



Optimising Virtual Network Performance with Super-Cobb-Douglas CRS-I DEA for Fair Cloud Tenant Accountability

Francisco Daladier Marques Júnior¹

IFPB/Campus Cajazeiras - UNINFO
Rua José Antônio da Silva, 300, Jardim Oásis.
Cajazeiras-Paraíba
e-mail: daladierjr@ifpb.edu.br

Ali Emrouznejad²

Surrey University - Surrey Business School
GU2 7XH
Guildford, Surrey, UK
e-mail: a.emrouznejad@surrey.ac.uk

ABSTRACT

Since the early days, cloud computing providers have charged their tenants based on the same premise used by PC manufacturers—that is, the greater the amount of CPU, RAM, and storage, the higher the final product price. However, in the virtualization world, this pricing model should be revised in favour of a methodology that allows tenants to observe the fractal patterns present in the configurations they select when contracting services. This would help them remain aware of the terms and conditions outlined in the software level agreements (SLAs) established between tenants and cloud service providers (CSPs).

This study demonstrates, through numerical results obtained from measurements on a type-I hypervisor, that tenants are incorrectly charged by CSPs, but also introduces as a contribution the multiplicative model of Data Envelopment Analysis (DEA), known as Super-Cobb-Douglas CRS-I. This model is proposed to predict, among the listed alternatives, which configuration exhibits the most suitable fractal pattern for service delivery at the transport layer of the TCP/IP stack over time.

KEYWORDS. KEYWORDS. Multiplicative Data Envelopment Analysis models. Fractal Analysis. Cloud Service Providers. Virtualization. Internet Service Prediction.

SE - DEA- Análise Envoltória de Dados

1. Introduction

Virtualization is a software abstraction that enables the execution of physical computers in software, using different architectures. These virtual computers, known as virtual machines, possess all virtual devices, such as CPU, memory, input and output devices, network cards, etc., similar to physical computers. Over time, various forms of enabling virtualization have been implemented, including type-I hypervisors, type-II hypervisors, para-virtualization, and, more recently, virtualization containers.

Cloud computing services utilise type-I hypervisors and virtualization containers, such as Docker and LXC. However, a virtual machine does not deliver the same performance as a physical machine due to bottlenecks in emulating hardware in software, as well as competition among tenants who share the finite resources of a cloud computing infrastructure. Therefore, when tenants are analysing and developing their systems, they should be aware of the performance of the entire set of tools used to create a suitable environment for delivering their services. However, this task is not trivial, as there is a myriad of different tools available to provide the same services on the Internet. Thus, all these alternatives must be compared to identify the optimal one for delivering a specific service.

This study is characterised by using measurements in virtual networks composed of two virtual machines, one acting as a client and the other as a server or sink node. The measurements were scaled, yielding results similar to the fractal patterns observed in our previous work. However, as different conclusions are drawn regarding how services are charged and evaluated in an Infrastructure as a Service (IaaS) model by cloud service providers, it is necessary to characterise the problem before presenting a new multiplicative formalism of Data Envelopment Analysis (DEA) and interpreting the results.

The results obtained here may have a significant impact on the accounting of on-demand services already paid for by cloud computing clients served by cloud service providers (CSPs) in recent years. This is because all CSPs follow the same pricing model as the PC/hardware industry, charging clients based on the quantity of certain configurations, such as CPU (version, clock, and type), amount of RAM, storage, and others. It should be noted that this study will compare only the three main CSPs in the world for illustrative purposes.

Amazon Web Services (AWS), considered the leading CSP globally, sets its prices based on the quantity of certain metrics related to an instance or virtual machine, as well as the chosen operating system. Each configuration has a distinct cost, primarily determined by the quantity of each configuration, the chosen operating system, and the continent where the service is delivered. See Figure 1 for a simplified pattern of accounting for computing or IaaS services by AWS.

According to the accounting policies of CSPs, if a virtual machine has a higher amount of vCPU and/or vRAM, the price of the computing service for that instance will be higher, similar to how the traditional PC industry sells its “physical” products.

Another relevant fact highlighted is that this accounting method does not account for discrepancies caused by self-similarity (SS) with long-range dependence (LRD) in TCP traffic during the delivery of services by these instances over time. In summary, both the impact of virtualization and SS with LRD must be evaluated for accounting and performance analysis in virtual networks within IaaS. Indeed, all CSPs follow the same tenant charging pattern presented by AWS.

Google Cloud¹, Microsoft Azure², and many other CSPs also charge their clients in the same manner shown in Figure 1. Thus, in all CSPs, each configuration has a distinct price, primarily determined by its configuration, metrics, operating system, and the continent where the service is provided.

Before discussing the impacts caused by the results, it is worth introducing the concept of service-level agreement (SLA) in cloud computing. According to Zhao et al. (2015), SLAs are declarative contracts between two parties—the contractor (tenant) and the contracted party (CSP)—that serve to establish parameters for the quality of services (quality-of-services (QoS)) that CSPs must provide to their clients.

If the agreed terms are breached, the CSP may face penalties, and vice versa. According to Ye et al. (2012), the bill paid by clients is closely tied to SLAs. Standard SLA metrics may include vCPU, vRAM, network bandwidth, system uptime, and depend on the service to be delivered. According to Ye et al. (2012), a third-party audit (TPA) should be employed to resolve the trust dilemma between both parties to the contract, as the CSP can easily conceal SLA details, providing fewer resources than agreed, thereby increasing their profits.

It is worth noting that in cloud computing environments, all services depend on the network to function properly. Thus, impacts on network performance in a cloud will reverberate across all cloud services, as it inherently carries AS with LRD, underscoring the importance of this research.

¹ Google Cloud – Prices for computing services can be found at the URL:
<https://cloud.google.com/compute/all-pricing?hl=pt-BR>

² Microsoft Azure Cloud – Prices for computing services can be found at the URL:
<https://azure.microsoft.com/en-us/pricing/details/virtual-machines/linux/>

Figure 1. Prices from Amazon Elastic Cloud Computing (EC2)

Instance name ▲	On-Demand hourly rate ▼	vCPU ▼	Memory ▼	Storage ▼	Network performance
t4g.nano	\$0.0042	2	0.5 GiB	EBS Only	Up to 5 Gigabit
t4g.micro	\$0.0084	2	1 GiB	EBS Only	Up to 5 Gigabit
t4g.small	\$0.0168	2	2 GiB	EBS Only	Up to 5 Gigabit
t4g.medium	\$0.0336	2	4 GiB	EBS Only	Up to 5 Gigabit
t4g.large	\$0.0672	2	8 GiB	EBS Only	Up to 5 Gigabit
t4g.xlarge	\$0.1344	4	16 GiB	EBS Only	Up to 5 Gigabit
t4g.2xlarge	\$0.2688	8	32 GiB	EBS Only	Up to 5 Gigabit
t3.nano	\$0.0052	2	0.5 GiB	EBS Only	Up to 5 Gigabit
t3.micro	\$0.0104	2	1 GiB	EBS Only	Up to 5 Gigabit
t3.small	\$0.0208	2	2 GiB	EBS Only	Up to 5 Gigabit

Source: Extracted from the Amazon Elastic Cloud Computing (EC2) pricing page³.

In summary, the contributions of this study are: a) evaluation of the fractal traffic of the transport layer (Layer 3 or L3) of TCP/IP in virtual networks using type-I hypervisors as client and server; b) formulation of a multiplicative DEA model called Super-Cobb-Douglas for predicting the best among the compared alternatives; and c) proof that CSPs are charging their clients incorrectly and should review their pricing policies for end users.

This paragraph outlines the sequence of this article. Next, we present the proposed Super-Cobb-Douglas model with CRS-I, detailing its operation and relevant formalisms. In section 3, we present the results and discussions. Finally, in section 4, we provide the summary and future work.

2. Super-Cobb-Douglas with CRS-I

Another name given to multiplicative DEA models is Cobb-Douglas models [COOK; ZHU, 2014]. For this reason, the model to be defined below is called Super-Cobb-Douglas with CRS, input-oriented. Multiplicative DEA models, introduced by Seiford et al. (1982), are so named because their initial objective function involves a product (which is transformed into a sum for computing the linear programming of decision variables in ratio form, as in equation (1).

The proposed formulation changes the piecewise linear efficiency frontier of CRS models to a distinct S-shaped frontier through a Cobb-Douglas strategy [COOK & ZHU, 2014]. Thus, the input-oriented Cobb-Douglas super-efficiency model with CRS, in its log-linear programming form, is presented in the equation as follows:

$$\min \phi_{j_0} \quad (1)$$

³ AWS Pricing Page – Available at URL: <https://aws.amazon.com/pt/ec2/pricing/on-demand/>

Subject to

$$\sum_{i=1 \neq 0}^n \lambda_j * \hat{x}_{i,j} + s^- - \phi * \hat{x}_{i,0} = 0, i = 1, \dots, m$$

$$\sum_{i=1 \neq 0}^n \lambda_j * \hat{y}_{r,j} - s^+ = \hat{y}_{r,0}, r = 1, \dots, s$$

$$\lambda_j \geq 0, j = 1, \dots, n$$

$$\phi_{j_0} \text{ free e } s_i^- \geq 0 \text{ \& } s_r^+ \geq 0$$

Consider a set of n DMUs under evaluation, such as DMU $_j$ (for each $j = 1, \dots, n$), associated with the Cobb–Douglas input vector $\hat{X}_j = (\hat{x}_{ij}, \dots, \hat{x}_{mj})$ and the Cobb–Douglas output vector $\hat{Y}_i = (\hat{y}_{ij}, \dots, \hat{y}_{sj})$. It should be noted that the input and output vectors have undergone a log-linear transformation, so as to conform to a multiplicative DEA model. Furthermore, the input slack and output surplus variables, denoted respectively as s^- and s^+ , were employed in formulation (1) to determine whether or not a DMU is fully efficient. To meet this condition, it is required that $efficiency_score \geq 1$, with zero in all slacks and surpluses (BANKER et al., 2004).

In DEA super-efficiency models [ANDERSEN; PETERSEN, 1993], the unit under evaluation, called DMU0, is removed from the set of observations during its evaluation. This deletion of the DMU under analysis increases its efficiency score to a value greater than one when the DMU is considered efficient by classical models. However, DMUs considered inefficient retain the same efficiency score in super-efficiency models.

The Super-Cobb-Douglas DEA CCR-I model was implemented in LINDO, yielding the same results and sensitivity analysis as other established DEA solvers. It should be noted that this model should be used when each DMU has only one time series to be evaluated. Furthermore, this model can be used to evaluate multi-criteria decision-making problems of DMUs with multiple input and output variables in ratio form, not restricted to the virtual network optimisation problem presented in this research.

3 Results and Discussion

A simple conclusion drawn from the problem characterisation presented earlier is that the greater the combined amount of vCPU and/or vRAM, the higher the price paid for an instance by tenants within an IaaS. Thus, according to the main accounting rules of these providers, the client will pay more for a VM when higher performance is required.

It should be highlighted that this study employed the same fractal traffic evaluation framework used by MARQUES JÚNIOR et al. (2019), including the same decision variables from that work. Regarding the decision variables listed in this research, they must be detailed for readers' understanding. The first is the fractal dimension, an index capable of comparing alternatives while

minimising the effect of outliers; in our case, we used the rodiogram method. The fractal dimension d varies between 1 and 2, where the closer d is to 1, the smoother the time series traffic; when $d=1.5$, the traffic is highly random or chaotic; and when d approaches 2, the traffic is irregular. Thus, d is an input variable. Another decision variable is the fractal memory or Hurst parameter (H), which, in our case, for future prediction, must be greater than 0.5, as this time series will have a positive autocorrelation function. Therefore, we must evaluate each time series per alternative to identify those with the highest H values, as this pattern indicates that the closer H is to 1, the longer the service pattern will persist. The other output variable, like H , is the TCP bandwidth.

However, this study mathematically demonstrates that the top three CSPs, and all other CSPs, mislead clients because the amount of vCPU and/or vRAM is not necessarily related to an increase in an instance's performance. On the contrary, it may be worse due to the effects of virtualization and, primarily, the effects of self-similarity (SS) with long-range dependence (LRD) over time per instance. SS with LRD is a contagion that spreads throughout the entire TCP/IP stack, as proven over the thirty years since its discovery by Leland et al. (1994). Thus, decision-makers or service providers must evaluate all alternatives for delivering any network service to select the one that offers the best fractal pattern among the compared alternatives.

Therefore, CSPs should provide their clients with a fractal analysis per configuration or instance, showing metrics related to fractal dimension, average TCP bandwidth (or another transport protocol), and fractal memory to comply with the contracts between tenants and CSPs, known as SLAs.

Consequently, consumers could be charged more fairly. This is because virtual network traffic services are mandatory for delivering all other types of cloud services. Therefore, according to the results previously presented in this chapter, if a tenant selects an instance with a poor fractal pattern to carry its TCP traffic, this stochastic pattern would act as a contagion across the entire virtual environment, degrading the performance of services offered by clients and the CSP itself, as this effect can reverberate within an IaaS, affecting other tenants like a domino effect.

Table 1 provides proof of concept (PoC) of how CSPs should evaluate performance to ensure SLAs, thereby increasing trust between parties. The public Mendeley dataset displays 288 distinct network hypervisor configurations, obtained through measurements to capture the fractal behaviour of TCP traffic over time. To comply with the event's page limit, Table 1 presents only a summary of the evaluated alternatives, limited to the first ten decision-making units (DMUs). However, all results can be found in our public Mendeley dataset, related to the Super-Cobb-Douglas CCR model with input orientation (CCR-I).

Table 1. Fractal performance analysis by configuration/DMU in a type-I hypervisor for the first ten DMUs.

#	DMU	(INPUT) Fractal Dimension	(OUTPUT) TCP Throughput	(OUTP UT) Hurst
1	Fedora24 – ESXI 6.5 – GG – E1000 – 1vCPU – 1GB VRAM	2,000000	599853,8	0,8622781
2	Fedora24 – ESXI 6.5 – GG – E1000 – 1vCPU – 2GB VRAM	2,000000	552623,8	0,8444985
3	Fedora24 – ESXI 6.5 – GG – E1000 – 1vCPU – 4GB VRAM	1,622085	378719,2	0,8634234
4	Fedora24 – ESXI 6.5 – GG – E1000 – 1vCPU – 8GB VRAM	1,756555	443682,1	0,7890162
5	Fedora24 – ESXI 6.5 – GG – VMXNET3 – 1vCPU – 1GB VRAM	1,365223	452180,4	0,7807921
6	Fedora24 – ESXI 6.5 – GG – VMXNET3 – 1vCPU – 2GB VRAM	1,675527	508714,7	0,7482798
7	Fedora24 – ESXI 6.5 – GG – VMXNET3 – 1vCPU – 4GB VRAM	1,409517	569266,3	0,7976480
8	Fedora24 – ESXI 6.5 – GG – VMXNET3 – 1vCPU – 8GB VRAM	1,438168	558170,6	0,7777546
9	Fedora24 – ESXI 6.5 – GGC – E1000 – Docker – 1vCPU – 1GB VRAM	1,840123	628785,1	0,8709777
10	Fedora24 – ESXI 6.5 – GGC – E1000 – Docker – 1vCPU – 2GB VRAM	1,697673	600281,3	0,8508286

Source: Dataset with all results available at: <http://dx.doi.org/10.17632/fx4rhxrdm.2>

This study reveals another significant discovery related to the impact of AS with LRD in virtual networks, which may alter how fractal behaviour is evaluated in a scaled manner for each configuration used as a network hypervisor for delivering TCP-based services.

It is noteworthy that the performance difference of the configuration with the highest average TCP bandwidth – DMU215 – is 294.630% greater than the configuration with the lowest TCP bandwidth – DMU75. It should be emphasised that this fractal pattern, with distinct values for all decision variables, is maintained across all evaluated virtual configurations.

These performance differences in TCP-related decision variables for each network hypervisor may raise questions about errors in client accounting by CSPs, where cloud tenants were likely charged incorrectly, certainly at higher rates, for the services contracted with CSPs.

Indeed, even if an instance is considered high-performing by a CSP, it may exhibit poorer fractal TCP traffic behaviour over time. This is because an instance's performance must comprise the triad of fractal dimension, average TCP bandwidth, and fractal memory over a period.

Therefore, an instance's performance is not solely related to the combination of certain metrics. For example, consider DMU75 (Fedora24 - ESXi 6.5 - GG - E1000 – 8 vCPUs – 4 GB

VRAM) and DMU215 (Ubuntu16 - ESXi 6.5 - GGC - VMXNET3 - LXC – 1 vCPU – 4 GB VRAM). According to CSP accounting policies, DMU75 should be priced higher than DMU215 because DMU75 has 8 vCPUs compared to 1 vCPU in DMU215, while both configurations have 4 GB of vRAM. Thus, all CSPs should adopt this fractal-based performance evaluation approach.

This study's proposal aims to promote a fairer tax policy, suggesting a contract or SLA amendment that is easy to audit and evaluate, as the information obtained from TCP traffic time series in cloud infrastructures indicates that network behaviour could be summarised in the three aforementioned fractal indices. However, other fractal variables, such as the alpha tail shape, Lyapunov index, etc., could also be used in decision-making.

The most critical factor to consider is the fractal dimension, where smoother or more stable TCP traffic is preferable to more irregular traffic. From a mathematical perspective, the most suitable TCP traffic behaviour over time is one where the fractal dimension indicates smoother TCP traffic, with superior TCP bandwidth performance and fractal memory, to provide optimised network services over time. Therefore, an input-oriented Cobb-Douglas DEA model was created to minimise the fractal dimension while maximising the average TCP bandwidth and fractal memory, as this DEA model is also non-radial.

However, the configuration with the highest average TCP in the mentioned dataset (DMU23) is not suitable for delivering transport services in virtual networks, as it has highly irregular TCP bandwidth related to its fractal dimension value. Thus, a multicriteria decision-making (MCDM) technique, such as DEA [BANKER et al., 1984; BANKER; MAINDIRATTA, 1986; CHARNES et al., 1978; CHARNES et al., 1984; EMROUZNEJAD; WITTE, 2010; EMROUZNEJAD; YANG, 2017], or a hybrid combination of key techniques, such as mixing DEA with Fuzzy logic, Analytical Hierarchy Process (AHP), among other strategies [WHAIDUZZAMAN et al., 2014], is necessary. The purpose of employing these various evaluation techniques is to provide alternatives to solve the complex multicriteria optimisation problem in virtual networks.

After running the input-oriented Super-Cobb-Douglas DEA model, the only two super-efficient configurations with zero slack are DMU114 and DMU165, with efficiency scores of 1.006526 and 1.004616, respectively, making them ideal for delivering TCP services in virtual networks over time. Due to space constraints, these results and the ranking of this model will only present the top two efficient DMUs.

However, if there is interest in viewing the complete results, the public Mendeley dataset should be accessed, containing a comprehensive comparison between the standard super-CCR-I model and the Super-Cobb-Douglas DEA CCR-I. The dataset includes efficiency frontiers, PPS

graphs, and the entire logarithmic variable transformation process mandatory in Cobb-Douglas DEA models.

It is worth noting that DMU114 has only 1 vCPU and 2 GB of vRAM, and DMU165 has 4 vCPUs and 1 GB of vRAM, proving that an instance's performance is not related to increasing the quantity of certain metrics related to virtual devices like vCPU, vRAM, and storage. Therefore, it is necessary to conduct a fractal analysis of as many network hypervisors as possible to accurately evaluate the fractal TCP behaviour in DMUs over time for more precise predictions.

Figures 2, 3, and 4 compare the PPS of the Super-Cobb-Douglas CCR-I versus the Super-CCR-I models. Thus, Figure 2 shows how the logarithmic transformation of variables achieves fair normalisation of input and output variables to attain proportionality, convexity, scale elasticity, and many other advantages of applying an S-shaped efficiency frontier.

Figure 2. Super-Cobb-Douglas DEA CCR-I

Figure 3. Super-CCR-I DEA

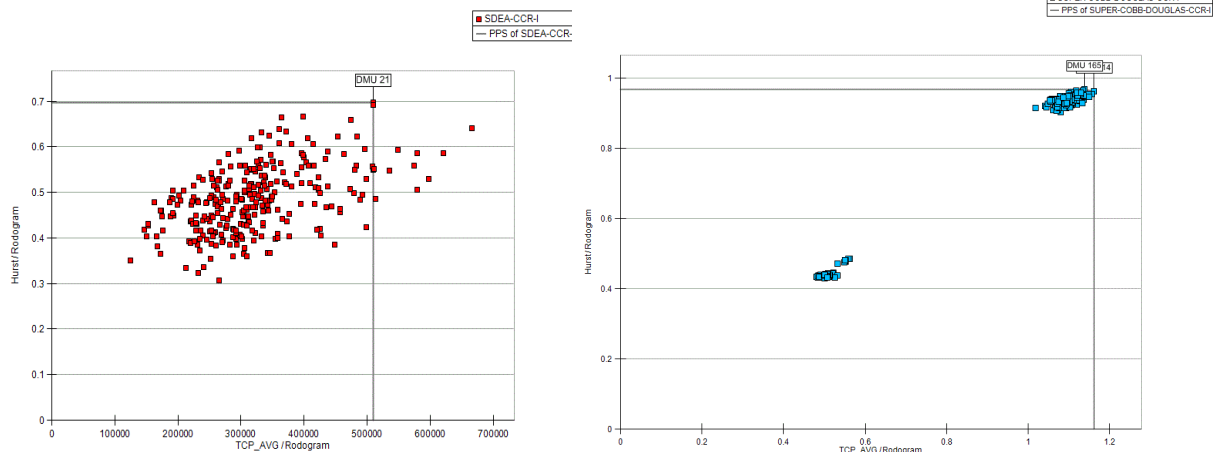
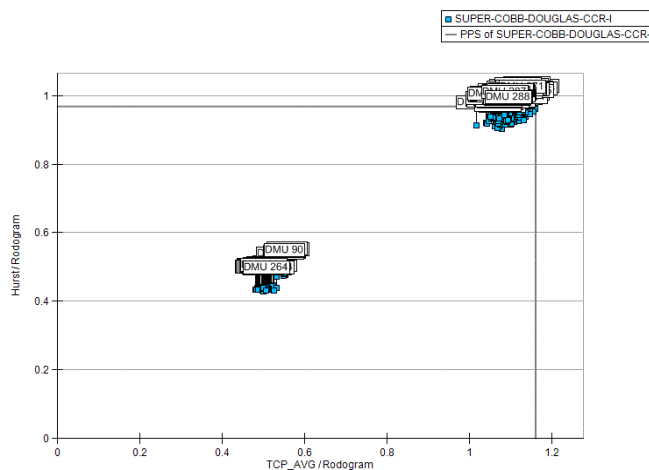


Figure 4. Super-Cobb-Douglas DEA CCR-I highlighting DMUs (with labels).



Instead, Figure 2 shows a dispersion of DMUs within the PPS because classical DEA models are not equipped to correctly evaluate DMUs operating above or below their capacities, making it mandatory to change the standard efficiency frontier to an S-shaped one. This PPS comparison is also available in the aforementioned online dataset.

Figure 3 demonstrates the increased discriminative power of this Super-Cobb-Douglas CCR-I DEA formulation, which places DMUs with poor fractal TCP behaviour—those with high irregularity, low average TCP bandwidth, lower fractal memory, or all simultaneously—far from the efficiency frontier. Indeed, the Super-Cobb-Douglas CCR-I model accommodates DMUs considered the most suitable near or within the efficiency frontier. Consider the labels of inefficient DMUs located in the middle of Figure 3, respectively, DMU90 and DMU264, where DMU90 has a rodogram value of 1.51728, an average TCP bandwidth of 403013.1 kbps, and a Hurst parameter of 0.8026025. The values for DMU264 are 1.342008, 448349.2 kbps, and 0.6324591, respectively. This same poor fractal behaviour is followed by DMUs numbered 24, 31, 52, 54, 201, etc.

Table 2 displays the results of the top two DMUs selected as super-efficient by the proposed Super-Cobb-Douglas CSR-I model. It should be noted that the most efficient DMU has only 1 vCPU and 2 GB of vRAM, while the second has 4 vCPUs and 1 GB of vRAM, demonstrating that having a high quantity of vCPU and vRAM does not guarantee better performance, as network traffic has fractal characteristics with chaotic memory over time. However, decision-makers with this information can make informed decisions to form an optimal set of tools for delivering an Internet service.

Table 2. Top 2 DMUs selected as super-efficient by the Super-Cobb-Douglas CRS-I model

Rank	DMU	Score
1	Ubuntu14 – ESXI 6.5 – GGC – VMXNET3 – Docker – 1vCPU – 2GB VRAM	100,652%
2	Ubuntu14 – ESXI 6.5 – GGC – VMXNET3 – LXC – 4vCPU – 1GB VRAM	100,470%

Finally, as there are many unicorn tenants (with a market value exceeding US\$1 billion) consuming services provided by cloud service providers, such as Uber, Dropbox, Airbnb, Booking, Spotify, and others, they could pressure CSPs to change how they charge clients to increase trust between parties.

It is important to highlight that Table 1 shows only the impact of virtualization and AS with LRD for a subset of evaluated VMs. Therefore, it is highly likely that performance decreases significantly more in the presence of multiple tenants simultaneously consuming all these infrastructure and network services offered by each VM or a set of VMs in virtual network environments.

4. Conclusion and future work

This study presented the Super-Cobb-Douglas CSR-I model as a multiplicative DEA model capable of acting as a mathematical hypervisor, selecting/predicting the most efficient DMU for delivering L3 TCP layer services. The results were satisfactory from the decision-maker's perspective. The methodology used was derived from the fractal evaluation framework proposed by Marques Júnior et al. (2019).

Another conclusion drawn from the results was that increasing the number of vCPUs, vRAM, or other virtual configurations in a virtual machine does not guarantee improved performance. Thus, decision-makers must evaluate all options and predict which is optimal for delivering a service.

Future work includes investigating the multifractal pattern in network traffic and service orchestration, as well as developing dynamic multiplicative models, with appropriate comparison of results.

References

- Andersen, P., & Petersen, N. C. (1993). A Procedure for Ranking Efficient Units in Data Envelopment Analysis. *Management Science*, 39(10), 1261–1264. <https://doi.org/10.1287/mnsc.39.10.1261>
- Banker, R. D., Charnes, A., & Cooper, W. W. (1984). Some models for estimating technical and scale inefficiencies in data envelopment analysis. *Management Science*, 30(9), 1078–1092. <https://doi.org/10.1287/mnsc.30.9.1078>
- Banker, R. D., & Maindiratta, A. (1986). Piecewise Loglinear Estimation of Efficient Production Surfaces. *Manage. Sci.*, 32(1), 126–135. <https://doi.org/10.1287/mnsc.32.1.126>
- Banker, R. D., Cooper, W. W., Seiford, L. M., Thrall, R. M., & Zhu, J. (2004). Returns to scale in different DEA models. *European Journal of Operational Research*, 154(2), 345–362. [https://doi.org/10.1016/S0377-2217\(03\)00174-7](https://doi.org/10.1016/S0377-2217(03)00174-7)
- Charnes, A., Cooper, W. W., & Rhodes, E. (1978). Measuring the efficiency of decision making units. *European Journal of Operational Research*, 2(6), 429–444. [https://doi.org/10.1016/0377-2217\(78\)90138-8](https://doi.org/10.1016/0377-2217(78)90138-8)



- Charnes, A., Clark, C. T., Cooper, W. W., & Golany, B. (1984). A developmental study of data envelopment analysis in measuring the efficiency of maintenance units in the U.S. air forces. *Annals of Operations Research*, 2(1), 95–112. <https://doi.org/10.1007/BF01874734>
- Cook, W. D., & Zhu, J. (2014). DEA Cobb–Douglas frontier and cross-efficiency. *Journal of the Operational Research Society*, 65(2), 265–268. <https://doi.org/10.1057/jors.2013.13>
- Leland, W. E., Taqqu, M. S., & Wilson, D. V. (1994). On the Self-Similar Nature of Ethernet Traffic (Extended Version). *IEEE/ACM Transactions on Networking*, 2(1), 1–15. <https://doi.org/10.1109/90.282603>
- Emrouznejad, A., & Witte, K. De. (2010). COOPER-framework: A unified process for non-parametric projects. *European Journal of Operational Research*, 207(3), 1573–1586. <https://doi.org/https://doi.org/10.1016/j.ejor.2010.07.025>
- Emrouznejad, A., & Yang, G. liang. (2017). A survey and analysis of the first 40 years of scholarly literature in DEA: 1978-2016. *Socio-Economic Planning Sciences*. <https://doi.org/10.1016/j.seps.2017.01.008>
- Marques Júnior, F. D., Emrouznejad, A., Dias, K. L., Cunha, P. R. F., & de Castro e Silva, J. L. (2019). Optimising virtual networks over time by using Windows Multiplicative DEA model. *Expert Systems with Applications*. <https://doi.org/10.1016/j.eswa.2019.05.005>
- Seiford, L., Charnes, A., W. Cooper, W., & A. Stutz, J. (1982). A Multiplicative Model for Efficiency Analysis. *Socio-Economic Planning Sciences*, 16, 223–224.
- Ye, L., Zhang, H., Shi, J., & Du, X. (2012). Verifying cloud Service Level Agreement. In 2012 IEEE Global Communications Conference (GLOBECOM) (pp. 777–782). <https://doi.org/10.1109/GLOCOM.2012.6503207>
- Whaiduzzaman, M., Gani, A., Anuar, N. B., Shiraz, M., Haque, M. N., & Haque, I. T. (2014). Cloud service selection using multicriteria decision analysis. *The Scientific World Journal*. <https://doi.org/10.1155/2014/459375>
- Zhao, X., Shen, L., Peng, X., & Zhao, W. (2015). Toward SLA-constrained service composition: An approach based on a fuzzy linguistic preference model and an evolutionary algorithm. *Information Sciences*, 316, 370–396. <https://doi.org/10.1016/J.INS.2014.11.016>