Serverless Edge Computing for Green Oil and Gas Industry

Razin Farhan Hussain[1], Mohsen Amini Salehi[1], and Omid Semiari[2]
[1]razinfarhan.hussain1, amini@louisiana.edu
[1]High Performance Cloud Computing (HPCC) Laboratory, School of Computing and Informatics,
University of Louisiana at Lafayette, Lafayette, LA, USA
[2]osemiari@georgiasouthern.edu
[2]Department of Electrical and Computer Engineering, Georgia Southern University, Statesboro, GA, USA

Abstract—Escalating demand of petroleum led the Oil and Gas (O&G) industry to extend oil extraction operation in the remote reservoirs. Oil extraction is a fault intolerant process where the maximum penalty is disaster impacting the environment seriously. Therefore, efficient and nature-friendly green oil extraction is a challenging operation, especially with location constrained in accessing the sites. To overcome these challenges and protect the environment from pollution, smart oil fields with numerous sensors (*e.g.,* for pipeline pressure, gas leakage, air pollution) are established to achieve clean O&G extraction. Conventionally, cloud datacenters are utilized to process the generated data. High-latency satellite communication are used for data transfer, which is not suitable for time-sensitive operations/tasks. To process such latency-sensitive tasks, edge computing can be a suitable candidate, however, their computational power goes downhill at disaster time due to surge demand of many coordinated activities. Therefore, we propose green smart oil fields that operate based on edge computing. To overcome shortage of resources and rapid deployment of the edge computing systems, we propose to use lightweight serverless computing on a federation of edge computing resources from nearby oil rigs. Our solution coordinates urgent coordinated operations/tasks to prevent disasters in oil fields and enable the idea of green smart oil fields. Evaluation results demonstrate the efficacy of our proposed solution in compare to conventional solutions for smart oil fields.

I. INTRODUCTION

Petroleum has been unquestionably one of the most important drivers of the world economy in recent decades. Due to high demand of petroleum, Oil & Gas (O&G) industry is expanding the extraction operations in remote and adverse locations (*e.g.,* Golf of Mexico, Persian Gulf, West Africa) [1] where giant reservoir exist, hence, several oil extraction sites are constructed within a short distance. The complex oil extraction process requires high reliability and extra safety measures to protect the surrounding environment. Moreover, governments (*e.g.,* U.S. environmental protection agency (EPA)) also enforcing regulations on O&G industries to reduce the adverse impact of oil extraction on the environment.

To protect the environment from disasters [2] that can occur due to flaws in oil extraction process, oil fields are geared up with many cyber devices (*e.g.,* sensors and actuators) and the concept of *smart oil field* has emerged with green oil extraction motto. Smart oil fields utilize various sensors (temperature,

Hydrogen Sulphide (H2S) gas emission, pipeline pressure, air pollution) which gather a large volume of data (up to two Terabytes per day [3]). To enable ultra-reliable and flawless green oil extraction these sensor generated datasets need to be analyzed and used in a real-time manner. The need for smart oil fields has been emphasized by both industry [3]–[5] and academia [6]–[8] to improve the efficiency of oil field and save the environment from pollution. However, existing smart oil field solutions cannot meet the requirements of *remote* oil fields for two specific reasons: (1) Lack of reliable and fast communications infrastructure to access to onshore management teams; (2) High cost of operations by human resources to perform real-time inspection and monitoring.

Current remote smart oil field solutions utilize satellite communication to cloud datacenters which is known to be unstable and imposes a significant latency. Hence, the goal of this study is to enable the idea of smart oil fields in remote sites for nature-friendly oil extraction. This research mainly focus on exploiting edge computing system in serverless manner for remote oil fields with unstable and weak connectivity to datacenter to enable the idea of Green O&G Industry. To handle latency-sensitive tasks, specifically, during a disaster when there is a surge for real-time computation, the edge computing system plays a vital role due to its locality to end user. However, the main challenges of utilizing edge computing are the resource constrained nature of edge nodes and difficulty of configuration and maintenance in remote areas. To overcome these problems and enable a robust system against the surge in demand, we harness the edge devices located in nearby oil rigs and propose a technique to federate them in an on-demand manner. For ease of federation and configuration, we propose to use serverless computing paradigm on the edge computing systems.

Federation of serverless edge computing systems can alleviate the shortage of resources, however, it introduces new challenges of processing tasks in the federated environment. As such, the research problem is *how to allocate surge requests to a serverless federated edge computing system, considering uncertainties exist during disaster times in these environments?*

To address this problem, we introduce a *service balancer*

for each edge nodes that provides Quality of Service (QoS) by considering the federation of edge nodes. Accordingly, the service balancer decides whether to allocate the arriving service request (aka task) locally (*i.e.,* on the receiving edge) or on a neighboring edge. Then, we propose a probabilistic model and develop a resource allocation heuristic for the service balancer to utilize the edge federation. As the serverless edge computing system has a central role in the smart oil field, its ability to cope with surge loads results in having a green energy production and O&G industry.

In summary, The contributions of this paper are as follows:

- Proposing a federation of serverless edge computing system to enable green oil extraction utilizing a robust resource allocation scheme with minimum connectivity to onshore cloud datacenter.
- Developing a model to capture the uncertainties exist in the federated edge environment.
- Analyzing the performance of the federated edge computing system under various oversubscribed conditions.

The rest of the paper is organized in the following manner. Section 2 presents the system model. Section 3 and 4 represent system architecture and task distribution in edge federation respectively. Section 5 demonstrates the performance evaluation and experiments. Section 6 presents related work. Finally, Section 7 concludes the paper with some future directions for exploration.

II. SYSTEM MODEL

In our system model, we consider utilizing edge machines which include storage, computational power, and communication capacity. The edge machines can be placed on the platform of oil rig above the water surface or can be mount on a floating boat near the site. Due to hardware limitation, edge machines are more appropriate for the real-time urgent task processing which typically has shorter deadline and delay-sensitive in nature. We consider utilizing serverless edge platform to facilitate management of resource allocation and optimal placement of smart oil field micro services (*i.e.,* database service, image processing). The sensors (*e.g.,* temperature, flow rate, tank level, gas leakage sensors) in a smart oil field generate a large amount of diverse data which is utilized by different applications. Accordingly, we classified these applications as task type (service type). These task types can vary from processing surveilled images (taken by unmanned aerial vehicles (UAVs) or embedded cameras) for detecting oil spill anomalies [9]; Analyzing large volume of data, streamed by sensors, to predict the oil spill spread direction and quantity [10]. Because the task types have various computational demands, they need processing machines with different characteristics (*i.e.,* heterogeneous machines). This form of HC systems is known as inconsistently heterogeneous systems [11]. We assume different machine provides various micro services in serverless manner. Upon arrival of a task of type i to a compute node j, it is assigned an individual deadline based on its arrival time and the end-to-end delay it can tolerate. The deadline can be defined as: $\delta_{ij} = arr_i +$

$\beta \times avg_i + \alpha \times d_{comm} + \epsilon$, where arr_i is arrival time of the task, avg_i is average of completion time in all edge nodes, d_{comm} is the communication latency, ϵ is system slack, β is computing constant, and α is the communication constant. During an incident, different task types have bust arrivals to the edge system and make the edge system oversubscribed. Hence, the system receives the number of tasks beyond its capacity. Consequently, some tasks are considered to miss their deadlines according to the level of oversubscription. For performance improvement of services in edge nodes, we assume utilizing serverless technology.
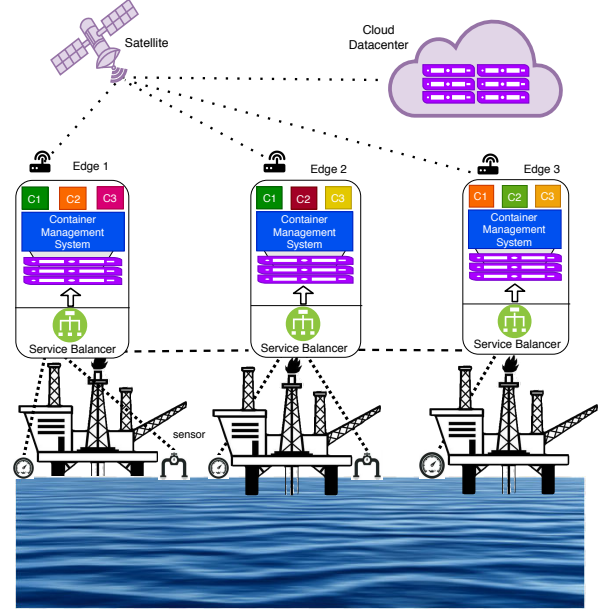

Fig. 1. System Architecture of serverless edge computing in smart oil field where services are provided through containerized technology to sensor generated data.

III. SYSTEM ARCHITECTURE

As stated in the system model, the proposed system architecture considers an edge device with a service balancer module in every oil extraction site as demonstrated in 1. The system architecture includes two-tier of computing nodes where edge nodes are located in local or in the first tier and cloud data centers are in the second tier. Physical sensors (*i.e.,* flow rate sensor, pressure sensor, tank level sensor, gas sensor) of smart oil fields takes physical quantity and convert it to the electrical signal. The physical sensors include microcontroller for getting the readings which are defined as sensor units. The sensor unit has a communication interface (*e.g.,* Ethernet, Bluetooth, Wi-Fi, Zigbee) to communicate with the edge device. The sensor units send tasks to the first tier (edge device) of the architecture where incoming tasks are sorted in terms of latency (*i.e.,* latency intolerant, latency tolerant) by the service balancer. This architecture support the serverless edge platform stated in system model combining the benefits of edge with the computational and storage capabilities of cloud.

IV. TASK ALLOCATION IN EDGE FEDERATION

Considering task allocation in edge federation, every edge node has a service balancer module which works in an immediate mode to allocate the incoming task to the appropriate computing node. We observe that the task completion time data from the historical record follows normal distribution according to the central limit theorem and can be applied to perform statistical modeling. Therefore, we consider every service balancer has access to this historical data which is stored in a matrix data structure and defined as Estimated Task Completion (ETC) time matrix. Every cell of this matrix represents normal distribution ($X \sim \mathcal{N}(\mu, \sigma^2)$) of a particular task type. Considering the task processing in neighbor edge device of the federation, communication overhead has a significant impact in service time. Therefore, the transfer time of a task from a one service balancer to other edge nodes is captured in Estimated Task Transfer (ETT) time matrix. ETT matrix basically captures the communication uncertainty using normal distribution ($Y \sim \mathcal{N}(\mu, \sigma^2)$). Both of the matrices are updated periodically to reflect the current situation which is utilized by the service balancer to estimate the probability of success.

1) Probabilistic Model: Upon arrival of a task t of type i, the service balancer calculates the probability ($P_i(t)$) of meeting the task's deadline δ_i in it's receiving edge node as well as it's neighbor edges. If receiving edge node is j, then the probability of success in j can be defined as:

$$P_i^j(E_c^j(t_i) < \delta_i) = P(Z < z), where \ z = (\delta_i - \mu_i^j)/\sigma_i^j \quad (1)$$

where $E_c^j(t_i)$ is the estimated task completion time of task type i in edge node j. μ_i^j and σ_i^j are respectively the average and standard deviation of the considering distribution. In Equation 1, z score is used to standardize the normal distribution. For calculating the probability of neighbor edge nodes, the ETC matrix's normal distribution of receiving task type is convolved with its ETT matrix distribution which incorporates the communication overhead with computing overhead. After calculating the probability with resulting distribution for all the edge nodes with respect to task type, the task is allocated to the edge node that offers the highest probability.

A. Heuristic based on Probabilistic Model

1) Highest Probability of Success (HPS): The heuristic allocates the arriving task based on its probability of success across the edge nodes. The service balancer utilizing HPS heuristic estimates the probabilities which represent success of the arriving task to meet its deadline across the edge nodes. The HPS heuristic chooses the maximum probability edge node to allocate receiving task.

V. EVALUATION

A. Experimental Setup & Workload

For performance evaluation of our proposed model, Edge-CloudSim [12] is considered which is a discrete event simulator specific to edge computing scenarios. Datacenters of EdgeCloudSim are considered as edge devices with limited computational capacity (1500-2500 MIPS) including 8 homogeneous cores. On the other hand, different data centers have different computational power (MIPS) which represents the heterogeneity across the edge nodes. A large cloud datacenter with massive computational power (40000 MIPS) is considered for non-urgent delay tolerant tasks. The WLAN bandwidth is set to 200Mbps and propagation delay is considered as 0.57 seconds [13] which occurs from satellite communication.

EdgeCloudSim's default workload includes four different task types, among which two of them are urgent (*i.e.,* latency intolerant) and the other two are non-urgent (*i.e.,* latency tolerant). The execution time of a task is represented in Million Instructions per seconds(MIPs) which is sampled out as a normal distribution from an average value for a particular task type.

B. Baseline Heuristics

In this paper, we consider using two baseline heuristics. They are Minimum Expected Completion Time (MECT) and Success with Computational Certainty (SCC). MECT heuristic considers minimizing expected completion time for receiving task in edge nodes of federation whereas SCC tends to maximize the difference between the deadline and average completion of arriving tasks for allocation decision.

C. Results and Analysis

As deadline miss rate is the fundamental rubric for maintaining QoS, we evaluate our system based on this standard. To investigate the performance of our scheme with increased oversubscription level we increase the number of applications submitted to the system. The number of submitted applications increased from 50 to 250 which generate approximately 500 to 10000 tasks for the system. The result reflects that with
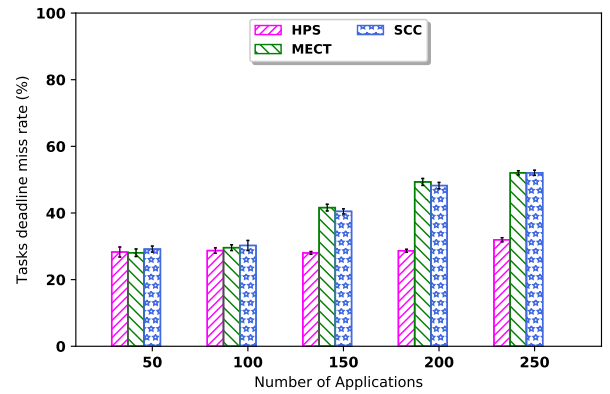

Fig. 2. Increasing oversubscription level with increased number of applications.

the increased number of applications, the deadline miss rate gets increased for all of the heuristics. When the system starts getting oversubscribed for 150 applications the difference in performance of the heuristics is significant. Specifically, when the system is fully oversubscribed with 250 applications HPS performs approximately 21% better than $MECT$ and SCC.

This is because our proposed heuristic considers both communication and computation overhead whereas other heuristics consider only one of them.

VI. RELATED WORK

The researcher has previously exploited the concept of edge computing for delay-tolerant networks. Lorenzo *et al.,* in [14] proposed an edge computing system with resource allocation design in the wireless network using mobile devices to mitigate the problem of network congestion. In [15] Chang *et al.,* proposed an optimized resource migration scheme from mobile IoT devices to heterogeneous Cloud-Fog-Edge computing environment which considers hardware limitation of edge devices. The serverless computing concept in edge level is explored by Nastic and Dustdar in [16]. Prior research works have been undertaken on resource allocation of edge computing systems with unreliable network connectivity [17]. However, these research works neither consider the heterogeneity of the edge resources nor have the self-organization and autonomy abilities [18]–[20]. The specific problem of resource provisioning in serverless edge federation with low-connectivity to the back-end datacenters has not been explored in the context of remote smart oil fields. Efforts towards smart oil fields have been predominantly focused on analyzing the big data extracted from oil wells [21], applying machine learning methods to reduce exploration and drilling costs [22], or warning systems for early prediction of disasters [23]. These solutions are all reliant on onshore datacenters, which is not viable for remote and offshore oil fields [24].

VII. CONCLUSION

In this paper, we propose federation of serverless edge computing systems to enable green oil extraction. The system utilizes an efficient resource allocation scheme to cope with the uncertainties that exist in remote offshore smart oil fields. We leveraged the edge federation to utilize underutilized neighbor edge devices and historical data to predict the completion time of an incoming service request (task) with a probabilistic model. The model is aware of resource constraint nature of edge devices as well as the uncertainties due to stochastic nature in communication. Both of the uncertainties were incorporated while predicting the probability of success within a strict deadline period of task completion. Experimental results demonstrate that our proposed model can improve the completion rate of urgent services compared to other conventional models. Simulation result reflects that for increasing service load, our proposed heuristic outperformed (up to 21%) the baseline heuristics. The future plan of this work is to utilize approximate computing in serverless edge to improve the performance.

REFERENCES

[1] D. Mathieson *et al.*, "Forces that will shape intelligent-wells development," *Journal of Petroleum Technology*, vol. 59, pp. 14–16, Aug. 2007.

[2] H. K. White, P.-Y. Hsing, W. Cho, T. M. Shank, E. E. Cordes, A. M. Quattrini, R. K. Nelson, R. Camilli, A. W. J. Demopoulos, C. R. German, J. M. Brooks, H. H. Roberts, W. Shedd, C. M. Reddy, and C. R. Fisher, "Impact of the deepwater horizon oil spill on a deep-water coral community in the gulf of mexico," *Proceedings of the National Academy of Sciences*, vol. 109, pp. 20303–20308, Feb. 2012.

[3] Cisco, "A New Reality for Oil & Gas: Data Management and Analytics," White paper, April 2015.

[4] WoodMackenzie, "Why are some deepwater plays still attractive?," White paper, Sep. 2017.

[5] Gartner, "Top 10 technology trends impacting the oil and gas industry in 2015," White paper, Mar. 2015.

[6] S. Prabhu, E. Gajendran, and N. Balakumar, "Smart oil field management using wireless communication techniques," *International Journal of Inventions in Engineering & Science Technology*, pp. 2454–9584, Jan. 2016.

[7] P. Liu, X. Li, J. J. Qu, W. Wang, C. Zhao, and W. Pichel, "Oil spill detection with fully polarimetric uavsar data," *Marine Pollution Bulletin*, vol. 62, pp. 2611 – 2618, Dec. 2011.

[8] N. G. Franconi, A. P. Bunger, E. Sejdi, and M. H. Mickle, "Wireless communication in oil and gas wells," *Energy Technology*, vol. 2, pp. 996–1005, Oct. 2014.

[9] P. Liu, X. Li, J. J. Qu, W. Wang, C. Zhao, and W. Pichel, "Oil spill detection with fully polarimetric uavsar data," *Journal of Marine Pollution Bulletin*, vol. 62, pp. 2611–2618, Dec. 2011.

[10] M. Fingas and C. E. Brown, "Oil spill remote sensing: a review," in *Oil spill science and technology*, pp. 111–169, Elsevier, 2011.

[11] M. Maheswaran, S. Ali, H. J. Siegel, D. Hensgen, and R. F. Freund, "Dynamic mapping of a class of independent tasks onto heterogeneous computing systems," *Journal of Parallel and Distributed Computing*, vol. 59, pp. 107 – 131, Jul. 1999.

[12] C. Sonmez, A. Ozgovde, and C. Ersoy, "Edgecloudsim: An environment for performance evaluation of edge computing systems," in *Proceedings of the 2nd International Conference on Fog and Mobile Edge Computing (FMEC)*, FMEC '17, pp. 39–44, May 2017.

[13] R. Goyal, S. Kota, R. Jain, S. Fahmy, B. Vandalore, and J. Kallaus, "Analysis and simulation of delay and buffer requirements of satellite-atm networks for tcp/ip traffic," *arXiv preprint cs/9809052*, Mar. 1998.

[14] B. Lorenzo, J. Garcia-Rois, X. Li, J. Gonzalez-Castano, and Y. Fang, "A robust dynamic edge network architecture for the internet-of-things," *Journal of IEEE Network*, vol. 32, pp. 8–15, Jan. 2017.

[15] C. Chang, A. Hadachi, and S. Srirama, "Adaptive edge process migration for iot in heterogeneous cloud-fog-edge computing environment," *arXiv preprint arXiv:1811.10939*, Nov. 2018.

[16] S. Nastic and S. Dustdar, "Towards deviceless edge computing: Challenges, design aspects, and models for serverless paradigm at the edge," in *The Essence of Software Engineering*, pp. 121–136, 2018.

[17] B. Lorenzo, J. Garcia-Rois, X. Li, J. Gonzalez-Castano, and Y. Fang, "A robust dynamic edge network architecture for the internet-of-things," *Journal of IEEE Network*, pp. 8–15, Jan 2017.

[18] S. Wunderlich, J. A. Cabrera, F. H. P. Fitzek, and M. Reisslein, "Network coding in heterogeneous multicore iot nodes with dag scheduling of parallel matrix block operations," *IEEE Internet of Things Journal*, vol. 4, pp. 917–933, Aug 2017.

[19] Y. Wang, X. Lin, and M. Pedram, "A nested two stage game-based optimization framework in mobile cloud computing system," in *Proceedings of the 7th IEEE International Symposium on Service-Oriented System Engineering*, pp. 494–502, March 2013.

[20] Z. Hu, Y. Wei, X. Wang, and M. Song, "Green relay station assisted cell zooming scheme for cellular networks," in *Proceedings of the 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery*, pp. 2030–2035, Aug 2016.

[21] D. Cameron *et al.*, "Big data in exploration and production: Silicon snake-oil, magic bullet, or useful tool?," in *SPE Intelligent Energy Conference & Exhibition*, Society of Petroleum Engineers, 2014.

[22] G. K. Parapuram, M. Mokhtari, J. B. Hmida, *et al.*, "Prediction and analysis of geomechanical properties of the upper bakken shale utilizing artificial intelligence and data mining," in *SPE/AAPG/SEG Unconventional Resources Technology Conference*, Unconventional Resources Technology Conference, 2017.

[23] B. Xu, W. Wang, Y. Wu, Y. Shi, and C. Lu, "Internet of things and big data analytics for smart oil field malfunction diagnosis," in *Proceedings of the 2nd IEEE International Conference on Big Data Analysis*, ICBDA '17, pp. 178–181, Mar. 2017.

[24] P. Bogaert, W. Yang, H. Meijers, C. van Dongen, M. Konopczynski, *et al.*, "Improving oil production using smart fields technology in the sf30 satellite oil development offshore malaysia," in *Proceedings of Conference on Offshore Technology*, May 2004.