Accepted Manuscript

Predicting host CPU utilization in the cloud using evolutionary neural networks

Karl Mason, Martin Duggan, Enda Barrett, Jim Duggan, Enda Howley

PII: S0167-739X(17)32279-3
DOI: https://doi.org/10.1016/j.future.2018.03.040
Reference: FUTURE 4053

To appear in: *Future Generation Computer Systems*

Received date : 5 October 2017
Revised date : 13 March 2018
Accepted date : 21 March 2018

Please cite this article as: K. Mason, M. Duggan, E. Barrett, J. Duggan, E. Howley, Predicting host CPU utilization in the cloud using evolutionary neural networks, *Future Generation Computer Systems* (2018), https://doi.org/10.1016/j.future.2018.03.040

Predicting Host CPU Utilization in the Cloud using Evolutionary Neural Networks

Author One, Author Two, Author Three, Author Four, Author Five

Address

Abstract

The Infrastructure as a Service (IaaS) platform in cloud computing provides resources as a service from a pool of compute, network, and storage resources. One of the major challenges facing cloud computing is to predict the usage of these resources in real time. By knowing future demands, cloud data centres can dynamically scale resources to decrease energy consumption while maintaining a high quality of service. However cloud resource consumption is ever changing, making it difficult for accurate predictions to be produced. This motivates the research presented in this paper which aims to predict in advance the level of CPU consumption of a host. This research implements evolutionary Neural Networks (NN), a powerful machine learning method, to make these predictions. A number of state of the art swarm and evolutionary optimisation algorithms are implemented to train the neural networks to predict host utilization: Particle Swarm Optimisation (PSO), Differential Evolution (DE) and Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES). The results of this research demonstrate that CMA-ES converges faster to a better solution on the training data. However when evaluated on the test data, DE performs statistically equal to CMA-ES. The results also demonstrate that the trained networks are still accurate when applied to CPU utilization data from different hosts with no further training needed. When evaluated to predict multiple steps into the future, the accuracy of the network understandably decreases but still performs well on average.

Keywords: Cloud Computing, CPU Prediction, Neural Networks, Optimisation, Differential Evolution, Particle Swarm Optimisation, Covariance Matrix Adaptation Evolutionary Strategy, Time Series, Neuroevolution,

1. Introduction

Predicting resource utilization has been listed as one of the ten biggest obstacles facing the growth of cloud computing [1]. One of the major difficulties for prediction algorithms in cloud computing is that cloud resources are constantly changing and exhibit a complex dynamic behavior. Artificial Intelligence (AI) and Machine learning (ML) algorithms such as Neural Networks have been shown to improve upon traditional models, e.g. ARIMA. Forecasting methods such as ARIMA rely on patterns in historical data to make future predictions. These approaches are not suitable when there is not a distinct pattern in the data or if there is significant amount of random variation in the data [2]. AI and ML methods are much more adaptable and robust than these traditional approaches. The robustness of ML approaches, such as neural networks, is due to the few assumptions made and requirements they have about the data set.

It is estimated that by 2020, there will be 51,974 GB of internet traffic generated per second [3]. This staggering amount of internet traffic combined with the trend of technology companies moving towards offering computing as a service via cloud computing will result in the generation of massive volumes of data being processed by hosts and Virtual Machines (VMs). The advantage of ML methods is that they can make use of this increased volume of data to make more accurate prediction models for cloud resources.

Email addresses: author.one@mail.com (Author One), author.two@mail.com (Author Two), author.three@mail.com (Author Three), author.three@mail.com (Author Four), author.three@mail.com (Author Five)