Filtering Multivariate Workload Non-Conformance in Shared IT-Infrastructures

–Names and affiliations are excluded from the original submission–

***Abstract*—Virtualized data centers are hosting virtual machines (VMs) with time varying resource demand behavior (workload) jointly on the same physical servers in order to increase server utilization. To avoid server overload situations, a data center operator needs to decide which VMs should be assigned to a physical server for a given period of time. As assignment decisions are based on historical workload traces, they are sensitive towards deviations from forecasted demands that potentially require VM reassignments. What renders this task difficult is that VM live migrations are overhead afflicted control operations. On the one hand, it is mandatory to anticipate overload situations in order to trigger VM migrations in a proactive manner. On the other hand, unnecessary migrations of VMs should be avoided. Hence, an autonomic controller should accurately predict situations where the aggregated workload of a set of collocated VMs will hit the capacity limit of a server for predictable as well as for rather erratic workload behavior without requiring manual adjustments of model parameters. In this paper we propose an automated, non-parametric approach to proactively filter multivariate workload behavior that requires VM migration. We learn an orthonormal projection from workload traces and extract a set of key metrics that concisely describe relevant developments in the joint workload behavior of physical servers. A geometric interpretation, in combination with simple short term forecasting techniques of these metrics allows for VM migration decision making. Based on a set of real world workload traces we conduct numerical experiments to evaluate the approach.**[1]

I. Introduction

Energy consumption has become a key driver for operational costs in today's virtualized data centers. Typically a set of different enterprise applications are run in virtual machines (VMs) jointly on a single physical server in order to increases the average server utilization [7]. Data center operators aiming at operational efficiency try to reduce the amount of physical servers required to a level that still ensures the availability of sufficient computational resources to all hosted virtual machines at any time [14], [17]. As a physical server's capacity is shared among collocated VMs, a decision maker is required to assign VMs to physical servers with the objective to reduce the overall server count while avoiding server overload situations that may result from aggregated VM resource demands. Although historical workload traces allow for resource demand prediction and efficient VM assignment, forecasting errors that result from short term fluctuations, demand level and time shifts, or trends in workload patterns of enterprise applications [2], [5] require the migration of VMs from one server to another to avoid extended periods of server overload as well as underutilization. However, VM migration entails significant overheads regarding CPU and network bandwidth usage, depending on a VM's actual resource demands [4]. Especially server overloads need to be predicted in a reliable way in order to trigger migrations in time. Missing the optimal time frame for migrations may lead to resource contention (executing a migration during overload may even exaggerate contention) on a physical server resulting in application performance degradation. Identifying beneficial points in time for VM reassignments represents a highly relevant problem that requires new methods to estimate the aggregated resource demand of arbitrary sets of VMs rather than the demand estimation of a single VM.

Although efficient resource management remains a top priority in data center management, in practice the sheer volume of data to be analyzed renders this task almost impossible for any but small VM set sizes. In summary, the determination of optimal planning period durations and beneficial points in time for VM migrations is a non-trivial task even for small sets of collocated VMs. In oder to illustrate the problem under consideration, consider the CPU demand of a set of 30 VMs in five-minute averages over a day observed in the past as shown in the upper part of Figure 1. Even if we assume only one scarce resource (CPU), the resulting aggregated workload of the set of 30 VMs exhibits complex and volatile behavior over time as depicted in the lower part of Figure 1. The determination of optimal planning period durations, for which the assignment of VMs shall be kept stable, is challenging if migration

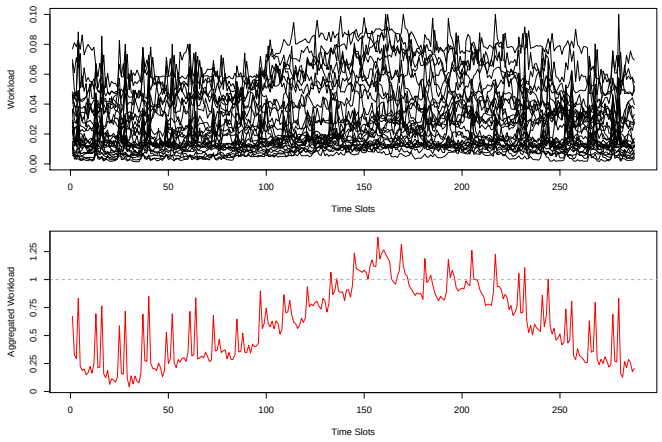

Fig. 1. Example Diurnal CPU demand of 30 VMs

[1]Acknowledgement: This work is supported by Siemens IT Solutions and Services (SIS). SIS provisioned the data for our experiments as well as valuable insights.

overheads need to be considered additionally. Accounting for the uncertainty in aggregated demand behavior and migration overheads would result in a probabilistic optimization model formulation, which is hard to solve. Time series models of aggregated demands depend on the constituents characteristics and the workload mix, which renders their usage inappropriate. Up to our knowledge, there exists no previous work on how to determine planning period durations and beneficial points in time for migrations not even for a single physical server. As will be shown in subsequent sections, the consideration of *normal* workload patterns of enterprise applications workloads can leverage migration decision making in order to minimize the amount of migrations and to use them effectively.

We conclude that knowledge of resource demand patterns of VMs is required for efficient dynamic resource management in virtualized data centers. However, for migration decision making the aggregated workload behavior of sets of VMs needs to be considered rather than individual VM workload developments. In shared infrastructures, contrasting dedicated application hosting, anomalous demands of a single application (VM) can be tolerated as long as the demands of collocated VMs compensate for the abnormal behavior (the aggregated workload behavior might range within acceptable bounds and may not require any control actions).

In this paper we propose an approach to reduce the complexity of managing large, dynamic systems without abandoning the opportunity to exploit useful information in available workload traces. We learn an orthonormal projection from the workload data of numerous VMs using a low-rank matrix approximation and extract a set of key metrics that concisely describe relevant structures and developments in workload of collocated VMs. A geometric interpretation of these metrics in combination with simple forecasting techniques is then used to automatically determine when VM reassignments are recommended. Based on workload data from a large professional data center, we demonstrate how the approach can be used as decision logic for an automated, non-parametric controller and how a simple plot of derived metrics can serve as a decision support tool. Starting wit a discussion of related work in section II, we will review our low-rank matrix approximation method in III and show how the reduced workload data description can be used to recommend migrations. Subsequently we demonstrate the approach and present experimental results. In section IV conclusions are drawn and future work is discussed.

II. Related Work

In the literature two distinct workload management techniques have been proposed that exploit VM live migrations for higher operational efficiency in data centers. First, approaches employing pre-determined static planning period; second, reactive, rule-based control techniques. Static techniques, as proposed by Rolia et al. [15] employ long planning periods such as days and weeks. At the end of each planning period the data center is transfered from the actual VM assignment to a new one that will be in place for the duration of the next planning period by triggering VM migrations in one large batch. Long planning periods are very sensible towards changes in the resource demand behavior of even a few VMs on a single physical servers. On the other hand, short planning periods of hours or minutes as proposed by Gmach et al. [8] or Zhu et al. [20] lead to frequent migrations of VMs entailing high control action overheads. Furthermore, short planning periods may result in unnecessary oscillations of VMs over time. Khanna et al. [12] and Wood et al. [19] propose reactive, rule-based approaches that define threshold levels regarding the utilization of specific physical server resources such as CPU-demand, memory allocation, or network bandwidth usage. Threshold-based approaches are also proposed and supported by commercial VM management software such as Xen-Center or VMware DRS. The idea of such rules is to trigger the migration of one or multiple VMs if the aggregated resource demands of VMs assigned to a server exceed a certain predefined utilization threshold. Similar to these methods, Seltzam et al. [16] propose fuzzy logic based reactive rules to trigger VM migrations. Static, definite and fuzzy rules are inherently inefficient if they are operationalized in a data and context agnostic fashion due to their myopic behavior.

While there exists a substantial body of related work on resource demand prediction, none of the mostly time series analysis based approaches takes into consideration the aggregated resource demands of consolidated servers. Hellerstein et al. [11] use ANOVA analysis to capture repeatable patterns in single application workloads and an autoregressive model for capturing the demand's deviation from normal patterns. Their method can be used to determine seasonal workload patterns from workload traces in a manual way. Vialta et al. [18] also propose the use of simple time series models to predict performance variables and to identify abnormal behavior on a medium and short term basis. We employ similar methods to anticipate threshold violations for aggregated resource demands on a single physical server. Since aggregated workloads do not expose clear patterns and different sets of workloads lead to different time series models, manual model selection and parameter estimation based on historical data is not viable in our problem setting. In contras to more established time series modeling, our approach mitigates the need for model and parameter selection as far as possible.

III. Normal and Non-Conformant Workload Behavior

The resource demands of most enterprise applications expose significant seasonal patterns on a daily, weekly, and even on a monthly basis [3], [9]. As an example, figure 2 illustrates the maximum hourly CPU utilization of an arbitrarily chosen enterprise application over five subsequent days of the week.

Figure 3 shows the average application workload per time slot (hours) as observed in the past, while figure 4 shows the uncertainty in the average pattern in each time slot. CPU demand

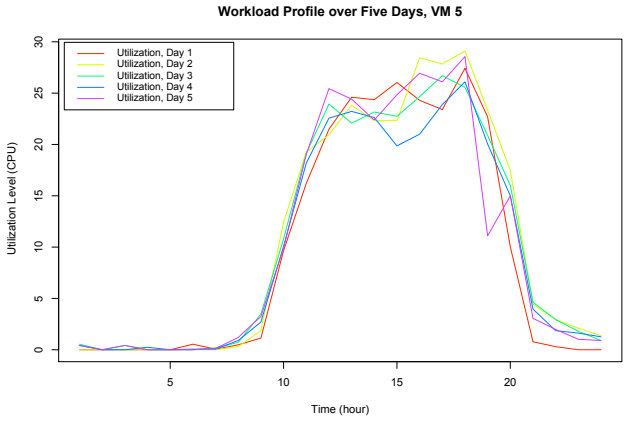

Fig. 2. Example Workload Profile of an Enterprise Application

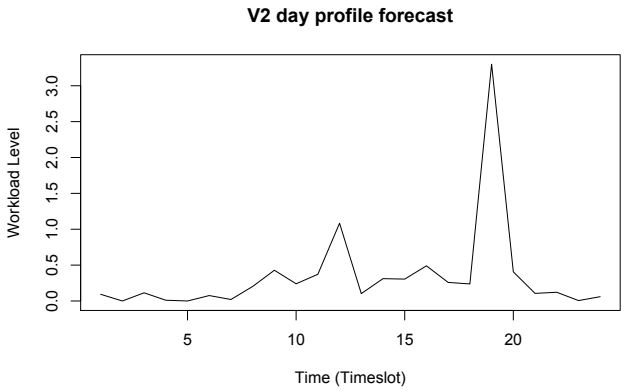

Fig. 4. Uncertainty in Workload Patterns

levels are expressed relative to th host server's maximum CPU capacity. Here, the residual workload behavior not explained by the major periodic component in figure 3 is indicated by the maximum observed deviance from the average workload in a time slot. The diagrams reveal that most uncertainty in workload is found around 7 p.m. (3%) and around 12 p.m. (1%). In other words, during these periods of time, the average application workload estimate needs to be handled with care. For the remaining time slots, the major deviation is below 0.5% of the workload. Although we found that a large class of applications are highly predictable, Figure 3 and Figure 4 indicate heteroskedacity even for such applications, which renders a large set of time series models problematic even for short term forecasting.

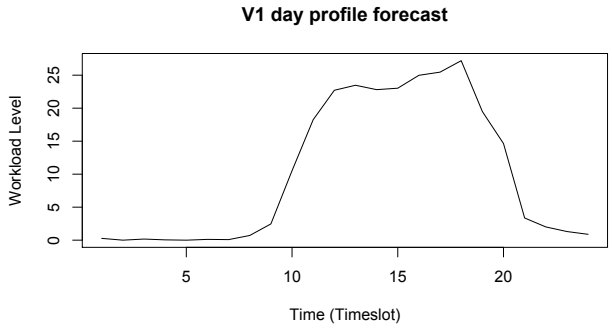

Fig. 3. Major Patterns in Workload

In general, periodicities in univariate time series data can be detected with autocorrelation functions, and forecasting mechanisms can be used to improve respective pattern forecasts and quantify uncertainty in pattern (instead of taking the average in each time slot as done in the example above) [11].

A. *Multivariate Time Series Analysis*

However, an autonomic controller is faced with multivariate workload time series of co-hosted VMs, where each VM exhibits individual, time-varying workload behavior. There is a large body of literature on multivariate time series analysis, which can be used for such purposes. We focus on singular value decomposition (SVD) for the following reasons. Golub and vanLoan [10] have shown that SVD derives the best approximation of a time series matrix given a maximum number of dimensions to approximate the matrix. In addition, SVD is a model-free approach and no parameters need to be adjusted. Furthermore, it is applicable to arbitrary time series matrices including non-square and not full-ranked workload matrices. Furthermore, the decomposition can be computed in quadratic time and fast SVD approximations exist.

Let us first introduce some notation. Let r_{jt} indicate the CPU demand pattern of a VM j ($j = 1, \ldots, J$), in a time interval t over a planning period devided into τ time slots (e.g., 144 ten-minute resource demand aggregates over a day). Let then R be the $J \times \tau$ matrix with workload time series of J co-hosted VMs as row-vectors.

Golub et al. [10] have proven that R can be factorized by singular value decomposition to $U\Sigma V^T$ (V^T denotes the conjugate transpose of V), where R's singular values σ_e in the diagonal matrix Σ (with $e \in \{1, \ldots, rank(R)\}$) are ordered in non-increasing fashion, U contains the left singular vectors (the *Eigenvectors* of $R^T R$), and V^T contains the right singular vectors (the *Eigenvectors* of RR^T). All left and right singular vectors are mutually orthogonal and normalized by their 'length', defined in terms of the l^2 norm.

The intuition of this decomposition is that $\{u_k\}$, the column vectors in U, are new axis spanning R's column space, the associated σ_k are scaling factors for these axis, and elements in v_k^T are coordinates of time slots along the new axes. As an illustration how SVD works, consider the simple scenario in Figure 5. The graphs show the diurnal demand patterns of 16 co-located VMs for CPU. CPU demand levels are shown relative (in percent) to the maximum CPU performance of the

host server. The entries in the first and second left singular

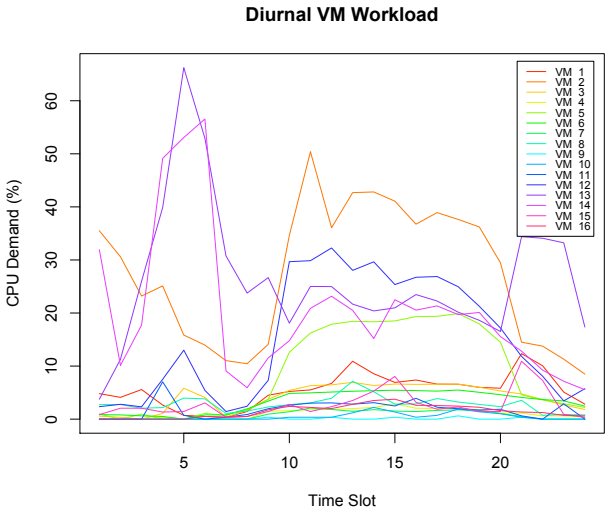

Fig. 5. Diurnal CPU Demand Patterns of 16 VMs

value of a SVD of the matrix underlying the curves in Figure 5 are shown in Figure 6 and in Figure 7, respectively. As the major new axis for R's columnspace, u_1 describes the most relevant workload mix or proportions between VMs over all time slots. The coordinates v_2^T regarding u_2, the second most dominant VM workload mix, capture the maximum variance after removing the data projection along u_1 (in this two-dimensional example, u_2 captures all of the remaining variance). The relevance of an axis is further quantifying by its associated singular value, and t's coordinate for a particular axis quantifies the impact of a particular workload mix in time slot t. However, usually higher-order singular values decrease rapidly and associated workload mixes can be ignored, as the singular values associated with the right singular vectors sort them in the order of the explained variation from most to least significant.

Using SVD, the workload mix in a time slot t can be

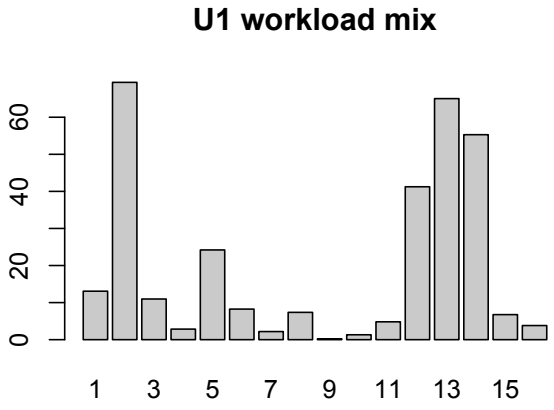

Fig. 6. u_1 Workload Mix

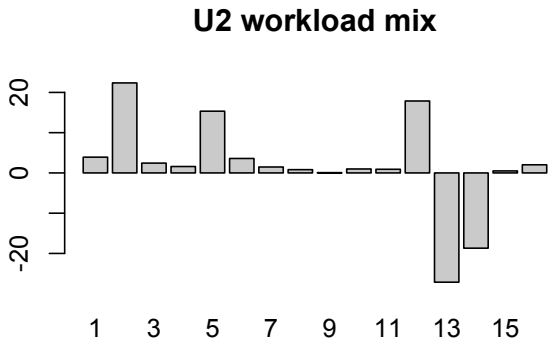

Fig. 7. u_2 Workload Mix

approximated by a super-composition of the first few left singular vectors (the dominant workload mixes) scaled by associated singular values and t's coordinates along the first few left singular vectors. Figure 8 shows the VM time series approximation derived by the first two singular vectors of the SVD. Even though the resulting graphs differ from the original curves, the principal workload behavior can be reconstructed by considering the first two singular vectors only. For each time slot t, the sum of all entries in first two the singular values, scaled by their associated singular values and multiplied by t's coordinates equals approximately the server's total CPU demand level in t.

B. Workload Modelling Using Dominant SVD Subspaces

Consider now a v_1^T v_2^T scatterplot (the coordinates of time slots along u_1 and u_2) as shown in figure 9. For each each hour of a day ($t = 1, \ldots, 24$) the predicted (normal) coordinates along the two dominant workload mixes of a set of VMs are drawn. The sum of entries (the weight) in the u_1 workload mix multiplied by the associated singular value, $W(u_1)$, is scaled along the horizontal axis and the vertical axis scales, $W(u_2)$

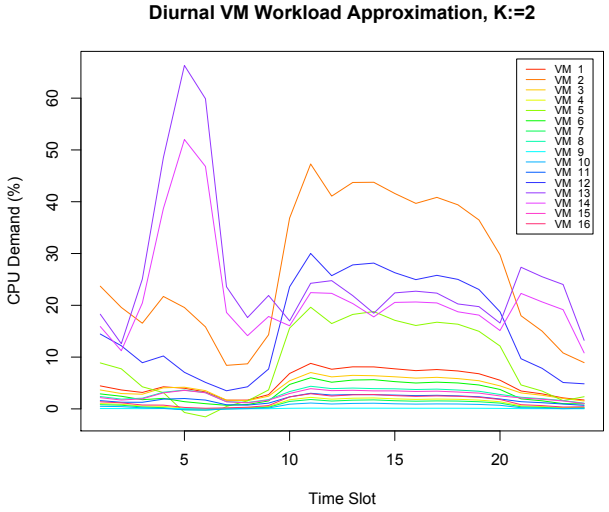

Fig. 8. Approximation of Diurnal CPU Demand Patterns of 16 VMs

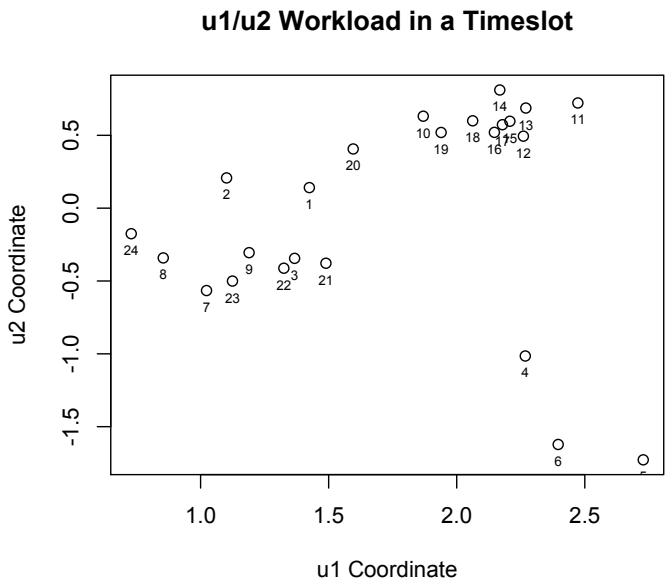

Fig. 9. v_1^T v_2^T Scatterplot

the sum of entries in the u_2 workload mix. Let $v_{1,t}^T$ ($v_{2,t}^T$) denote the t-th entry in v_1^T (v_2^T). To avoid server overlaod, for each t the following inequation must hold: $(W(u_1) \cdot v_{1,t}^T + (W(u_2) \cdot v_{2,t}^T) \leq 1$. Using this inequation, a line can be added to the plot, which separates the allowed utilization region from the region where a server is overloaded as shown in figure 10.

Furthermore, a second line (the dotted line in the plot) can be drawn, which indicates a CPU demand limit for migrations. If the demand in a time slot t is passing the line, no migrations are feasible without experiencing in server overload. The distance between both lines indicates the migration overhead of a VM with the lowest expected CPU migration overhead) of all VMs assigned to the server under consideration. Migration overheads can be learned from log traces obtained from past migrations.

The plot indicates that, assuming normal workload behavior of all VMs, no overload situation will occur. Reconsidering Figure 2, where we derived an indicator for uncertainty in workload behavior over time for univariate workload behavior of a VM, we can derive the expected uncertainty in the aggregate of a set of VMs assigned to a single server. In scatterplots we represent the level of workload uncertainty within a time slot by the diameter of a data point. Figure 11 shows the resulting plot. The points representing the time slots 5 and 11 are located beyond the dotted line. During these phases, workload intensity is high and a planner must be confident that the workload prediction is accurate, as here even small deviations from expected behavior might result in non-preventable overload situations. However, given stable workload behavior and depending on the risk-attitude of the planner, co-hosting these VMs might be a valid option. Significant long-term trends or workload shifts can be easily determined by applying traditional time series forecasting mechanisms such as spectral decomposition or ARMA models on the coordinates along u_1 and u_2 as applied for example by Fan [6].

C. Anomaly Filtering Based on SVD Coordinates

As an illustration, consider aggregated VM workloads with an upward trend as shown in Figure 12. For reasons of brevity, we connect consecutive points and derive a respective daily loop. Using data of the first three days, the overload situation on the forth day can be anticipated with simple trend analysis and corrective actions can be triggered in a timely manner before approaching the overload situation.

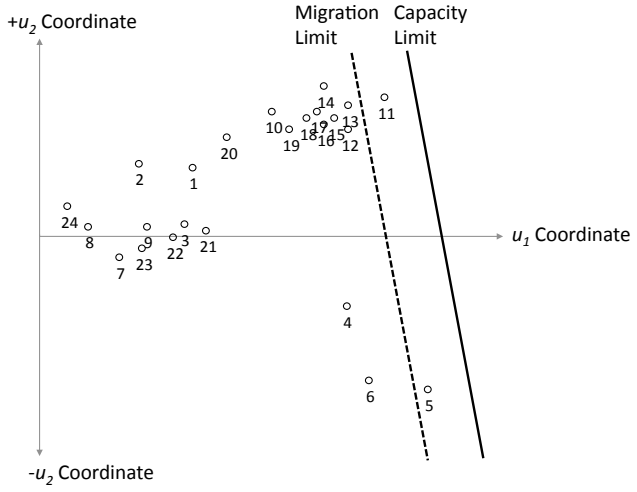

Fig. 10. v_1^T v_2^T Scatterplot with Capacity Limit

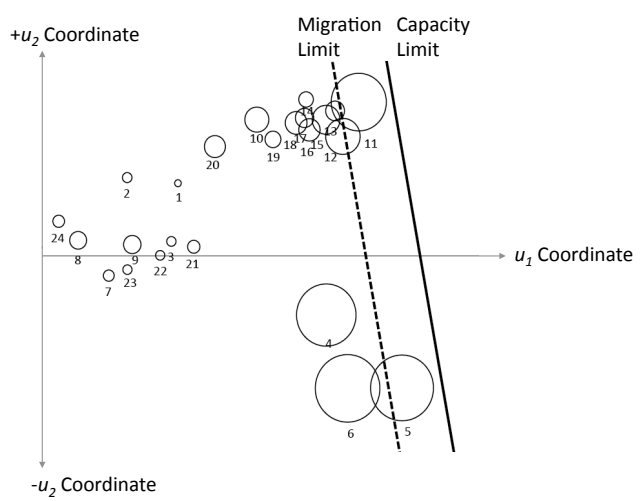

Fig. 11. v_1^T v_2^T Scatterplot considering Uncertainty

Short-term, intra-period workload anomalies that might result

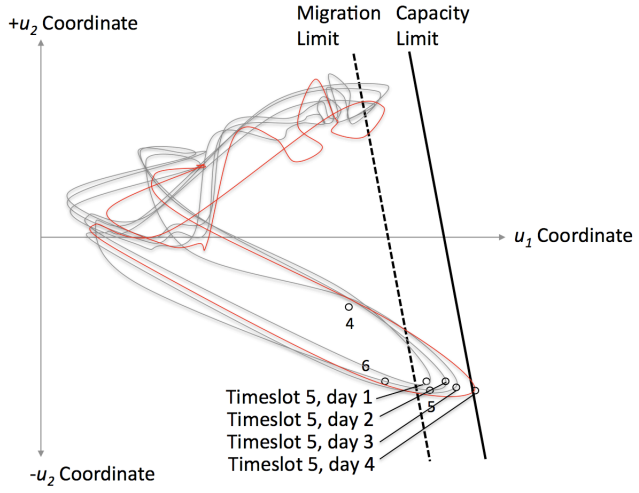

Fig. 12. v_1^T v_2^T - Long-Term Non-Conformance Detection

in overload situations are harder to predict. Based on the definition of normal workload behavior for a single VM, we use exponential smoothing for proactive anomaly detection. Exponential smoothing has been successfully used on high frequency data in a similar context in [1] and allows us to extrapolate the workload behavior of a single physical server along u_1 and u_2 axes.

As an example, consider Figure 13. The underlying workload data exhibit a non-conformant workload increase in aggregated workload beginning at time slot 1. The normal behavior is reflected by the shape of the grey loop, the current workload behavior is reflected by the red line. Using exponential smoothing as defined in [13] for trend extrapolation, it is possible to detect a deviance from normal behavior in coordinates of time slots 1-3. With $\alpha := 0.3$, the overload situation in time slot 5 (the forecast is 5') can be anticipated and a migrations can be triggered to avoid the future overload situation. Besides the automated detection and anticipation of deviations in workload that might lead to overload, the resulting scatterplots can be used as dashboards to track workload and to support semi-automated or manual decision making. In order to reduce the managerial complexity of monitoring hundreds or thousands of servers in parallel, mostly an IT manager will focus on non-conformant server workload that might hit the capacity limit of a server, while fading out normal, uncritical workload behavior. In addition, scatterplots of multiple servers can be integrated in fewer scatterplots that can be easily inspected.

Figure 14 sketches how multiple server scatterplots might be integrated. The figure shows the current workload of four different servers including a one time slot ahead workload forecast. The visualization draws the coordinates of server 1, 2, and 4 in light-grey color as these currently exhibit uncritical behavior. The state of host server 3 is marked in red as the current coordinates indicate an anomalous workload behavior beyond the tolerance limit (the circle around the server's expected coordinates) and the slot-ahead forecast of its coordinates reveals a potential overload situation.

D. Experimental Evaluation

We now study the efficiency of the proposed SVD-based approach. We conduct numerical experiments with workloads of 370 VM over a month. The VM workload time series are taken from workload traces from a large professional data center in Europe, containing CPU-demand averages over five-minute time slots of VMs hosting web and application servers of various customers. The first three weeks are used as training data to learn the diurnal profiles of the VMs, while data of the last week is used as the test data set.

We conduct numerical experiments using 2000 different scenarios, where a scenario is a set of arbitrarily chosen VM workloads collocated an a physical server. In each scenario, we select a set of VM workloads that does not exceed the capacity limit of a host server when assuming deterministic workload behavior derived by the average workload profiles learned in the training phase. With our data sample, on average eight VMs are hosted jointly on a single server.

The false positive rate and the false negative rate is used as our evaluation criterion. As usual in statical analysis, the false positive rate is the proportion of predicted overload situations where no overload occurred, while the false negative rate is the proportion of realized overload situations not predicted by the detection method. Please notice that we only consider an overload situation as anticipated if it is predicted before passing the threshold line, as later no migrations are possible due to resource shortages. As proposed by the data center operator, in our experiments migration overhead of a VM is assumed to be 35% of its current CPU-demand.

Outcomes are benchmarked against results obtained when setting static threshold levels from 50% to 90% CPU utilization for triggering control actions and observing the actual

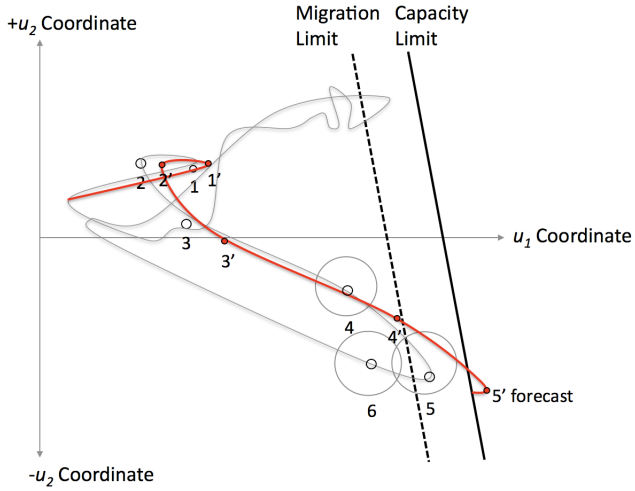

Fig. 13. $\{v_1^T, v_2^T\}$ Short-Term Non-Conformance Detection

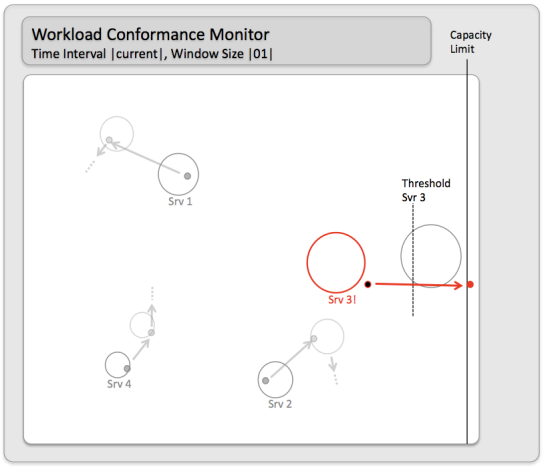

Fig. 14. Server Workload Monitoring Dashboard

workload on a physical server. In contrast to our method the threshold based detection method does not make use of forecasting. Instead, if a threshold is violated an overload alarm is raised immediately. If the workload development resulted in an overload situation as long as the current upward trend is not permanently (within $m = 6$ time slots) inverted, a true positive is detected, otherwise a false positive is raised. If a threshold is violated and the workload development leads immediately (within the same time slot) to an overload, a false negative is reported.

Table I shows the aggregated experimental outcomes. As

TABLE I
EXPERIMENTAL RESULTS

Overload Filter	False Negative Rate	False Positive Rate
50% Threshold	0.00	0.71
60% Threshold	0.00	0.55
70% Threshold	0.02	0.21
80% Threshold	0.04	0.11
90% Threshold	0.14	0.06
SVD based	0.03	0.02

could be expected, the false positive rate of the simple threshold-based approach increased with decreasing threshold levels, while the false negative rate increased with increasing threshold level. Both sensitivities are not surprising, as with static thresholds both metrics cannot be optimized simultaneously. However, both, high rates for false positives and false negatives are critical as in the first case ineffective migrations are triggered that result in artificially created resource demand increases and potentially more frequent overload situations, while false negatives entail immediate overload situations that could not be dealt with in a proactive way.

In contrast, the consideration of typical workload patterns and the dynamic detection of critical changes in workload behavior as done with the proposed SVD-based approach leads to much better migration decision making as it lowered both error rates to rather moderate levels. An ex-post analysis revealed that around 94% of overload situation have been predicted using short-term forecasting, and around 6% have been prediced using longer-term forecasts (beyond 12 time slots into the future). We did not find significant deviance in error rates between both, overload and under-load situations (for under-load prevention the thresholds were inverted).

IV. SUMMARY AND OUTLOOK

We considered the problem of proactive filtering multi-variate workload behavior of co-hosted VMs that requires migration to avoid overload. We proposed a non-parametric approach based on singular value decomposition of workload data to describe principal workload behavior and significant workload deviance. The approach can be used for automatically triggering VM migration and the resulting visualization of parameters and parameter changes can serve as a decision support tool for IT managers. First numerical experiments show that the approach outperforms static threshold-based approach often used in practice. We demonstrated the viability of our method on VMs co-hosted on a single server. The exponential smoothing forecasting of coordinates along the dominant singular vectors for anticipatory overload detection has shown good results in comparison to threshold-based approaches typically applied in practice. However we plan to implement other, more adaptive short term trend prediction methods that take the volatility in resource demands into consideration. In the future, we plan to evaluate the approach in a real test data center to consider for the uncertainty in migration overheads and system workloads.

Furthermore, we are currently implementing a data center controller that determines the optimal physical target server for a VM migration which aims at locking up further energy savings.

REFERENCES

[1] Mauro Andreolini, Sara Casolari, and Michele Colajanni. Models and framework for supporting runtime decisions in web-based systems. *ACM Trans. Web*, 2(3):1–43, 2008.

[2] M. Bichler, T. Setzer, and B. Speitkamp. Capacity management for virtualized servers. In *Proceedings of International Conference on Information Systems (ICIS), Workshop on Information Technologies and Systems (WITS)*, 2006.

[3] M. Bichler, T. Setzer, and B. Speitkamp. Provisioning of resources in a data processing system for services requested, 2006.

[4] Christopher Clark, Keir Fraser, Steven H, Jacob Gorm Hansen, Eric Jul, Christian Limpach, Ian Pratt, and Andrew Warfield. Live migration of virtual machines. In *In Proceedings of the 2nd ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI*, pages 273–286, 2005.

[5] X. Zhu et al. 1000 islands: Integrated capacity and workload management for the next generation data center. In *Proceedings of the Int. Conference on Autonomic Computing*, 2008.

[6] Li Fan. Singular value decomposition expansion for electrical demand analysis. *IMA Journal of Mathematics Applied in Business and Industry*, 11:37–48, 2000.

[7] D. Filani, J. He, and S. Gao. Technology with the environment in mind. Technical report, INTEL Corporation., 2008.

[8] D. Gmach, J. Rolia, L. Cherkasova, G. Belrose, T. Turicchi, and A. Kemper. An integrated approach to resource pool management: Policies, efficiency and quality metrics. In *Dependable Systems and Networks With FTCS and DCC, 2008. DSN 2008. IEEE International Conference on*, pages 326–335, June 2008.

[9] Daniel Gmach, Jerry Rolia, Ludmila Cherkasova, and Alfons Kemper. Workload analysis and demand prediction of enterprise data center applications. In *IISWC '07: Proceedings of the 2007 IEEE 10th International Symposium on Workload Characterization*, pages 171–180, Washington, DC, USA, 2007. IEEE Computer Society.

[10] G. Golub and C. v. Loan. *Matrix Computations*. Johns Hopkins Univ. Press, 1990.

[11] Joseph L. Hellerstein, Fan Zhang, and Perwez Shahabuddin. A statistical approach to predictive detection. *Comput. Netw.*, 35(1):77–95, 2001.

[12] G. Khanna, K. Beaty, G. Kar, and A. Kochut. Application performance management in vm environments. In *IEEE/IFIP NOMS*, Vancouver, BC, Ca, 2006.

[13] David J. Lilja. *Measuring computer performance: a practitioner's guide*. Cambridge University Press, New York, NY, USA, 2000.

[14] OGC. *Service Delivery (IT Infrastructure Library Series)*. The Stationery Office, London, UK, 2000.

[15] Jerry Rolia, Ludmila Cherkasova, Martin Arlitt, and Artur Andrzejak. A capacity management service for resource pools. In *WOSP '05: Proceedings of the 5th international workshop on Software and performance*, pages 229–237, New York, NY, USA, 2005. ACM.

[16] S. Seltzam, D. Gmach, S. Krompass, and A. Kemper. Autoglobe: An automatic administration concept for service-oriented database applications. In *Int. Conference on Data Engineering (ICDE 2006)*, Atlanta, USA, 2006.

[17] H. Stevens and C. Petty. Data centers focus on green, but many neglect metrics. Technical report, Gartner, 2009.

[18] R. Vilalta, C. V. Apte, J. L. Hellerstein, S. Ma, and S. M. Weiss. Predictive algorithms in the management of computer systems. *IBM Syst. J.*, 41(3):461–474, 2002.

[19] Timothy Wood, Prashant Shenoy, Arun Venkataramani, and Mazin Yousif. Black-box and gray-box strategies for virtual machine migration. In *4th USENIX Symposium on Networked Systems Design and Implementation*, pages 229 – 242, 2007.

[20] Xiaoyun Zhu, Cipriano Santos, Dirk Beyer, Julie Ward, and Sharad Singhal. Automated application component placement in data centers using mathematical programming. *Int. J. Netw. Manag.*, 18(6):467–483, 2008.