Survey on Resource Management / Scheduling Techniques

Fog/Edge Computing Emphasis and Summary

Marcus Casey
University of Nevada - Reno
CS 491
Reno, Nevada
marcuscasey@unr.edu

***Abstract*—This paper's purpose is to inform the reader on the basics of fog/edge computing, develop more understanding of fog/edge computing management and scheduling techniques, implementations of fog/edge computing, and to determine what exactly can be created from further investment and implementations of fog computing. Initially this paper was to survey only on scheduling/management techniques, however due to lack of material to populate a paper on both from the author's understanding of the subject and inability to research new methods of scheduling management, the paper evolved to form an encompassing piece on the basis of fog/edge computing and its intricacies. (With the research proposal's acceptance permission)**

Keywords*—*fog node; edge node; Internet of Things (IoT); cloud; server; server implementation

I. Characteristics

Fog & Edge computing help to extend cloud computing to the edge of networks; mostly wireless networks for the Internet of Things (IoT). Fog/Edge computing is closer to data and input compared to most large datacenters. The fog/edge computing implementations right now are found to be in two or three regions around the world. [4] Fog/edge computing has many different characteristics and descriptions, some of which include:

1. Computing nodes are spread across the world in large numbers.
2. Code runs on fog nodes that is part of a distributed cloud application. (DCA)
3. Nodes help applications that rely on geographical location and context with awareness more efficiently.
4. Nodes are not in the main cloud datacenters, located at the edge.
5. Edge and Fog nodes provide lower latency as well as predictable latency as a result.
6. Edge and Fog nodes help the portability of devices when a device is too far away by redirecting a node to a more appropriate location.

II. Need Of Technology

The need for Fog/Edge computing relies on the current system of IoT applications being locked into a two-tiered system, which can't meet needs for low latency, locational capabilities, and portability. [3] Additionally, the current system can only run the first operation on its own, then get sent to a datacenter. They can't complete operations independently.

With Fog/Edge computing—multi tiered architecture with the physical object, sent to a fog node, then finished on the data center allows for a low latency, efficient, and scalable platform in which to create and deploy technologies which can meet the needs of the present and future in server networking. [9]

III. Modern Utilization

Currently not many systems have embraced fog/edge computing into their server and cloud management. This is due to the fact that not many organizations or companies currently understand how fog/edge computing can be handled effectively without too much cost into reworking and re-implementing systems.

Most implementations are academic or classified as R&D. Currently only three major entities have implemented successful research into the field. [8] These are:

1. Carnegie Mellon University—Carnegie has developed a system known as a "cloudlet" which embraces edge computing to create a system in which the cloud is closer to the overall data system and network system implemented in the main technology.
2. Microsoft Research—Microsoft has utilized Microsoft Azure to experiment with edge networking capabilities with success and have started creating so called "micro datacenters" which can help decrease latency and increase power for developers and end-users. These experiments are small in scale but have the potential for big results when integrated properly due to the fog/edging technology.
3. The European Telecommunications Standards Institute—The ETSI has now classified a specification for "Mobile Edge Computing" which is specifically designed towards creating and opening APIs for mobile edge cloud computing.

Essentially—From these listed the conclusion can be drawn that while fog computing is powerful, there are other alternatives that are viable. However there currently is not many, and while the technology is currently there to implement, many are hesitant as there is little groundwork done for widespread implementations in systems without major restructure. [1]

IV. Deployment at Edge

In deploying at the edge, there are two types of environments: Curated and Non-Curated. Both attempt to achieve a different aspect of edge/fog implementations. [7]

Non-Curated:
Non-Curated environments help to employ a type of edge applications into any of the main clouds that currently exist. It is considered fast, easy, and efficient. Most software companies that are based online would want to use this type of deployment in order to better benefit from reduced latency, and of course, the implementation of the fog now in their scheme. [7]

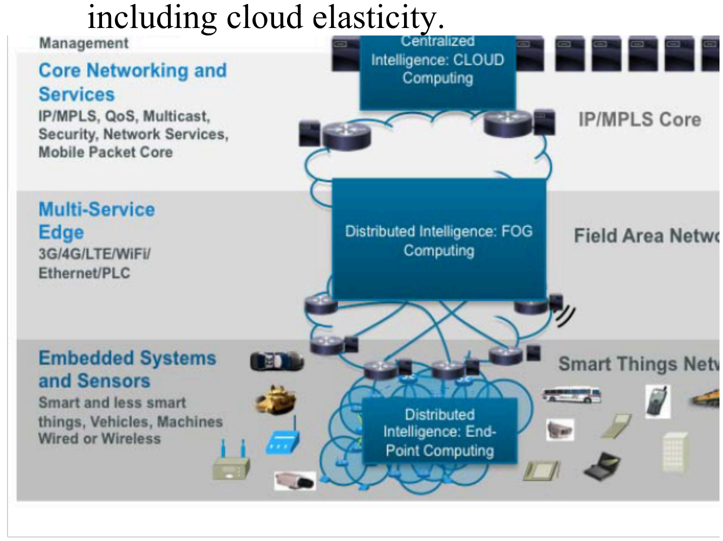
However, the difficulty of deploying in a Non-Curated environment is the method in which applications are able to compute and process beyond domain boundaries. [10] Additionally, resourcing and application deployment would need to be restructured and worked out. So, while the process is "quick and seamless" there are a lot of issues that could arrive when deploying a non-curated environment, specifically in resource management. [6]

Curated environment:
Curated environments require a third-party edge application entity to work with the infrastructure of the provider who sends edge/fog platforms, as well as the operators of the platform.[3] Edge applications will then be tested for security and integrated in a specific manner for the individual client's needs. Curated environments take much longer to implement but require less resource management as the details are figured out with each party and should be able to be adjusted based on need and required implementation techniques. [4]

V. FOG / EDGE RESOURCE CONFIGURATIONS AND MANAGEMENT.

In implementing fog/edge systems resource management and configuration is incredibly important in how a system operates correctly. Below are a some of the major components in ensuring resources are handled correctly in a system as well as their respective physical capabilities when running: [3]

1. Fast Computation—fast computational utility in a fog/edge environment. Using fast computation processing, memory size, and algorithms can all be improved.
2. Security—Fog/edge computing is considered secure in its implantations correctly, which is why it's important to ensure that all software and hardware is secure and correctly configured.
3. Communication—Contains a ton of physical devices such as sensors, hosts, and communication links. Due to the importance of low latency in fog/edge computing implementations, bandwidth and network links are prioritized.
4. Power—Power consumption is reduced by utilizing cooling and UPS devices. Since fog/edge computing requires so many servers powering thousands of devices, power is incredibly important in implementations.
5. Storage—Rather than containing data on a storage device, they are stored at a remote location. A storage utility implementation has thousands of drives, servers, and configurations. Since systems will eventually fail, it is important to be repetitive in implementation. Since the cloud has many different variants in service models, utility has to have features including cloud elasticity.



Fog/edge computing resource management's main goal is to provide high availability of resources, sharing of resources, fulfilling time variant service model, providing efficiency and reliability on resource. Looking from the computing perspective, resource management helps provide stability and clarity of mind in implemented systems. [6]

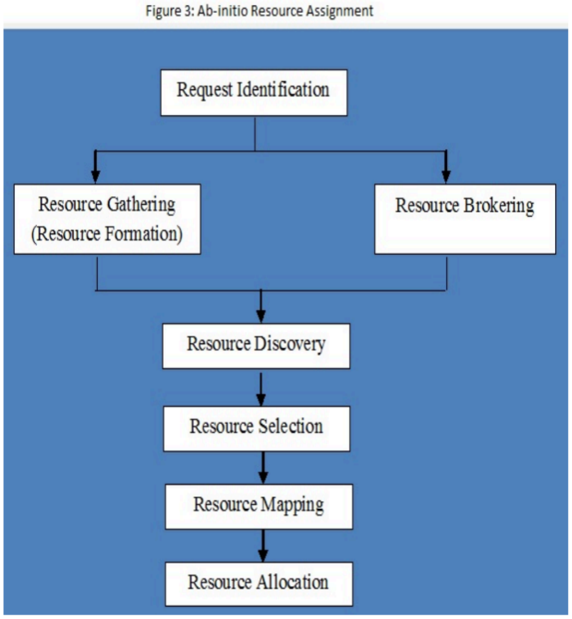
VI. How Resource Management Is Performed in Fog/Edge Computing

There are two major types of resourcing, virtualized resourcing and non-virtualized resourcing. Each has their own pro's and con's and are outlined in their logical steps below: [5]

Non-Periodic Resource Optimization:

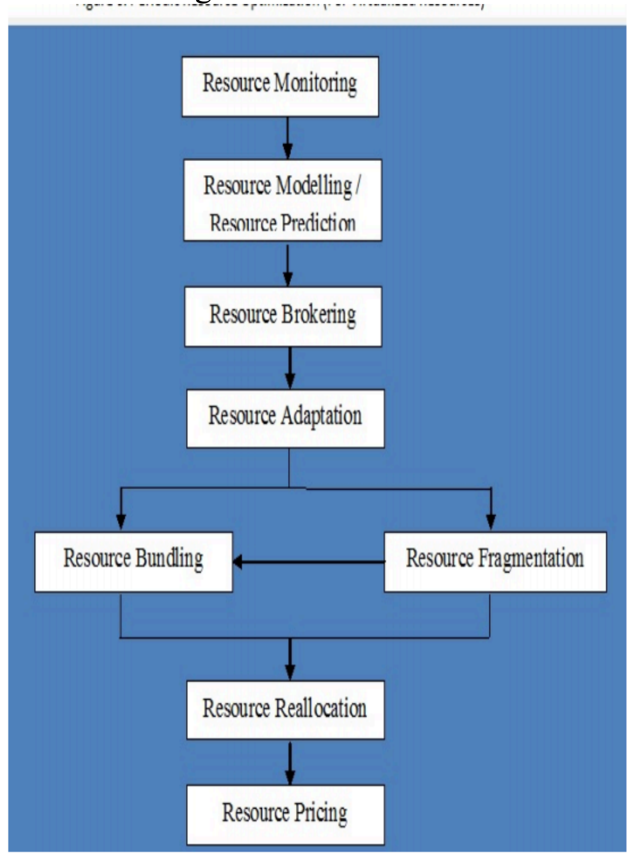
1. Resource Selection—choose best resources that are available that meet requirement provided by request.
2. Resource Mapping—map the best virtual resources with physical resources like fog nodes that are requested.
3. Resource Allocation—Collect then distribute correct resources to cloud requesters. Correctly handle requests so that the system can operate correctly.

Figure 3: Ab-initio Resource Assignment



Periodic Resource Optimization

1. Resource Modeling—Predict non-virtualized resources that are requested. It is very difficult to predict these requests no matter what the traffic on the server currently is.
2. Resource Brokering— Determine that resources are available based on the request.
3. Resource Adaptation—Determine number of requests and scale up or down amount of cloud resources. Can be very costly.
4. Resource Reallocation—Redistribute resources to the cloud customer, make sure implementation works correctly.
5. Resource Pricing—determine pricing on resources so implementations do not go over budget.



VII. PRACTICAL IMPLEMENTATIONS

Fog/edge computing implementations can vary from almost all server-side devices and current implementations. These can extend from your refrigerator to your watch, to your car. A more commonly implemented system (researched) is website performance using fog/edge computing. More dynamic adaptation to the user's conditions can also be accomplished with network edge specific knowledge.

In helping speed up a website with fog/edge computing, webpage rendering performance is improved beyond that achieved by simply applying traditional methods at a webhosting service or CDN. [4]

Device	Edge or fog layer	Main cloud (datacentre)
Provides user interface (I/O, rendering of output)	Hosts as 3rd party apps "network" functions like video acceleration	Provides data storage (long permanence)
Hosts micro-control of actuators	Hosts middleware like registry for edge applications, inter-edge-application communications, services that provide access radio network information. These may be considered components of PaaS at the edge.	Provides human to machine interface for overall application management (e.g. dashboard, deployment, provisioning)
Hosts micro-control of on-board sensors	Offers APIs to 3rd party applications executed at the edge	Hosts visualisation and reporting for operations
Hosts local compute, storage, network stack	Hosts compute-heavy parts of the overall application (e.g. object recognition, motion classification)	Provides off-line, batch data analytics software (maybe real-time analytics as well)
Others	Hosts real-time analytics software	Hosts machine-learning software
	Hosts latency-sensitive control-loop software	Serves queries from device with response time > 100ms 1)
	Issues control commands to devices and actuators	Hosts enterprise integration components
	Collects M2M/IoT data incl. sensor data	Others

The primary focus for practical implementations however revolves around more commonly encountered devices and how they communicate with the cloud. At the Open Fog Consortium, research is being propelled towards electronic clocks, building electronic system implementations, AI labor, medical drones, cellular devices, cars, stoplights, cameras, and airplanes.

All of which benefit from low latency, which is a point of importance in this paper, as the main goal of achieving advancements in fog computing is surrounded by this main goal. [1]

Figure 5: Periodic Resource Optimization (For Non-Virtualized Resources

VIII. CONCLUSION

Throughout researching fog/edge computing, it has been discovered that the topic is still very early in development. Edge and fog computing have very intricate technologies associated with any implementation that may occur. Since server and data-based computations when implemented can be so expensive, it's even more of a deterrent to invest more potential methods and technology in regard to fog/edge computing due to existing systems already being established.

Fog/Edge computing resource management is a very complex and complicated subject, with many different aspects to ensuring successful builds of servers. Due to many different aspects of how queuing and management is handled, fog/edge computing can be very efficient and more beneficial to most systems currently in existence. More research and testing need to be done however, in order for the technology to reach optimum stages for mass implementation.

Acknowledgment

This survey paper acknowledges Dr. Feng Yan's lectures as well as all papers sourced below utilized in the creation of this document. As well as the CS 491 course materials defining more research into the cloud and server management/resources.

References

[1] The Emergence of Edge Computing

[2] Cost Efficient Resource Management in Fog Computing Supported Medical Cyber-Physical System

[3] Combining Mobile and Fog Computing: Using CoAP to Link Mobile Device Clouds with Fog Computing

[4] Improving Web Sites Performance Using Edge Servers in Fog Computing Architecture

[5] Fog computing: A cloud to the ground support for smart things and machine-to-machine networks

[6] Towards power consumption-delay tradeoff by workload allocation in cloud-fog computing

[7] Mobile edge computing Fog et al.: A survey and analysis of security threats and challenges

[8] A proximal algorithm for joint resource allocation and minimizing carbon footprint in geo-distributed fog computing

[9] Fog Computing and Mobile Edge Cloud Gain Momentum Open Fog Consortium

[10] ETSI MEC and Cloudlets