IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 8, Issue 12, December 2020

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.488

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 8, Issue 12, December 2020 ||

| DOI: 10.15680/IJIRCCE.2020.0812023 |

Study of Edge Computing

Shubhangi S Sutar[1], **Prof.Mrs.Rama S.Bansode**[2]

P.G. Student, Department of Computer Application, P.E.S. Modern College of Engineering College, Pune, Maharashtra, India[1]

Research Scholar, TMV Pune & Asst. Professor, P.E.S. Modern College of Engineering, Pune, Maharashtra, India [2]

ABSTRACT: Early goals of edge computing were to address the costs of bandwidth for data travelling long distance because of the growth of loT-generated data, the rise of real-time applications that need processing at the edge will drive the technology ahead.

KEYWORDS:IOT, CDN(content Deliver Network), VPN(Virtual Private Network)

I. INTRODUCTION

Edge computing is a computing paradigm which brings computation and data storage closer to the location where it's being gathered, rather than relying on a central location that can be thousands of miles away. This is done so that data, especially real-time data, does not suffer latency issues, that can affect an application's performance. It improve response time and also save bandwidth.

The origins of edge computing were created in the 1990s to serve web and video content from edge servers that were deployed close to users. In the early 2000s, these networks evolved to host applications and application components, which resulted in the first commercial edge computing services that hosted applications such as dealer locators, shopping carts, real-time data aggregators, and ad insertion engines. Modern edge computing significantly extends this approach through virtualization technology that makes it easier to deploy and run a wider range of applications on the edge servers.

II. HOW EDGE COMPUTING WORKS

Edge computing works by pushing data, applications and computing power away from the centralized network to its extremes and enables fragments of information to lie scattered across distributed networks of the server. Its target users remain any internet client using commercial internet application services. Earlier available to large-scale organizations, it's now available to small and medium organizations because of the cost reductions in large-scale implementations.

What is Edge Conputing

Placing edge computing workload is easy, especially if you're familiar with setting up a content delivery network (CDN). The concept is the same. You're distributing assets across the globe to reduce latency. The main difference is that, with edge computing, you're distributing software and code instead of static assets, as you would distribute static assets with a CDN.When you have your software and code, you can utilize as many virtual machines or container instances. You can also run code at the edge with server less functions.

In addition, companies can save money by having the processing done locally, reducing the amount of data that needs to be processed in a centralized or cloud-based location.For example devices that monitor manufacturing equipment on a factory floor, or an internet-connected video camera that sends live footage. While a single device producing data can transmit it across a network quite easily, problems arise when the number of devices increases which transmits data at the same time. Instead of one video camera transmitting live footage, multiply that by hundreds or thousands of devices. Not only quality will suffer due to latency, but the costs in bandwidth can be tremendous.

Edge-computing hardware and services help solve this problem by being a local source of processing and storage for many of these systems.Edge devices can include many different things, such as an IoT sensor, an employee's notebook computer, their latest smartphone, the security camera or even the internet-connected microwave oven in the office break room.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |
|| Volume 8, Issue 12, December 2020 ||
| DOI: 10.15680/IJIRCCE.2020.0812023 |

Features

A. Decreased latency
B. Decrease in bandwidth use and associated cost
C. Decrease in server resources and associated cost
D. Added functionality.
E. Improved Performance
F. Reducing Operational Costs

Points	**Edge Computing**
Suitable Companies	Edge Computing is regarded as ideal for operations with extreme latency concerns. Thus, medium scale companies that have budget limitations can use edge computing to save financial resources.
Programming	Several different platforms may be used for programming, all having different runtimes.
Security	Edge Computing requires a robust security plan including advanced authentication methods and proactively tackling attacks.

What are Advantages of using Edge Computing

A. Faster Response Time
B. Reliable operations with intermittent connectivity
C. Security and compliance
D. Cost effective solutions
E. Interoperability between legacy & modern devices

What are disadvantages

A. It can increase attack vectors
B. It requires more local hardware.
C. It requires less of a robust security plan compared to cloud computing

III. EXAMPLE OF EDGE COMPUTING

Consider a building secured with dozens of high-definition IoT video cameras. These are 'dumb' cameras that simply output a raw video signal and continuously stream that signal to a cloud server. On the cloud server, the video output from all the cameras is put through a motion-detection application to ensure that only clips featuring activity are saved to the server's database. This means there is a constant and significant strain on the building's Internet infrastructure, as significant bandwidth gets consumed by the high volume of video footage being transferred. Additionally, there is very heavy load on the cloud server that has to process the video footage from all the cameras simultaneously.

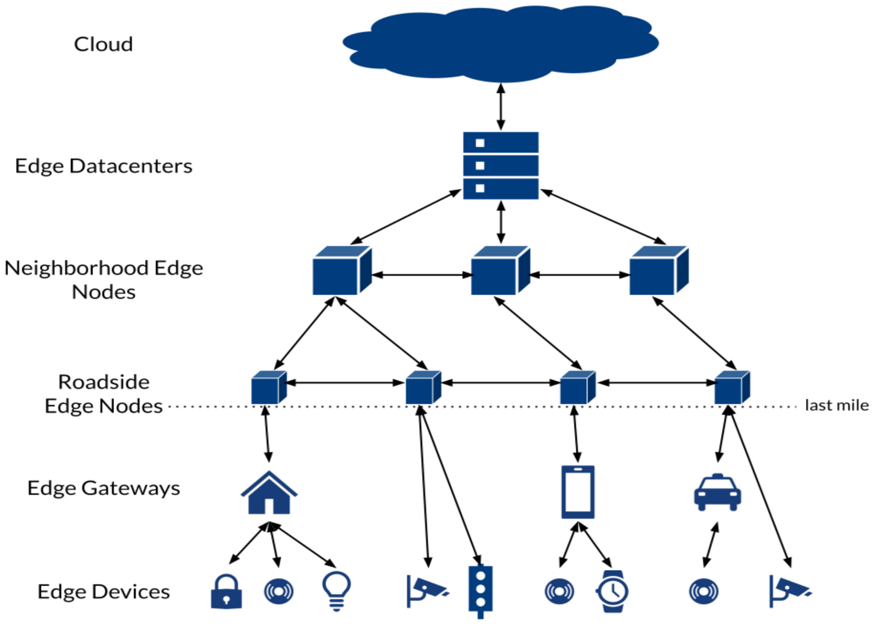
Now imagine that the motion sensor computation is moved to the network edge. What if each camera used its own internal computer to run the motion-detecting application and then sent footage to the cloud server as needed? This would result in a significant reduction in bandwidth use, because much of the camera footage will never have to travel to the cloud server. Additionally, the cloud server would now only be responsible for storing the important footage, meaning that the server could communicate with a higher number of cameras without getting overloaded. This is what edge computing looks like.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 8, Issue 12, December 2020 ||

| DOI: 10.15680/IJIRCCE.2020.0812023 |

IV. CONFIGURATIONS



Edge application services reduce the volumes of data that must be moved, the consequent traffic, and the distance that data must travel. That provides lower latency and reduces transmission costs. Computation offloading for real-time applications, such as facial recognition algorithms, showed considerable improvements in response times, as demonstrated in early research. Further research showed that using resource-rich machines called cloudlets near mobile users, which offer services typically found in the cloud, provided improvements in execution time when some of the tasks are offloaded to the edge node. On the other hand, offloading every task may result in a slowdown due to transfer times between device and nodes, so depending on the workload an optimal configuration can be defined.

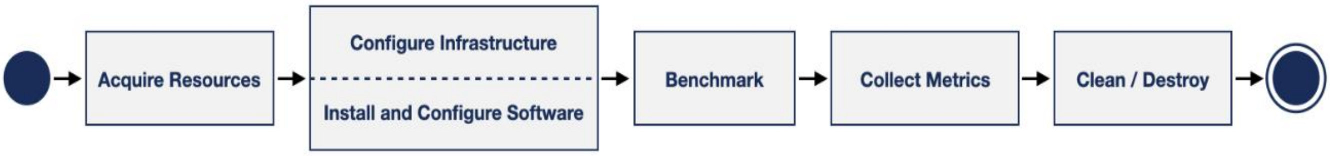
Another use of the architecture is cloud gaming, where some aspects of a game could run in the cloud, while the rendered video is transferred to lightweight clients running on devices such as mobile phones, VR glasses, etc. This type of streaming is also known as pixel streaming.
Other notable applications include connected cars, autonomous cars,smart cities, Industry 4.0 (smart industry) and home automation systems.
Edge computing can help lower dependence on the cloud and improve the speed of data processing as a result. Besides, there are already many modern IoT devices that have processing power and storage available. The move to edge processing power makes it possible to utilize these devices to their fullest potential. It helps in faster response time, Reliable operations with intermittent connectivity, Security & compliance, cost effective solutions, Interoperability between legacy & modern devices.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 8, Issue 12, December 2020 ||

| DOI: 10.15680/IJIRCCE.2020.0812023 |

V. EDGE COMPUTING ENROLLMENT PROCESS



The first step describing the acquisition of resources from a testbed is not specific to edge computing scenarios. The assigned resources (e.g., compute, storage, network) represent the physical infrastructure that will be used to conduct the evaluation.

The second phase is more difficult. It incorporates multiple sub steps to prepare the physical infrastructure as well as the deployment of the system under test (SUT). As edge environments can be very complex, they also need to be tested for their ability to be prepared for circumstances such as an unreliable network connection. Therefore, having a deployment tool that supports a declarative approach is preferred to specify the characteristics of the infrastructure such as latency, throughput and network packet loss ratio to emulate the targeted real life scenario and circumstances.
Once the deployment plan has been created and the resources have been selected, it needs to be confirmed that the infrastructure is configured correctly during the pre-deployment phase before installing the applications and services on top. This is especially true in edge architectures where resources must be available over complex networking topologies. The purpose of this procedure is to ensure that the deployment step will be completed successfully and result in a test environment that is aligned with the requirements and plans.
The next step is to deploy the software applications on the infrastructure. Majority of these tools must be designed with the limitations of one datacentres as their scope, which means that there is an assumption that the environment can scale further during operation, while edge infrastructures are geographically distributed and often have limited resources in the remote nodes. In addition, the configuration options are significantly different among the different models. To ensure the success of testing, the installation itself needs to be verified.
When all the preparations are done, the next step is benchmarking the entire integrated framework. Benchmarking is often defined as performance testing, but here it applies to a broader scope that includes integration and functional testing as well. It is also important to note that the test suites can be heavily dependent on the use case, so they need to be fine tuned for the architecture model being used. While a few tools exist to perform network traffic shaping and fault injections, the challenge lies more in the identification of values that are representative to the aforementioned edge use cases.

Building an edge infrastructure consists of various well known components that were not implemented specifically for edge use cases originally. Because of that, there are situations where there will be a need to test basic functionality in these environments as well to make sure they work as expected in other scenarios. Example functions include:

- create/delete a resource (user, flavor, image, etc); scope: one or more edge sites
- list instances (VM, container); scope: an edge site or 'single pane of glass' dashboard
- create resources for cross-data-center networks

Further testing of the edge infrastructure needs to take the choice of architectural model into consideration:

- Using OpenStack in the centralized control plane model depends on the distributed virtual router (DVR) feature of the OpenStack Network Connectivity as a Service (Neutron) component.
- The behavior of the edge data centers in case of a network connection loss might be different based on the architectural models. In some cases, the decision might be to choose to configure the system to keep the instances running while in other cases, the right approach would be to destroy the workloads in case the site becomes isolated. In addition to these considerations, the expectations on functions such as auto-scaling will also be different due to possible resource constraints, which need to be reflected in the test suites as well.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 8, Issue 12, December 2020 ||

| DOI: 10.15680/IJIRCCE.2020.0812023 |

The final two steps are trivial. The test results need to be collected and evaluated, before returning the SUT infrastructure to its original state.

Tools such as Enos, Enos-Kubernetes and enoslib are available in the experiment-driven research community to evaluate Open Stack and Kubernetes in a distributed environment over Wide Area Network (WAN) connection. They can be extended or leveraged as examples of solutions that can be used to perform the above described process to evaluate some of the architecture options for edge. Further components are needed to ensure the ability to test more complex environments where growing numbers of building blocks are integrated with each other.

VI. CONCLUSION

This paper deals with the all aspects of Edge Computing., as what we have examined in this paper, we can clear that a good and established well-defined network can be created by edge computing rather than cloud computing. We believe that next-gen computing will be influenced a lot by Edge Computing and will continue to explore new use-cases that will be made possible by the Edge.

REFERENCES

1. Hamilton, Eric (27 December 2018). "What is Edge Computing: The Network Edge Explained". cloudwards.net. Retrieved 2019-05-14.
2. "Globally Distributed Content Delivery, by J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman and B. Weihl, IEEE Internet Computing, Volume 6, Issue 5, November 2002"(PDF). Archived (PDF) from the original on 2017-08-09. Retrieved 2019-10-25.
3. Jump up to:[a] [b] Nygren., E.; Sitaraman R. K.; Sun, J. (2010). "The Akamai Network: A Platform for High-Performance Internet Applications" (PDF). ACM SIGOPS Operating Systems Review. **44** (3): 2–19. doi:10.1145/1842733.1842736. S2CID 207181702. Archived(PDF) from the original on September 13, 2012. Retrieved November 19, 2012. See Section 6.2: Distributing Applications to the Edge
4. ^ Davis, A.; Parikh, J.; Weihl, W. (2004). "EdgeComputing: Extending Enterprise Applications to the Edge of the Internet". 13th International World Wide Web Conference. doi:10.1145/1013367.1013397. S2CID 578337.
5. ^ IEEE DAC 2014 Keynote: Mobile Computing Opportunities, Challenges and Technology Drivers
6. ^ MIT MTL Seminar: Trends, Opportunities and Challenges Driving Architecture and Design of Next Generation Mobile Computing and IoT Devices