Renewable energy-aware heuristic algorithms for edge server selection for stream data processing

Tomasz Szydlo[1] and Chris Gniady[2]

[1] AGH University of Science and Technology,
Institute of Computer Science, Krakow, Poland
[2] Department of Computer Science, University of Arizona,
Tucson, Arizona 85719

Abstract. Internet of Things and Edge computing are evolving, bringing data processing closer to the source and a result closer to the network's edge. This distributed processing can increase energy consumption and carbon footprint. One solution to overcome the environment's impact is using renewable energy sources such as photovoltaic panels to power both cloud and edge servers. Since solar energy is not available at night or can vary with cloudiness, the centers still rely on conventional energy sources. Any solar energy that exceeds the demand power of the computing infrastructure is put back to the grid. Fluctuations in energy output due to moving clouds can have a negative impact on conventional energy suppliers as they have to maintain a constant energy supply. This paper presents heuristic algorithms for selecting edge servers for data stream processing to manage renewable energy utilization and smooth out energy fluctuations.

Keywords: IoT, edge computing, renewable energy

1 Introduction

Internet of Things consists of smart homes, smart cities, and the fourth industrial revolution based on cyber-physical systems is becoming ubiquitous. The IoT devices surround us and generate a vast amount of data that requires analysis to reap the benefits of smart infrastructures. Processing data directly on the end device provides a short response time but may have limited scope due to a global view. This scope vs. response challenge is faced by Industry 4.0 that requires response times in the millisecond range. Pushing processing into the network increases response time but broadens the decision-making process's scope by enabling aggregation of data from multiple sensors in the factory. Finally, centralized cloud processing enables the development of truly global systems at the cost of larger data transfer delays from the cloud. The challenge is exacerbated by large bandwidth requirements from hundreds or even thousands of concurrent data streams to the cloud. The approach to solve the challenge relies on intelligent processing distribution between the edge and the cloud to minimize delays and maximize the global decision-making process.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77970-2_49

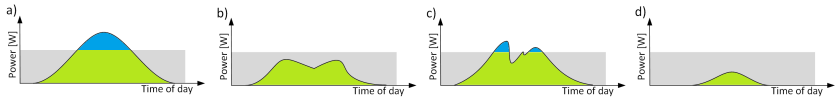

Fig. 1: PV energy generation: a) summer, b)summer during cloudy weather, c) spring with dynamic cloud movements, d) winter.

Such distributed computing can have a negative impact on energy consumption and carbon footprint since the computing hardware has to be distributed between multiple sites. Energy-efficient software and hardware help curb carbon footprint, and once combined with renewable energy sources, the carbon footprint can be eliminated. While solar energy is most promising, it is only available during the day and can have large variability due to cloud movements. Subsequently, utilization of solar energy in computing is challenging as it required adaptation from computing infrastructure to the changing energy supply. Real-time stream processing in Industry 4.0 or other smart environments make it even more challenging as unexpected delays can result in severe consequences. In this paper, we focus on data stream processing, also called dataflow processing, and how we can maximize solar energy utilization based on its availability.

2 Related work

Computing infrastructure is expected to generate up to 14% of global gas emissions by 2040, with datacenters and networking infrastructure accounting for 33% by 2025 and becoming more dominant in the future. The use of renewable energy can contribute to the sustainable development of computing infrastructure. However, renewable energy sources are characterized by high dynamicity of change and uncertainty in their supply, making their use in computing infrastructure a challenge. The challenges are tackled by datacenter design to accommodate renewable energy [3, 5] and to manage workflow scheduling based on energy availability [6]. Alternatively, the workload can be migrated between datacenters based on renewable energy availability [8].

In this paper, we focus on stream data processing systems based on a programming paradigm that emphasizes data flows [1, 4] generated by IoT devices. This class of systems' characteristic property is that they define applications as directed graphs where vertices correspond to processes while edges represent data streams. Flow-based programming (FBP) [7] and proposes architectures consistent with this paradigm – such as the staged event-driven architecture (SEDA) [10], or reactive programming [2] try to tackle challenges associated with distributed stream processing to provide scalable and efficient systems. Subsequently, asynchronous communication between processes coupled with the SEDA architecture ensures scalability and reduces response time. Flow-based processing makes energy optimizations challenging as it is not delay-tolerant and can't be postponed until renewable energy becomes available.

Solar energy is one of the most cost-efficient and readily deployable renewable energy sources. Fig. 1 shows solar energy production and consumption as

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77970-2_49

Table 1: Notations used in the system model.

Symbol	Meaning
G	Edge processing network
k	Number of edge processing locations
n	Number of stream data sources
M	Set of edge datacenters
S	Set of stream data sources
e_j^m	Power necessary for processing on m_j
e_j^{re}	Renewable energy available at m_j
e_i^s	Power necessary to process stream data from s_i
$x_{i,j}$	binary variable indicating assignment of s_i to m_j
$d_{i,j}$	network latency between s_i and m_j

related to seasons and cloudiness. The grey color on a chart represents the grid's consumed energy; green shows the solar energy produced and consumed by the system; while blue is the energy sent to the grid and as it is overproduced. As we observe, winter and cloudiness result in low energy production that cannot satisfy the demand at any time. Overproduction of energy by the solar system can be stored in the grid and drawn from the grid as needed. While such a solution seems practical as overproduction can be delivered somewhere else, solar energy production variability makes it challenging for utility companies to provide high quality of energy, requiring them to either overproduce or keep expensive natural gas generators on standby in case the solar output drops. Furthermore, grid usage cost is usually structured into the customer's energy cost and solar energy production, and resulting grid usage needs to be accounted for, leading to complex purchase/sale agreements. Subsequently, maximization of solar energy usage is critical to mitigating costs associated with energy trade.

3 Problem and algorithms description

In the context of edge processing, the problem of selecting the appropriate edge location of stream data processing can be considered as an undirected graph $G = (V, E)$ consisting of many stream data sources, and locations of edge datacenters thus $V = M \cup S$, where M is a set of edge datacenters and S is a set of stream data sources. E represents the links enabling network communication between M and S. Computing centers use backbone computer networks with high capacity, usually based on fiber optic technology. Assume that there are k edge server locations, and each of them has a similar performance for processing data streams. We use the notation e_j^m to denote the power used by the datacenter and e_j^{re} to denote the amount of energy from photovoltaic panels. To process data stream $s_i \in S$ in the edge datacenter it is necessary to consume e_i^s of power. Every edge datacenter can process streams from several sources as long as it does not exceed the computing capacity. Each data stream can be processed only by one edge datacenter at a particular time. Nevertheless, the processing location's choice impacts communication delays $d_{i,j}$ due to the geographical distance and the computer network's extent.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77970-2_49

Local consumption of renewable energy generated by solar panels used to power edge datacenter m_j is calculated as:

$$localconsumption = \begin{cases} 1 & e_j^{re} < e_j^m \\ \frac{e_j^m}{e_j^{re}} & e_j^{re} \geqslant e_j^m \end{cases} \quad (1)$$

When local consumption is less than 100%, the surplus of energy is sent to the grid. To increase the local energy consumption, it is necessary to process more data to increase the e_j^m. It can be achieved by moving stream processing from other edge datacenters. In general, choosing an edge datacenter is an NP optimization problem[9] solvable only for small cases.

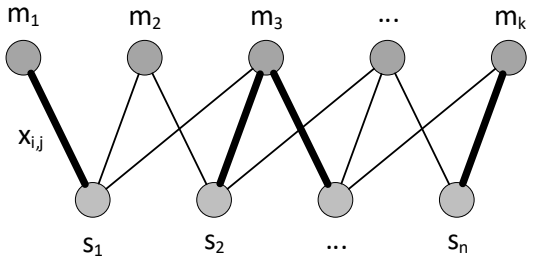

Fig. 2: Adaptability mechanisms for stream processing.

The problem of edge selection can be represented graphically, as in Fig.2. We introduce the binary decision variable $x_{i,j} \in \{0, 1\}$ to indicate whether the data stream from source s_i is processed by m_j edge datacenter. The decision to choose an edge datacenter can impact the quality of data processing and the processing delays. In the paper, we propose three heuristic algorithm, shown below, for edge server selection: 1) BEAS selects the processing center that has the lowest latency; 2) GEAS aims to distribute the load on edge servers evenly, potentially suffering from increased latency; 3) k-REAS takes into account information about the availability of renewable energy and selects the edge server with the highest surplus of renewable energy (to control latency k specifies that no more than the k-th server can be selected in terms of distance).

Algorithm 1 Best Effort Assignment Strategy (BEAS).

```
1: function BEAS(G)
2:     x ← ∅
3:     for each s_i ∈ S do
4:         sort m_j ∈ M in order of increasing distance d_{i,j}
5:         x[i, j] ← 1 for m_j with the smallest distance d_{i,j}
6:     return x
```

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77970-2_49

Algorithm 2 Greedy equal assignment strategy (GEAS).

```
1: function GEAS(G)
2:     x ← ∅
3:     max ← ⌈n/k⌉
4:     counter ← ∅
5:     for each s_i ∈ S do
6:         sort m_j ∈ M in order of increasing distance d_{i,j}
7:         for each m_j ∈ M with increasing distance d_{i,j} do
8:             if counter[i, j] < max then
9:                 x[i, j] ← 1
10:                counter[i, j] = counter[i, j] + 1
       return x
```

Algorithm 3 k-Renewable Energy-aware Assignment Strategy (k-REAS).

```
1: function K-REAS(G, k)
2:     x ← ∅
3:     energy ← ∅
4:     for each s_i ∈ S do
5:         sort m_j ∈ M in order of increasing distance d_{i,j}
6:         temp ← ∅
7:         for k stations m_j ∈ M with lowest distance d_{i,j} do
8:             temp[j] ← e_j^re − (e_j^m + energy[j] + e_i^s)
9:         select m_j with highest value of temp[j]
10:        energy[j] = energy[j] + e_i^s
11:        x[i, j] ← 1
       return x
```

4 Evaluation

To evaluate the solution, we set up a synthetic application representing a typical IoT system's operation that processes the sensors' data streams and requires real-time interactions between sensors and actuators. We consider a system with data stream sources in 50 US states. The location of edge data processing centers was adopted from Amazon Web Services - CloudFront services computing centers. We ssume that edge servers do not process other tasks than those currently analyzed.We also assume that the photovoltaic installation provides the annual energy budget for processing 3 data streams at each edge datacenter, with the energy necessary to process a single data stream is equal to 100W. This leads to variable sizes of solar panels at each location. We used PVGIS, an European Commissions' Project, that gives information about solar radiation and photovoltaic system performance based on satellite image analysis, with hourly resolution. Network delays are calculated based on the distance between stream data sources and the datacenters, assuming that the connection is based on the fiber technology.

We perfromed simulation of the IoT system based on the presented assumptions. For the *k-REAS* algorithm, we analyzed the operation for $k = \{1..5\}$.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77970-2_49

Algorithm	Latency				Localconsumption			
	min	max	avg	stdev	min	max	avg	stdev
GEAS	0.48	66.29	10.03	11.50	0.0	41.0	29.57	13.06
BEAS	0.47	66.29	7.40	10.80	0.0	73.0	28.0	20.91
1-REAS	0.47	66.29	7.40	10.69	0.0	73.0	28.0	20.91
2-REAS	0.47	66.56	8.67	10.70	6.0	57.0	31.28	15.90
3-REAS	0.47	66.61	10.35	11.06	5.0	52.0	32.76	14.58
4-REAS	0.47	70.18	11.45	11.23	12.0	48.0	33.47	12.85
5-REAS	0.47	72.41	12.53	11.50	13.0	46.0	34.14	10.61

Table 2: Detailed results of the experiments

Detailed results are presented in Tab.2. In the case of the *BEAS* algorithm, the lowest communication delay of 7.40 ms was obtained. Local consumption of solar energy was in the range of $[0\%; 73\%]$. This means that some processing centers were underutilized, delivering energy to the grid, and some were overloaded. We note that the decision to choose the edge datacenter was made once - at the beginning of the simulation. The *GEAS* algorithm distributed the load evenly among the edge servers, resulting in a maximum local consumption of 41%. As a result, this led to an increase in the average latency of 10.03ms. As before, the selection of the edge datacenters was performed once at the beginning of the simulation.

The third algorithm, *k-REAS*, is based on the current and time-varying energy balance available in each edge data center. It aimed to increase local consumption of solar energy at the expense of increased processing latency. The algorithm's aggressiveness depends on the parameter k and increases the local consumption of solar energy. As a result, for $k = 5$, the lowest achieved level of local consumption was 13%. Contrary to the previous algorithms, *k-REAS* dynamically changes the edge server's selection based on information about the instantaneous load on the servers and the amount of available energy. Fig. 3 visualizes energy informations from Tab. 2. We were able to increase the average local consumption by 6.14% (BEAS/5-REAS), but resulting delays increased by 5.13ms. Notably, the choice between processing delays and the local energy consumption is a trade-off and should be selected according to the system application. At the same time, the increase in the delay by a few milliseconds, as was the case with the *k-REAS* algorithms, is in many cases insignificant but should be consciously chosen.

5 Summary and future work

In this paper, we have presented the concept of heuristic management algorithms. The results show that the dynamic choice of where to process the streaming data impacts the use of energy obtained from renewable sources. The use of this class of algorithms as presented in the paper is a step towards designing and constructing sustainable computer systems to minimize the impact on the natural environment. We have focused on the problem of selecting the place for process-

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77970-2_49

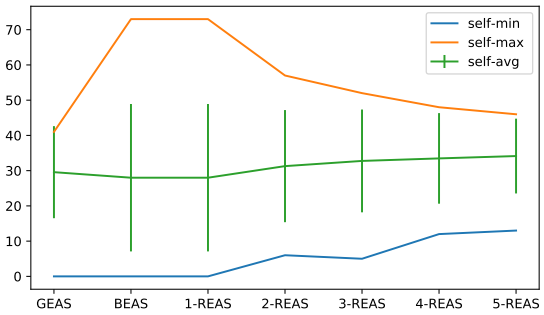

Fig. 3: Average local consumption of renewable-energy at the edge datacenters.

ing data streams. An interesting direction for further work would be the analysis of transferring processing between devices, edge computing, and cloud computing. This would provide more heterogeneous systems with different performance and energy trade-offs, potentially improving the overall system efficiency.

Acknowledgment The research presented in this paper was supported by the National Centre for Research and Development (NCBiR) under Grant No. Gospostrateg1/385085/21/NCBR/2019.

References

1. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., Ayyash, M.: Internet of things: A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys & Tutorials **17**(4), 2347–2376 (2015)
2. Bainomugisha, E., Carreton, A.L., Cutsem, T.v., Mostinckx, S., Meuter, W.d.: A survey on reactive programming. ACM Comput. Surv. **45**(4), 52:1–52:34 (2013)
3. Chonglin Gu, Zhenlong Li, Chunyan Liu, Huang, H.: Planning for green cloud data centers using sustainable energy. In: 2016 IEEE Symposium on Computers and Communication (ISCC). pp. 804–809 (2016)
4. Giang, N.K., Blackstock, M., Lea, R., Leung, V.C.M.: Developing iot applications in the fog: A distributed dataflow approach. In: 2015 5th International Conference on the Internet of Things (IOT). pp. 155–162 (Oct 2015)
5. Gill, S.S., Buyya, R.: A taxonomy and future directions for sustainable cloud computing: 360 degree view. ACM Comput. Surv. **51**(5) (Dec 2018)
6. Grange, L., Costa, G.D., Stolf, P.: Green IT scheduling for data center powered with renewable energy. Future Gener. Comput. Syst. **86**, 99–120 (2018)
7. Morrison, J.P.: Flow-Based Programming, 2Nd Edition: A New Approach to Application Development. CreateSpace, Paramount, CA (2010)
8. Shuja, J., Gani, A., Shamshirband, S., Ahmad, R.W., Bilal, K.: Sustainable cloud data centers: A survey of enabling techniques and technologies. Renewable and Sustainable Energy Reviews **62**, 195 – 214 (2016)
9. Wang, S., Zhao, Y., Xu, J., Yuan, J., Hsu, C.: Edge server placement in mobile edge computing. J. Parallel Distributed Comput. **127**, 160–168 (2019)
10. Welsh, M., Culler, D., Brewer, E.: Seda: An architecture for well-conditioned, scalable internet services. In: Proceedings of the Eighteenth ACM Symposium on Operating Systems Principles. pp. 230–243. SOSP '01, ACM, New York, NY (2001)

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77970-2_49