Controller. Furthermore, when electrical power is constrained, the power resources for the Node Controller and the Payload Processor can be independently managed. Such independence will be central to developing more sophisticated node behavior, such as low power modes to manage solar/battery installations, putting the Application Processor into low power mode (or fully powered down) when lighting conditions are too low to support computer vision, or to implement a "last gasp" transmission upon loss of power.

IV. Privacy and the Waggle Plugin Architecture

Cameras are everywhere; private devices peer down from the ceilings of coffee shops and hotel lobbies while municipal devices record images on sidewalks, subways, and streets. The cameras are usually used for crime deterrence, and therefore the images must be preserved as evidence. However, if high-resolution images are to be used in public spaces for sensing, and not crime prevention, the large majority of the images do not need to be preserved. As noted earlier, street activity such as pedestrian or vehicle flow, are already in common use. However, in an urban landscape, many other opportunities for privacy-preserving edge computing exist. Flooding is a significant problem for cities with aging infrastructures. With the appropriate edge- computing support, computer vision algorithms can detect standing water on streets. When collected across a city, a hydrological model can be built. Similarly, computer vision algorithms that detect when cars have difficulty stopping at intersections in the winter might indicate areas where more road salt could be used to improve driving conditions. To enable these and other privacy-preserving applications, the Waggle architecture is designed to carefully restrict, monitor, and manage the connectivity and data stream from the Payload Processor.

The Payload Processor is connected via Ethernet directly to the Node Controller. A software plug-in architecture allows data from sensors to be forwarded to the Node Controller, where it is cached and then eventually forwarded to the "beehive", our server that maintains the database of recorded sensor values. The simplest plug-ins read from serial line-connected sensors such as temperature, humidity, or ozone, and push their data to the data cache. Edge-computing plug-ins, such as those that can count pedestrians, process the images in the Payload Processor, delete the processed image, and then call a routine to push the summarized data, for example "10 pedestrians in the field of view" or "20% of roadway covered with ice" over the Ethernet link to the data cache on the Node Controller.

The Payload Processor has no access to the WAN, and is only connected to the Node Controller via a monitored data path to the Node Controller cache. Rogue plug-ins processing image data cannot therefore leak private data, but only send short summary datagrams. This architecture allows experimental image processing pipelines to be safely executed on the Payload Processor.

An "UglyBox" at Indian Boundary Prairie

V. Deployments

Waggle-based sensor nodes have been deployed in several locations, including an Argonne National Laboratory test site, the University of Chicago campus, DePaul University in Chicago, the Illinois Indian Boundary Prairie managed by the Nature Conservancy and Northeastern Illinois University. Relaxing the aesthetic and quick-mount requirements for urban deployments such as Array of Things, Waggle is often deployed in a more utilitarian "UglyBox" kits, using a simple waterproof enclosure and Stevenson shield for the sensors monitoring the atmosphere. The data being collected is improving weather and climate modeling [8].

The largest Waggle deployment planned to date is the Chicago Array of Things project, with 50-80 to be deployed in 2016 and a total of 500 by 2018 [4]. In addition to measuring air temperature, relative humidity, barometric pressure, UV light, IR light, and magnetic field, prototype electrochemical sensors will sense carbon monoxide, nitrogen dioxide, hydrogen sulfide, ozone, total oxidizing gases, and total reducing gases. Two cameras and an microphone will use the edge-computing capabilities to explore how smart city technologies can improve the energy efficiency and quality of life for city residents.

VI. Conclusions

Waggle is an open source platform designed for adaptation and extension, enabling new sensors, computing, and other technologies to be readily integrated. Architecturally, its defining characteristic is the ability to support sophisticated edge computing. The design of Waggle addresses three important needs for wireless sensor platforms that support programmable in-situ computing: resilience, performance isolation, and data privacy. Resilience is managed via a custom-designed microcontroller board that treats the full-OS boards as managed devices. Waggle uses a physically distinct Payload Processor to support edge computing without introducing resilience issues in the Node Controller. Privacy is managed by implementing a sandbox for the Payload Processor so only summarized data can be injected into the cache and sent to the Beehive server. Initial AoT data will be available for presentation at Sensors 2016.

References

[1] http://www.evs.anl.gov/research-areas/highlights/ecospec.cfm

[2] Mydlarz, C., Nacach, S., Roginska, A., Rosenthal, E., & Temple, M. (2014). The implementation of MEMS microphones for urban sound sensing. AES 137th Convention.

[3] Munder, Stefan, and Dariu M. Gavrila. "An experimental study on pedestrian classification." Pattern Analysis and Machine Intelligence, IEEE Transactions on 28.11 (2006): 1863-1868.

[4] NSF Award 1532133, "MRI: Development of an Urban-Scale Instrument for Interdisciplinary Research," 2015-2018, Catlett, Beckman, Papka, Cagney, Work. https://arrayofthings.github.io/

[5] http://opencv.org

[6] http://caffe

[7] Jia, Yangqing and Shelhamer, Evan and Donahue, Jeff and Karayev, Sergey and Long, Jonathan and Girshick, Ross and Guadarrama, Sergio and Darrell, Trevor. "Caffe: Convolutional Architecture for Fast Feature Embedding" 2014

[8] Sankaran, Rajesh, et al. "Waggle: A Framework for Intelligent Attentive Sensing and Actuation." AGU Fall Meeting Abstracts. Vol. 1. 2014