Distributed Cloud Intelligence: Implementing An ETSI MANO-Compliant Predictive Cloud Bursting Solution using Openstack and Kubernetes

Francescomaria Faticanti[1,2], Jason Zormpas[3], Sergey Drozdov[3], Kewin Rausch[1], Orlando Avila García[4], Fragkiskos Sardis[3], Silvio Cretti[1], Mohsen Amiribesheli[3], and Domenico Siracusa[1]

[1] Fondazione Bruno Kessler, Italy
[2] University of Trento, Italy
[3] Konica Minolta Laboratory Europe
[4] Atos, Spain

Abstract. While solutions for cloud bursting already exist and are commercially available, they often rely on a limited set of metrics that are monitored and acted upon when user-defined thresholds are exceeded. In this paper, we present an ETSI MANO compliant approach that performs proactive bursting of applications based on infrastructure and application metrics. The proposed solution implements Machine Learning (ML) techniques to realise a proactive offloading of tasks in anticipation of peak utilisation that is based on pattern recognition from historical data. Experimental results comparing several forecasting algorithms show that the proposed approach can improve upon reactive cloud bursting solutions by responding quicker to system load changes. This approach is applicable to both traditional datacentres and applications as well as 5G telco infrastructures that run Virtual Network Functions (VNF) at the edge.

Keywords: Cloud Bursting · Proactive Control · Application Metrics · Workload Orchestration.

1 Introduction

Todays diverse utility-based computing ecosystem cannot function without relying on the cloud paradigm. The paradigm has disturbed all the existing computing tasks. At its core, it decouples applications from hardware and allows for increased and elastic scaling of compute and storage. It achieves this, through the implementation of virtualised infrastructures and platforms on top of commodity hardware. It is worthy to note that, although legacy monolith applications can be migrated to the cloud, only cloud-native ones can fully benefit from cloud computing features such as automatic scaling, failover and self-healing.

To enable the users to take full advantage of the cloud computing paradigm, Konica Minolta is working on an advanced all-in-one data-driven cloud platform called Distributed Cloud Intelligence (DCI). DCI is an optimised Platform as a Service (PaaS) for the particular needs of the next chapter of applications in areas such as smart cities, data analytics, computer vision, IoT and robotics. In the following study, in close collaboration with Fondazione Bruno Kessler and Atos, DCI portrays a PaaS capable of edge-centric cloud bursting. The following work will showcase how DCI can enable businesses to efficiently handle peak IT demands. As an instance, if all of the on-premise resource capacity of an organisation is utilised, the overflow traffic is directed to a centralised cloud (e.g., public) so theres no interruption of services. Additionally, given the agreed Service Level Agreements (SLAs), DCI removes the costs of raw data transfer to the centralised cloud by performing the heavy processes and pre-processes at the edge locations. The work illustrates that leveraging deep learning techniques, DCI will tremendously lower the data transfer costs and delays. Deep learning methods perform a proactive control of system workload in order to prevent overflow situations in the resource utilisation and requirements' violations in the applications' performances.

The remainder of the paper is structured as follows. The System Architecture is shown in Section 2. The predictive cloud bursting method and experimental results are presented in Section 3. A concluding section ends the paper.

2 System Architecture

The system is comprised of three components: i) the edge datacentre where end-user applications are hosted by an organisation that wishes to employ cloud-bursting, ii) the remote cloud which can be a public cloud or a remote hosting facility that can offer its resources for task offloading, iii) and the Jump server which is a hardened host that performs light-weight orchestration functions in the form of collecting and processing performance metrics from the application and the infrastructure. The implementation presented in this paper uses Openstack as the cloud platform on the edge and remote clouds and Kubernetes as the container orchestration engine for hosting applications. Specifically, we use Openstack VMs to deploy a Kubernetes cluster that will host the end-user applications subject to cloud-bursting. The resource utilisation metrics from Openstack's VMs are monitored at the Jump server. Once the resource utilisation conditions are met on the edge, the Jump server is able to initiate a cloud burst of microservices from the edge to the remote cloud. When certain conditions are met, the Jump server will communicate directly with Openstack on the remote cloud to deploy additional Kubernetes slave instances for task offloading or delete instances that are no longer required.

3 Predictive Cloud Bursting

In this section, we analyse the integration of machine learning components to the Cloud environment in order to achieve a predictive Cloud bursting system and we present some experimental results. In what follows we describe how the data

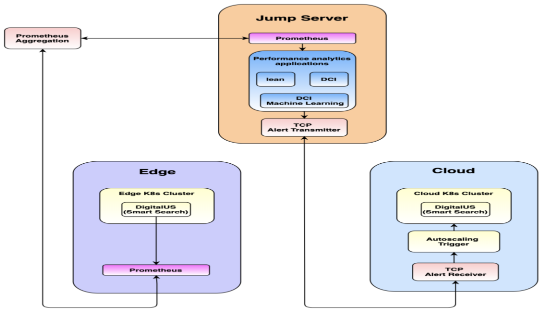

Fig. 1. System Architecture Based on Konica Minolta DCI

were collected and pre-processed for the application of the Machine Learning models. Finally, we describe the model selected to perform the forecast of peak demands and the obtained results.

Data Collection. After the metrics were identified and their effects on the behaviour of the application (Konica Minolta Semantic Search) and the system (Cloud environment) was analysed, the InfluxDB and the Prometheus APIs were utilised to collect the timeseries data from these metrics. Historic data for the systems CPU Usage and Load Average were collected utilising the InfluxDB for the purpose of training the machine learning algorithm, while real-time data were collected utilising the Prometheus APIs for the purpose of predicting a possible overloading of the system.

Data Pre-processing. The following three data transforms are performed on the dataset prior to fitting a model and making a forecast:
1)**Transform the time series data so that it is stationary**. Specifically, we aim to remove the increasing trends in the data. This can be skipped since our data looks stable.
2)**Transform the time series into a supervised learning problem**. Specifically, the organisation of data into input and output patterns where the observation at the previous time step is used as an input to forecast the observation at the current time time-step.
3)**Feature scaling** (also known as data normalization [1]) is the method used to standardize the range of features of data. Since, the range of values of data may vary widely, it becomes a necessary step in data pre-processing while using machine learning algorithms. This method is used to automatically scale up or down the number of resources based on demand at any time. Essentially, the process entails transforming the values of the data from the original range to a value that is within the range of 0 to 1. The formula for feature scaling is $X_{scaled} = \frac{X - \min X}{\max X - \min X}$, where X is the original feature value, and X_{scaled} is the normalized one [1].

LSTM Model Description. A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence [2].This allows it to exhibit temporal dynamic behaviour. Derived from feedforward neural networks, RNNs can use their internal

Table 1. LSTM Model for real-time load average prediction

Upload Freq	**Job**	**Metric Prediction Time**	**ML Training Time**
Slow (3-10 files/m)	System Load Average	0.85 sec	3 min 56 sec
Medium (1-3 files/s)	System Load Average	1.02 sec	4 min

state (memory) to process variable length sequences of inputs. Long short-term memory is an artificial recurrent neural network architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points, but also entire sequences of data.

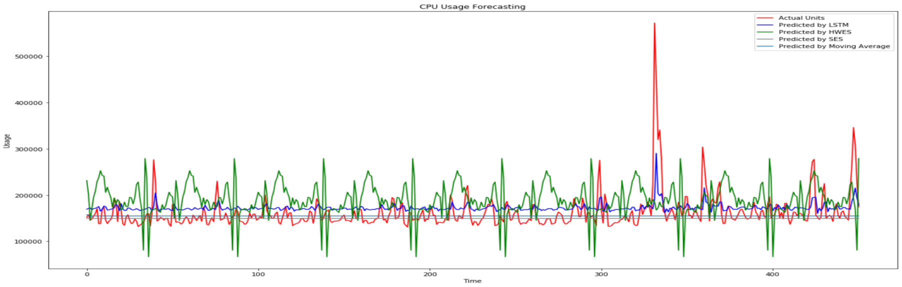

Fig. 2. An Illustration of the Time Series forecasting models predicting values in actual data (CPU Usage vs Time).

Experimental Results. The purpose of our experiment was to prove that a Machine Learning algorithm could predict, in real-time, whether a system (DCI Private Cloud) would overload, thus alerting it in advance to trigger the Cloud Bursting functionality (to the DCI Public Cloud). In order to achieve this, the recurrent neural network called Long Short Term Memory was selected amongst multiple Statistical (e.g. Moving Average, Holt-Winter Exponential Smoothing) and Machine Learning networks (Sarima and Sarimax), as its accuracy of predicting Time-Series data outperformed all the other models as seen in Figure 2.

The methodology used to conduct the experiment was the following:

- Multiple files (pdf, txt) were uploaded to the file server.
- The LSTM model was executed to collect new historical Time-Series data (Systems Load Averages) from InfluxDB and was tasked to train on them.
- Using API calls to Prometheus the real-time Time-Series data (System Load Averages) were received to the program.
- The LSTM model trained on the historical data from InfluxDB and made predictions/forecasts on the real-time Time-Series data that were collected from the API calls to Prometheus.

Test Results. The model was trained [Table 1] and tested [Figure 3] using real Time-Series data from the systems load averages. As we can see in Figure 2

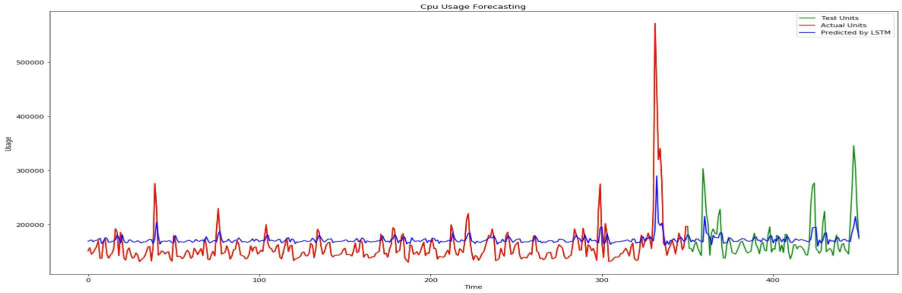

Fig. 3. CPU Usage Forecasting (CPU Usage vs Time).

and Figure 3, LSTM accurately predicts most of the trends in our data.

4 Conclusions & Future Work

In recent years, cloud computing paradigm has attracted a growing amount of attention from industry and academic players. This has occurred because of cloud computings ability to dynamically provision resources on-demand. In this context cloud bursting techniques are crucial to avoid overflows in resource utilisation and to monitor applications' performances. In this work we described a new cloud platform (DCI) able to perform cloud bursting using predictive control methods based on deep learning approaches. DCI demonstrates a real-time prediction-based solution that controls the auto-scaling process of cloud infrastructures by modifying the configuration of service mesh underlying the computing resources. This deep learning-based solution is independent of distributed frameworks (i.e., Kubernetes and OpenStack) and therefore applies to any other cloud infrastructures. Future works involve the support for new emerging technologies such as 5G and Edge-centring Machine Learning such (e.g., Federated Learning).

Acknowledgments

This work has received funding from the European Unions Horizon 2020 Research and Innovation Programme under grant agreement no. 815141 (DECENTER: Decentralised technologies for orchestrated Cloud-to-Edge intelligence) and internal funding from Konica Minolta by providing support for the Light-Edge project in collaboration with FBK.

References

1. Data Science on Prometheus Metrics. https://github.com/AICoE/prometheus-data-science
2. Understanding LSTM Networks. https://colah.github.io/posts/2015-08-Understanding-LSTMs/