Identification of Resource Utilisation Patterns in Data Centers using Tensor Decomposition

Waltenegus Dargie

Faculty of Computer Science, Technical University of Dresden, 01062 Dresden, Germany
waltenegus.dargie@tu-dresden.de

***Abstract*—The worldwide workload of the public cloud infrastructure has been increasing steadily for the past many years and the latest statistics indicate that it will remain so for the coming years. At the same time, the energy consumption of the cloud infrastructure is considerably high. In this respect, the efficient utilisation of computing resources is of profound importance. In contemporary data centres tens of thousands of virtual machines execute simultaneously. Considering the number and heterogeneity of the virtual machines, balancing the demand for and the supply of resources is one of the challenges facing Cloud and Edge Computing. Often, infrastructure providers over-provision resources to ensure that service level agreements are respected. This, however, is not sustainable and its long-term impact on the environment cannot be overlooked. In this paper, we propose the use of tensor decomposition to analyse the resource utilisation metrics of a large number of hosted virtual machines and to identify complementary and contentious features which can be vital for efficient resource utilisation.**

***Index Terms*—Cloud computing, consolidation, energy-efficient computing, resource utilization, tensor decomposition**

I. Introduction

Cloud computing and Edge Computing provide complementary advantages to distributed data processing, the former providing high reliability whilst the latter great flexibility. At present, Cloud Computing is playing an indispensable role in computing, even though its reliability often comes at the expense of a disproportionate amount of energy consumption and an over-provisioning of computing resources [1]. The scope and usefulness of Edge Computing is still being studied, there are, nevertheless, strong indications that it will play a critical role in the years to come. One of the areas where it will serve as a backbone is in self-driving cars and vehicular communication.

Both paradigms are built upon the virtualization concept which enables a large number of virtual machines (containers) to execute on the same platform without compromising on their autonomy and security. Hence, in contemporary data centres tens of thousands of virtual machines execute simultaneously. Considering the number and heterogeneity of the virtual machines, balancing the demand for and the supply of resources is one of the challenges facing modern data centres [2]. Often, infrastructure providers over-provision resources to ensure that service level agreements are respected [3]. As long as they are making profits, the disproportionate amount of energy the physical servers consume may be of secondary concern to infrastructure providers, however, in light of the unprecedented scale at which the workload of the Cloud infrastructure grows worldwide, this practice is not sustainable and its drastic and long-term impact on the environment cannot be overlooked.

Offline and online solutions have been proposed to efficiently utilise virtualized resources. A substantial body of these focuses on the seamless consolidation (aggregation) of virtual machines on a few number of physical machines. The underlying strategies are workload characterisation and prediction, identifying virtual machines having complementary characteristics, and the live migration of virtual machines [4]. Most of the approaches rely on multi-objective optimisations which take into account, among other things, the number of resources to be utilised (CPU, memory capacity, network bandwidth, storage capacity, storage bandwidth, multi-level cache, memory bandwidth) and various execution costs (cooling, performance, energy, and migration costs). For data centres hosting a large number of virtual machines, multi-objective optimisations are NP-hard [5].

In this paper, we aim to characterise virtual machines according to their resource utilisation characteristics using multi-way tensor decomposition [6], [7]. This approach enables to achieve two opposing objectives at the same time. Firstly, it enables to efficiently process a large amount of statistical data pertaining to the resource utilisation of a large number of virtual machines. Secondly, it enables to uncover hidden features which can be vital to identify virtual machines having complementary as well as contentious resource utilisation characteristics. Our analysis is based on measurement traces of 44 active virtual machines which are currently running on a medium-scale data centre consisting of 59 physical computing servers and 29 storage servers organised into 9 clusters.

Originally arising in the fields of psychometrics and chemometrics, tensor decomposition and analysis has induced a great deal of interest to model and reason about complex relationships in a wide range of research areas, including neurology [8], digital signal processing [9], knowledge retrieval and data mining [10], and many others. Their popularity lies in their capacity to:

- structure a large amount of data in a comprehensible way;
- exploit multi-dimensional correlations in order to significantly reduce the dimensions of the original data; and,
- extract latent features which can be examined from different vantage points.

The remaining part of the paper is organized as follows: In Section II, we explain how we obtained the measurement sets we used for our analysis. In Section III, we introduce dimensionality reduction techniques, highlighting tensor decomposition and its relevance to characterise resource utilisation. In Section IV, we demonstrate how we apply tensor decomposition to analyse the resource utilisation characteristics of 44 active virtual machines and identify "spatial" and temporal characteristics. Finally, in Section V, we provide concluding remarks and outline future work.

II. Background

For the analysis we present in this paper, we rely on statistics obtained from the Enterprise Cloud Infrastructure at the Centre for Information Services and High-Performance Computing (ZIH)[1] at the TU Dresden. The data centre consists of 59 physical computing servers organised into 9 clusters and 29 physical storage servers. Altogether, it has 120 multi-core processors with an equivalent total capacity of 1029 GHz CPU cycles, 1606 MB RAM and 65.7 TB disk space. In 2018, its approximate average annual resource utilisation is: 40 % CPU, 55 % MEM and 91 % disk. Currently, it hosts 1190 commercial virtual machines.

Fig. 2 displays the utilisation of three resources (CPU, memory, and network bandwidth) by the 44 most active virtual machines. If we consider the volume formed by the three axes as the overall capacity of the data centre (its actual capacity much exceeds this and we are considering only three types of resources in order to visualise the problem), it is easy to notice that the data centre is underutilised. By contrast, the idle power consumption of each server accounts for more than 60 % of its full-load (peak) power consumption. Secondly, if we examine the utilisation distributions along the three axes, we can easily observe that the resources are not utilised with comparable efficiency.

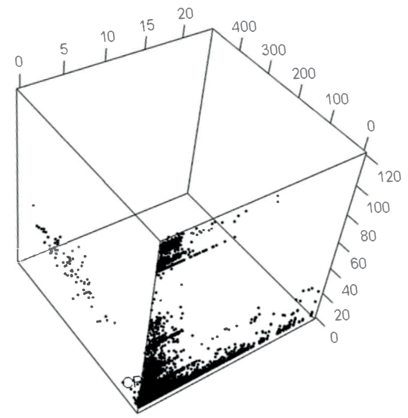


Fig. 1: A three-dimensional view of the resource utilisation of the 44 virtual machines. The dimensions refer to the utilisation of memory (in MB), the utilisation of network bandwidth (in Kbps), and the utilisation of CPU (in percent).

In light of this observation, it is important to raise and address the following research questions:

1) Given a large set of hosted virtual machines (containers) having stochastic workloads and a corresponding set of computing resource demands (CPU, memory bandwidth, memory capacity, network bandwidth, storage size, disk read/write bandwidth, etc.), is it possible to identify virtual machines having complementary as well as contentious resource utilisation characteristics?
2) Considering the large amount of hosted virtual machines and the large amount of statistical samples required pertaining to resource utilisation, is it possible to develop analytic strategies which (1) are efficient to compute, (2) yield tractable solutions, and (3) are intuitive to identify virtual machines exhibiting higher-level features?
3) Using the same sets of analytic tools and sets of data, is it possible to simultaneously uncover hidden, non-overlapping features and predict the temporal evolution of the resource demand of hosted virtual machines? This aspect will be useful to perform dynamic virtual machine consolidation based on anticipated resource demands, two complex tasks which are often carried out in two separate stages.

III. Dimensionality Reduction

Fig. 2 summarises what we aim to achieve. Given n hosted virtual machines and statistics pertaining to their resource demand, our strategy is intended to uncover hidden characteristics which can be useful for categorising the virtual machines into m clusters. Thereafter, a consolidation algorithm will select virtual machines from each cluster to consolidate them in one and the same physical server, so that:

(a) the virtual machines utilise different resources at any given time and,
(b) all the available resources of the physical server are utilised with comparable efficiency.

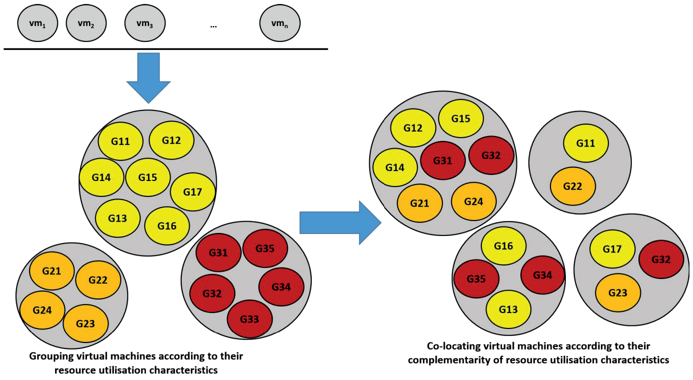

vm_1
vm_2
vm_3
...
vm_n
G11
G12
G14
G15
G17
G13
G16
G21
G22
G24
G23
G31
G35
G32
G34
G33
Grouping virtual machines according to their resource utilisation characteristics
G12
G15
G14
G31
G32
G21
G24
G11
G22
G16
G35
G34
G13
G17
G32
G23
Co-locating virtual machines according to their complementarity of resource utilisation characteristics

Fig. 2: Identification of hidden features in the resource utilisation statistics for consolidating virtual machines.

[1] https://tu-dresden.de/zih (Last accessed on 14 May 2019, 15:15 CET).

In case (b) cannot be achieved completely, the algorithm should attempt to first consolidate those virtual machines utilising resources which are not amenable to dynamic power management, because the others (for example, CPU sockets and individual CPU cores) can be switched off completely or set to low power modes. The step from clustering to consolidation will be made based on anticipated resource utilisation.

A. Singular Value Decomposition

The metrics describing the resource utilisation history of hosted virtual machines can be analysed to determine the existence of latent (hidden) features which can be vital for identifying virtual machines showing contentious and complementary features. The hidden features may relate to temporal as well as "spatial" aspects. Temporal aspects reveal how resources are utilised in time whereas "spatial" aspects reveal which resources are utilised at any given time. Knowledge of these two aspects enables to decide which virtual machines should be placed together.

Suppose we have an n by m matrix $\mathbf{X}$ containing statistics pertaining to the CPU utilisation of all hosted virtual machines in a data centre (n refers to the hosted virtual machines and m to the number of samples). Since we have only a single dimension (time) to consider, the hidden features we are seeking to uncover can be resolved along this dimension only. Decomposing this matrix using the Singular Value Decomposition (SVD) yields:

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^{\intercal} \tag{1}$$

$\mathbf{U}$ and $\mathbf{V}$ are orthogonal (uncorrelated) and orthonormal matrices and Σ is a diagonal matrix having entries which are naturally arranged according to their magnitude (i.e., $\sigma_{11} \geq \sigma_{22} \geq ...\sigma_{33}$ and so on). The matrix σ reveals how many unique features are hidden or embedded in the utilisation matrix. The matrix $\mathbf{U}$ encodes the relationship of the virtual machines with the hidden features whereas the matrix $\mathbf{V}$ encodes the relationship of the hidden features with the samples (temporal features, for our case). Some of the advantages of using SVD for analysing the utilisation matrix are the following:

1) Firstly, one does not need to make any assumption as regards the hidden features. Their number and significance is dynamically revealed by the diagonal matrix (refer to Fig. 3).
2) Secondly, one can express the original utilisation matrix $\mathbf{X}$ as the summation of many matrices as follows (refer also to Fig. 4):

$$\mathbf{X} = \sum_{r=1}^{R} \sigma_{rr}\mathbf{u}_r \circ \mathbf{v}_r \tag{2}$$

where $\circ$ indicates the vector (outer) product and $\mathbf{u}_r$ and $\mathbf{v}_r$ refer to the r-th column of the matrices $\mathbf{U}$ and $\mathbf{V}$, respectively. We refer each matrix on the right side as a component. Note that the relevance of each component is associated with the relevance of σ_{ii}.
3) Thirdly, if the samples of the utilisation matrix exhibit strong correlations, then, $\mathbf{X}$ can be approximated by taking the first K components only:

$$\mathbf{X} \approx \sum_{r=1}^{K} \sigma_{rr}\mathbf{u}_r \circ \mathbf{v}_r \tag{3}$$

for $K < R$. If the difference in magnitude between the successive σ_{rr} entries is significantly large, then a strong correlation is identified in the original utilisation matrix and, hence, the error resulting from our approximation will be significantly small.

B. Utilisation Tensor

One of the limitations of working with SVD is that it is two dimensional. In other words, we can attempt to uncover hidden features along one dimension only. This forces us to analyse the utilisation of a single resource at a time. If we wish to analyse the utilisation of multiple resources using SVD, we have to analyse the average utilisation. But the average utilisation disregards the temporal variation of resource utilisation and leads to a considerable resource overload or underutilisation should the virtual machines be consolidated without the knowledge of this aspect.

The most plausible alternative is to model resource utilisation using a three-way tensor, as shown in Fig. 5. As can be seen, the tensor is a three-dimensional array consisting of elements intersecting three orthogonal axes. Hence, in the same way every element of a matrix can be referred to by two indices (the row index i and the column index j), every element of the tensor can be referred to by three indices. So, for example, x_{ijk} refers to the utilisation of the j-th element by the i-th VM in the k-th time slot. Now we have two dimensions along which we can search for hidden features (and, hence, two degrees-of-freedom to cluster the virtual machines), namely, the resource and the time dimensions.

Similarly, in the same way a matrix can be decomposed (factorised) into basic constituting elements, a tensor can be decomposed into basic constituting elements. However, unlike decomposing a matrix, decomposing a tensor is not straightforward. To start with, an assumption has to be made about the number of the hidden factors, whereas this is done

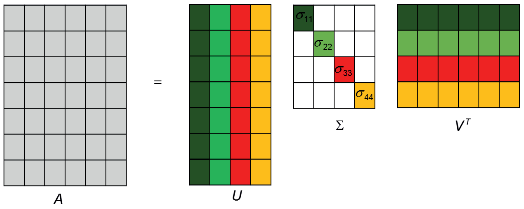


Fig. 3: Decomposing a resource utilisation matrix (VMs vs. sample statistics) using the Singular Value Decomposition (SVD).

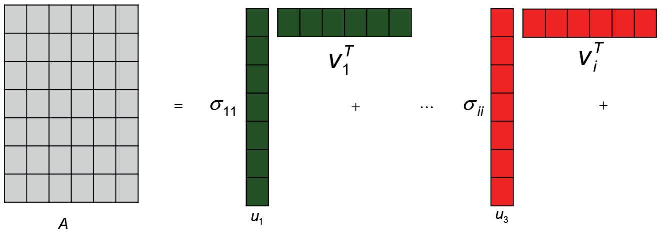


Fig. 4: Expressing the utilisation matrix as the summation of SVD components.

automatically with SVD. Secondly, a tensor has to be unfolded (or flattened) into a matrix before it can be decomposed, which influences the outcome of the decomposition.

A closer look into the utilisation tensor reveals that it provides three orthogonal views which can serve different purposes. For example, the front view (borrowing an expression from architecture) provides a matrix describing the utilisation of all resources by all hosted virtual machines at the k-th sampling interval – i.e, (VM versus time)$_k$. This view is called the front slice. Likewise, the top view provides a matrix describing the utilisation of all resources by the i-th virtual machine over a period of time – i.e., (resources vs. time)$_i$. This is called the horizontal slice. Finally, the side view provides a matrix describing the utilisation of the j-th resource by all virtual machines over a period of time – i.e., (VM vs. time)$_j$. This is called the lateral slice. It is this flexibility, among others, which makes a tensor desirable.

C. Tensor Decomposition

The chief task of a tensor decomposition is to identify multidimensional features in terms of which the virtual machines can be categorized. Compared to the size of the tensor, the basic features should be significantly small in size, so that the clustering process is computationally tractable. A tensor analysis begins by unfolding (flattening) the tensor into a matrix. The unfolding can take place in different ways, but whichever way is chosen, the entries along each dimension form a column vector. Afterwards, the unfolded matrix can be decomposed as if it were a normal matrix.

There are different tensor decomposition strategies, but we use the canonical decomposition/parameter factorisation (referred in the literature as CANDECOM/PARAFAC, or, in short, CP) [7] which decomposes a tensor into three matrices:

$$\mathcal{X} = \mathbf{ABC} \tag{4}$$

or

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \tag{5}$$

where $\mathbf{a}_r$, $\mathbf{b}_r$, and $\mathbf{c}_r$, are the r-th columns of the matrices $\mathbf{A}$ $\mathbf{B}$, and $\mathbf{C}$, respectively. In the existence of a strong correlation in the utilised resources, the utilisation tensor can be approximated only by the outer product of the first K column vectors of the matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, respectively (refer to Fig. **??**).

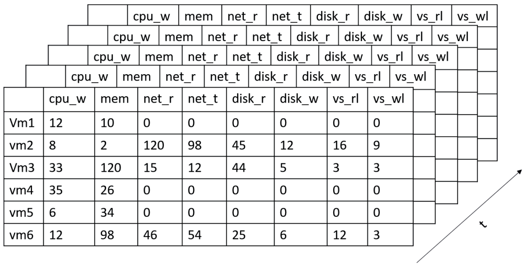
The three basic matrices have the following significance: The matrix $\mathbf{A}$ characterises the hosted virtual machines in terms of the unique features. The matrix $\mathbf{B}$ associates the unique features with the resources utilised and the matrix $\mathbf{C}$ reveals the temporal characteristics of the virtual machines without explicitly referring to the particular resources they utilise. For the consolidation task, the most relevant matrices are $\mathbf{A}$ and $\mathbf{C}$, because the former reveals the "spatial" characteristics whereas the latter reveals the "temporal" characteristics of the virtual machines.

IV. Evaluation

The VMware managing the data centre provides a large number of key performance indicators (KPI) to monitor resource utilisation. We selected 13 metrics (listed in Table I) to build our utilisation tensor. The value of these metrics is updated every 5 minutes and stored in a database. Thus, for the 44 most active VMs, the 24-hour resource utilisation results in a tensor having a dimension of $44 \times 13 \times 1440$.

Amongst the 13 distinct resources listed in Table I, a given VM can use either all or a subset of these in a specific time period. So, we can attempt to characterise the VM according to which of these resources it predominantly utilises. However, some of the resources cannot be utilised in isolation. For example, the utilisation of a memory bandwidth inevitably involves the CPU and the memory. Similarly, the utilisation of a network bandwidth involves the utilisation of the memory and, potentially, a virtual storage. It is this dependency the tensor decomposition exploits in order to uncover distinct utilisation features.

As we mentioned in the previous section, a tensor decomposition requires the estimation of the unique underlying



	cpu_w	mem	net_r	net_t	disk_r	disk_w	vs_rl	vs_wl
Vm1	12	10	0	0	0	0	0	0
vm2	8	2	120	98	45	12	16	9
Vm3	33	120	15	12	44	5	3	3
vm4	35	26	0	0	0	0	0	0
vm5	6	34	0	0	0	0	0	0
vm6	12	98	46	54	25	6	12	3

Fig. 5: A three-way tensor representing the resource utilisation statistics of hosted virtual machines.

TABLE I: A summary of the utilisation metrics used to construct the utilisation tensor.

Metric	Metric
Average CPU usage (in MHz)	
MEM usage	NET received average
NET transmit average	NET broadcast TX summation
Datastore read average	Datastore write average
Disk read average	Disk write average
Storage total read latency	Storage total write latency
Virtual disk read average	Virtual disk write average

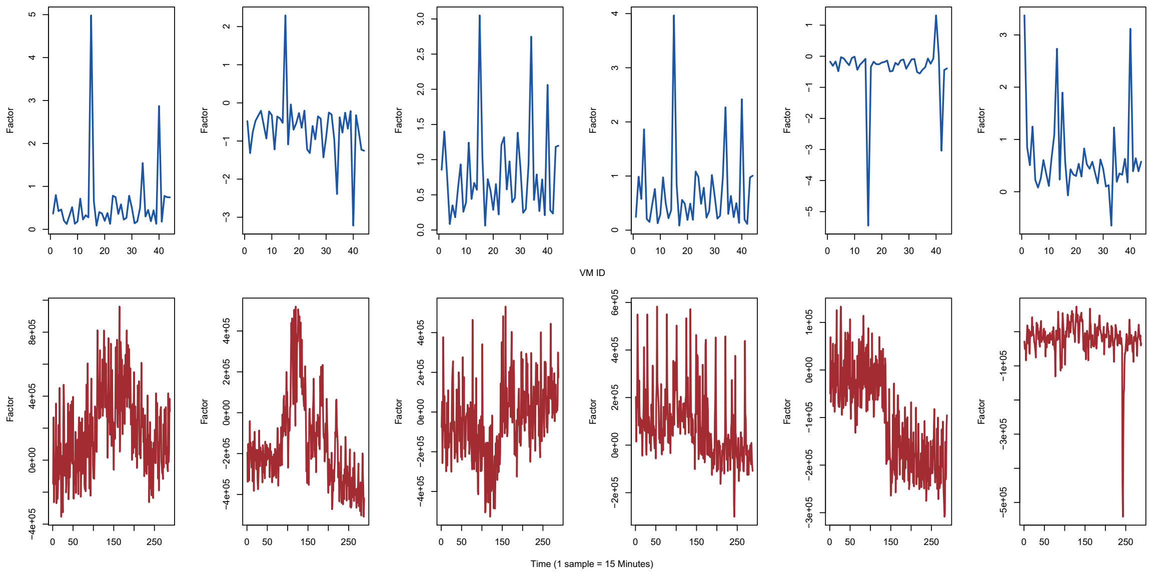


Fig. 6: The relationship between the six factors describing the resource utilisation statistics. Top: VMs vs Factors. Bottom: Samples vs Factors.

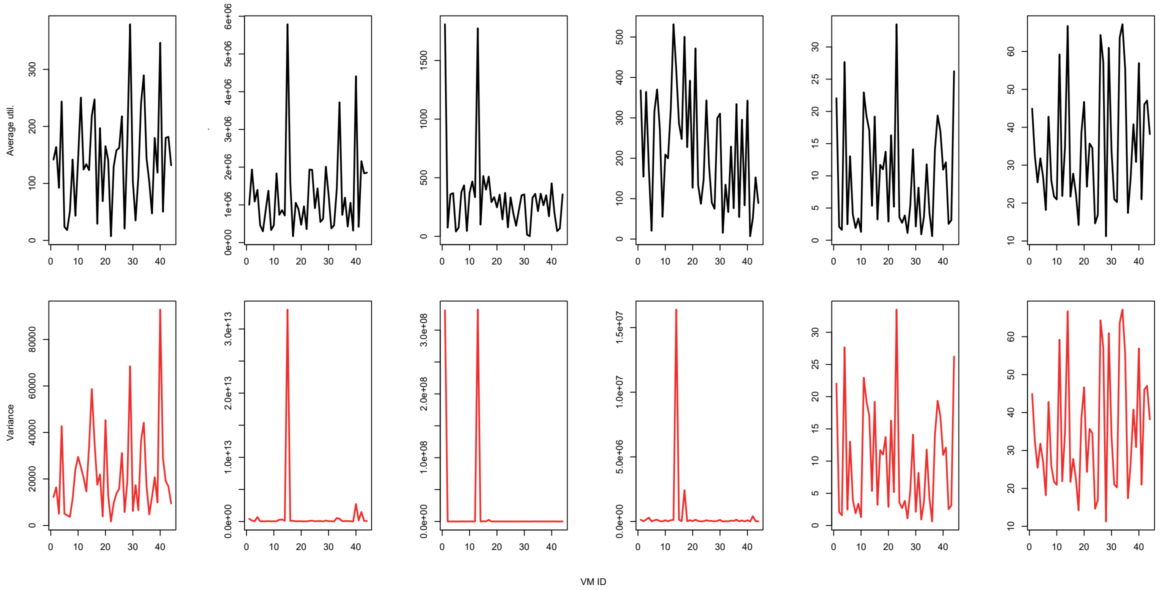


Fig. 7: The average values and variances of five key performance indicators: CPU utilisation (in %), memory utilisation (in KB), network (received, in kbps), network (transferred, in kbps), storage total read latency (in ms), storage total write latency (in ms) of the 44 hosted VMs (along the x-axis).

features (factors) before a decomposition takes place. In order to ensure that we uncover all the hidden features relevant to characterise the VMs, we examined the coefficient of determination (R^2) for different number of factors[2]. The CANDECOMP/PARAFAC decomposition yields a reconstruction accuracy of greater than $98\,\%$ ($R^2 = 0.986$) when the number of factors are set to 6. The accuracy remained by and large unchanged when we gradually increased the number of factors. Similarly, the accuracy deteriorates quickly when the factors were less than 5. Thus, we decided to decompose the utilisation tensor by assuming that six distinct factors adequately describe the resource utilisation characteristics of the hosted VMS. This yields a 44×6 $\mathbf{A}$ matrix (VMs vs hidden features), a 14×6 $\mathbf{B}$ matrix (resources vs hidden features) and 1440×6 $\mathbf{C}$ matrix (samples vs hidden features).

Fig. 6 displays the "spatial" (top) and the "temporal"

[2] The reconstruction of the original tensor from the decomposed matrices entails some error. One way of measuring the difference between the reconstructed tensor and the original tensor is by using the coefficient of determination.

(bottom) aspects of the tensor decomposition. The former are generated from the columns of the **A** matrix encoding the relationships of the factors with the virtual machines while the latter are generated from the columns of the **C** matrix encoding the relationships of the factors with all the samples. In order to make our analysis comprehensible, we have plotted in Fig. 7 the average values and variances of the most important key performance indicators, namely: CPU utilisation, memory utilisation, average received packets, average transferred packets, storage total read latency, storage total write latency. High utilisation variance can be taken as an indication of fluctuation of resource demand over time. Low variance simply means resource demand does not change appreciably over time. It does not, however, speak much about how much resources individual VMs utilise. For this, one has to look at the average utilisation. Having this in mind, it is possible to observe that almost all the VMs have relatively high demand for and variation in CPU and disk read/write operations.

If we closely study the temporal aspects of the tensor decomposition (Fig. 6, bottom), we can divide the plots into three groups: The first two plots show a resource utilisation pattern which is relatively intense during the day time whereas the next three plots exhibit activities which vary throughout the day and the night. Likewise, the last plot exhibits activities persisting throughout the day and the night, but here their intensity is modest compared to the ones revealed in the previous two plots (i.e, a low variance). The key performance indicators having a high variance are related to CPU and disk read/write operations. Therefore, the three plots (three to five) in Fig. 6 (bottom) refer to these resources. The last plot in Fig. 6 (bottom) refers to a memory operation, because, as can be seen in Fig. 7, it has a very small variance but almost all the VMs have appreciable memory demands. So, to what type of operations the first two plots in Fig. 6 (bottom) refer? They must refer to network receive and transfer operations. The fourth plot in Fig. 7 (top) indicates that almost all the VMs have considerable packet transfer operations, but one of them has a high variance, suggesting that it must have something to do with day time activity. Possibly, it is an email application, serving the university community. Since packet transfer must be complemented with packet reception, the fist two plots in Fig. 6 (bottom) refer to network operations.

Having used the temporal aspects of the tensor decomposition to understand the basic factors, it is now possible to categorise the virtual machines in terms of their resource consumption, because the same factors which are used to describe the temporal aspects of resource utilisation in the **C** matrix are used to describe the virtual machines in the **A** matrix. For example, it can be seen that VM 15 has high memory demand and high network receive/transfer operations but low CPU operations. This clearly qualifies it to be labelled as an IO-intensive VM. We can likewise classify each virtual machine according to its resource consumption pattern which is the first step towards identifying complementary as well as contentious characteristics.

V. Conclusion

In this paper we propose the use of tensors and tensor decomposition to analyse the resource utilisation characteristics of hosted virtual machines in large-scale data centres and to identify virtual machines exhibiting complementary and contentious features. The outcome of a tensor decomposition provides three different but complementary views into the temporal and spatial characteristics of resource utilisation. This can be useful for utilising available resources with comparable intensity. We demonstrated that the CANDECOMP/PARAFAC tensor decomposition identifies both "spatial" and temporal resource utilisation characteristics in a single step of decomposition.

For the analysis of our work we relied on measurement sets obtained from the Enterprise Data Centre of the Centre for Information Services and High-Performance Computing at the TU Dresden, Germany. The data centre consists of 59 physical computing servers organised into 9 clusters and 29 physical storage servers. Altogether, it has 120 multi-core processors with a total capacity of 1029 GHz CPU cycles, 1606 MB RAM and 65.7 TB disk space. Currently, it hosts 1190 commercial virtual machines. We chose the utilisation of 44 of the most active virtual machines for our analysis.

This work focused on analysis. The next step will be clustering virtual machines according to their "spatial" and temporal properties and scheduling them according to their complementary aspect. So, for example,

References

[1] C. Mobius, W. Dargie, and A. Schill, "Power consumption estimation models for processors, virtual machines, and servers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 6, pp. 1600–1614, 2014.

[2] T. H. Nguyen, M. Di Francesco, and A. Yla-Jaaski, "Virtual machine consolidation with multiple usage prediction for energy-efficient cloud data centers," *IEEE Transactions on Services Computing*, 2017.

[3] A. Floratou, A. Agrawal, B. Graham, S. Rao, and K. Ramasamy, "Dhalion: self-regulating stream processing in heron," *Proceedings of the VLDB Endowment*, vol. 10, no. 12, pp. 1825–1836, 2017.

[4] A. Strunk and W. Dargie, "Does live migration of virtual machines cost energy?," in *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*, pp. 514–521, IEEE, 2013.

[5] F. Hao, M. Kodialam, T. Lakshman, and S. Mukherjee, "Online allocation of virtual machines in a distributed cloud," *IEEE/ACM Transactions on Networking (TON)*, vol. 25, no. 1, pp. 238–249, 2017.

[6] P. McCullagh, *Tensor methods in statistics*. Courier Dover Publications, 2018.

[7] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[8] S. Kouchaki, S. Sanei, E. L. Arbon, and D.-J. Dijk, "Tensor based singular spectrum analysis for automatic scoring of sleep eeg," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 1, pp. 1–9, 2015.

[9] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, 2017.

[10] U. Kang, E. Papalexakis, A. Harpale, and C. Faloutsos, "Gigatensor: scaling tensor analysis up by 100 times-algorithms and discoveries," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 316–324, ACM, 2012.