Osmotic Computing Literature Study

Maarten Petit
University of Amsterdam
maarten.petit@student.uva.nl

April 30, 2020

Abstract

Osmotic Computing is a paradigm for Cloud and Edge integration proposed in 2016 by Villari et al. [1]. Osmotic Computing aims at decomposing an IoT application into microservices and involves the dynamic management of these services across the cloud, edge/fog and IoT layers. The paradigm is inspired by the concept of osmosis in chemistry. The Osmotic Computing paradigm inherits aspects of Cloud and Edge computing but extends this and adds several feature.

1 Introduction

The growth of the Internet of Things (IoT) rapidly increases the number and type of devices connected to the internet [2, 3]. These devices induce an enormous growth of the data volume, velocity, variety and veracity. [4, 5] Whilst Cloud computing has become fundamental to delivering services over the internet and multiple mature cloud-centric IoT solutions such as Amazon IoT and Google Cloud Dataflow exist. The consolidation of the ever growing data volume generated by IoT devices and the use of these centralized IoT cloud solutions puts pressure on the internet networking infrastructure which can cause congestion of networking paths leading to centralized cloud datacenters. Especially time, latency and bandwidth sensitive IoT application can suffer from this. To overcome this problem paradigms that disseminate the analysis of raw data to the the network edge have been proposed. Edge [6, 7] and Fog [8–10] computing focus on decentralizing the data processing to reduce the volume of data being transferred to and from the cloud and enable time and privacy sensitive data processing for IoT applications. The goal of the Osmotic Computing paradigm is to provide integration of cloud, edge/fog and IoT layers. This paper describes a literature study on Osmotic Computing and is structured as follows. The literature study research design is discussed first. The Osmotic Computing paradigm and it's scope are discussed in section 3. In Section 4 the Osmotic Computing architecture and it's basic principles are described. To achieve a complete application of the paradigm a number of issues must be solved. These implementation issues of the paradigm are discussed in Section 5. Both IoT and non-IoT applications of the paradigm both are discussed in Section 6. Last, the paper is concluded in Section 7

2 Research Design

In this section the research design consisting of the motivation for the research, the research questions and literature search strategy is discussed.

2.1 Motivation

Osmotic Computing is a paradigm for Cloud and Edge integration proposed in 2016 by Villari et al. [1]. The proposal outlines a high-level description of the paradigm, it's open challenges and proposed research directions. The goal of this literature research is to determine what the paradigm entails, how the paradigm solves the challenges of Cloud/Edge integration in general and for specific case studies.

2.2 Research questions

Proposed research question: *Is Osmotic Computing a pragmatic solution to the challenges of Cloud and Edge integration?*

- What does the Osmotic Computing paradigm entail?
- How does Osmotic Computing solve the challenges of Cloud and Edge integration?
- How is the Osmotic Computing Paradigm applied to specific case studies and if challenges occurred how were they solved?

2.3 Literature Search Strategy

For this literature research a systematic search is applied to find the required and relevant literature. A review protocol

Table 1: Selection Criteria

ID	Criterion
I1	A study that proposes or explains or applies the architecture or architectural elements of Osmotic Computing.
I2	A study that is carried out by either academics or practitioners.
I3	A study that is peer-reviewed
E1	A study surveys edge/fog/cloud computing paradigms
E2	A study that does not cite the paper that introduces the Osmotic Computing paradigm that is considered in this literature study

was set up to perform a consistent literature search. The review protocol consists of a research query and selection criteria.

2.3.1 Research Query

The research query consist of the following keywords:

- Osmotic Computing
- **AND** (Edge **OR** Fog) Computing
- **AND** IoT
- **AND** Cloud

Google Scholar is used to conduct the literature search query.

2.3.2 Selection Criteria

The candidate literature that is collected using the search query is filtered based on a list of selection criteria. The selection criteria are list in Table **??**. The inclusion and exclusion criteria have an identifier starting with I and E respectively. An extended version of the table that includes the rationale for the criteria is included in appendix Z.

3 Osmotic Computing Paradigm

In this section the Osmotic Computing paradigm is discussed. The motivations to develop the paradigm are discussed first. The paradigm and it's concept are discussed. Last an outline of the scope of the paradigm is described.

3.1 Motivation

The centralized processing of data is a fundamental limitation of cloud-centric IoT solutions. Edge and Fog computing aim to solve this limitation but introduce issues related

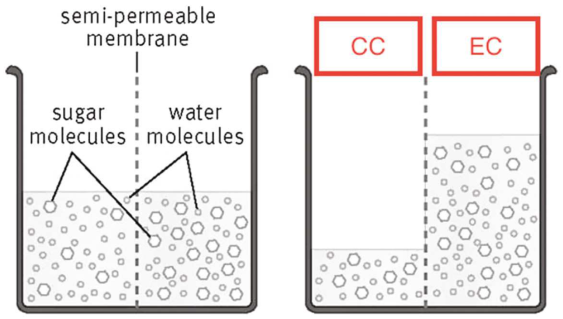

Figure 1: The concept of Osmotic Computing [12]

to application and infrastructure heterogeneity, utilization cost-effectiveness and redundancy. Osmotic Computing, proposed in [1], aims to provide an architecture to develop hybrid virtual environments over the cloud, edge/fog and IoT layers.

3.2 The Paradigm

Osmotic Computing aims at decomposing an IoT application into microservices and involves the dynamic management of these services across the cloud, edge/fog and IoT layers [1]. The paradigm is inspired by the concept of osmosis in chemistry as depicted in Figure 1. Osmosis is the movement of molecules from a lower concentrated solute trough a semipermeable membrane solution to a region of higher solute concentration which results in a equalized solute concentration. In the Osmotic Computing paradigm the solvent component is defined to be the microservices that can move between the different computing layers. The solute is defined application specific but involves metrics on the infrastructure and environment such as processing time, current load and energy. The semipermeable membrane is defined as a service that takes decisions about the movement of the services across the layers. [1, 5, 11] The services in the different layers evolve towards a balanced deployment that satisfies a specified configuration. Osmotic Computing deployment strategies consider (dynamic) requirements related to both infrastructure and applications. [1]

3.3 Scope of Paradigm

The Osmotic Computing paradigm inherits aspects of Cloud an Edge computing but extends this and adds several features which are discussed below.

- **Microservice Configuration**: Osmotic Computing requires holistic decision-making frameworks. To meet the Quality-of-Service constraints these frameworks muse be able to automate the configuration selection of microservices and resources in the cloud and edge datacenters.

- **Networking:** Osmotic Computing incorporate a network abstraction that spawns from the cloud to edge. A interoperability layer that enables distributed microservice orchestration and microservice intercommunication over the different layers is required.
- **Security:** Osmotic Computing requires a coherent security policy supported by both the cloud and edge computing layer.
- **Offloading** Osmotic Computing advocates to combine the mobile offloading and cloud offloading approaches. To effectively combine these approaches one needs understanding of which types of microservices would be relevant to execute on which layer. So to optimize microservice deployment the understanding of the types of microservices and their optimal execution layers is required.
- **Workload Contention and Interference Evaluation:** workload contention among deployed microservices should be detected and handled. Evaluation is required to support the orchestrator in deciding which microservices can be deployed in parallel.
- **Monitoring:** Osmotic Computing involves monitoring of heterogeneous (hardware) resources on both layers, (virtual) networks as well as microservices on both layers. A monitoring framework that is able to support the monitoring of all layers is required.
- **Orchestration:** Osmotic Computing involves feedback-driven runtime orchestration of microservices over both the Cloud and Edge layer. Quality-of-Service, workload-input and contention aware resource orchestration is required. The paradigm proposes that Machine Learning techniques can be used to predict the workload-input and performance of the distributed microservices.

4 Osmotic Computing Architecture

In this section the architecture of the Osmotic Computing paradigm is described. The specific architectural elements are described of which an outline of was discussed in the previous section 3.3.

4.1 Infrastructure Layers

Osmotic Computing defines two main infrastructure layers. The Cloud layer (L1) and Edge Layer (L2). The Cloud layer consists of centralized cloud datacenters and L2 identifies the edge computing layer which constitutes of all other infrastructure available to the application. What constitutes the edge layer is application specific and ranges from Cloudlets (small-scale cloud datacenters located at the network edge), to gateways, to the IoT devices themselves. [1, 11] The layers are composed of different types of resources with specific properties that must be considered when deploying the microservices. For example, most devices at L2 have resource-constraints while resources at L1 are more likely to be limited by the network properties such as latency and bandwidth.

4.2 Osmotic Resources and Services

Osmotic resources are defined to be any IT resource within the Osmotic Computing framework spread among the different internet layers. The different layers can include many different osmotic resources such as a virtual machine in a private cloud data in the cloud layer, a Cloudlet in the Fog layer or a Raspberry Pi in the IoT layer. All resources that are used to dynamically run microservices withing the osmotic application are considered osmotic resources

4.3 Application Decomposition

An osmotic application is decomposed into microservices that are not confined in a specific layers but can find collocation among the different layers. Requirements on the microservices is that they are mobile, portable and cross-platform The osmotic computing paradigm introduces the concept of Micro Elements (MELS) that decomposes into two main components the Micro Services (MS) and Micro Data (MD). The Micro Services is partitioned into Micro Operational Service (MOD) and Micro User Service (MUS). The Micro Data is partitioned into Micro Operational Data (MOD) and Micro User Data (MUD). Examples of a MS and MD are a Docker container containing an application and a JSON data entity respectively.

4.4 General Purpose MELS

An Osmotic application is decomposed in to application specific MELS tha fulfill the functional requirements of the application. To fulfill the non-funcitonal requirements of the osmotic application such as network and security management, general purpose MELS are required. General purpose MELS have the requirement of being cross-platform as

An Osmotic application consists of application specific and general purpose microservices. General purpose microservices are required to fulfill non-functional requirements of the Osmotic application such as network and security management. Unlike in Cloud

4.5 Software Defined Membranes

The Software Defined Membranes (SDMem) are an essential part of any osmotic application. The software defined membrane is responsible for the decision making on the inner en inter layer movement of MELS. The SDMem is a special type of MEL that allows movement of MELS.

4.6 Feedback Driven Orchestration

Microservices must be automatically adapted to the deployment sites, considering location and context.

5 Open Challenges

6 Applications

6.1 IoT Applications

6.2 Non-IoT Applications

The Osmotic Computing paradigm is not only applied to IoT related computing architectures.

7 Conclusion

References

[1] Massimo Villari et al. “Osmotic computing: A new paradigm for edge/cloud integration”. In: *IEEE Cloud Computing* 3.6 (2016), pp. 76–83.

[2] Dave Evans. “The internet of things: How the next evolution of the internet is changing everything”. In: *CISCO white paper* 1.2011 (2011), pp. 1–11.

[3] Statista Research Department. *IoT: number of connected devices worldwide 2012-2025*. Nov. 2019. URL: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/.

[4] Andy Daecher and Robert Schmid. “Internet of Things: From sensing to doing”. In: *Wall Street Journal* (2016).

[5] Mirjana Maksimović. “The role of Osmotic computing in Internet of Things”. In: *2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH)*. IEEE. 2018, pp. 1–4.

[6] Mahadev Satyanarayanan et al. “Edge analytics in the internet of things”. In: *IEEE Pervasive Computing* 14.2 (2015), pp. 24–31.

[7] Weisong Shi and Schahram Dustdar. “The promise of edge computing”. In: *Computer* 49.5 (2016), pp. 78–81.

[8] Flavio Bonomi et al. “Fog computing: A platform for internet of things and analytics”. In: *Big data and internet of things: A roadmap for smart environments*. Springer, 2014, pp. 169–186.

[9] Flavio Bonomi et al. “Fog computing and its role in the internet of things”. In: *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. 2012, pp. 13–16.

[10] Shanhe Yi et al. “Fog computing: Platform and applications”. In: *2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb)*. IEEE. 2015, pp. 73–78.

[11] Vishal Sharma et al. “Managing service-heterogeneity using osmotic computing”. In: *arXiv preprint arXiv:1704.04213* (2017).

[12] Massimo Villari, Antonio Celesti, and Maria Fazio. “Towards osmotic computing: Looking at basic principles and technologies”. In: *Conference on Complex, Intelligent, and Software Intensive Systems*. Springer. 2017, pp. 906–915.