CSSE7014 Distributed Computing
Assignment 2
Semester 1, 2017

Paul Kogel (44644743), Ramdas Ramani (44743767), Andi Nuruljihad (44159069)

May 10, 2017

Contents

1 Introduction **3**

2 Architectures and Models **3**

3 Common Issues **4**

3.1 Networking 4

3.2 Optimal resource use 4

3.3 Fault tolerance 5

3.4 Application Development 5

3.5 Security and Privacy 5

4 Applications **7**

4.1 Real-time health monitoring 7

4.2 Web Site Performance 7

4.3 Smart Vehicles 7

5 References **8**

1 Introduction

Fog computing is a new, exciting computing paradigm [1].

The introduction is clear with several definitions of the computing paradigm under study for comparison. The structure of the report is presented.

2 Architectures and Models

Compare and contrast different architectures and models with examples to back the arguments.

RAM

3 Common Issues

Though still an emerging field, previous research, such as Yi et al.'s "survey on fog computing" [2], has been able to identify multiple potential issues related to fog computing. In this section, we summarise their main findings, and provide additional discussion and research related to them whenever possible.

To improve clarity, we organise issues around 5 main areas: networking, optimal use of resources, fault tolerance, application development, and security and privacy. Note that we focus purely on technical issues. Business-related aspects, such as implementation of a viable business model, and billing mechanisms, are not covered.

3.1 Networking

In order for the fog to function properly, the network has to provide nodes with connectivity, and additional network services, such as routing. The particular nature of the fog, though, makes the implementation of these functions difficult.

For example, the network has to be highly scalable, providing support for a large number of potential nodes. In addition, it should account for constant topology changes due to node mobility. Ensuring that the system is fully distributed is also an important aspect [2].

Using virtualisation mechanisms, such as SDN, has been deemed as viable solution to these issues [2]. In their proposal of a general architecture for the fog, Bonomi et al. [3] state that the fog should use virtualisation for "key resources", including networking. Providing an implementation of SDN for the fog, however, is still an open issue [2]. Partly, this appears to be the case because SDN itself does not put "a high emphasis" on distribution [4].

3.2 Optimal resource use

As stated before, the fog is highly heterogeneous. This heterogeneity is greatly reflected by different degrees of resource availability throughout the system. Important resources are storage, computation power and bandwidth. For example, in parts of the system, available bandwidth might be high due to the presence of more powerful network links, while in others, it can be a scare resource.

Naturally, these resources should be "optimally" used. However, the actual optimisation goal is highly dependent on the use case: for example, in a real-time application, the main objective is to ensure a small delay. Using more bandwidth or computation power to meet this goal is a valid trade-off. For a computation-heavy application running on a mobile device, in contrast, reducing the amount of computation performed locally on the device might be most important.

In their paper, Yi et al. [2] present several strategies that might be used to optimise resource use under different circumstances. Firstly, they suggest that the adequate placement of data can help optimising bandwidth use. In the previously given example of a real-time application, storing data on nodes that are well-connected to the consumer could significantly reduce delay. This placement, however, has to take the dynamic nature of the fog into account. If a node changes location, for example, data placement on the same node can result in small latencies at one time, but introduce great delays at another time. Even if the location of the consumer does not change, available bandwidth at a link might, e.g. if more nodes are interested in the data. Besides placing data, the authors also suggest to place computation. Using "computation offloading", an operation can be partly or fully delegated to a different node in the network. In the aforementioned example of the computation-heavy application, for instance, a more powerful node could perform most of the computation-intense work. Determining which parts of the computation to offload to which nodes, though, can be challenging. As for data placement, this is largely due to the dynamic nature of the fog. Lastly, they present different methods based on the concept of adjusting the network topology. For example, they suggest that effectively choosing the relay nodes for one or multiple endpoints could help reducing delay, while increasing throughput. Again, however, constant changes in the environment, especially the topology, make an implementation challenging.

Focusing less on choosing locations for data and operations, and more on resources actually available at a given node, Aazam and Huh [5] present a resource management method that has been developed especially for the dynamic environment of the fog. At its core, their method predicts the resources required by a consumer for the use of a specific service, and uses these predictions to give "guarantees"

about resource availability. For example, a consumer might get a guarantee of 80% availability for a particular service, meaning that it will have access to all resources required to run the service for most of the time. Predictions are largely dependent on past consumer behaviour. If the node in the previous example had, for instance, frequently disconnected from the service provider, its resource guarantee would be lower. Basically, this means that if not sufficient resources are available, they would preferably be given to nodes that make better use of them. As it can be easily seen, this makes resource allocation considerably fair.

3.3 Fault tolerance

As describe before, the fog mainly uses unreliable wireless network links. In addition, nodes are highly mobile. Being able to ensure availability of services, and provide reliability in general are therefore important aspects.

To improve service availability, Yi et al. [2] suggest to adjust the network topology (see section 3.2). For instance, they present the idea of dividing a network into several clusters, with each cluster centred around a "rich-resource" node.

Traditionally, reliability in a distributed system can be provided by the means of techniques such as checkpointing or rescheduling (see [6]). According to Yi et al. [2], though, most of these techniques are unfit for the fog, as they introduce too much delay. They conclude that replication might work, but they expect it to be difficult to implement due to the distributed nature of the system. Additional research on the topic does not seem to exist. Madsen et al. [7] claim to provide such, but fail to give any actual fog computing-related insights.

3.4 Application Development

As stated in section XX, the fog is dynamic in regards to network topology, and resource availability. In addition, fog nodes might run on different platforms and system architectures [2]. Developing applications that are able to run in this environment, and provide high compatibility, can be expected to be difficult.

To ease development, Bonomi et al. [3] propose a "fog abstraction layer" that hides the underlying heterogeneity, and provides developers with a "uniform and programmable interface". Yi et al. make a similar suggestion by calling for a "unified interfacing and programming model" [2].

Due to issues mentioned in the beginning, though, we expect that the implementation of such a layer is challenging.

3.5 Security and Privacy

Many applications that have been proposed for fog computing are safety-critical, and/or process sensitive data. For example, in vehicle-to-vehicle communication, an insecure system that allows attackers to remotely control the car could have disastrous consequences. In home automation, users might be worried about giving third parties insights into their daily routine.

Stojmenovic and Wen [8] find that providing authentication throughout the system is one of the "main security issues" for fog computing. As an example, they describe a smart meter that is modified by a user, and reports then, due to a lack of proper authentication, false readings. As a possible solution, they suggest encryption at node-level. For this, the meter would encrypt its data, and another node would decrypt it before further forwarding the data. Similarly to this, the OpenFog consortium deems access control (to which it counts authentication) as "key to building a secure system" [9].

In addition to advocating access control, the consortium's reference architecture for fog computing also defines a hardware component called "root of trust" that is "at the heart of the [...] security of the fog node". This component is tamper proof, and required to be implemented by every fog node. It provides security by creating a "chain of trust", i.e., selecting other components such as hardware, software, or other nodes that it considers trustworthy. If a component is compromised, like the smart meter in the example above, it would not gain trust from the root, and therefore not do any harm.

Though access control and the chain of trust promise to provide a solid foundation to a secure fog system, they are both rather general methods. In order to improve security in a given context, it has been suggested in [9] to select additional measures based on the specific use case.

To protect privacy, Yi et al. [2] suggest to run "privacy-preserving" algorithms before data is transferred from the fog to the cloud. As examples, they mention techniques based on differential privacy and homomorphic encryption. Gerla [10] makes an interesting point by stating that moving processing from the cloud to mobile devices alone gives users more control over their data. It can be easily seen, however, that this requires the implementation of adequate control mechanisms. The aforementioned reference architecture [9] vaguely describes "privacy attributes" that a user can assign to his/her data, suggesting that these might be used to control its use.

4 Applications

Fog computing was conceptualized as an extension of the cloud to address services and applications for which the cloud paradigm is not entirely suitable [11]. As a relatively new model, the potential applications and likely infrastructure and design challenges for the fog are still being explored. However, there is a wide range of possible uses of a paradigm that enables real-time, low-latency processing, reduces bandwidth costs, with the benefits of improved security and governance.

4.1 Real-time health monitoring

Wireless Body Area Networks (WBAN) is an important technology in healthcare Internet-of-Things (IoT) applications that allows for the unobtrusive monitoring and recording of various vital signs of a patient in real-time. Many health-monitoring systems employ cloud servers for their low cost, processing power, and high storage volume. The primary drawbacks of relying on the cloud paradigm for such a system are the high latency and the volume of data transmitted over the network by such a large number of sensor nodes. For such a system that is highly dependent on accurate, real-time processing of large amounts of data, there is a need for the reduction of transmitted data that still guarantees quality of service (QoS) [12].

Gia et al. [12] suggest the "provision of an extra layer in between a conventional gateway and a remote cloud server". This added layer, the "fog layer", would serve to preprocess data from the various sensor nodes, expediting the response time of vital applications while simultaneously reducing the volume of data transmitted to the cloud.

4.2 Web Site Performance

There are a great number of factors that influence web page performance over the Internet. Many companies offer website optimization services and tools for web page optimization. However, such services only perform server-side optimizations like smarter organization of stylesheets and scripts, HTTP request optimization, and cache usage maximization without any understanding or knowledge of the network conditions of the user. Recent research shows that, contrary to previous belief, 80-90% of load-time occurs at the front end rather than at the server, primarily during the rendering and execution of a page by the user's browser (Souders, cited in Zhu et al. 2013 [13]). There is presently no implementation of dynamic optimization of a website based on client information that can only be measured near the client's network [13].

4.3 Smart Vehicles

5 References

[1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pp. 13–16, ACM, 2012.

[2] S. Yi, C. Li, and Q. Li, "A survey of fog computing: concepts, applications and issues," in *Proceedings of the 2015 Workshop on Mobile Big Data*, pp. 37–42, ACM, 2015.

[3] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: A platform for internet of things and analytics," in *Big Data and Internet of Things: A Roadmap for Smart Environments*, pp. 169–186, Springer, 2014.

[4] M. Peng, S. Yan, K. Zhang, and C. Wang, "Fog-computing-based radio access networks: issues and challenges," *IEEE Network*, vol. 30, no. 4, pp. 46–53, 2016.

[5] M. Aazam and E.-N. Huh, "Dynamic resource provisioning through fog micro datacenter," in *Pervasive Computing and Communication Workshops (PerCom Workshops), 2015 IEEE International Conference on*, pp. 105–110, IEEE, 2015.

[6] M. Van Steen and A. S. Tanenbaum, *Distributed Systems*. Pearson Higher Education, 2013.

[7] H. Madsen, B. Burtschy, G. Albeanu, and F. Popentiu-Vladicescu, "Reliability in the utility computing era: Towards reliable fog computing," in *Systems, Signals and Image Processing (IWSSIP), 2013 20th International Conference on*, pp. 43–46, IEEE, 2013.

[8] I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," in *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on*, pp. 1–8, IEEE, 2014.

[9] "Openfog reference architecture for fog computing," tech. rep., OpenFog Consortium, 2017.

[10] M. Gerla, "Vehicular cloud computing," in *Ad Hoc Networking Workshop (Med-Hoc-Net), 2012 The 11th Annual Mediterranean*, pp. 152–155, IEEE, 2012.

[11] N. Bessis and C. Dobre, *Big data and internet of things: a roadmap for smart environments*, vol. 546. Springer, 2014.

[12] T. N. Gia, M. Jiang, A.-M. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, "Fog computing in healthcare internet of things: A case study on ecg feature extraction," in *Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on*, pp. 356–363, IEEE, 2015.

[13] J. Zhu, D. S. Chan, M. S. Prabhu, P. Natarajan, H. Hu, and F. Bonomi, "Improving web sites performance using edge servers in fog computing architecture," in *Service Oriented System Engineering (SOSE), 2013 IEEE 7th International Symposium on*, pp. 320–323, IEEE, 2013.