# A Fog Computing-oriented, highly scalable IoT framework for Monitoring Public Educational Buildings

Orestis Akrivopoulos[1], Na Zhu[2], Dimitrios Amaxilatis[3,4], Christos Tselios[4], Aris Anagnostopoulos[2] and Ioannis Chatzigiannakis[2]

[1] Spark Works ITC Ltd, UK mail: akribopo@sparkworks.net
[2] Sapienza University of Rome, Italy, mail: zhu.1706409@studenti.uniroma1.it, {aris,ichatz}@diag.uniroma1.it
[3] Computer Technology Institute & Press (CTI), Greece, mail: amaxilat@cti.gr
[4] University of Patras, Greece, mail: tselios@ece.upatras.gr

***Abstract*—We present here an IoT-based platform that provides an integrated solution for real-time monitoring and management of educational buildings at a national scale. The proposed system follows the Fog Computing paradigm so that sensor data processing takes place at the edge devices of the network. In this way, the system significantly reduces the network traffic across the network core layers. The architecture and implementation of the system are presented in details in relation to existing use-case scenaria. The performance of the prototype architecture is evaluated in a real-world environment using a range of edge devices available in a pilot deployment spanning across 18 school buildings. The evaluation indicates that existing resources are sufficient to accommodate traffic that can increase up to $5$ times higher from the existing one even in sites where low-end devices (e.g., such as Raspberry Pi) are available. The results provide evidence that Fog Computing can address the ever-increasing amount of data that is inherent in an IoT world by effective communication among all elements of the architecture.**

## I. Introduction

Public educational systems on a national level involve the operation and management of a massive number of buildings that possess vastly different characteristics in terms of size, age, location, construction, thermal behavior and user communities, among other, while at the same time such buildings can be situated in a very fragmented manner. At national level, public educational systems operate thousands of buildings, i.e., one could envision a system to handle energy-related aspects at a massive scale. In many cases, the size of such institutions' assets outclasses even those of multinational corporations. The energy and environmental impact caused by public educational buildings via their complex activities and operations in teaching and research, as well as provision of support services could be considerably reduced by an effective choice of organizational and managerial measures. To design and operate a set of sustainable public educational buildings it is necessary to factor a number of parameters such as the interaction of indoor-outdoor environment, strategic planning and operational processes.

An interesting aspect regarding the energy efficiency of schools is the fact that historically, energy expenses in public educational systems have been treated as relatively fixed and inevitable. Evidence shows that a focus on energy use in schools yields an array of important rewards in concert with educational excellence and a healthful learning environment [1]. Since energy costs are the second largest expenditure within public educational systems budgets, exceeded only by personnel costs [2], significant savings can be carved out, if energy consumption can be reduced. We expect that an IoT infrastructure can help improve the organizational and managerial measures that will lead to a reduction of energy consumption by actively involving the school staff (e.g., school directors, building managers, custodial staff and teachers) in fostering a culture of energy conservation. If empowered to do so, building managers and custodial staff can offer critical insights about ways to lower a buildings energy footprint through effectively managing building operations [3]–[5].

The dominant approach followed by large industries focused SMEs and startups is the development of cloud-based IoT platforms that simplify the interconnection of smart devices, the collection of data generated to the cloud, and the central processing of the information utilizing other cloud-based services [6]. This work studies a cutting-edge IoT infrastructure deployed over 18 educational buildings across 3 countries, comprising over 900 IoT monitoring points, providing data for energy consumption in several points in each building, as well as outdoor and indoor environmental data [7]. The energy consumption meters monitor the overall electricity consumption in these buildings, as well as in specific building sectors/floors, or teaching rooms. The electricity data are complemented by indoor/outdoor environmental data for temperature, humidity, room occupancy, noise levels, and other sensors that can paint an accurate picture of the environmental conditions inside and outside the aforementioned buildings. In its current setup, this pilot deployment produces daily over 400MB of data, resulting in a yearly data volume of approximately 140GB. However, the current setup uses averaging to minimize the required storage space. For obtaining near real-time information on the building status, it is necessary to shorten the averaging period (now set to 5 minutes), an approach that will lead to significant data volume increase. In this context, sensors

978-1-5386-3180-5/18/$31.00 ©2018 IEEE

comprising an IoT infrastructure deployed at a national level over a large number of public educational buildings will generate, handle, transfer and store a tremendous amount of data, which cannot be processed in an efficient manner using current cloud platforms and techniques.

It is evident that the cloud-based approach needs to address multiple performance issues appearing at all levels of the network architecture while transferring massive datasets collected from the sensors and delivered to distant machine clouds: (a) network bandwidth issues at the network edge, (b) network energy consumption as traffic flows through the network core, (c) continuous I/O operations on the data centers where datasets are stored, (d) increased exposure of data across third-party cloud-based services. As stated in [8] minimal possible latency, network bandwidth preservation, increased security and enhanced reliability are elements of paramount importance for any IoT-related application. The necessity for data collection, storage and availability across large areas, the demand for uninterrupted services even with intermittent cloud connectivity and resource constrained devices [9], along with the necessity of sometimes near-real-time data processing in an optimal manner, create an amalgam of challenges where only radical and holistic solutions apply.

Fog computing was conceived as a distributed computing paradigm delivering computational resources, storage and control to consumers, through an intermediate operations layer strategically placed between the secluded cloud data centers and end-user equipment. This approach greatly alleviates bandwidth consumption, increases the data processing capacity of isolated nodes, reduces latency and provides additional security and reliability while accelerating system responsiveness.

In this paper, a Fog Computing based approach is presented that takes advantage of the resources available at the edge of the network. Data arriving from the sensors are analyzed directly at the edges of the network and only aggregated information are forwarded to the cloud. The prototype architecture proposed is generic enough to cover a broad range of applications. The performance of the solution is evaluated based on a real-world deployment of 18 school buildings using the edge devices available. The results indicate that the resources are sufficient to process even 5 times higher traffic than the one currently accommodated by the real-world deployment even in cases where low-end devices (e.g., such as Raspberry Pi) are available.

## II. Related work

Obtaining reliable access to the potentially huge number of sensor data originating from the IoT domain has always been a challenging issue. Since several low-end devices are involved, the notion of in-network aggregation and on-the-spot data management was considered as a viable solution due to its inherent ability to combine heterogeneous datasets from a broad spectrum of sources within a specific timeframe and deliver an elevated end-user quality of experience [10], [11]. The aforementioned techniques were always considered working in parallel with lower-level medium access control protocols as well as network-level routing ones. An overview of different techniques and existing protocols is presented in [12].

With the advent of Big Data, several map-reduce-related approaches were introduced [13] (e.g., Apache Spark[1]) that essentially conduct the overall analysis in distinct batches. A survey of possible stream processing optimizations and variations is presented in [14]. These solutions take into consideration the internal logic of the components that constitute the high-level application [15] and appear capable of confronting several of the intrinsic limitations of cloud thus alleviating the deployment of services with low or even zero tolerance for latency delays.

In addition to sensor-originating traffic management and dataset handling, precise energy monitoring and conservation methods are aspects of great interest, mostly due to the imbalance between power generation and demand. Smart Grids [16] appear to be an excellent playground for smart power meters based on advanced sensors and IoT-related technologies, therefore an independent layer of communication and information handling like the one existing in the Fog Computing paradigm would enhance the monitoring frameworks' overall robustness and efficiency. When it comes to private residential energy monitoring solutions, several prototypes for domestic power consumption estimation have been presented [17], unfortunately with the limited ability to calculate expenditure patterns and relatively small-scale deployment capabilities. Other prototypes such as [18] only focus on low price, therefore offer a limited set of features, totally lacking the ability of data manipulation and local storage. A more holistic approach to real-time building energy modeling through IoT device integration along with structural information extraction per building from dedicated databases is presented in [19], [20]. This work exploits the advances in physical and environmental information sensing, communication and processing offered by IoT together with the now available digital repositories of buildings and districts. The authors present an interesting prototype which supports near-real-time energy consumption visualization, yet with limited scaling ability without an explicit provisioning process.

The notion of local data pre-processing to reduce data transfer between nodes was considered by [21], [22]. This approach appears to be much more suitable for the somehow limited transfer capacity intermediate layer of Fog Computing, where the vastly increased number of interconnected nodes will render dataset transmission inefficient and significantly expensive in terms of resources. This is the compelling reason that our approach fully exploits local preprocessing thus alleviating the shortages in both bandwidth and throughput every network occasionally faces.

## III. Prototype Architecture for Fog-Computing-Aware Deployments

Designing an IoT infrastructure deployment over Public Education Systems on a national level entails a broad range

[1]http://spark.apache.org/

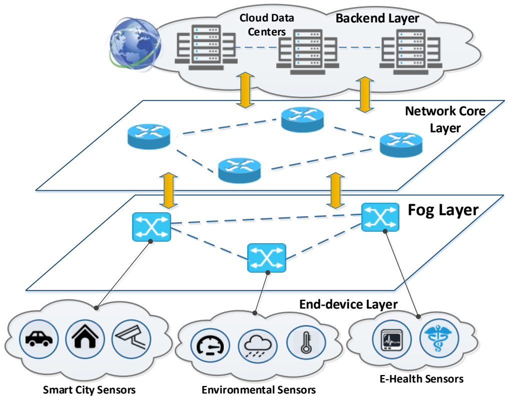

Fig. 1: Fog Computing IoT architecture

of functional and non-functional requirements. The platform will be deployed over literally thousands of buildings spread throughout a country (*multi-site deployments*). It is reasonable to expect a diverse set of device providers working under the same interoperability framework (*device heterogeneity*). A number of different sensors will be installed ranging from energy sensors to indoor and outdoor environmental sensors, motion sensors, etc., generating different data (*data variety*). Such a platform will be used by a broad variety of people-centric applications [23] that will facilitate the educational sector towards improving the energy efficiency of school buildings (*user diversity*). Certain users (e.g., building managers) will require real-time monitoring of the buildings while other users (e.g., students, teachers) will require aggregated numbers as educational material and student projects (*data velocity*).

### A. Use-case Scenaria & Data Access

The availability of actual measurements of environmental parameters, such as energy consumption, indoor and outdoor luminosity, temperature, noise, pollution, etc., can feed a plethora of people-centric information, education, and involvement initiatives in order to effectively change the ways people live and work inside school buildings and achieve better energy efficiency. This means, on the one hand, to better inform people and enable them to make educated decisions, and on the other hand to enable a whole different set of applications, like gamification apps that bridge the virtual world with the real one, towards the end-goal of such systems. We here focus on three specific use-case scenaria:

*Education:* Teachers use collected data and analytics during the class to explain to pupils basic phenomena related to the parameters monitored.

*Students' engagement:* Teachers organize student projects where students (or group of students) monitor specific environmental parameters.

*Building management:* Collected data are analyzed and used to profile the energy performance of the building and specific equipment. The availability of data from similar schools and/or similar equipment allows to do some benchmarking and supporting decisions for preventive maintenance or substitution of existing equipment.

### B. Data Engineering on the Fog layer

Given the heterogeneity of the hardware devices, data collected from the sensors are in different units, measure slightly different physical phenomena (e.g., carbon dioxide $CO_2$ vs equivalent carbon dioxide $eCO_2$) and with different calibrations. It is usual for sensor devices to experience transient disconnections leading to missing data over certain periods. In other cases, the low-quality of the sensors produce erroneous data that may affect studies of any time series. It is thus important that data received from the sensors are continuously processed and curated automatically. A graphical representation of the internal services of the fog layer is included in Fig. 2.

*Stream Processing Pipeline:* All data messages retrieved from the IoT end-devices are circulated within the edge device over a message bus system that is responsible for distributing the information gathered to the various subsystems responsible for storing and processing the data or generating alerts. The message bus system offers us the flexibility to introduce data processing, transformation, and aggregation mechanisms at different layers of the architecture depending on the needs of the high-end application. Internally the message bus assigns a dedicated *stream processing pipeline* for each separate sensor supported by each end device. Sensor data streams are assigned unique names that are derived from the unique network address of the sensor device producing the data, the type of sensor and the unique ID of the edge device.

The message bus layer offers us the ability to store data in a scalable way at different layers of the architecture: (a) data are formatted for near-instantaneous retrieval, thus avoiding time-consuming queries and aggregations; (b) a cache-like mechanism is available for providing recent data for a specific sensor devices or collection of end-devices using a single lookup in near-located storage service; (c) older historical measurements are stored in a deep storage service that is used only when measurements that are older than the ones provided by the first service. This decoupling of the data generators and the data storage services allows us to implement a broad range of services that need to optimize different performance criteria.

*Generic Data Processing:* After the creation of the stream processing pipeline, all data received are initially processed based on a set of generic data processing functions. In this sense the edge device becomes the "primary" processing point for the curation and aggregation of all data arriving from the end devices (see Fig. 2). The *first step* of the processing pipeline is to identify the presence of erroneous data that adversely affects the study of any time series. At this step, we identify outliers (i.e., new observation points that are distant from the historic values) that may be due to transient errors occurring on the sensor device and should be excluded from the data set. The *second step* applies a 5-minute moving window to average and smooth out short-term fluctuations. The *third step* is used to address temporary disconnections

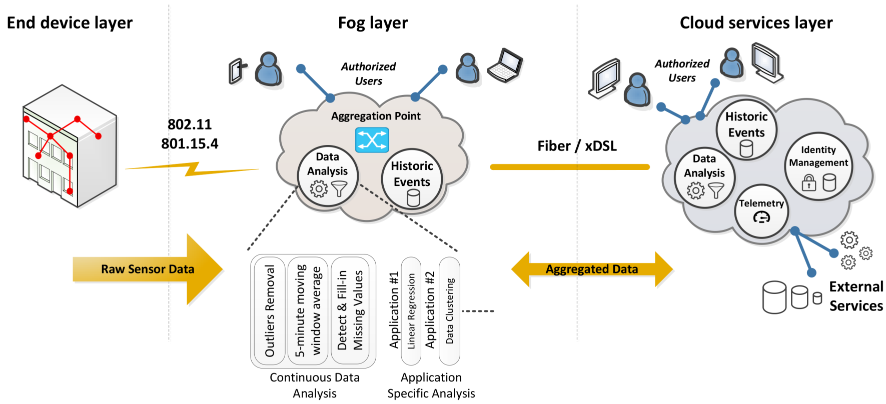


Fig. 2: High-level Architecture, Data analytics at the Fog layer and Data flow

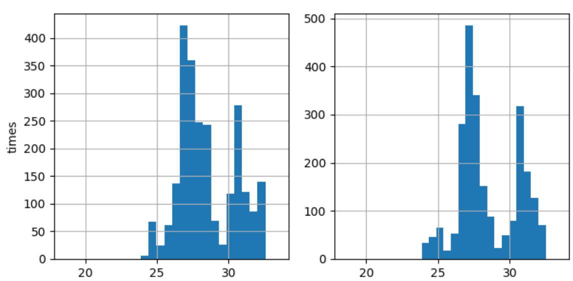


Fig. 3: Histogram of indoor temperature for the evaluation of thermal comfort of two different classrooms

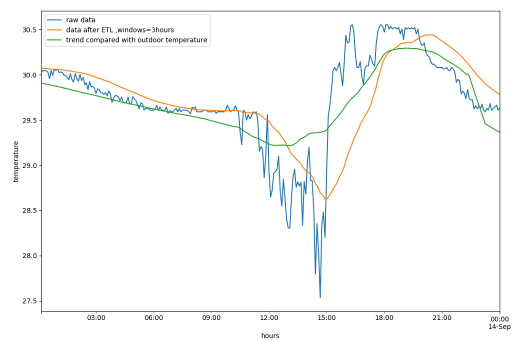


Fig. 4: Application specific sensor data processing that are included in classroom educational material

of the sensor devices that lead to missing values. A simple local algorithm identifies the absence of data and based on the historic values introduces mean values to fill in the missing data.

*Continuous Data Analysis:* As soon as the 5-minute values are processed for each separate sensor stream, they are aggregated based on a variety of functions (e.g., min/max/mean), and different intervals (e.g., 1-hour, 1-day). Furthermore first-order and second-order analysis is applied to each separate stream to identify local maxima/minima and basic measures on the slope of the data. The resulting data are persisted in the storage available at the edge device and are also forwarded to the cloud layer. As an example of the continuous data aggregation, Fig. 3 depicts a histogram of the indoor temperature of a specific classroom that is generated automatically as part of the evaluation of the thermal comfort of each classroom.

*Application Specific Processing:* Depending on the application different data processing is required. Users specify the rules for data-driven processing and event-based response using JSR-000335 lambda expressions[2]. These specifications are injected into the edge device and decomposed based on the sensor data that they process as well as the time-windows used for processing the stream of data. As part of educational activities, teachers need to examine the relationship between indoor and outdoor environmental conditions to address topics related insulation, air quality, etc. A common approach is to apply a linear regression for modeling the relationship between scalar dependent variables and one or more explanatory variables (or independent variables). For example, in Fig. 4 indoor and outdoor temperature are compared to explain how the orientation of a building affects the thermal comfort.

### C. Services on the Cloud layer

The cloud layer is providing certain features to facilitate uniformity of security and information exchanges across all edge devices and in general facilitate the overall compliance of the prototype to the fog computing principles of security, efficiency, and enhanced reliability.

*Identity Management*. The Identity Management service is used to provide authentication to all users and covers a number of aspects involving users' access to services and applications, including secure and private authentication from mobile devices and the web application or user profile management and privacy-preserving disposition of personal data. Generating a new identity requires providing a minimum set of information that is stored on an encrypted database. The information associated with the account is only accessible by the user and

[2]https://jcp.org/aboutJava/communityprocess/final/jsr335/index.html

TABLE I: Technical specifications of edge devices

| | Raspberry | Zotac | VPS |
|---|---|---|---|
| Processor | BCM2836 Arm7 | i3-3120M | E5-2630V4 |
| Frequency | 900MHz | 2.2GHz | 2.2GHz |
| Cores | 1 | 2 | 6 |
| Memory | 736 MB | 8 GB | 24 GB |
| Disk Type | 64GB SD Card class10 | 120GB SSD | 600GB SSD |

those users that are authorized to do so. All other services (and users) cannot decrypt the information and thus all information accessed is anonymized.

*Historic Events:* Accessing historical data is crucial for building monitoring applications, e.g., when comparing historical data from different time spans and building areas. This cloud service ensures that data from different buildings (i.e., arriving from different edge devices) are stored on a common logical place and can be accessed without delays independently of the targeted time interval.

*Telemetry:* Monitoring and managing multiple buildings of a Public Educational System also requires real-time access to remote buildings. The Telemetry service allows the interaction with specific edge devices in order to provide a direct link to the raw sensor data collected. This interaction is two-way, allowing building managers to remotely operate devices (e.g., switch off/on lights, HVAC).

*Data Analysis:* The Data Analysis service allows to organize large volumes of data and visualize them from different points of view. In order to improve the flexibility of business intelligence this service stores *sensor annotations and their metadata* (e.g., observed properties, units of measurements or locations) as part of a graph database and thus allows to do complex relation queries like retrieving the list of available *temperature* sensors in *multiple locations* simply by running a graph traversal query. In this way, we completely avoid the need for the preconfiguration of the IoT infrastructure and the fixed routing of data through the various cloud-based services that constitute the resulting system.

## IV. Performance evaluation

We evaluate the prototype architecture by deploying an experimental implementation on the edges of the pilot IoT deployment of 18 school buildings provided spread in 7 locations [7]. Three different edge devices were used: two low-power single-board devices (a Raspberry PI and a Zotac a low-end Intel-based system), and also an off-the-shelf Intel-based edge-based server box. Their exact specifications are summarized in Table I. The software deployed on the devices is bundled using a collection of Docker containers that guarantee that the exact same software (and configuration parameters) is executed on each edge device regardless of the actual operating system type and version used. The software is configured to use only 1 core so that we can produce comparable results among the three different edge devices used. For the case of the Zotac and VPS devices we can improve the performance by increasing the number of cores. However this requires further analysis that due to the space limitations are not included.

TABLE II: NET I/O in MB

| Service | Raspberry I/O | Zotac I/O | VPS I/O |
|---|---|---|---|
| Stream Procesing Pipeline | 101 / 26.9 | 88.6 / 17.3 | 89.1 / 15.9 |
| Generic Data Processing | 79.8 / 82 | 78.2 / 79.9 | 79.4 / 80.4 |
| Continuous Data Analysis | 6.62 / 5.37 | 2.03 / 1.97 | 0.45 / 0.46 |

On the pilot IoT deployment, on average 15 sensing devices are available on each school. Each of them is equipped with 1 to 5 sensors, providing data for attributes like energy consumption, outdoor and indoor environmental data [7] in several points in each building. The actual sensing rate is $30sec$ thus on average each edge device is receiving 2.5 sensor updates every second. Given this real-world deployment, we record a total of 70000 actual messages containing sensor data generated from 100 different end devices located at different places in the pilot IoT deployment. These recordings of message traffic are organized in one data set that is injected on the incoming message queues of each individual edge device. In order to stress the performance of each edge device, the sampling rate is increased to 12 messages per second - which is almost 5 times higher than the actual requirements of the existing IoT deployment. The experiment is repeated for each different type of edge device available.

The first goal of the evaluation is to measure the network bandwidth required, the average processing rate, and the processing latency of services executed on the Fog layer per sensor measurement. Table II indicates the values of the network I/O traffic for each individual service operated at the Fog layer. Recall from Sec. III that the *Stream Processing Pipeline* organizes the exchange of the sensor data across the rest of the services. The *Generic Data Processing* module does the hard part of calculating the updates on the sensor data in real time. Finally, the *Continuous Data Analysis* module finalizes the computation and coordinates the persistence of the resulting data in a uniform way across all levels of the edge and the cloud, as needed. Observe that all services achieve very similar traffic levels. Since the same message traffic is replayed to each edge device and since all three devices are executing exactly the same software stack, it is reasonable to expect that some minor fluctuations will exist.

The second goal is to assess the CPU utilization of the edge devices. During the experiment, we record the CPU utilization time of the processors of each edge device. As expected, the Raspberry Pi is processing the message stream at its maximum capacity - the CPU utilization is continuously at $100\%$ while for the Zotac and the VPS this is not the case, as the CPU utilization is always around $10\% - 25\%$. Furthermore, the average processing rate of the Fog layer is recorded for each edge device. The results are depicted in Table III. In the case of Raspberry Pi the average processing rate is 15.36, slightly above the message injection rate used and well above the actual sensing rate used in the actual deployment. The Zotac and VPS have a much larger processing rate, $2,692.31$ and $5,833.33$ messages per second respectively.

The last objective is to quantify the processing latency

TABLE III: Processing Rate in Messages per Second

| Device | Processing Rate |
|---|---|
| Raspberry | 15,36 |
| Zotac | 2,692.31 |
| VPS | 5,833.33 |

TABLE IV: Services Processing latency (ms)

| Service | Raspberry | Zotac | VPS |
|---|---|---|---|
| Stream Processing Pipeline | 1.620 | 0.016 | 0.004 |
| Generic Data Processing | 52.183 | 0.400 | 0.244 |
| Continuous Data Analysis | 63.817 | 0.186 | 0.145 |

of each different service of the Fog Layer. In Table IV the processing latency is measured (in $ms$) for each different service and for each edge device. In the case of the Raspberry Pi, both the Generic Data Processing and Continuous Data Analysis are dominating the CPU usage. On the other hand, for the Zotac and VPS, the more powerful processor provides faster integer and floating point calculations thus leading to a much shorter processing latency for the Continuous Data Analysis module.

## V. Conclusions

The paper presented a prototype architecture for IoT deployments where the processing of the sensor data is conducted at the edges of the network. The prototype architecture is applied on a real-world deployment for monitoring a fleet of educational building at a national level. The performance of the prototype is evaluated using the available infrastructure, indicating that even low-power edge devices such as the Raspberry Pi platform, can easily accommodate existing network loads. The results provide evidence that Fog Computing, even when using edge devices of very limited resources, can address the current need of IoT deployments and sustain up to 5 times higher amount of traffic by effective communication among all elements of the architecture.

## VI. Acknowledgments

This work has been partially supported by the EU research project "Green Awareness In Action" (GAIA), funded under contract number 696029 and the research project Designing Human-Agent Collectives for Sustainable Future Societies (C26A15TXCF) of Sapienza University of Rome. This document reflects only the authors' view and the EC and EASME are not responsible for any use that may be made of the information it contains.

## References


[1] K. Crosby and A. B. Metzger, "Powering down: A toolkit for behavior-based energy conservation in k-12 schools," U.S. Green Building Council (USGBC), Washington DC, USA, Tech. Rep., 2012.

[2] "Energy star building upgrade manual," US Environmental Protection Agency (EPA), Washington DC, USA, Tech. Rep., 2008.

[3] J. E. Cross, "Shifting organizational culture - innovation for energy conservation," Garrison Institute, New York, USA, Tech. Rep., 2012.

[4] C. Schelly, J. E. Cross, W. S. Franzen, P. Hall, and S. Reeve, "Reducing energy consumption and creating a conservation culture in organizations: A case study of one public school district," *Environment & Behavior*, vol. 43, no. 3, pp. 316–343, May 2011.

[5] ——, "How to go green: Creating a conservation culture in a public high school through education, modeling, and communication," *Journal of Environmental Education*, vol. 43, no. 3, pp. 143–161, 2012.

[6] I. Chatzigiannakis, G. Mylonas, and S. E. Nikoletseas, "jwebdust : A java-based generic application environment for wireless sensor networks," in *DCOSS*, ser. Lecture Notes in Computer Science, vol. 3560. Springer, 2005, pp. 376–386.

[7] D. Amaxilatis, O. Akrivopoulos, G. Mylonas, and I. Chatzigiannakis, "An iot-based solution for monitoring a fleet of educational buildings focusing on energy efficiency," *Sensors*, vol. 17, no. 10, 2017.

[8] Cisco Systems Inc., "Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are," 2015.

[9] M. Chiang and T. Zhang, "Fog and iot: An overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, Dec 2016.

[10] C. Tselios and G. Tsolis, "On QoE-awareness through Virtualized Probes in 5G Networks," in *Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2016 IEEE 21st International Workshop on*, 2016, pp. 1–5.

[11] I. Chatzigiannakis, H. Hasemann, M. Karnstedt, O. Kleine, A. Kroller, M. Leggieri, D. Pfisterer, K. Romer, and C. Truong, "True self-configuration for the IoT," in *2012 3rd IEEE International Conference on the Internet of Things*. IEEE, Oct. 2012, pp. 9–15.

[12] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "In-network aggregation techniques for wireless sensor networks: A survey," *Wireless Commun.*, vol. 14, no. 2, pp. 70–87, Apr. 2007. [Online]. Available: http://dx.doi.org/10.1109/MWC.2007.358967

[13] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008. [Online]. Available: http://doi.acm.org/10.1145/1327452.1327492

[14] M. Hirzel, R. Soulé, S. Schneider, B. Gedik, and R. Grimm, "A catalog of stream processing optimizations," *ACM Comput. Surv.*, vol. 46, no. 4, pp. 46:1–46:34, Mar. 2014. [Online]. Available: http://doi.acm.org/10.1145/2528412

[15] A. Papageorgiou, E. Poormohammady, and B. Cheng, "Edge-computing-aware deployment of stream processing tasks based on topology-external information: Model, algorithms, and a storm-based prototype," in *2016 IEEE International Congress on Big Data, San Francisco, CA, USA, June 27 - July 2, 2016*, 2016, pp. 259–266. [Online]. Available: http://dx.doi.org/10.1109/BigDataCongress.2016.40

[16] R. Morello, C. D. Capua, G. Fulco, and S. C. Mukhopadhyay, "A smart power meter to monitor energy flow in smart grids: The role of advanced sensing and iot in the electric grid of the future," *IEEE Sensors Journal*, vol. PP, no. 99, pp. 1–1, 2017.

[17] S. Karthikeyan and P. T. V. Bhuvaneswari, "Iot based real-time residential energy meter monitoring system," in *2017 Trends in Industrial Measurement and Automation (TIMA)*, Jan 2017, pp. 1–5.

[18] W. Hlaing, S. Thepphaeng, V. Nontaboot, N. Tangsunantham, T. Sangsuwan, and C. Pira, "Implementation of wifi-based single phase smart meter for internet of things (iot)," in *2017 International Electrical Engineering Congress (iEECON)*, March 2017, pp. 1–4.

[19] L. Bottaccioli, A. Aliberti, F. Ugliotti, E. Patti, A. Osello, E. Macii, and A. Acquaviva, "Building energy modelling and monitoring by integration of iot devices and building information models," in *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, vol. 1, July 2017, pp. 914–922.

[20] O. Akribopoulos, I. Chatzigiannakis, C. Koninis, and E. Theodoridis, "A web services-oriented architecture for integrating small programmable objects in the web of things," in *Proceedings of the 2010 Developments in E-systems Engineering*, ser. DESE '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 70–75.

[21] O. Akrivopoulos, I. Chatzigiannakis, C. Tselios, and A. Antoniou, "On the deployment of healthcare applications over fog computing infrastructure," in *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, vol. 2, July 2017, pp. 288–293.

[22] G. Xu, E. C. H. Ngai, and J. Liu, "Ubiquitous transmission of multimedia sensor data in internet-of-things," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2017.

[23] F. e. a. Boavida, *People-Centric Internet of Things—Challenges, Approach, and Enabling Technologies*. Springer International Publishing, 2016, pp. 463–474.