Experiments with Fog Applications in the Cloud: Automated Execution

K Iyswaryalakshmi, S Priyanka, M.O Ramkumar and
D Jayakumar

May 22, 2022

Experiments with Fog Applications in the Cloud: Automated Execution

Iyswaryalakshmi K
Computer Science and Engineering
IFET College Of Engineering
Villupuram,India
iysmurthy@gmail.com

Priyanka S
Computer Science and Engineering
IFET College Of Engineering
Villupuram,India
priyaswami9057@gmail.com

Ramkumar M O/Associate Professor
Computer Science and Engineering
IFET College Of Engineering
Villupuram, India
ramkumar.mo86@gmail.com

Jayakumar D/Associate Professor
Computer Science and Engineering
IFET College Of Engineering
Villupura,India
jayakumarifetd@gmail.com

Abstract—Fog computing is a novel processing and storage paradigm for computing resources on the ground, in the cloud, and perhaps anyplace in between. However, the fog of testing and benchmarking packages is a concern since runtime infrastructure will be required or may not yet exist. Although procedures for modelling infrastructure test beds exist, they are mostly focused on mimicking area devices. Other strategies employ the intermediary to imitate infrastructure; they can leverage the network or the cloud. As a result, automated test orchestration is not possible. We suggest assessing fog packages on a cloud-based, emulated infrastructure test bed that may be altered according to pre-defined orchestration strategy in this research. Developers have total control over the layout of the infrastructure.

Keywords—Experiment orchestration, fog computing, mockfog, testing, benchmarking, infrastructure emulation.

I. INTRODUCTION

Fog computing evolved to overcome these issues by utilizing many compute nodes between the cloud and smart things, as shown in Fig. 1. Compute nodes include cloudlets and fog nodes, as well as base stations and access points at the network's edge [1], [3]. Fog computing employs such nodes to handle IoT data that is local to the data source and, when necessary, cloud-based processing resources [4]. Many fog computing architectures have been presented so far, with the purpose of dispersing IoT data processing and reducing transmission delay using the network's edge [16]. The bulk of these ideas are layers of compute nodes in fog computing systems [17], [18]. The top of the hierarchy is cloud computing nodes, the middle is network edge compute nodes, and the bottom is IoT devices [19]. IoT devices frequently send data to edge computing nodes. It is also possible to connect to other nodes (for example, the cloud). There have been ideas for alternatives to hierarchical architecture in fog computing. One of the options is flat typologies, in which computational nodes communicate and interact without the need of layers [20]. We suggest that fog apps be evaluated on a cloud-based, simulated infrastructure test bed that may be updated according to a pee-determined orchestration time frame. In an emulated fog environment, virtual cloud computers are set up to closely mirror the real(or-intended)fog infrastructure.Interconnections between the simulated fog machines can be changed to have equivalent characteristics depending on core network parameters obtained from the production environment or assumptions and past application experiences. Actual fog machine performance measurements might also be used to determine resource restrictions for Dockerized1 application containers 2. This allows fog applications to be totally deployed in the cloud, with performance and failure characteristics that are comparable to a real fog deployment. The top of the hierarchical architecture is cloud computing nodes, the centre is compute nodes at the network's edge, and the bottom is IoT devices [19]. The physical fog infrastructure, on the other hand, will only be available for a limited time: between the completion of the actual fog infrastructure and the completion of the actual fog infrastructure.Device deployment and live running The infrastructure is likely to be non-existent before that period, and after that, its whole capacity is put to use in production. Other options, whether on a network or in the cloud, basically replicate infrastructure, but they lack the ability to decide tests autonomously.

II LITERATURE SURVEY

Because they appear to have endless resources, elastic scalability, and a simple pay-per-use pricing plan, cloud services are extensively utilized to deploy cutting-edge applications. While this is great for engineers, it will cause some access delays for end users. On the other side, future application areas such as the Internet of Things, autonomous driving, and future 5G mobile apps require low latency access, which is typically achieved by shifting computing to the network's edge. Fog Computing refers to the natural extension of the cloud to the edge, which has received a lot of attention recently. However, we believe that applying the service-oriented computing paradigm consistently to fog infrastructure services would enable Fog Computing gain widespread adoption as a deployment platform. This study explores the key impediments to fog computing adoption and suggests open research challenges based on this motivation [1]. In recent years, the number of IoT devices and sensors has exploded. To handle the processing requirements of mostly geo-distributed IoT devices/sensors, a novel computing paradigm called as "Fog computing" has been developed. Fog computing adds Cloud-based processing, storage, and networking capabilities to IoT devices and sensors in general. The function of the cloud as an intermediary layer between IoT devices/sensors and cloud data centers is discussed in this chapter, as well as recent advancements in the sector. Based on the mentioned difficulties and key traits, we present a Fog computing taxonomy. Previous events are mapped out. This study delves into the notion of fog computing, looks at the main impediments to adoption, and outlines outstanding research issues including enormous volumes of data and quick streams from the edge. As a complement to Edge and Cloud computing, fog processing of various observation streams. We examine many aspects of system design, application characteristics, and platform abstractions in the Edge, Fog, and Cloud settings in this research. Physical and application mobility, privacy sensitivity, and a developing runtime environment are highlighted as distinguishing features of the Edge and Fog layers. This category is driven by IoT application case studies based on first-hand experiences from a variety of fields. We also examine the gap between the promise of fog computing and its actuality, as well as the challenges that must be overcome for the solution to be long-term feasible. Our essay, when read as a whole, can help platform and application developers bridge the gap that still exists in making fog computing practical [6].

III EXISTING SYSTEM

In this functional system, MockFog is made up of three modules: infrastructure emulation, application management, and experiment orchestration.Developers simulate the parameters of their expected (simulated) fog infrastructure in the first module, including the quantity and types of computers and the features of their interconnections.. This setting is used by the infrastructure emulation module is used by the application management module for application container deployment, results gathering, and application shutdown. It is also used by the infrastructure bootstrapping and tear-down modules. The node manager advises the node agents to adjust the network settings of their VMs so that an application's network traffic appears to be routed completely through the cloud VM in this situation. In addition, the node agents employ a specialized management network to ensure that network activities do not interfere with communication with the node manager.The first Wireless Sensor Nodes (WSNs), dubbed motes [1,], were designed to operate on very little power in order to extend battery life or even collect energy. The majority of these WSNs are made up of a large number of microscopic memory motes that act as unidirectional sources for a sink and have limited bandwidth, low energy, and poor processing power (collector). This type of sensor network's responsibilities include environmental sensing, minimal processing, and data forwarding to a static sink. TinyOS2 has become the de-facto standard for this sort of sensor network. Motes have shown to be effective in collecting environmental data in a range of circumstances (humidity, temperature, rainfall amount, light intensity, and so on). In consecutive rounds, To meet the needs of new applications for energy-constrained WSN, multiple sinks, mobile sinks, multiple mobile sinks, and mobile sensors were proposed. However, in situations where actuators must execute real actions in addition to sensing and tracking, they fall short. The term "fog computing" refers to a highly virtualized platform for providing compute, storage, and networking services between end devices and conventional Cloud Computing Data Centers, which are frequently, but not always, located at the network's edge. The graphic below depicts the idealized information and computing architecture for future IoT applications, as well as the fog computing function.

IV MODULE

A typical fog infrastructure consists of a number of fog machines, including part machines, cloud machines, and maybe machines inside the part-cloud community [1]. Builders can use the principles, exceptional practices, or reference designs provided in [20], [21], [22], [23], and [24] if no physical infrastructure exists yet. The infrastructure may be visualized as a graph at a high level, with machines acting as vertices and the community connecting them functioning as edges [25]. This graph can also house machines and community links, which are made up of a device's computational power or available bandwidth. The developer provides such a summary graph for the infrastructure emulation module before allocating homes to vertices and edges. We discuss the device and the community homes that are MockFog is assisting.

A.MACHINE PROPERITIES

Machines are the infrastructural components that run utility programmes. Fog machines are available in a variety of shapes and sizes, ranging from small to large devices like Raspberry Pis5 to computers inside a server rack, such as Cloudlets [2], [3], and AWS EC2 is a public cloud service that provides digital computers. To replicate these types of computers in the cloud, their homes must be properly defined. Compute energy, memories, and garage are common places for machines to live. Network I/O may be any other trendy characteristic; nevertheless, we choose to model it solely as a part of the machine-to-machine community. While the memory and garage houses are self-explanatory, it's worth noting that computational energy may be evaluated in a variety of ways. The number of vCPUs, for example, is used by AWS EC2 to assess a machine's processing capability. Because ordinary fog utility installations seldom exceed 100% CPU load,This, or the number of cores, is a rough approximation that is enough for many application cases. More common overall performance metrics, such as instructions per second (IPS) or floating-factor operations per second (FFOPS), can also be employed (FLOPS). We use Docker's help restrictions in our present proof-of-concept prototype.

B.APPLICATION MANAGING MODULE

A unique container name is specified in the container configuration. Fog applications have a lot of moving parts with a lot of inter-dependencies. For each container in the deployment configuration, developers supply a deployment mapping of application components to virtual machines They can also limit a container's access to CPU and RAM, for example, to balance the resource needs of several containers running on the same VM. The node manager installs dependencies on the VMs, transfers files, and executes the configured containers during the application container deployment process.

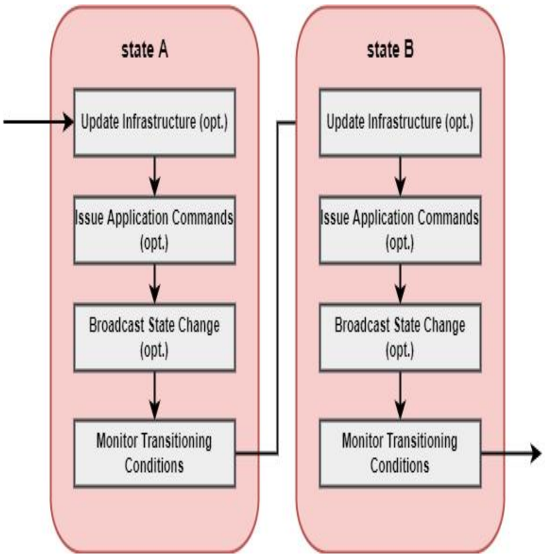

FIG:**APPLICATION MODULES**

All application components must be Dockerized as a precondition. Developers must also explain how application containers will be deployed on infrastructure. SERVER IP and SERVER PORT are configured to the values supplied.in the environment variables when the container is launched, and the application executing inside the container has access to them.

B ORCHESTRATION MODULE

This is in particular beneficial as screw ups are not unusual place in actual deployments however will now no longer always occur even as an usefulness is being evaluated. As a result, simulated screw-ups are the go-to approach for assessing a utility's fault-tolerance and resilience [26]. While MockFog visual displaying units the simulated system to detect deviations from what it put up, extra tracking data is likely of relevance. We recommend either using the cloud vendor's tools, such as Amazon CloudWatch7 when operating on AWS, or deploying bespoke tooling, such as Prometheus, alongside the application using the application management module.

V SMART FACTORY EXAMPLE

At least one additional component of the smart factory application connects with each other. The camera sends its records to production control to check for flaws, which alerts them to items that should be rejected. Adapt packaging transfers the goal packing rate to package control based on data from heating elements and production control.The packing rate is estimated by Adapt packaging depending on the present production pace as well as the backlog of products that have been manufactured but have not yet been packed.The temperature of the input temperature is higher than a set point. For anticipating pickup and aggregation, packaging control provides the current rate and backlog.Forecasts when the next set of things will be available for pickup and alerts logistical outlook. To save bandwidth, Aggregate aggregates several rate and backlog numbers and broadcasts the results to build a dashboard.The information is entered into a database, an

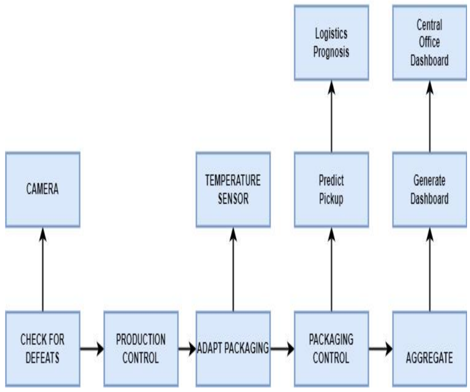

FIG:SMARTFACTORY APPLICATION DIAGRAM

executive summary is prepared, and the information is forwarded to the central office dashboard.

CONCLUSION

We've detailed the concept and core features a platform for fog computing for delivering a diverse at the network's edge a collection of new services and applications .We encourage partnerships on the large volume of Infrastructure of computing ,storage and networking Devices is complex to design .Fog node orchestration and resources management .The Fog will provide support for new services and apps.

REFERENCE

[1] L. Santos, J. Pereira, E. Silva, T. Batista, E. Cavalcante, and J. Leite,
"Identifying requirements for architectural modeling in internet of things applications," in 2019 IEEE International Conference on Software Architecture Companion. IEEE, 2019, pp. 19–26.

[2] V. Karagiannis and S. Schulte, "Comparison of alternative architec
tures in fog computing," in 2020 IEEE 4th International Conference
on Fog and Edge Computing. IEEE, 2020, pp. 19–28.

[3] L. Santos, E. Silva, T. Batista, E. Cavalcante, J. Leite, and F. Oquendo, "An architectural style for internet of things systems," in Proceedings of the 35th Annual ACM Symposium on Applied Computing. ACM, 2020, pp. 1488–1497.

[4] T. Rausch, C. Lachner, P. A. Frangoudis, S. Dustdar, and P. Raith,
"Synthesizing plausible infrastructure confifigurations for evaluat ing edge computing systems," in 3rd USENIX Workshop on Hot Topics in Edge Computing. USENIX, 2020.

[5] A. Kajackas and R. Rainys, "Internet infrastructure topology as sessment," Electronics and Electrical Engineering, vol. 7, no. 103, pp.
91–94, 2010.

[6] D. Bermbach, L. Zhao, and S. Sakr, "Towards comprehensive mea
surement of consistency guarantees for cloud-hosted data storage services," in Technology Conference on Performance Characterization
and Benchmarking. Springer, 2014, pp. 32–47.

[7] K. Morris, Infrastructure as Code: Managing Servers in the Cloud,
1st ed. O'Reilly, 2016.

[8] T. Pfandzelter, J. Hasenburg, and D. Bermbach, "From zero to fog:

Effificient engineering of fog-based iot applications," 2020.
[9] P. Ammann and J. Offutt, Introduction to Software Testing, fifirst
edition ed. Cambridge University, 2008.
[10] O. Skarlat, M. Nardelli, S. Schulte, M. Borkowski, and P. Leitner,
"Optimized IoT service placement in the fog," Service Oriented Computing and Applications, vol. 11, no. 4, pp. 427–443, 2017.
[11] Xiong Z, Feng S, Wang W, Niyato D, Wang P, Han Z (2018) Cloud/fog computing resource management and pricing for blockchain networks. IEEE Internet Things J 6(3):4585–4600.
[12] Mahmud R, Koch FL, Buyya R (2018) Cloud-fog interoperability in IoT-enabled healthcare solutions In: Proceedings of the 19th International Conference on Distributed Computing and Networking, 32.. ACM, Varanasi.
[13] Rahmani AM, Liljeberg P, Preden J-S, Jantsch A (2017) Fog computing in the internet of things: Intelligence at the edge, 17–31.. Springer International Publishing.
[14] Wu C, Li W, Wang L, Zomaya A (2018) Hybrid Evolutionary Scheduling for Energy-efficient Fog-enhanced Internet of Things. IEEE Trans Cloud Comput:1–1.
[15] Mahmud R, Ramamohanarao K, Buyya R (2020) Application Management in Fog Computing Environments: A Taxonomy, Review and Future Directions. ACM Comput
[16]Dastjerdi AV, Gupta H, Calheiros RN, Ghosh SK, Buyya R (2016) Chapter 4 - fog computing: principles, architectures, and applications. In: Buyya R Vahid Dastjerdi A (eds)Internet of Things, 61–75.. Morgan Kaufmann.
[17] Bellavista P, Berrocal J, Corradi A, Das SK, Foschini L, Zanni A (2019) A survey on fog computing for the Internet of Things. Pervasive Mob Comput 52:71–99.
[18]Okafor KC, Achumba IE, Chukwudebe GA, Ononiwu GC (2017) Leveraging fog computing for scalable IoT .
datacenter using spine-leaf network topology. J Electr Comput Eng 5(4):3246–3257.
[19] Morabito R (2017) Virtualization on Internet of Things edge devices with container technologies: a performance evaluation. IEEE Access 5:8835–8850.
[20]Geerling J (2020) Power Consumption Benchmarks.

[21] Yousefpour, G. Ishigaki, R. Gour, and J. P. Jue, "On reducing IoT service delay via fog offloading," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 998–1010, 2018.

[22] H.Mahini, A. M. Rahmani, and S. M. Mousavirad, "An evolutionary game approach to IoT task offloading in fog-cloud computing," *The Journal of Supercomputing*, vol. 77, no. 6, pp. 5398–5425, 2021.

[23] R Somula, "A load and distance aware cloudlet selection strategy in multi-cloudlet environment," *International Journal of Grid and High Performance Computing*, vol. 11, no. 2, pp. 85–102, 2019.

[24] S. P. Ahuja and N. Deval, *Research Anthology on Architectures, Frameworks, and Integration Strategies for Distributed and Cloud Computing*, IGI Global, Pennsylvania, United States, 2021.
[25] A.H. Ben Hassen, N. Ayari, and B. Hamdi, "A home hospitalization system based on the Internet of things, Fog computing and cloud computing," *Informatics in Medicine Unlocked*, vol. 20, p. 100368, 2020.

[26] A.V. Dastjerdi, R. Buyya Fog computing: helping the internet of things realize its potential
Computer, 49 (8) (2016),

[27]K. Velasquez, D.P. Abreu, M.R.M. Assis, C. Senna, D.F. Aranha, L.F. Bittencourt, N. Laranjeiro, M. Curado, M. Vieira, E. Monteiro, E. Madeira
J. Internet Serv. Appl., 9 (1) (2018),

[28]J. Byrne, S. Svorobej, K. Giannoutakis, D. Tzovaras, P. Byrne, P.-O. Östberg, A. Gourinovitch, T. Lynn
A review of cloud computing simulation platforms and related environments
Closer 2017: The 7th International Conference on Cloud Computing and Services Science, SCITEPRESS Digital Library, Porto, Portugal (2017),

[29]B. Wickremasinghe, R.N. Calheiros, R. Buyya
Cloudanalyst: a cloudsim-based visual modeller for analysing cloud computing environments and applications
2010 24th IEEE International Conference on Advanced Information Networking and Applications, IEEE, Perth, WA, Australia (2010),

[30]J. Son, A.V. Dastjerdi, R.N. Calheiros, X. Ji, Y. Yoon, R. Buyya
CloudSimSDN: modeling and simulation of software-defined cloud data centers
2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, IEEE, Shenzhen, China (2015),

[31]A. Brogi, S. Forti, A. Ibrahim
How to best deploy your fog applications, probably
2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC), IEEE, Madrid, Spain (2017),

[32]. G. Vinueza Naranjo, E. Baccarelli, M. Scarpiniti
Design and energy-efficient resource management of virtualized networked fog architectures for the real-time support of iot applications
J. Supercomput., 74 (6) (2018).