

Ensemble-based Network Edge Processing

Ioan Petri¹, Ali Reza Zamani², Daniel Balouek-Thomert², Omer Rana³, Yacine Rezgui¹, and Manish Parashar²

¹*School of Engineering, Cardiff University, UK*

²*Rutgers Discovery Informatics Institute, Rutgers University, USA*

³*School of Computer Science & Informatics, Cardiff University, UK*

contact author: petrii@cardiff.ac.uk

Abstract—Estimating energy costs for an industrial process can be computationally intensive and time consuming, especially as it can involve data collection from different (distributed) monitoring sensors. Industrial processes have an implicit complexity involving the use of multiple appliances (devices/ sub-systems) attached to operation schedules, electrical capacity and optimisation setpoints which need to be determined for achieving operational cost objectives.

Addressing the complexity associated with an industrial workflow (i.e. range and type of tasks) leads to increased requirements on the computing infrastructure. Such requirements can include achieving execution performance targets per processing unit within a particular size of infrastructure i.e. processing & data storage nodes to complete a computational analysis task within a specific deadline. The use of ensemble-based edge processing is identified to meet these Quality of Service targets, whereby edge nodes can be used to distribute the computational load across a distributed infrastructure. Rather than relying on a single edge node, we propose the combined use of an ensemble of such nodes to overcome processing, data privacy/ security and reliability constraints. We propose an ensemble-based network processing model to facilitate distributed execution of energy simulations tasks within an industrial process. A scenario based on energy profiling within a fisheries plant is used to illustrate the use of an edge ensemble. The suggested approach is however general in scope and can be used in other similar application domains.

Keywords-Edge computing, Energy Efficiency, Internet of Things, Industrial Processes

I. INTRODUCTION

The integration of industrial workflows/ processes with edge devices provides numerous opportunities in automation, optimisation, intelligent manufacturing and smart industry, moving towards an on-demand service model. This leads to potentially new revenue models and facilitates industrial transformation. Integration of manufacturing industries with Cloud-based analysis has been extensively investigated, considering the abundance of potential data generated from sensors integrated into industrial processes. Edge computing can represent a new solution to enhance and complement cloud-based data centers with support for real-time analysis of such data, enabling distributed execution of tasks and enabling support for security/ privacy policies. Significant work has emerged in recent years on how edge devices can be used to extend the capability of cloud-based analysis for other latency sensitive applications,

such as health care, security, smart cities, traffic control, transportation, production automation and many others all looking into how to implement the edge/fog models [1].

There are a number of industries and businesses which act as energy *prosumers* (both users and producers) often located close to large urban centres. To survive, they must be innovative in their business practices, controlling their cost base by the use of intelligent techniques for managing their energy consumption (a major factor in operational expenditure). For example, the fish processing industry is going through a paradigm shift from a unidirectional, demand driven industry with large centralised power generation to a market driven operational environment making use of smart grids, where supply and demand will be balanced with variable and intermittent renewable energies in a more localised manner. This will require intelligent systems to enable end users to satisfy demand within the peaks and troughs of the energy market. Therefore, supporting energy efficiency in industrial plants represents a prime objective for energy policy at regional, national and international levels. Studies have also indicated that although people and organisations are often aware of the benefits of using energy more efficiently, a variety of social, cultural, and economic factors often prevent them from doing so [2], [3].

Energy optimisation demonstrates real time use of sensor data, where a number of parameters need to be optimised based on a particular model representation. Based on such real-time readings from sensors it has become possible for site managers to take decisions in order to reduce energy consumption. As sensors can provide readings within an interval of 15-30 minutes, it is necessary for any simulation/ optimisation to also be carried out over a similar interval. The efficiency of the optimisation process depends on the capacity of the computing infrastructure used to execute demand analysis tasks. There has also been recent focus on the integration of sensor networks with decentralised distributed systems based on the emergence of various network and IP-based technologies, where monitoring devices do not simply act as sensors, but feature computational, storage, and networking resources. Decentralized processing of data on Internet of Things (IoT) devices supported by cloud technologies and virtualization has proved to be an efficacious method for reducing communication overheads and data

transfer times in an IoT context. To realize decentralized data processing, an orchestrated use of computational and storage resources is needed in order to process IoT data closer to data sources and service consumers (i.e., end users or data sinks) [4], [5], [6]. Grouping of edge devices and resources based on (logical) proximity and/ or geographic location in order to mitigate risks and vulnerabilities and to address security and performance indicators is referred to as an “edge-ensemble”.

We describe how an edge ensemble-based infrastructure can be used to scale out energy simulation of a fish processing industrial workflow. Our application workflow performs real-time energy optimisation and makes use of multiple edge resources to carry out simulation(s). In this work, we determine which of these techniques can be executed in-transit or at the network edge in the presence of multiple constraints such as execution time and quality of results. Such discrete time energy simulation can be used to model heating, cooling, lighting, ventilation and other energy flows within an industrial site. In particular, we show how an edge-resource ensemble can be efficiently used for running and deploying distributed EnergyPlus simulation-based optimisation in a fish processing plant to meet a number of energy related objectives. The remainder of this paper is organised as follows: sections I, II and III introduce and explain the notion of an edge-ensemble, providing a key motivation for our research and analysing several related approaches. Section VI presents the model we make use of and explains how the edge resource ensemble has been utilized in our scenario. The evaluation of our implemented system is presented in section VII. We conclude and identify future work in section VIII.

II. RELATED WORK

Edge computing builds on the significant growth in IoT infrastructures where heterogeneous and networked devices collaborate to achieve particular data monitoring/ processing objectives [7]. *Fog cells* have been utilised as a solution for grouping IoT devices based on vicinity, i.e., single IoT device coordinating a group of other IoT devices and provides benefits related to security, performance and data analysis. Such fog cells can lead to the development of IoT services to process data in close vicinity to data sources/ sinks as an alternative to the traditional usage of cloud systems. The use of Fog (edge) cells can reduce communication delays, facilitating a more efficient utilization of computational, storage, and networking resources. Application scenarios include supporting pre-processing of data streams from sensor nodes [8], or data processing in smart systems [9].

Edge computing involves aspects related to mobility, geo-distribution, low latency, and a communication network [10]. Determining appropriate models for aggregating computational capacity from IoT devices is complex, therefore virtual machines can be deployed in order to serve as a

capacity extension. In edge processing, performance metrics associated with quality of service such as network latency are key objectives which can be fulfilled by the use of cloudlets, and which can bring edge capacity closer to IoT and mobile devices. Complementary to edge computing, Cloud of Things [11] aligns with *in-network* capability as an interaction between IoT and Cloud Computing. Operations on data (such as data sample or filtering) as it is being transmitted from an IoT device to a Cloud data center can be carried out using in-transit processing, thereby reducing data volumes and limiting the size of the data being moved across the network [12], [13], [14].

Recently researchers have investigated the integration of WSNs (Wireless Sensor Networks) with large-scale distributed computing infrastructures to support data analysis and decision support. Examples include an architecture for integrating Cloud computing and WSNs [15], Sensor-Web [16], SensorGrid [17], [18], Sensor-Cloud infrastructure [19], the BodyCloud architecture [20] and use of wireless sensors in buildings [21] etc. These approaches can vary complexity from: (i) development of specialist middleware approaches that can aggregate capability across a range of different (often heterogeneous) sensors; (ii) operating system-based approaches that attempt to support commonly used operations on sensor devices; (iii) gateway/ hub based approaches which involve integrating data across different sensor nodes (supporting sensor fusion, for instance) before an aggregate data stream is sent to a cloud platform.

SensorGrid is a middleware-based framework for providing approximate answers to aggregate queries on summarized sensor network data based on data compression and approximation paradigms. Aggregate queries are the basis for achieving Online Analytical Processing (OLAP) over sensor network readings in Data Grid environments. OLAP has a number of interesting applications for eScience, covering aspects such as visualization of scientific data, multi-dimensional analysis of data streams, privacy of multi-dimensional data [17], [18], etc.

In [19], the authors propose a Sensor-Cloud infrastructure which can manage physical sensors accessible over a network. The Sensor-Cloud Infrastructure virtualizes a physical sensor on the Cloud computing platform, enabling such virtual sensors to be dynamically grouped and provisioned on-demand, primarily through a portal server interacting with (i) provisioning server performing resource management; and (ii) a monitoring server recording the state of real/virtual sensors.

SAaaS [22] is a cloud-enabled SaaS architecture for the management of wireless sensor and actuator networks (WSAN). SAaaS is a software stack that implements the following main functionalities: use of specialist (W)SNs, smartphones or other devices endowed with sensors and/or actuators, and their enablement for interoperation and management in a cloud environment. It also enables exploita-

tion of volunteer-based methods for node involvement, along with interfaces for federating SAaaS Clouds, either volunteer-based or those at a commercial organisation.

There is also significant work in support adaptation of IoT/sensor devices based on their operating environment. This includes provisioning for specialist operating systems (the most popular of which is Contiki/CoAP) which enable dynamic modification of code running on sensor nodes within a network. Extensions to this work have included LooCI (LOOsely coupled Component Interfaces) which support integration of software components executing across different sensor nodes. Edge nodes can also be viewed as hosting environments for code, the adaptation of which can change the functionality of such nodes dynamically. Examples of sensor-hosted, lightweight Virtual Machine environments include DAViM (Dynamically Adaptable Virtual Machine) [23], and Mate [24] – a Java byte-code interpreter that executed over Contiki. Mate breaks an application into *capsules* that can be distributed throughout the network at runtime. This approach is similar to the *Fog cells* idea being proposed in this paper.

This paper aims to determine how an edge ensemble could be used for a particular application use case from the area of fish processing industry. We utilise an edge ensemble to deploy EnergyPlus simulation instances and make use of an actual deployment to conduct energy optimisation within an edge distributed infrastructure. A key focus is on the use of a group of edge resources to enable: (i) fault tolerance, i.e. ensuring that failure/unavailability of edge resources can be compensated for by other nodes in the group; (ii) concurrency through the use of multiple resources within a group to improve performance; (iii) aggregation of different types of resources to offer service enactment at the network edge, with each resource offering different types of computational *functions*.

III. APPROACH

Energy costs represent a significant portion of operational expenditure in many industrial processes, it is therefore necessary to make use of intelligent techniques for managing energy demand – both production and consumption is important. This is also the case for the fish processing industry, where in order to improve efficiency and enable more active monitoring of operations, energy management systems have been developed that provide advanced controls such as motion sensors and other wireless sensors and metering devices that allow a high frequency of data capture. Such devices also enable automated generation and instant distribution of receiver-tailored and pre-processed information (raw data, consumption trends, deviation alarms, etc.), which can subsequently be used to improve operational processes within the industrial plant. Smart metering devices can perform triggered measurement and recording of electricity, water, or gas consumption at different levels within

a built environment/ facility (sub-metering) and allow for remote access to the consumption data (e.g. using power line, GSM, or standard wired communication protocols). In such systems, it is also possible to dynamically alter the rate at which data capture takes place, and in some instances provide mechanisms for automated actuation of systems within the industrial plant.

In such real-time energy systems, time associated with carrying out the energy optimisation process represents a key aspect for those managing the industrial process, aiming to minimise time and generate optimised set-points (identifying particular control objectives that need to be met by the industry managers). The time parameter is also important in real-time optimisation where delays can bring additional costs for the managers especially when site related parameters (such as temperature, occupancy, ice-flake, cold room, box-washer consumption, etc.) are frequently changing. The complexity of this process increases depending on the size of the fish processing site involved. In practice, such an optimisation process will require multiple executions of an optimisation simulation, using software such as EnergyPlus, with different parameter ranges. We consider two key parameters here: (i) *Complexity of the industrial model* has a direct impact on the overall simulation time; (ii) *Simulation period*, i.e. the time interval over which the energy optimisation is carried out, which can range from 1 week to 1 year, for instance. Therefore, the computing infrastructure required to support such complexity must comply with two parameters:

- Time-to-complete: An optimisation plan needs to be completed by a particular deadline. Assuming that sensors can deliver readings every 15 minutes, the optimisation process also needs to be carried out over an equivalent period. Each new execution uses as input the last configuration of the industrial process and set points (for various control outputs) associated with the industrial site.
- Results quality: An optimisation process, as identified in this study, involves running a number of EnergyPlus simulations. Depending on the complexity of the site and the period to simulate, a time interval is associated with each simulation. If suitable computational resources are not available, it may become necessary to sacrifice the quality of results and complete only a part of the required rounds of simulation in order to comply with the time deadline. Returning a partial optimisation result may have a twofold impact: (i) it reduces the number of resources needed to carry out the simulation/ optimisation; (ii) it influences the accuracy of the energy optimisation plan.

For instance, if a computing resource provider decides to stop the optimisation process after a certain number of simulations – lower quality results are returned to the user.

IV. EDGE ENSEMBLE AND SIMULATION

In this section we explain the overall process of the fish processing industry with particular emphasis on appliance consumption, simulation constraints and the use of edge resource ensembles. Our hypothesis is that an edge ensemble can support the execution of a workflow with a better performance, higher security and lower cost. An edge ensemble is therefore formed based on a number of functional criteria related to proximity, cost, performance and security.

- Proximity: the edge resources are identified based on geographical proximity (i.e. same building) facilitating advantages related to cost, latency and security.
- Cost: an edge ensemble can represent a more cost efficient solution and can bring significant benefits in relation to cost. While a regular edge system would use resources dispersed geographically, an edge ensemble can be formed based on cost constraints where the ownership of the edge resources can lead to reduced costs. An example is where a building user will form an ensemble with all the devices that exist within the building, obtaining a *no cost* (i.e. no additional resources need to be acquired from external sources) computing ecosystem.
- Performance (Latency): an edge ensemble can reduce the latency associated with data transfer due to proximity between the resources in the ensemble (i.e. distance measured in terms of no. of hops or aggregate transfer time/delay).
- Security: using an ensemble will reduce security implications as data will not be transferred outside of a local network (intranet).

A. Simulations with EnergyPlus

The simulation process represents a part of the optimisation workflow where simulation instances are launched using a number of different *what-if* scenarios, using different input parameter range(s). In the fish processing plant being considered here, this is achieved by the use of specialised software applications such as EnergyPlus, Simlink, Transys etc [25], [26]. This technique involves reducing the number of EnergyPlus instances by reducing the number of iterations (loops) used within this simulation. As EnergyPlus execution needs to be carried out over a particular time frame, we can limit the number of iterations based on the time available, thereby leading to a reduction in time over which EnergyPlus executions are carried out (within some pre-defined error threshold). In addition, EnergyPlus can be used as an energy simulation engine employing a simultaneous load/system/plant simulation methodology. In load calculation, various methods are available to calculate heat conduction through envelopes and then a heat balance method for zone load. Moreover, EnergyPlus makes use of a modular, loop-based method to simulate HVAC systems

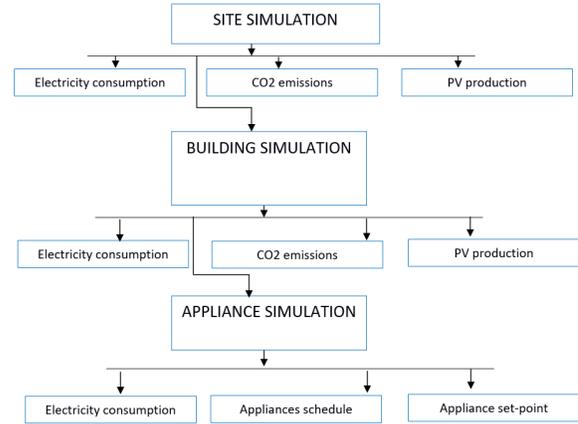


Figure 1: Overall simulation workflow for determining the optimised schedule

(cold rooms, compressors) which helps accelerate the modeling of an industrial process [27]. Through the use of a “Setpoint Manager” in EnergyPlus, many different kinds of parameters, such as supply air temperature and cold room temperature can be controlled and this function enables the implementation of a supervisory control capability.

The output of the simulation process are the optimised control set-points which make use of energy consumption forecasts (taking account of both energy production/availability and price/cost). These outputs can be used to determine the corresponding schedule of appliances, i.e. time intervals over which appliances should be turned on/off. An illustration of the overall process involved is outlined in Figure 1 and additional description of the process is provided in Section VI.

B. Workflow Approximation

We identify the following edge-ensemble approximation techniques applicable to the appliance optimisation/scheduling process:

EnergyPlus loop reduction: This technique implies the reduction in the number of EnergyPlus instances by updating the looping mechanism. As a simulation refers to period to simulate and the number of loops to be repeated, we can reduce the number of loops in order to reduce the number of EnergyPlus execution time while keeping the quality of results in a reasonable error interval. The error rate is: $err = 1/\frac{1}{times}$, where $times(loops)$ is the number of iterations over which the simulation is repeated.

Artificial Neural Networks (ANN): Such method involves the use of learning strategies and algorithms for replacing the EnergyPlus simulation. An ANN is trained on historical data and used to replace the EnergyPlus simulation – acting as a functional approximator for

EnergyPlus. The corresponding error rate is based on the size and variability of historical data and on the efficiency of the ANN algorithm. The error rate is approx.0.001.

Parameter value skipping: Based on a set of parameters that the simulation requires, this methods reduces the number of the parameter values which are used as input for the EnergyPlus simulation. The corresponding error rate of this method is based on the skipping interval. The associated error rate is $\frac{k}{100}$, where k is the number of parameter values skipped.

- Execution time without approximation techniques:
 $total\ time = n * m * time;$
- Execution time with approximation techniques
 $total\ approx.\ time = (n * m * total\ time) - (k * total\ time),$ n represents total number of parameters values, m is the number of parameters, and k represents the number of parameter values skipped;
- $error\ rate = \frac{k}{100}$

Parameter interval reduction: From the interval associated with a parameter we reduce the interval limits so the simulation would use only values from a defined centrality of the intervals. The error of this method depends on the remaining number of parameter values to use as input in the simulation. The error rate is: $\frac{n+k}{100}$, where n is the number of the total parameters and k is the number of intervals reduced.

- Execution time without approximation techniques:
 $total\ time = n * m * time;$
- Execution time with approximation techniques
 $total\ approx.\ time = (n * m * total\ time) - ((n - k) * total\ time),$ n represents total number of parameters values, m is the number of parameters, and k represents the number of parameter intervals reduced;
- $error\ rate = \frac{n+k}{100};$

V. EDGE ENSEMBLE MODELING

We consider a set of edge resources $E = \{e_1, e_2, e_3, \dots, e_m\}$, where each edge resource e_i has CPU, storage capacity and a *type*. We call an edge-ensemble E_e , any sub-set $E_e = \{e_1, e_2, e_3, \dots, e_s\}$, $E_i \in E$, that is formed based on pre-established functional criteria such as proximity, latency and cost. A simulation job is structured as a set of simulation task instances $S = \{i_1, i_2, i_3, \dots, i_p\}$. We use an edge ensemble $E_e = \{e_1, e_2, e_3, \dots, e_s\}$, $E_i \in E$, where on each edge ensemble resource e_i we allocate a simulation instance i_j . From the total p simulation instances, p will be submitted for execution on the edge resources existing within the ensemble E_i . To enable replication and make more efficient use of E_e we also consider that a simulation instance s_j can be replicated t times over the

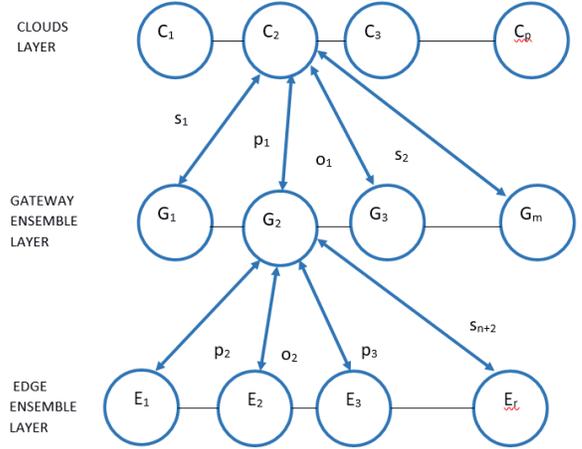


Figure 2: In edge computational layers

s edge resources in the ensemble E_i . We consider that all simulation task instances need to be accommodated within an ensemble and that the deployment of tasks outside the ensemble is limited (this is assumed to improve potential utilisation of edge resources). We consider different types of simulations, per appliance, per building and per site, o_i refers to per appliance tasks, p_i refers to per building simulation tasks and s_i refers to per site simulation tasks. The scenario associated with the simulation process is presented in Figure 3.

VI. MILFORD HAVEN PILOT PROJECT

Milford Haven port is considered the largest energy port, situated in the West of the UK, and the largest handler of oil and gas with capability of delivering about 30% of UK gas demand. The Milford Haven site also manages a fish processing plant where large quantities of fish are stored and delivered to other factories and supermarkets. In the port there are four main buildings producing energy from local Photo Voltaic (PV) systems and consume energy from the solar farm and from the national grid. Such buildings are: Packaway, K SHed, M Shed, F Shed. In the Packaway building (main port building) there are several appliances consuming energy: ice flake machine, ice store freezer, box washing machine, lighting systems, smart meter also PV generation solar system. The port also has a solar farm with ~ 5 Megawatt PV power capacity containing about 20000 panels that in combination with the local PVs provide the required quantity of electricity for the port. To optimise the entire site energy process including operating appliances, a simulation process is required where various input and output variables are optimised in order to meet a set of objectives.

The evaluation is based on the Packaway building which is the main building and contains several appliances consuming

energy: a flake ice machine, ice store freezer, box washing machine, lighting systems and smart meter. There are four storage room in the Packaway building and each storage room has a double tube lighting system. The box washing machine has a power capacity of 50kWh and it works only when fish storage boxes are cleaned during the day. Ice storage is under operation all the time in order to meet the demands for fish storage with the quantity required. The Packaway building has installed a PV System on the roof of the building with 50kW panels that feed the building at day time. They have a total power output of 275W per panel with two DC-AC inverters. The list of appliances in Packaway and associated constraints are presented in Table I

VII. EVALUATION

In our evaluation we have the following objectives: (a) investigate how to complete a bag of EnergyPlus tasks under no criteria or policies for tasks allocation (i.e. execute tasks on any available computational resource); (b) investigate how to execute jobs on an edge-ensemble basis where the task execution is distributed on ensemble of edge resources and (c) investigate the edge-ensemble with replication approach where sub-tasks are allocated on the resources in the ensemble, making use of a replication mechanism.

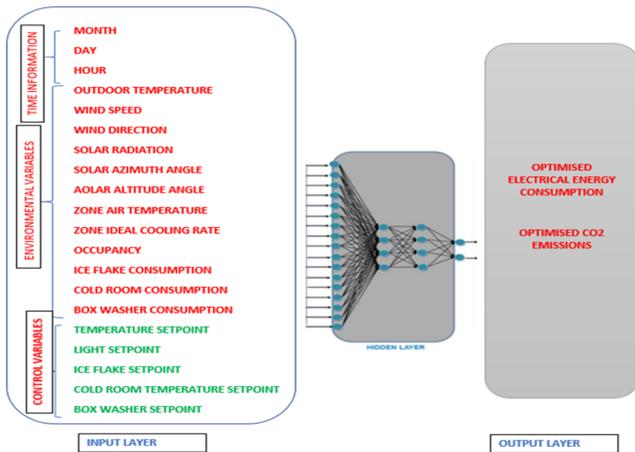


Figure 3: Simulation scenario

A. Simulation scenarios

In the Packaway pilot optimisation we have used the following objectives: (i) reduce electrical energy consumption and (ii) reduce total carbon emissions from the plant. These objectives relate to reducing energy consumption needed, whilst maintaining normal operation of all appliances involved in the the fish processing process. The Packaway energy consumption is the total consumption of all appliances used: (i) cold room consumption (E_c), (ii) ice flake consumption (E_i), (iii) box washer consumption (E_b) and (iv) lighting system consumption (E_l), therefore

energy consumption function can be described as follows: $E = E_c + E_i + E_b + E_l$, where E represents total energy consumed by the factory facility. In a simulation scenario, we are interested to determine what is the optimum consumption and optimum appliance set-point to use in the application scheduling decision process. The range of decisions are as follow: (i) Schedule appliances, (ii) Use for operation, (iii) Store in battery, (iv) Sell to national grid (i.e. return generated energy to the national grid via an inverter) or (v) a combination of these options such as: Sell and Store.

B. Experiments

This section presents the overall setup for our infrastructure for evaluation purposes, identifying the scenarios and results that show the benefits of using an edge ensemble approach.

C. Configuration of Testbed and Scenarios

The CometCloud [28] federation has been deployed on CloudLab [30] which is a distributed testbed for the computer science research community. We generally considered two types of resources: (i) resources at the Cloud that are located within the well-provisioned data center and are far away from the data source. (ii) Edge resources which are located within proximity of the data source. Hierarchy Token Bucket (HTB) has been utilized to set the bandwidth between resources and emulate geographic distribution of the resources. Thus, 40 Mbit/s and 120 Mbit/s has been considered for the network connection between source and core, and source and edge, respectively.

To model the various scenarios, we used the characteristics of Amazon EC2 VM instances in our model. Summary of resource characteristics have been shown in table II.

Based on the computational capabilities at the edge and core resource, we considered three different types infrastructure:

- Core Resources: That are located within well-provisioned data centers. We considered 8 instance of c4.4xlarge VMs for our core infrastructure.
- Edge Clouds: These are small-scaled cloud resources located at the edge of the network. 3 c4.4xlarge VMs has been allocated to create our edge clouds.
- Edge Devices: These are IoT devices that are located very close to the data source and within the industrial fish factory. Compared to edge clouds and core resources, these devices are weaker with small amount of computational capabilities. Hence, we allocated 24 c4.2xlarge VMs for edge devices. Each VM represents one IoT device.

Our infrastructure network of devices is depicted in Figure 4.

Our federated marketplace spans across all available resources. Once a data processing request is inserted to our federation, each available resource puts a bid on the job

Table I: Description of appliances

Appliance	Power Rating	Minimum Running Time	Interruption	Required Usage	Required Start Time
Box washing machine	50 KWh	30	Not Possible	Once a day	Between (6:00-16:00)
Ice Flake machine	30kWh	60	Possible	Twice a day	Between (6:00-16:00)
Cold storage room	30kWh	180	Possible	Twice a day	Between (0:00-23:45)
Lighting system	25 W/per tube	60	Possible	Twice a day	Between (0:00-23:45)

Table II: Resource Properties

Resource Type	vCPU	ECU	Memory	Price (\$/Hour)
c4.2xlarge	8	31	15	0.464
c4.4xlarge	16	62	30	0.928

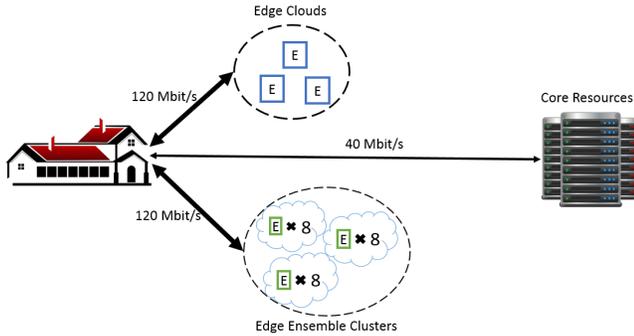


Figure 4: Infrastructure Setup.

which is based on the status of the resources, i.e., whether they are able to process such request or not.

D. Experiments

For the use case, we considered that there are multiple buildings in the industrial site requesting to evaluate and optimise their energy consumption. Considering the Milford Haven site, three types of jobs have been considered in this work. Table III summarize the characteristics of different job types. To eliminate imposed cost limitation on the system, budget has been chosen high enough for the jobs. Hence, the deadline is the only limiting factor in our experiments. Table IV collects the estimated execution time of the job on two available VM types.

Table III: Job Information.

JobType	Data Size(MB)	Budget	Deadline(s)	Tasks [†]
JobType1	50	20	120	16
JobType2	100	30	150	24
JobType3	150	40	180	32

[†] – A job is composed of a set of tasks

In total we explore four different scenarios in this work, as listed below:

- **Traditional Approach- Core Only (T-C):** In this scenario, Core resources are used to complete a bag of EnergyPlus tasks under no criteria or policies for

Table IV: Time to completion of EnergyPlus job types.

JobType	c4.2xlarge	c4.4xlarge
JobType1	160 s	80 s
JobType2	200 s	100 s
JobType3	240 s	120 s

tasks allocation. We assume that a bag of tasks is submitted for execution on one of the available resources. One of the Core resources which can process the job within deadline and budget is picked randomly. In this scenario, edge clouds and edge ensemble clusters are disabled.

- **Traditional Approach- Edge and Core Only (T-E-C):** In this scenario we enabled all of the edge resources (edge cloud and devices within edge ensemble clusters) to contribute to task execution. However, since edge devices within edge ensemble clusters are operating individually, their computational capabilities are not enough to process the job within the deadline. Hence, in practice edge devices cannot execute any of the jobs and only edge clouds and core resource can execute the requests.
- **Edge Ensemble (EE):** In this scenario, the edge clouds and core resources are disabled. However, IoT devices are able to (dynamically) create edge ensemble clusters. We create three edge ensemble clusters (each has 8 VMs). Although these devices are not powerful enough to process the job as stand alone resource within the deadline, the cluster of devices are powerful enough to process the jobs within deadline.
- **Edge Ensemble Replication (EER):** This strategy is similar with the edge-ensemble approach (EE) where subtasks are allocated on the resources in the ensemble devices but adding a replication mechanism where another ensemble devices within the cluster is also used to ensure that the tasks execution are replicated on other edge devices within the same ensemble cluster. This would be implemented based on different probabilities of failure (failure models, i.e. stack models). In this scenario, part of the jobs are replicated to overcome failures. Hence each individual edge node executes more tasks per job compared to previous scenario. In case of failure at each device, the replica tasks can be used for execution.

For all of the scenarios mentioned above, we conducted

the experiments and compared these scenarios using various parameter ranges. In each experiment, 382 jobs were inserted from source/BMS (Building Management System) to a federated marketplace, generated using a Poisson distribution. Once a job was inserted in the federation, all of the available resources offered their services using a blind auction mechanism. If edge ensemble clustering is enabled, one of the devices within the cluster is responsible for offering the services made available within the cluster. From all available bids, we chose one of the services randomly and send the job and its data to the corresponding resource(s) for processing.

In order to evaluate the impact of using the edge ensemble approach, we measured job acceptance ratio, which is the ability of resources to accept and execute the jobs within both deadline and budget constraints. Figure 5 demonstrates the percentage of the jobs that the system is able to process. Although edge devices within the edge ensemble are not able to process the jobs within the deadline on their own individually, once they form the edge ensemble cluster (form a cluster containing 8 IoT devices) they create an aggregate resource capable of undertaking task execution. Figure 5 shows that EE scenario is able to process all of the jobs compared to T-C and T-E-C strategies which can process around 32% and 52%, respectively. EER strategy performs better compared to the traditional approach. However, since EER replicates the task within a job by a factor of 2, part of its resources are doing redundant executions and as a result EER accepts around 72% of the jobs which is 28% less than the EE approach.

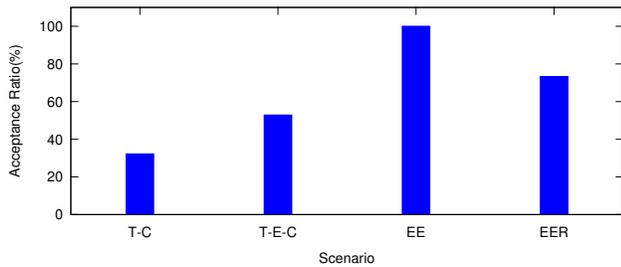


Figure 5: Job Acceptance Ratio.

Figure 6 compares the completion time of all jobs in the system, calculated as the time since a job was inserted until it was processed. It is observed that all of the accepted jobs are completed before the required deadline. In general, edge ensemble approaches (EE and EER) show more variance compared to the traditional approaches since dividing tasks across the available IoT devices causes concurrent execution of the tasks, leading to jobs executing faster. However, the average execution time in all strategies are similar because in both EE and EER approaches, more jobs are accepted which cause larger queue time for the accepted jobs. In conclusion, in both EE and EER scenarios jobs have a large queuing delay.

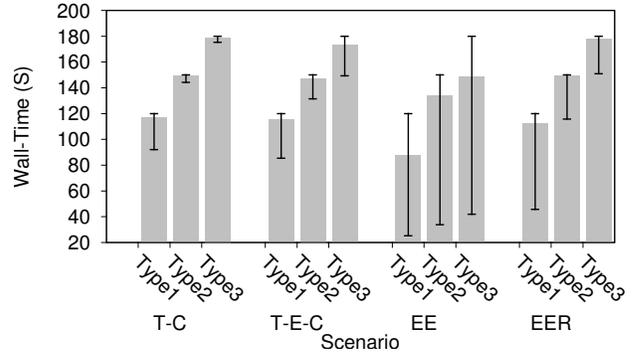


Figure 6: Average Completion Time.

Next, we studied the data transfer time for these different strategies. As shows in Figure 7, the inclusion of edge resources can reduce the data transfer time because the network connection between source and edge resource has higher bandwidth. In T-E-C approach, since both of the core and edge are available, the average data transfer time is lower than T-C and higher than EE and EER strategies where core resources are disabled.

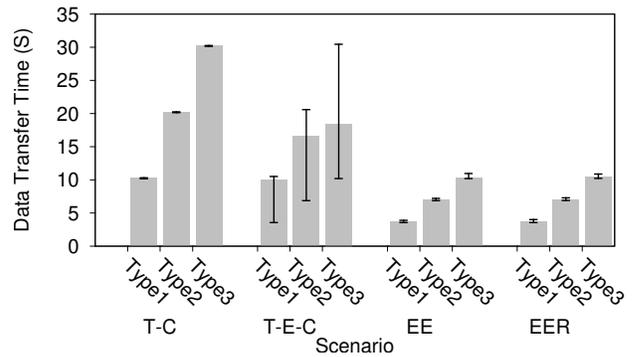


Figure 7: Data Transfer Time.

Finally, we conducted another set of experiments and included random failures in edge resource while the jobs are being executed. This is motivated by the observation that edge resources are likely to have the greatest *churn* rate due to their limited battery life or configuration parameters. Figure 8 presents the job acceptance ratio while 10%, 20%, 30% and 40% random failures are considered within such edge resources. Other than EER scenario, as possibility of the failure increases, the job acceptance ratio decreases since failure causes less successful job completion, as expected. In EER approach since each task is replicated by a factor of two, it means that if one task fails, its replica will be use instead. Hence, failures do not affect the job acceptance ratio in EER. For example, EER strategy performs better if possibility of failure exceeds 30%. These simulations can be used to support capacity planning in a system comprising

edge resources, where the expected (or observed) failure rate can be used to determine the number of replicas needed.

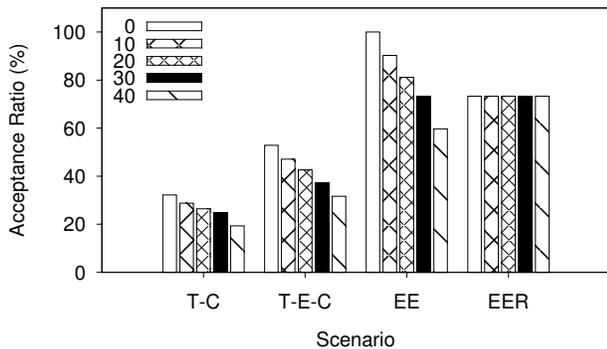


Figure 8: Job Acceptance Ratio with Random Failure Addition. Random Failure of 0% means no failuer.

VIII. CONCLUSIONS

Supporting cost effective operations of industrial processes requires intelligent techniques for the controlling and scheduling of available hardware and systems. Energy use is a major factor in the overall operational costs of such systems. Developing timely forecasts for energy consumption and generation has a significant influence on managing these costs. Recently available independent and wireless sensor technologies provide finer grained (and higher frequency) monitoring of operational parameters within industrial processes, leading to improved capture and processing of information (raw data, consumption trends, deviation alarms, etc.). Edge-ensemble proposes a more close-to-source method to aggregate available computational, storage, and networking resources for supporting data captured from such industry sensors/systems. Increasing use of IoT devices within industrial processes, and subsequent use of data from these to inform decision support, has also increased in recent years. The approach proposed in this work can be directly made use, enabling greater computational capacity to be made available alongside the deployed IoT devices (near the network edge).

Understanding how an ensemble (group) of edge resources can be used at the network edge to support real-time energy simulation has been investigated. Utilising multiple resources within the ensemble enable: (i) distribution of workload across different devices at the network edge, enabling the dynamic formation of an edge resource computation/storage cluster; (ii) enable fault tolerance to be realised through the replication of functions across edge resources. We have presented the design and implementation of the proposed approach and experimentally evaluated a number of scenarios for supporting edge-ensemble based optimisation. The experimental results show a number of benefits that our system provides with regards to task completion and data

transfer. We demonstrate how edge-ensembles can represent a reliable solution for coordinating a complex industrial process. In addition, edge ensembles can enable a more secure and efficient way to execute tasks with significant improved coordination and reliability.

Acknowledgements: The research presented in this work is supported by EU INTEREG *piSCES* project, Smart Cluster Energy System for Fish Processing Industries. Also, this research is supported in part by NSF via grant number OCE 1745246.

REFERENCES

- [1] A. R. Zamani, I. Petri, J. Diaz-Montes, O. Rana and M. Parashar. Edge-Supported Approximate Analysis for Long Running Computations. 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud), Prague, Czech Republic, 2017, pp. 321-328. doi:10.1109/FiCloud.2017.24
- [2] A. Fazeli, E. Christopher, C. M. Johnson, M. Gillott and M. Sumner. Investigating the effects of dynamic demand side management within intelligent Smart Energy communities of future decentralized power system. 2011 2nd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies, Manchester, 2011, pp. 1-8. doi: 10.1109/IS-GTEurope.2011.6162619.
- [3] Dong Sig Chai, John Z. Wen, Jatin Nathwani, Simulation of cogeneration within the concept of smart energy networks, Energy Conversion and Management, Volume 75,2013,Pages 453-465, ISSN 0196-8904, https://doi.org/10.1016/j.enconman.2013.06.045.
- [4] Dastjerdi AV, Gupta H, Calheiros RN, Ghosh SK, Buyya R (2016) Fog computing: principles, architectures, and applications. In: Internet of things: principles and paradigms, chap. 4, MorganKaufmann
- [5] Fernando N, LokeSW, RahayuW(2013) Mobile cloud computing: a survey. Future Generation Computer Systems 29:84106
- [6] Elmroth E, Leitner P, Schulte S, Venugopal S (2017) Connecting fog and cloud computing. IEEE Cloud Comput 4(2):2225
- [7] Atzori L, Iera A, MorabitoG(2010) The internet of things: a survey. Computer Networks 54:27872805
- [8] Dastjerdi AV, Gupta H, Calheiros RN, Ghosh SK, Buyya R (2016) Fog computing: principles, architectures, and applications. In: Internet of things: principles and paradigms, chap. 4, MorganKaufmann
- [9] Rohjans S,DnekasC,UslarM(2012) Requirements for Smart-Grid ICT-architectures. In: 3rd IEEE PES international conference and exhibition on innovative smart grid technologies. Berlin, Germany, pp 18
- [10] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, Fog computing: A platform for internet of things and analytics, in Big Data and Internet of Things: A Roadmap for Smart Environments. Springer, 2014, pp. 169186.

- [11] M. Aazam and E.-N. Huh, Fog computing and smart gateway based communication for cloud of things, in Intl. Conference on Future Internet of Things and Cloud (FiCloud). IEEE, 2014, pp. 464470.
- [12] A. R. Zamani, M. Zou, J. Diaz-Montes, I. Petri, O. Rana, A. Anjum and M. Parashar. Deadline constrained video analysis via in-transit computational environments. IEEE Transactions on Services Computing, 2017.
- [13] A. R. Zamani, M. Zou, J. Diaz-Montes, I. Petri, O. Rana, and M. Parashar. A computational model to support in-network data analysis in federated ecosystems. Future Generation Computer Systems, 80 (2018): 342-354.
- [14] M. Zou, A. R. Zamani, J. Diaz-Montes, I. Petri, O. Rana, and M. Parashar. Leveraging in-transit computational capabilities in federated ecosystems. 2016 IEEE Symposium on Service Oriented System Engineering, pp. 81-90.
- [15] W. Kurschl and W. Beer, "Combining Cloud Computing and Wireless Sensor Networks". In Proc. of iiWAS09, 2009.
- [16] X. Chu, and R. Buyya, "Service-Oriented Sensor Web". In: Sensor Networks and Configuration. Springer, 2007.
- [17] A. Cuzzocrea, F. Furfaro, S. Greco, E. Masciari, G.M. Mazzeo and D. Sacca, "A Distributed System for Answering Range Queries on Sensor Network Data", in Proc. of PerComW05, 2005.
- [18] A. Cuzzocrea and D. Sacca, "Exploiting Compression and Approximation Paradigms for Effective and Efficient OLAP over Sensor Network Readings in Data Grid Environments", Concurrency and Computation: Practice and Experience, 2013.
- [19] M. Yuriyama and T. Kushida, "Sensor-Cloud Infrastructure-Physical Sensor Management with Virtualized Sensors on Cloud Computing", In Proc. of NBiS'10, 2010.
- [20] G. Fortino, D. Parisi, V. Pirrone, G. Di Fatta, "BodyCloud: A SaaS Approach for Community Body Sensor Networks". Future Generation Computer Systems, 2014, Vol. 35, n. 6, pp. 62-79. Elsevier.
- [21] G. Fortino, A. Guerrieri, G. O'Hare, A. Ruzzelli, "A Flexible Building Management Framework based on Wireless Sensor and Actuator Networks", Journal of Network and Computer Applications, 2012, Vol. 35, n. 6, pp. 1934-1952.
- [22] S. Distefano, G. Merlino, and A. Puliafito, "SAaaS: a Framework for Volunteer-Based Sensing Clouds". Parallel and Cloud Computing 1(2), 2012.
- [23] S. Michiels, W. Horre, W. Joosen and P. Verbaeten, "DAViM: a dynamically adaptable virtual machine for sensor networks" Proc. of Int. Work. on Middleware for sensor networks (Mid-Sens), pp 7-12, Melbourne, Australia, November 28 2006. ACM Press.
- [24] P. Levis and D. Culler, "Mate: A Tiny Virtual Machine for Sensor Networks", Proc. of the 10th Int. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS X), Oct. 2002. Available at: <https://sing.stanford.edu/site/publications/2>
- [25] N. Fumo, P. Mago and R. Luck, "Methodology to Estimate Building Energy Consumption Using EnergyPlus Benchmark Models." Energy and Buildings, (42:12); pp. 2331-2337, 2010.
- [26] V. Garg, K. Chandrasen, S. Tetali and J. Mathur, "Energy-plus Simulation Speedup Using Data Parallelization Concept", ASME Energy Sustainability Conference, New York: American Society of Mechanical Engineers. pp. 1041-1048, 2010.
- [27] R. K. Strand, "Modularization and simulation techniques for heat balance-based energy and load calculation programs: the experience". In: of the ASHRAE Loads Toolkits and Energy-Plus, in: Proceedings of Building Simulation 2001, IBPSA, Rio de Janeiro, 747-753.
- [28] CometCloud Project. <http://nscac.rutgers.edu/CometCloud/>. Last accessed: July 2018.
- [29] M. Parashar, M. Abdelbaky, I. Rodero, A. Devarakonda, "Cloud Paradigms and Practices for Computational and Data-Enabled Science and Engineering", IEEE Computing in Science and Engineering (CiSE) Magazine, to appear.
- [30] CLOUDLab. <https://www.cloudlab.us> (Last accessed on May 2017.).