From Insight to Impact: Building a Sustainable Edge Computing Platform for Smart Homes

Xiaomin Chang[1], Wei Li[1], Chunqiu Xia[1], Jin Ma[2], Junwei Cao[3], Samee U. Khan[4], and Albert Y. Zomaya[1]
[1]Centre for Distributed and High Performance Computing, School of Information Technologies, The University of Sydney, Australia
[2] School of Electrical and Information Engineering, The University of Sydney, Australia
[3] Research Institute of Information Technology, Tsinghua University, China
[4] Department of Electrical and Computer Engineering, North Dakota State University, USA
xcha8737@uni.sydney.edu.au, weiwilson.li@sydney.edu.au, cxia3271@uni.sydney.edu.au, j.ma@sydney.edu.au,
jcao@tsinghua.edu.cn, samee.khan@ndsu.edu, albert.zomaya@sydney.edu.au

***Abstract*—There is a growing trend for engaging edge computing to help smart homes to improve the living comfort of residents. However, the rapidly rising interest in such deployments does not normally integrate with energy management schemes, which is a central issue that smart homes have to face. In this paper, we propose a unified energy management framework for enabling a sustainable edge computing paradigm while meeting the needs of home energy management and smart home applications. This framework aims to enable the full use of renewable energy while reducing electricity bills for households. A prototype system was implemented by using low-cost and easy-to-get hardware. The experiment results demonstrated that renewable energy is fully capable of supporting the reliable running of edge computing devices and electricity bills could be cut by up to 86% when our proposed framework was employed.**

***Index Terms*—Internet of Things, Edge Computing, Smart Homes, Renewable Energy, Sustainable Computing, Scheduling, Energy Management**

I. Introduction

With the advent of Internet of Things (IoT) [1], the increasing use of smart devices opens up new avenues to promote the development of smart homes, which aim to improve living comfort and make daily life easier. The rapid growth of IoT-based smart home systems will certainly produce a large amount of streaming data, and such data is expected to be processed on easily accessible computing resources for the sake of performance and efficiency. Edge computing [2], as an emerging distributed computing paradigm, has recently been introduced to the field to enable data processing and on-demand services at the edge of networks where such functions and/or services were normally available at far-end data centres. It can thus reduce service latency and improve QoS (Quality of Service) for time-constrained IoT applications, and provide better overall experience to end users [3] [4] [5].

Currently, significant research efforts are focused on improving the features of flexibility [6], scalability [7], [8] security [9] [10] [11], programmability [12] and real-time processing [13] [14] of edge computing systems better cope with IoT applications. However, the energy consumption of edge computing systems is becoming a noticeable issue if they are all powered by the utility grid. As reported in [15], a significant portion (over 80%) of today's energy is still generated by fossil fuels (brown energy). As is well known, the widespread use of fossil fuels is implicated in global climate warming. Carbon taxes have helped to reduce the rapid anthropogenic release of carbon dioxide from fossil fuel. However, they have not only driven up electricity prices for residents, they have also incentivized reductions in total energy demand as well as shifts toward using renewable energy.

The development of home energy systems [16] has successfully helped users to reduce electricity bills by shifting loads from peak hours to non-peak hours. To take a step further, lowering the carbon footprint of smart homes requires using renewable energy as the primary and brown energy as the secondary energy supply. The relatively small scale of the deployments of at-home edge computing systems puts them in a better position to make effective use of renewable energy. Moreover, if smart homes are equipped with one or more energy storage device(s), energy generated from local renewable energy sources can be used to charge the storage when the electricity cost is low and discharge the storage during high-cost periods. The use of storage in conjunction with renewable energy sources is helpful to optimise the cost effectiveness of smart homes. Unfortunately, local renewable energy generation and its proper use are misaligned with most home energy systems in smart homes.

To address these issues, it is a strong motivation for us to propose a unified energy management framework on edge computing systems to handle both renewable energy management and smart home applications. The proposed energy management framework fully utilises the local renewable energy of a household to increase the effectiveness and utilisation of energy resources while still meeting the requirements of IoT applications for smart homes. The energy management framework is designed to run on single-chip computers at the consumers' premises, so that privacy-sensitive data can be processed and stored locally. Besides, our approach is highly scalable once the computational resource is needed and has low cost of capital investment. The main contributions of this paper are:

(i) We explore the idea of using edge computing systems to provide an integrated platform for managing the use of renewable energy and brown energy, as well as processing smart home applications.

(ii) We propose a unified energy management framework that minimises negative environmental impact of using brown energy and significantly reduce the cost of running edge computing systems in homes.

(iii) We conduct a real-world empirical evaluation of the proposed framework using low cost and easy-to-get hardware.

In the rest of this paper, we first provide a brief overview on edge computing and its development in Section II. Then, we introduce the architecture of our designed sustainable edge computing platform for households, and show how the key modules enable the features of data processing and energy management in Section III. In Section IV, we present an optimal forecasting method integrated with receding horizon control mechanism, which significantly supports to reduce forecasting errors. Moreover, in this section, we also present two energy scheduling strategies, which cover different working scenarios, and a cost-effective algorithm for task scheduling on the edge computing system. After that, Section V presents the experimental studies for the proposed system architecture and the scheduling algorithms, and then the experiment results are analysed. Finally, the conclusion and future work are given in Section VI.

II. Related Work

In the past few years, edge computing has gained popularity as a promising technology to be integrated with IoT and Cloud. Most of existing works on this area is driven by the purpose of using edge computing paradigm to meet the needs of IoT applications [17], such as: location-aware processing, privacy and energy consumption. As one of such IoT applications, the existing home energy management systems (HEMS) can be certainly benefited from the involvement of edge computing.

A. Edge Computing

To realise an edge computing platform that provides the needed services and be well integrated with the IoT ecosystem, there are a few work on utilising different technologies to build up the standalone edge computing platforms.

In [18], Cumulus is introduced as a distributed edge computing testbed, which is designed for edge cloud computational offloading. Several performance metrics are taken into consideration during the design, including task execution time, CPU usage and average memory usage. However, the intrinsic features on low compatibility and poor scalability limits its applicability to individual cases. To avoid the same issue, [19] used the popular virtualisation technique - Docker in their work. Compared to the traditional virtual machines, Docker can reduce the boot-up overhead and avoid the conflict of resource allocation to a large extent. It also addresses the limitation of compatibility and scalability, which enables the testbed can better capture the orchestration patterns, conduct the cost-effective resource management, and perform various types of IoT applications regardless of their running environment. This design is implemented on a Raspberry Pi cluster to demonstrate and quantify the hardware feasibility, but more experiments need to be performed to evaluate the stability and energy consumption of the framework. Another docker-based edge computing architecture proposed by [20]. In this work, the docker containerisation is introduced in support of implementing IoT gateway, which can be dynamically configured with respect to multiple scheduling demands, and responds to various service requests.

B. Home Energy Management Systems

Due to the needs of saving energy and reducing electricity bill for each household, HEMS has become one of the most popular research topics in the context of smart homes. Despite of current solutions, the presence of either edge computing or sustainable energy is largely missing from both open-source platforms, e.g. OpenHAB [21], Dreamwatts® [22] and commercial platforms, e.g. Samsung SmartThings [23], LG Smart Thinq™ [24].

In [25], an initial attempt on enabling energy management on edge/fog computing platform is presented. The authors conducted a study of performing energy management and the related control algorithms over a Fog computing platform, which aims at demonstrating the feasibility, scalability, adaptability and interoperability of the design. In addition, such a design is implemented as a HEMS prototype to coordinate with a lab-level microgrid for the HVAC (Heating, ventilation, and air conditioning) applications. The numerical studies of the above implementation show that the results are promising. However, this platform does not address the integration of sustainable energy.

In an effort to address the issue of using sustainable energy to support the running of HEMS, we propose a low-cost sustainable edge computing system design as a home-level extension of the exploratory design of our sustainable edge computing systems [26], which can not just reduce the electricity tariff, and also meet the needs of smart home applications and home energy management. The motivations of the deployment of our edge-based HEMS framework include (i) well utilise the possibility of accessing the share resources from the edge devices within the same house so as to processing the data streams produced by smart devices in a timely manner, (ii) lowering the energy cost of edge computing systems by employing the optimal scheduling strategies, and (iii) allocating available renewable energy to edge device(s) in a cost-effective way with the support of an optimal forecasting method.

III. System Architecture Overview

Figure 1 represents a systematic overview on the functional modules developed in our design, and their interactions in an edge device. There are in total four major modules that provide all of the functionality required by the energy management as well as the applications of smart home, namely, weather forecast module, energy generation prediction module, energy scheduler module and data processing module.

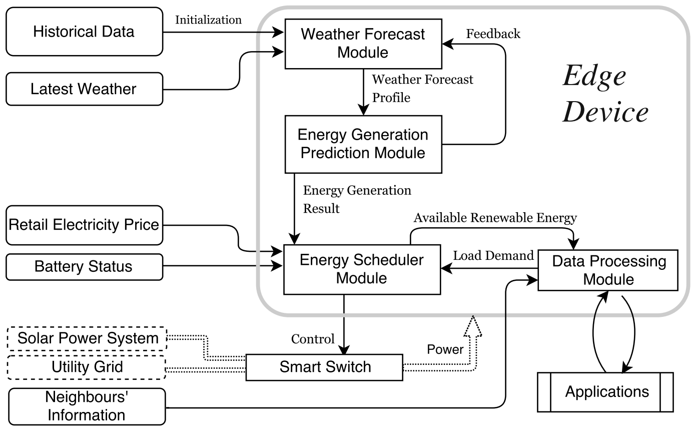

Fig. 1. System Architecture

The weather forecast module is used to predict weather parameters that impact solar energy generation, such as global horizontal irradiance (GHI) and temperature. At the beginning of a day, the raw historical weather data of the present location is uploaded to this module. An operational mesoscale weather forecast model is responsible for day-ahead forecast and then the initial weather forecast profile is generated. However, solar irradiance variations are tightly coupled with cloud movement, cloud formation, and dissipation [27] that are hard to predict day-ahead accurately, which means the day-ahead forecast error is often far from desirable and less than ideal for on-site use. To address this issue, solar forecasts are expected to span over different time horizons and granularity. In this paper, we adopt receding horizon control [28] strategy to handle the highly dynamic and partially known environments. By using this strategy, the intra-day forecast will be performed repeatedly on user-defined granularity once the up-to-date weather information is retrieved. In each intra-day forecast, the full time horizon forecasting is still conducted, but only the first user-defined granularity of the newly generated profile will be used so as to increase forecast accuracy to a large extent.

The energy generation prediction module is used to estimate the solar generation based on the profile forwarded from the weather forecast module, which contains the relevant weather variables on solar power generation. Photovoltaics is the process of converting sunlight directly into electricity using solar cells. According to the photovoltaic properties of solar cells described in [29], the open-circuit voltage decreases along with increasing temperature whereas current output remains unchanged, almost equal to short-circuit current which just increases slightly with increasing temperature. For the dependency of light intensity, the generated current is linear to the solar radiation. With these principles, working current and working voltage can be first obtained from the V-I characteristics of solar cells respectively, and then the solar power generation can be determined from them.

Once the result from the energy generation prediction module is ready, it will be passed to the energy scheduler module as an input for deciding whether the electricity should be drawn from the solar power system or the utility grid. Besides the energy generation result, the status of the energy storage, load demand and the retail electricity price are the other major inputs for making the feasible energy scheduling decision. To achieve the goal of minimising electricity cost and carbon emissions, we develop two energy scheduling algorithms based on dynamic programming and greedy-like approach, each of which is used in different scenarios. The use of renewable energy is thus maximised without jeopardising the functionality and the reliability of the edge computing system. Moreover, new scheduling scheme is periodically derived for each time slot, implicitly guaranteeing that receding horizon control mechanism is applied throughout the whole system and each derived energy scheduling scheme contributes to minimising the energy cost. More details will be discussed in Section IV.

The data processing module is used for completing the processing of smart home applications in a timely and effective manner while still meeting the needs of users. To make our solution scalable and adaptive, we are inspired by the proposal of Distributed Weighted Backpressure (DWB) [30], which is an online and distributed system control policy that asymptotically minimises the computation and communication cost of a Fog-integrated IoT ecosystem. By refining and extending this policy, our design is able to fully exploit the capabilities of edge devices so as to complete the smart home applications collaboratively. Furthermore, it aims to achieve load balancing among edge devices used in our system, which naturally leads to the maximum usage of green energy.

IV. Green Energy Management Design

The overall goal of our proposed energy management framework is to maximise utilisation of renewable energy while still meeting the needs of users. However, it is challenging to determine the optimal energy allocation of renewable energy because of the power fluctuation caused by the intermittency of renewable energy generation. In this section, we introduce our preliminary skeleton to facilitate the allocation of available renewable energy for edge computing systems by using the rolling control strategy for both forecasting and scheduling functionalities to achieve the aforementioned goal.

A. Forecasting

Most existing forecasting models using day-ahead forecast manner are not able to reflect actual environment changes perfectly, and will incur considerable errors leading to huge deviations from real measurements, which is highly likely to increase inaccuracy of energy scheduling and the cost of electric power drawn from utility grid. To alleviate these existing issues, we adopt receding horizon control in our design. According to [28], receding horizon control is an optimal control technique, by using which system is expected to update action plan and calculation results periodically based on feedback from last time step in order to minimise errors in runtime. In this control manner, we introduce intra-day forecast in the forecasting module, combined with day-ahead forecast.

As mentioned before, our weather forecast module consists of two processing stages, a day-ahead solar forecasting stage and an intra-day solar forecasting stage. For day-ahead solar forecasting, the numerical weather prediction (NWP) models [31] are preferably used for the early quantitative prediction of solar irradiation forecasts. However, NWP models could exhibit substantial biases when conducting the long term prediction (from six hours up to days ahead). To refine the results of NWP models, post-processing methods are routinely employed to reduce these biases since the local weather features are generally not involved in NWP predictions [32]. Furthermore, the method of spatial averaging is effective at reducing forecast errors, particularly for the situations with variable clouds that are hard to predict precisely [33].

To further improve the reliability and validity of solar forecast, the receding horizon control strategy is employed. This strategy allows us to improve the result of day-ahead forecast over a fine granularity time horizon, starting from the currently predicted data, and complete the refinement, then move into the next time granularity and re-optimise. To conduct the intra-day prediction, we employed a well-known and proven linear technique in the field of solar forecast, auto regressive moving average (ARMA) model [34], in our system. The standard setting of the general form of ARMA is based on the least squares method and thus a training data set is used to estimate the set of parameters. To reduce the computational cost for estimating the model's parameters, we chose recursive least square method, which is a variation of the least squares method but no training set is required. Hence, the recursive least square method is particularly well suited for on-site scenarios where forecasts have to be performed in a timely manner.

B. Energy Scheduling

To achieve the design goal of the system, we employed two different scheduling strategies to cooperate with the forecasting module, one is for the energy scheduler module and the other is for the data processing module.

For the energy scheduler module, the edge device is expected to generate a cost-effective schedule with respect to solar forecast, electricity price and battery status. As mentioned before, the solar forecast is generated according to the user-defined time granularity, which makes once the scheduling decision is determined, the rest of that time in a granularity will follow the same decision. At the beginning of each user-defined time granularity $t_i \in (t_1, \ldots, t_n)$ over a day, the energy scheduler module first extracts the related information from the inputs, and then such information is stored in a tuple $(l_{t_i}, p_{t_i}, g_{t_i})$. In this tuple, l_{t_i} represents the energy demand over t_i; p_{t_i} is electricity price over t_i, and g_{t_i} represents solar generation forecast over t_i. It is worth mentioning that most existing work assumes that residential users can get the realtime or near-realtime electricity price from the electricity market. However, in Australia, the electricity market is operated by the Australian Energy Market Operator, which offers electricity to the energy retailers at every five-minute interval with the average price over the past 30 minutes. The final electricity price that residential users get from energy retailers is stepped over a day, which can be denoted as $p_{t_i} \in (p_{peak}, p_{shoulder}, p_{offpeak})$. If the edge computing system draws power from solar panels, the price is set as zero. To maximise the utilisation of solar energy, the excess power generated from the solar panel is expected to be stored into battery as much as possible, thus we use $E_{battery}(t)$ to denote the amount of the energy left in the battery at the start of time granularity t_i, which can also be deemed as battery status. The battery capacity is represented as $E_{battery}^{max}$, and $E_{battery}(t)$ is expected to be equal to or less than the value of $E_{battery}^{max}$ at any time. Based on the above notations, we can formally define the objective of the energy scheduling problem as follows.

$$\textbf{Minimize} \left\{ \sum_{t_i = t_1}^{t_n} l_{t_i} \times t_i \times p_{t_i} \times x_{t_i} \right\} \tag{1}$$

subject to

$$l_{t_1} \times t_1 \times \overline{x_{t_1}} \le g_{t_1} \times t_1 + E_{battery}(t_1); \tag{2}$$
$$l_{t_2} \times t_2 \times \overline{x_{t_2}} \le g_{t_2} \times t_2 + E_{battery}(t_2); \tag{3}$$
$$\ldots$$
$$l_{t_{n-1}} \times t_{n-1} \times \overline{x_{t_{n-1}}} \le g_{t_{n-1}} \times t_n + E_{battery}(t_{n-1}); \tag{4}$$
$$l_{t_n} \times t_n \times \overline{x_{t_n}} \le g_{t_n} \times t_n + E_{battery}(t_n); \tag{5}$$
$$E_{battery}(t_1) \le E_{battery}^{max} \tag{6}$$
$$E_{battery}(t_2) \le E_{battery}^{max} \tag{7}$$
$$\ldots$$
$$E_{battery}(t_{n-1}) \le E_{battery}^{max} \tag{8}$$
$$E_{battery}(t_n) \le E_{battery}^{max} \tag{9}$$

$$\begin{aligned} E_{battery}(t_i) = E_{battery}(t_{i-1}) + (g_{t_{i-1}} \times t_{i-1}) \\ -(l_{t_{i-1}} \times t_{i-1} \times \overline{x_{t_{i-1}}}) \; t_i \in \{t_1 \ldots t_n\} \end{aligned} \tag{10}$$

The objective as shown in Eq. (1) is to produce an optimal scheduling scheme on selecting energy source, which finally leads to the least electricity cost over a certain long period of time. In the objective function Eq. (1), $x_{t_i} \in \{0, 1\}$ represents the type of the energy to be used in the current time granularity, where 1 denotes that the system selects energy from utility grid and 0 denotes that the system selects energy from solar panels. Meanwhile x_{t_i} is the key decision variable in this optimisation problem. Moreover, there are several critical constraints (from (2) to (9)) that need to be satisfied. The constraints from (2) to (5) require that the total energy consumption in each user-defined time granularity cannot exceed the available renewable energy when the solar energy is selected as the energy source. In our design, the available renewable energy in a certain time

Algorithm 1 The energy scheduling approach under the full support of energy storage devices

```
Input: $t_1$: the first time granularity;
       $t_n$: the last time granularity;
         : the length of a user-defined time granularity;
       $l_{t_i}$: energy demand over $t_i$;
       $p_{t_i}$: electricity price over $t_i$;
       $g_{t_i}$: the solar generation forecasting over $t_i$;
       $E_{battery}(t_i)$: the battery status at the beginning of time
    granularity $t_i$
Output: the value of each decision variable $x_{t_i}$
 1: for $t_i = t_1$ to $t_n$ do
 2:     $P[t_i]$   $p_{t_i}$
 3:     $D[t_i]$   $l_{t_i}$
 4:     $G[t_i]$   $g_{t_i}$
 5: end for
 6: Sort elements in the price array P[ ] in a descending order
 7: for each $P[t_k]$ 2 $P[\,]$ (with the maximum electricity price
    selected first) do
 8:     if $D[t_k]$     $G[t_k]$   $+ E_{battery}(t_k)$ then
 9:         Use renewable energy as energy source
10:         $E_{battery}(t_{k+1}) = G[t_k]$     $D[t_k]$   $+ E_{battery}(t_k)$
11:     else
12:         Use utility grid as energy source
13:         $E_{battery}(t_{k+1}) = G[t_k]$   $+ E_{battery}(t_k)$
14:     end if
15: end for
16: return the value of each decision variable $x_{t_i}$
```

granularity mostly comes from two energy sources. The first one is solar panels and the other one is battery storage. The energy left in the battery can be calculated by Eq. (10) and it must necessarily satisfy the constraints (6) to (9) at any time.

To generate cost-effective scheduling schemes, we developed two low-complexity algorithms for energy scheduling, each of which fits to a specific deployment environment. Moreover, we utilised the time-slot based scheduling strategy, which neatly matches up with the user-defined time granularity used in the intra-day forecast. By doing so, it is convenient for us to make the scheduling decision with the latest input.

Case 1: We focus on large-scale deployment of our designed edge computing platform, which is normally set up in residential buildings or even communities. In this scenario, sufficient energy storage devices are generally deployed locally, thus the battery is always capable of storing all surplus energy generated from the solar energy generator, and we do not need to concern about the constraints (6) to (9). To realise the energy scheduling in this case, we can simply employ a greedy-like strategy as shown in **Algorithm 1** to maximise the solar energy utilisation over a day, particularly for the time periods when the electricity price is high.

With respect to the routine of deriving energy schedules, which is depicted in **Algorithm 1**, the scheduler needs to first recursively store each type of collected input information into the corresponding array and then sort the electricity price of

Algorithm 2 The energy scheduling approach under the partial support of energy storage devices

```
Input: $t_1$: the first time granularity;
       $t_n$: the last time granularity;
         : the length of a user-defined time granularity;
       $l_{t_i}$: energy demand over $t_i$;
       $p_{t_i}$: electricity price over $t_i$;
       $g_{t_i}$: the solar generation forecasting over $t_i$;
       $E_{battery}(t_i)$: the battery status at the beginning of time
    granularity $t_i$
Output: the value of each decision variable $x_{t_i}$;
 1: for $_i = {}_1$ to $_n$ do
 2:     $_i$   the end time of the time granularity $t_i$
 3: end for
 4: $E(\ _0) = 0$
 5: $E(\ _n) = 0$
 6: $F(\ _0; E(\ _0)) = 0$
 7: for $_i = {}_0$ to $_n$ do
 8:     for each possible battery status $E(\ _i)$ at time point $_i$
        do
 9:         if $l_{t_{i+1}}$  $t_{i+1}$   $g_{t_{i+1}}$  $t_{i+1} + E(\ _i)$ then
10:             Choose renewable energy or utility grid ($x_{t_{i+1}}$
                1 or 0)
11:         else
12:             Only choose utility grid ($x_{t_{i+1}}$   1 )
13:         end if
14:         Calculate all possible battery statuses satisfying bat-
            tery capability constraints at time point $_{i+1}$
15:     end for
16:     for each possible battery status $E(\ _{i+1})$ at time point
        $_{i+1}$ do
17:         $F(\ _{i+1}; E(\ _{i+1}))$     $\min f F(\ _i; E(\ _i))$ +
            $W(E(\ _i); E(\ _{i+1})) g;$
18:         Put the value of $x_{t_{i+1}}$ into the corresponding decision
            variable array $X[\,]$
19:     end for
20: end for
21: return the optimal $X_{final}[\,]$ that leads to $E(\ _n)$
```

the time slots in a descending order over a day as presented in lines 1-6. After that, the scheduler will determine solar energy on different time granularities based on the sorted electricity price as depicted in Lines 7-15. By jointly considering the status of energy storage device, solar energy forecasting and the estimated load demands, the module filters out those time slots when the solar energy is insufficient to support the running of edge computing systems. Once the scheduling scheme is generated, this scheme will then be used for the following time slot.

Case 2: When our edge computing platform deployed in households, there is a high chance that not enough energy storage devices are installed in each family. The capacity of the battery is now becoming a critical factor that makes effects on energy scheduling over different time granularities.

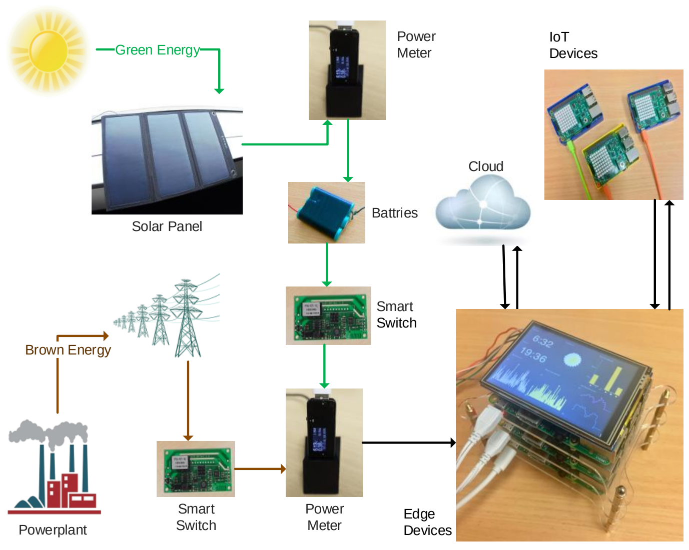

Fig. 2. Testbed Setup

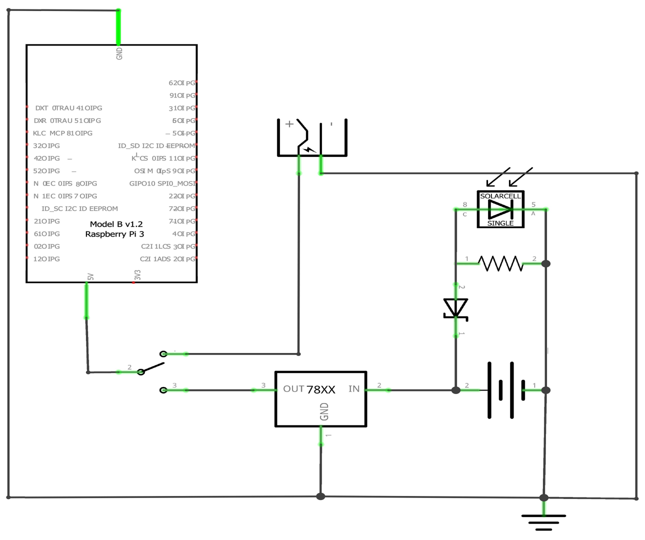

Fig. 3. The External Circuit Connection of a Raspberry Pi 3B

In this case, the constraints (6) to (9) are therefore expected to be taken into consideration. In order to generate an optimal scheduling scheme for this case scenario over a long-term time period, we developed a dynamic programming based algorithm as shown in **Algorithm 2**.

In **Algorithm 2**, $E(\ _i)$ denotes the battery status at time instant $\ _i$. As the dynamic programming (DP) strategy is employed in this algorithm, we use $E(\ _i)$ to match up with the system state in the DP problem. $F(\ _i; E(\ _i))$ is a cost-to-go function in this DP problem, which represents the minimum total energy costs charged by the utility grid from $\ _0$ to $\ _i$. Initially, the scheduler specifies the start time and the end time of each user-defined time granularity as depicted in lines 1-3. We also assumed that the energy stored in the battery is empty at the system initialisation and the energy charged into the battery during the schedule is expected to be used up in the end. Therefore, lines 4-6 set the values of battery storage at time points $\ _0$ and $\ _n$ to 0, and initialise total energy cost at the starting time point $\ _0$ to 0. After that, lines 8-14 contribute to calculating all possible battery states at time point $\ _{i+1}$ once $E(\ _i)$ is determined. For each possible battery state at $\ _{i+1}$, the corresponding minimum energy cost can be calculated based on the state-transition equation shown in line 17, where $W(E(\ _i); E(\ _{i+1}))g$ is the transition function capable of calculating transition cost from state $E(\ _i)$ to state $E(\ _{i+1})$. With this approach, the system can finish all iterative calculations at different stages as shown in lines 7-20. Based on the results, we can eventually obtain the optimal values of decision variables, which lead to the minimum electricity cost.

C. Task Scheduling

The data processing module employed a distributed task scheduling policy. Our approach is built on top of our previous work of DWB policy [30]. The original DWB policy is a time-slot based Lyapunov optimisation approach for making task offloading decisions among edge devices. However, the original design of the DWB policy only focused on minimisation of computing and communication cost in runtime, which is not well aligned to the design objective of our system. To tackle this issue, we practically generalise it to further take the solar energy into consideration. In the extended DWB policy, each edge computing device is firstly expected to retrieve the latest information from its connected neighbours at the beginning of each time slot, including the number of tasks, battery status and solar power generation. Once the information is retrieved, the data processing module will dynamically determine the incoming tasks of that time slot to be performed locally or on the neighbours. In order to meet the needs of the applications and to maximise the use of solar energy, the tasks are preferably allocated to the edge device that has the most available renewable energy and computation resource. Based on this principle, the utilisation of the resources of the edge devices can thus be maximised.

V. A Proof-of-concept Case Study

To demonstrate the feasibility of our proposed energy management framework, a real-world testbed made of low-cost and easy-to-get hardware is built. We first provide the details of our prototype implementation. After that, we present the results of experimental studies.

A. Testbed Setup

Figure 2 depicts our testbed setup that recreates the aforementioned sustainable edge computing systems in smart home environments. In this testbed, three Raspberry Pi 3B devices with a quad-core CPU and 1 GB RAM are used as edge servers that are responsible for receiving requests, processing data and making decisions for schedulers, and another three Raspberry Pi 3B devices equipped with Sense Hat are used as IoT devices to emulate smart devices at home.

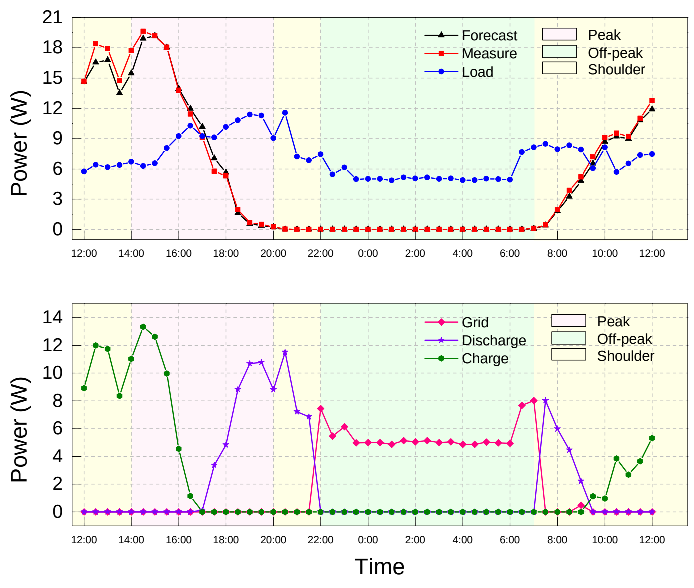

Fig. 4. Energy Generation and Consumption

We enabled the peer-to-peer wireless communication in our testbed with the help of WiFi, including the server to server connection and IoT devices to server connection. We used sockets in between every two edge servers for message passing so that each edge server is able to relay the key information to others. Meanwhile, each server is responsible for at least one IoT device, so the data collected from Sense Hat will then be forwarded to the corresponding edge server for further processing. For those tasks with complex computation or a large amount of data, they are supposed to be uploaded to the cloud. In our experiment, we connected the edge servers to IBM Watson IoT platform [35], which provides a set of component cloud services, such as visualised data analysis, and tools for remote control, to form a user-oriented application.

To power up the edge devices with renewable energy, we firstly connected the solar panel to three 20000mAh rechargeable batteries, each of which is used to power a single PI via GPIO (General-Purpose Input/Output). Besides, we expect that the surplus renewable energy can be stored in the batteries while brown energy can be drawn when a power deficit occurs. To do this, each PI can draw power from utility grid at its micro-USB port. Both GPIO pins and micro-USB port are wired to smart switches that are able to access WiFi and can be remotely controlled by APP, and the circuit connection of a single Raspberry Pi 3B is shown in Figure 3. If the system selects to use renewable energy, the switch wired with GPIO is connected and the switch connected to micro-USB is disconnected. The reverse operation occurs if the system selects to use brown energy. To measure the power usage of different energy sources, a power meter is also connected in between the smart switch and the PI as shown in Figure 2.

B. Evaluation of Energy scheduling

Figure 4 shows the comparison between the forecast data and the measured data of power generation within a day, and illustrates the performance of energy scheduler module under load fluctuation. As shown in upper part of the figure, the forecast data of solar generation nicely matches the measured ones across a day. This is due to the fact that the employed receding horizon control method allows us to calibrate forecasts with the latest information. Compared with traditional day-ahead forecasting, such interval-ahead forecasting literally shrinks the gaps between forecast and measurement, making our forecast more accurate. The lower part of Figure 4 depicts the operations on battery and smart switch for energy scheduling in response to the load variation and electricity tariffs.

As shown in Figure 4, from 12:00 to 17:00, the electricity generation from solar panel is sufficient to solely power the edge servers and the surplus solar energy is used to charge the battery. From 17:00, the solar energy cannot meet the load demands, and the batteries stops charging since then. To reduce the electricity bill, the energy scheduler aims to maximise the utilisation of renewable energy at peak hours, then shoulder hours. Based on this principle, from 17:00 onward, battery starts discharging to meet the needs of load demands. From 22:00 to 07:00, the offpeak hours of a day, the electricity cost is much cheaper. To ensure the reliability of edge device, the system uses utility grid to support the system running in these hours. Once moving into the shoulder hours staring from 07:00 of the next day, electricity price goes up and the batteries start discharging again to power the edge devices. With solar energy increases, the system will again to choose solar panel as energy source from 09:30. It is obvious that the utility grid is used during the off-peak hours most of the time, while the renewable energy is able to support the reliable running of edge computing systems during the rest of the time. Thus, we can reduce up to 86% electricity tariffs every day for edge computing systems.

C. Evaluation of Task Scheduling

To evaluate how the data processing module contributes to the energy management framework, we employed two different strategies to distribute incoming tasks among connected edge devices. The first one used no collaboration mechanism, and the incoming tasks were allocated to edge servers according to a pre-defined ratio, which is 10%, 20%, and 70%, respectively. As shown in Figure 5, from 12:00 to 20:00, all these three Raspberry Pis purely draw power from renewable energy sources, either the solar panel or the connected batteries. However, from 20:00 onwards, the third Raspberry Pi started to draw power from the utility grid, since tremendous tasks were released during the peak hours and the tasks allocated to this edge device need to be completed in time. This leads to the case that the connected battery was depleted rapidly. By contrast, the stored energy in the batteries for the other two Raspberry Pis is more than enough during that period. Furthermore, the less utilisation of the

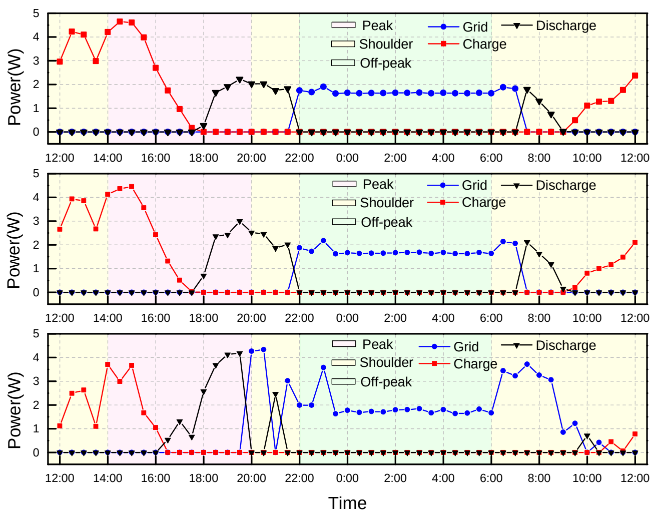

Fig. 5. Load Imbalance

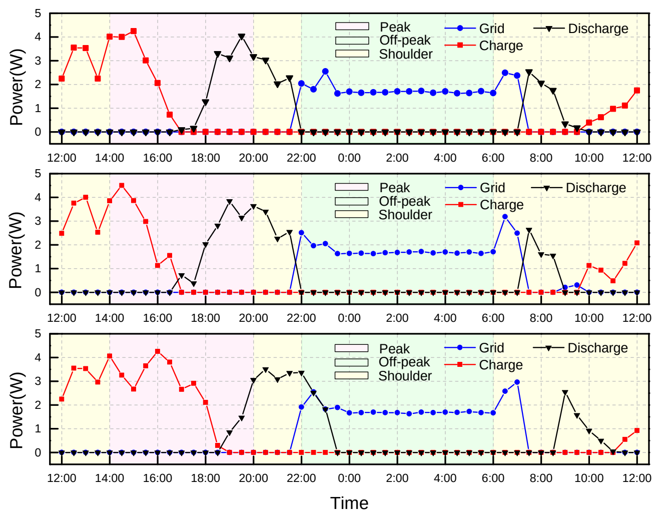

Fig. 6. Load Balance

stored energy makes the energy cost can be further reduced by balancing the peak demands among the three edge devices.

In the second approach, we used the extended DWB policy, which not only fully exploited the computing capabilities and storage resources of the edge devices, but also balanced the loads among these devices. As shown in Figure 6, three edge devices fully drew power from renewable energy source before 22:00, thus the system maximises the utilisation of surplus energy stored in the batteries during the peak hours and the time periods of drawing power from utility grid is significantly reduced. More importantly, no obvious load imbalance is found among the Raspberry Pis. This approach saves 23.5% electricity tariff compared to the load imbalance case.

VI. Conclusion and Future Work

In this work, we explore the possibilities of developing a sustainable Edge Computing platform that fulfills the needs of home energy management systems and smart home applications, while are missing from both open-source platforms and commercial platforms. As sustainable computing becomes a rapidly expanding research area spanning many areas in computer science, e.g. cloud computing, IoT, mobile computing and blockchain, we believe this topic will surely become of significant interest to the edge computing community. We also illustrate the design choices needed to realise a sustainable edge computing system for handling the needs of home energy management and smart home applications. This paper introduces the high-level design and the required functionalities of the system instead of delving into the intricate details of our proposed framework.

To make our design more robust, it is important to fulfill the following two tasks:

Green cloud computing has been studied for years and numerous mechanisms have been proposed for real-world practices. It is important to distil from the accumulated experience and the learned lessons that are valuable and relevant to edge computing.

The works on smart grids and energy Internet can greatly benefit from ICT support to balance supply and demand in near real-time with frequent price updates. Edge computing can provide the needed mechanisms to achieve this but more studies are necessary to come up with practical solutions.

In the future, we plan to address multiple challenges on receding horizon control, resource management on edge devices, demand response and load disaggregation. The technique of receding horizon control is employed in both energy scheduling and task scheduling of our framework. However, there are still existing a delimma on this control method in need of further research. It is well known that the accuracy of forecast and scheduling increases when the length of time granularities is decreased. However, the size of the problem increases at the same time, which leads to higher computational complexity and more energy consumption. Therefore, it is necessary to explore a solution to find a practical balance of the accuracy and computation overhead in the future study.

On the other hand, peak load shifting is a popular approach to minimise the energy cost at households, by selectively shifting the jobs of the appliances from peak hours to off-peak hours. However, to improve the effectiveness of such approach, edge computing paradigm can be used to determine which types of jobs can be shifted without jeopardising users' comfort during the peak hours in a real-time or quasi-real time manner. Besides, the multimodal human-machine interaction technique can also be integrated into the system so that user's preference and feedback can be identified immediately before performing peak load shifting. Furthermore, the location-aware users' demand prediction and analytics models need to be explored to further reduce the search space of the problem.

Acknowledgement

The work of Samee U. Khan was supported by (while serving at) the National Science Foundation. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787 – 2805, 2010.

[2] P. Garcia Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino, A. Iamnitchi, M. Barcellos, P. Felber, and E. Riviere, "Edge-centric computing: Vision and challenges," *SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 5, pp. 37 – 42, Sep. 2015.

[3] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[4] M. Satyanarayanan, "Edge computing: Vision and challenges," *USENIX Association, Santa Clara, USA*, 2017.

[5] H. Chang, A. Hari, S. Mukherjee, and T. Lakshman, "Bringing the cloud to the edge," in *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*. IEEE, 2014, pp. 346–351.

[6] H. Yin, X. Zhang, H. H. Liu, Y. Luo, C. Tian, S. Zhao, and F. Li, "Edge provisioning with flexible server placement," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 4, pp. 1031–1045, 2017.

[7] K. Bhardwaj, M.-W. Shih, P. Agarwal, A. Gavrilovska, T. Kim, and K. Schwan, "Fast, scalable and secure onloading of edge functions using airbox," in *Edge Computing (SEC), IEEE/ACM Symposium on*. IEEE, 2016, pp. 14–27.

[8] Y. Xiao and M. Krunz, "Qoe and power efficiency tradeoff for fog computing networks with fog node cooperation," in *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*. IEEE, 2017, pp. 1–9.

[9] Z. Liu, K.-K. R. Choo, and J. Grossschadl, "Securing edge devices in the post-quantum internet of things using lattice-based cryptography," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 158–162, 2018.

[10] E. Fernandes, J. Paupore, A. Rahmati, D. Simionato, M. Conti, and A. Prakash, "Flowfence: Practical data protection for emerging iot application frameworks." in *USENIX Security Symposium*, 2016, pp. 531–548.

[11] E. Zeng, S. Mare, and F. Roesner, "End user security & privacy concerns with smart homes," in *Symposium on Usable Privacy and Security (SOUPS)*, 2017.

[12] J. Cao, L. Xu, R. Abdallah, and W. Shi, "Edgeos_h: A home operating system for internet of everything," in *Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on*. IEEE, 2017, pp. 1756–1764.

[13] 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.

[14] S. Yi, Z. Hao, Q. Zhang, Q. Zhang, W. Shi, and Q. Li, "Lavea: Latency-aware video analytics on edge computing platform," in *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, ser. SEC '17, 2017, pp. 15:1–15:13.

[15] "Renewable energy: A world turned upside down, the economist," http://www.economist.com/news/briefing/21717365-wind-and-solar-energy-are-disrupting-century-old-model-providing-electricity-what-will, accessed: 2017-08-17.

[16] B. Zhou, W. Li, K. W. Chan, Y. Cao, Y. Kuang, X. Liu, and X. Wang, "Smart home energy management systems: Concept, configurations, and scheduling strategies," *Renewable and Sustainable Energy Reviews*, vol. 61, pp. 30–40, 2016.

[17] S. Flores, "Divide-and-conquer: How edge processing will open a new door for iot applications," *Newsletter*, vol. 2014, 2014.

[18] H. Gedawy, S. Tariq, A. Mtibaa, and K. Harras, "Cumulus: A distributed and flexible computing testbed for edge cloud computational offloading," in *Cloudification of the Internet of Things (CIoT)*. IEEE, 2016, pp. 1–6.

[19] C. Pahl, S. Helmer, L. Miori, J. Sanin, and B. Lee, "A container-based edge cloud paas architecture based on raspberry pi clusters," in *Future Internet of Things and Cloud Workshops (FiCloudW), IEEE International Conference on*. IEEE, 2016, pp. 117–124.

[20] P. Bellavista and A. Zanni, "Feasibility of fog computing deployment based on docker containerization over raspberrypi," in *Proceedings of the 18th international conference on distributed computing and networking*. ACM, 2017, p. 16.

[21] "Openhab-empowering the smart home," https://www.openhab.org/.

[22] "Makad energy - dreamwatts energy management system," http://www.makadenergy.com, accessed: 2018-03-25.

[23] "Smart home - smartthings — samsung us," https://www.samsung.com/us/smart-home/smartthings/, accessed: 2018-03-25.

[24] "LG home appliances: Kitchen, laundry and vacuum appliances — LG australia," view-source:http://www.lg.com/au/home-appliances, accessed: 2018-03-25.

[25] M. A. Al Faruque and K. Vatanparvar, "Energy management-as-a-service over fog computing platform," *IEEE internet of things journal*, vol. 3, no. 2, pp. 161–169, 2016.

[26] W. Li, T. Yang, F. C. Delicato, P. F. Pires, Z. Tari, S. U. Khan, and A. Y. Zomaya, "On enabling sustainable edge computing with renewable energy resources," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 94–101, May 2018.

[27] J. Zhang, B.-M. Hodge, and A. Florita, "Investigating the correlation between wind and solar power forecast errors in the western interconnection," in *ASME 2013 7th International Conference on Energy Sustainability collocated with the ASME 2013 Heat Transfer Summer Conference and the ASME 2013 11th International Conference on Fuel Cell Science, Engineering and Technology*. American Society of Mechanical Engineers, 2013, pp. V001T16A003–V001T16A003.

[28] J. Mattingley, Y. Wang, and S. Boyd, "Receding horizon control," *IEEE Control Systems*, vol. 31, no. 3, pp. 52–65, 2011.

[29] E. Cuce, P. M. Cuce, and T. Bali, "An experimental analysis of illumination intensity and temperature dependency of photovoltaic cell parameters," *Applied Energy*, vol. 111, pp. 374–382, 2013.

[30] W. Bao, W. Li, F. C. Delicato, P. F. Pires, D. Yuan, B. B. Zhou, and A. Y. Zomaya, "Cost-effective processing in fog-integrated internet of things ecosystems," in *Proceedings of the 20th ACM International Conference on Modelling, Analysis and Simulation of Wireless and Mobile Systems*. ACM, 2017, pp. 99–108.

[31] R. Perez, E. Lorenz, S. Pelland, M. Beauharnois, G. Van Knowe, K. Hemker Jr, D. Heinemann, J. Remund, S. C. Müller, W. Traunmüller *et al.*, "Comparison of numerical weather prediction solar irradiance forecasts in the us, canada and europe," *Solar Energy*, vol. 94, pp. 305–326, 2013.

[32] E. Lorenz, J. Hurka, D. Heinemann, and H. G. Beyer, "Irradiance forecasting for the power prediction of grid-connected photovoltaic systems," *IEEE Journal of selected topics in applied earth observations and remote sensing*, vol. 2, no. 1, pp. 2–10, 2009.

[33] M. Sengupta, A. Habte, C. Gueymard, S. Wilbert, and D. Renne, "Best practices handbook for the collection and use of solar resource data for solar energy applications," National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2017.

[34] J. D. Hamilton, "State-space models," *Handbook of econometrics*, vol. 4, pp. 3039–3080, 1994.

[35] "IBM Watson IoT platform," https://www.ibm.com/internet-of-things/spotlight/watson-iot-platform, accessed: 2018-03-25.