DATA CENTER WORKLOAD CONSOLIDATION BASED ON TRUNCATED SINGULAR VALUE DECOMPOSITION OF WORKLOAD PROFILES

Thomas Setzer
Technische Universität München
setzer@in.tum.de

Abstract

In today's data centers, typically thousands of enterprise applications with varying workload behaviors are hosted. As energy usage is one of the key cost drivers in data centers, workload consolidation is increasingly used to host multiple applications on a single server, sharing and multiplexing a server's capacity over time. To minimize the number of required, energy-consuming servers, IT managers need to decide which applications should be combined on which server. For that purpose, typically application workload levels are predicted for a planning period such as a month in a defined granularity (e.g., over 5-minute intervals). Then integer programs are used to minimize the amount of required servers, while for each interval constraints ensure that the aggregated workloads of applications assigned to a server must not exceed a server's capacity. As such problems are NP-hard and computationally intractable for data centers with hundreds of servers and fine-grained workload data, approximations are applied to find at least a good solution, often abandoning the chance to find the optimum. In this paper we propose a novel approach based on applying Singular Value Decomposition to the workload data to reduce the dimensionality of the problem by capturing workload features in order to make the problem computationally tractable. We interpret the coordinates of the time-series projections along the first right singular vectors as indicators for workload levels and complementarities and propose a model to solve the consolidation problem with these few indicators only. We evaluate the model using industry data.

Keywords: Workload Management, Virtualization, Consolidation, Multivariate Data Analysis, SVD

1. Introduction

In today's data centers, typically thousands of enterprise applications like ERP modules or databases with complex and varying workload behaviors are hosted. Server virtualization based workload consolidation is increasingly used to raise server utilization levels. Server virtualization allows for hosting multiple virtual servers (or virtual machines (VM)) including application plus underlying operating system on a single physical server (target). A target's capacity is then shared and multiplexed over time amongst VMs. As specifically energy costs account for 30–50% of the total data center operation costs, IT managers need to decide which VMs should be combined (consolidated) on which target to minimize the number of required, energy-consuming targets (Filani et al. 2008).

Existing consolidation decision models typically first predict VM workload over a planning period such as a day or a month in a defined granularity (e.g., maximum workload over 5-minute intervals) based on past observations. Usually, workloads show recurring patterns on a daily or weekly basis. For example, payroll accounting is performed at the end of the week, while workload of an OLAP application has a daily peak in the morning when managers access their reports. More advanced consolidation models leverage these cycles by first determining representative e.g. daily VM workload profiles describing the workloads expected in each time interval (e.g. maximum over a 5-minute interval) for different resource types such as CPU and memory. Second, an integer program (IP) attempts to assign those VMs together on targets whose workloads are complementary, i.e. peaks are at different times of the day to smoothen and increase overall target workload in order to reduce the number of targets. One constraint per resource and interval ensures that the aggregated workload of VMs assigned to a target must not exceed the target's capacity.

As an example consolidation model we describe the *Static Server Allocation Problem considering Varying Workload* (*SSAPv*) model published by Bichler et al. (Bichler et al. 2006). Suppose that we are given J VMs j, $j \in \{1,\ldots J\}$ to be hosted by I or less target servers i, $i \in \{1,\ldots I\}$. Different types of resources k, $k \in \{1, \ldots, K\}$, may be considered and each target has a certain capacity s_{ik} of resource k. y_i is a binary decision variable indicating if target i is used, c_i describes the cost of a target (e.g. energy costs over a planning period), and the binary decision variable x_{ij} indicates which VM is allocated to which target. The planning period is divided into time intervals indexed by $t=\{1, \ldots, \tau\}$. Let further u_{jkt} describe how much capacity j requires of k in t. Techniques how to derive u_{jkt} are described in (Bichler et al. 2008). The resulting consolidation problem is formulated in equation (1).

$$\begin{aligned} &\min \sum_i c_i y_i \\ &s.t. \\ &\quad \sum_{i \le I} x_{ij} = 1 && \forall j \le J \\ &\quad \sum_{j \le J} u_{jkt} x_{ij} \le s_{ik} y_i && \forall i \le I, \forall k \le K, \forall t \le \tau \\ &\quad y_i, x_{ij} \in \{0,1\} && \forall j \le J, \forall i \le I \end{aligned} \qquad (1)$$

The objective function minimizes server costs, the first constraint ensures that each VM is allocated exactly once, and the second constraint ensures that the aggregated workload of multiple VMs does not exceed a target's capacity.

As the problem is strongly NP-hard, it cannot be solved optimally for larger instances, in particular as the number of constraints grows linearly with τ multiplied by K (Garey et al. 1979). Therefore, usually intervals are coarsened to reduce the number of constraints, e.g., hourly workload intervals are used by taking maxima over 12 5-minute intervals. However, coarsening intervals reduces the problem size but also the ability to exploit workload complementarities and therefore impacts the solution quality. Additionally, there are inherent inefficiencies with time intervals: for a certain period of time an interval might be coarse for VMs with volatile workload during that period, while it might be unnecessarily fine-grained for other VMs with smoother workload during that period (v.v. in other periods).

In this paper we propose a consolidation model based on multivariate statistics to circumvent the computational problems resulting from fine-grained workload data as well as the trade-off between fine- and coarse-grained time resolution. In section 2 we apply truncated Singular Value Decomposition (SVD) to the original workload matrix (with workload time series as row vectors) and project the time series onto data points in the space spanned by the first right singular vectors of the SVD. In section 3 we give an interpretation of these points, where coordinates along the first right singular vector indicate workload levels, and subsequent coordinates indicate workload complementarities. Subsequently we develop a mathematical model to solve the consolidation problem with only the few indicators derived. In section 4 we evaluate the model using industry data. Related work is discussed in section 5. In section 6 conclusions are drawn and future work is discussed.

2. Dimensionality Reduction of Workload Data

The $K\tau$-dimensional tuples describing VM workload time series can be represented as points in a $K\tau$-dimensional space, where a VM workload level of a resource k in t is indicated as a coordinate along a dimension (k, t). To reduce dimensionality, these points need to be projected into an E-dimensional space so that $E<<K\tau$. We apply truncated SVD for that purpose as it is applicable to non-square and not full-ranked workload matrices and fast SVD approximations exist.

Let R be the original J by $K\tau$ matrix of J VMs, with time series (per k) of length τ as row-vectors (elements of R are u_{jkt}). Let $U \Sigma V^T$ be R's factorization using standard SVD, where R's singular values σ_e in Σ are ordered in non-increasing fashion, U contains the left singular vectors, and V^T contains the right singular vectors. The intuition of this factorization is that the right singular vectors are the axis of a new space, the associated singular values are scaling factors for these axis, and the row-vectors in U represent the coordinates of VM workloads in the new space. As an illustration, consider workloads u_{jkt} of VMs j for $\tau = 2$ (maximum during daytimes (t=1) and nighttimes (t=2)) for one resource k=1. The resulting data points are shown in Figure 1.

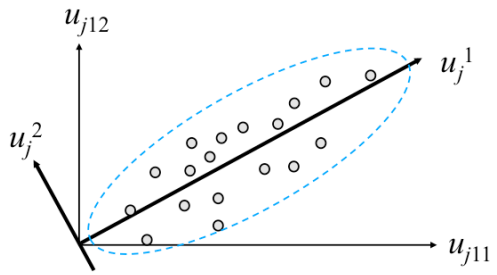

Figure 1: Workload Time Series Projections

For each j, coordinates u_j^1 are calculated by perpendicular projection of the points onto u^1, the first right singular vector. These coordinates show the best 1-dimensional approximation of the data because u^1 captures as much of data variation as possible by one direction. VM coordinates u_j^2 regarding the second right singular vector u^2 ($u^2 \perp u^1$) captures the maximum variance after removing the projection of the data along u^1 (in this 2-dimensional example, u^2 captures all of the remaining variance; in general the number of singular vectors equals R's rank).

What makes SVD practical is that variation below a particular threshold E can be ignored as the singular values associated with the right singular vectors sort them in "goodness" order with explained variation from most to least. This is the idea of truncated SVD where only the first E column vectors of U and the first E row vectors of V^T are considered.

3. Dimensionality Reduction of Workload Data

A Principal Direction of Workload and Capacity Limits

As a regression line running through the data points, u^1's direction approaches dimensions (k, t) with high aggregated workload where overload situations are likely to appear. Hence, we interpret u^1 as major workload direction. Consider the scenario depicted on the left-hand side of Figure 2.

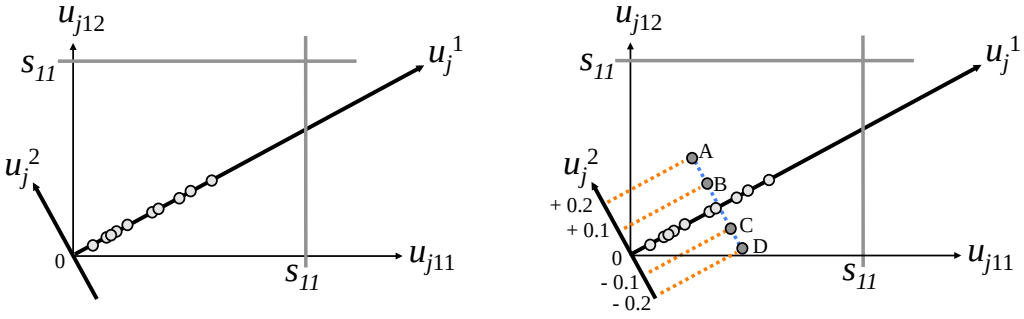

Figure 2: Major Workload direction and Complementarity

The coordinate u_j^1 of a VM j along u^1 fully describes j's workload as $\sigma_e = 0 \; \forall \; e > 0$. Here, the problem can be solved as a variant of the bin-packing problem, with u_j^1 as object sizes, and the projection of the target capacity limits as bin sizes. We determine the bin sizes as follows: for each of the $K\tau$ original dimensions the capacity constraint for resource k of target i is s_{ik} (for all t). Hence, for each target we obtain hyperplanes which form a convex polyhedron indicating its capacity limits (the grey lines in the pictorials show the hyperplanes of a target i=1; a rectangle in the 2-dimensional case).

As a point (e.g. the aggregated u_j^1- coordinates of combined VMs) outranging this rectangle indicates target overload, the capacity limit is the intersection point P_i of u^1 and a hyperlane of target i. Hence, i's bin size equals $\|P_i\|$, the Euclidian norm of the vector from origin to P_i.

B Workload Complementarities

However, usually $\sigma_2, \sigma_3, \ldots$ are non-zero and u^1-based workload estimation is inaccurate. In the scenario depicted on the right-hand side of Figure 2, additional VMs *A-D* with equal u_j^1 but different u_j^2 coordinates are considered. u_j^2 captures "distances" to u^1, i.e., u^1 workload approximation errors. Workload in t=1 (t=2) is overestimated (underestimated) by u_j^1 for VMs with positive u_j^2 (A and B); the opposite for VMs with negative u_j^2 (C and D). Hence, when combining VMs with positive and negative u_j^2 - for example A and D – A's workload is overrated in intervals where B's workload is underestimated and v.v., which reduces a target's aggregated u^1 workload estimation error. For example, when combining A and D, A's higher workload in t=1 is compensated by D's lower workload in t=1 in order to avoid overload due to u^1 approximation errors. Therefore, VMs j with u_j^2-coordinates that add to zero can be considered as complementary.

C Model Formulation

On the other hand, combining VMs with positive u_j^2 like A and B on a target further intensifies u^1 workload underrating in t=2. Let z_{i2} be the absolute sum over u_j^2 values of VMs assigned to a target. To avoid target overload when using a bin-packing formulation, z_{i2} must be added to the aggregated u_j^1 coordinates of assigned VMs to ensure sufficient capacity in all time intervals. The resulting IP entitled *Thin Workload Consolidation Model* (*ThinWCM*) is shown in equation (2).

$$
\begin{aligned}
&\min \sum_{i \le I} c_i y_i \\
&s.t. \\
&\quad \sum_{i \le I} x_{ij} = 1 && \forall j \le J \\
&\quad \sum_{j \le J} (u_j^1 \, x_{ij}) + z_{i2} \le \|P_i\| y_i && \forall i \le I \\
&\quad \left| \sum_{j \le J} (u_j^2 x_{ij}) \right| - z_{i2} = 0 && \forall i \le I \\
&\quad y_i, \; x_{ij} \in \{0,1\} && \forall j \le J, \; \forall i \le I \\
&\quad z_{i2} > 0 && \forall i \le I
\end{aligned}
\tag{2}
$$

Again, the objective is to minimize server costs and the first constraint ensures that each VM is allocated exactly once. The second constraint ensures that the aggregated u^1 workload estimate of VMs assigned to a target plus z_{i2}, their aggregated u^2-coordinates do not exceed the target's capacity limit. The third constraint calculates z_{i2} required in the second constraint (replacing the third constraint by two linear constraints is straightforward). Although not shown for reasons of clarity, variation along u^3, u^4, … is

considered just as variance along u^2. For each u^e, $2 < e < E$, we introduce a constraint to determine z_{ie}, and add each z_{ie} to z_{i2} in the second constraint. As a conservative estimator for the remaining variance in u^{E+1}, u^{E+2}, … for each j we add the sum of j's absolute coordinates u_j^e, $e>E$, to u_j^E, and ignore further complementarity in u^e, $e>E$.

4. Experimental Analysis

From a professional data center we obtained data describing 5-minute averages for CPU and memory workload of hundreds of VMs over multiple months. Most of these workloads exhibit rather deterministic daily patterns without a significant trend. Thus, we consider daily workload profiles. We consider scenarios from 10 to 160 VMs to be consolidated, where each scenario consists of 10 arbitrarily chosen VM subsets. We assumed targets with identical capacity.

In our experiments we analyze *ThinWCM* regarding solution quality (no. of targets) and computational time to solve the model using *SSAPv* as a benchmark. We set E=5 as over 90% of total workload variance was described in the directions of the first 5 eigenvectors. As *SSAPv* with 5-minute intervals (*SSAPv 5 minute*) is intractable for larger problem instances, we solve *SSAPv* additionally for 1-hour (*SSAPv hourly*) and 1-day intervals (*SSAPv daily*). For 1-hour intervals we derive maxima over 12 5-minute intervals and for 1-day intervals workload is represented by its maximum. As mentioned before, the ability to exploit complementarities decreases with increasing interval lengths. Using *SSAPv daily* for each scenario an upper bound I for the number of targets was obtained; solutions of *SSAPv 5 minute* indicated lowest bounds. Calculations were performed on a 2.4Ghz Intel Duo, 4GB RAM using R for SVD calculation and Lp_solve v.5.5 (with defaults) as solver. Figure 3 shows the aggregated experimental results.

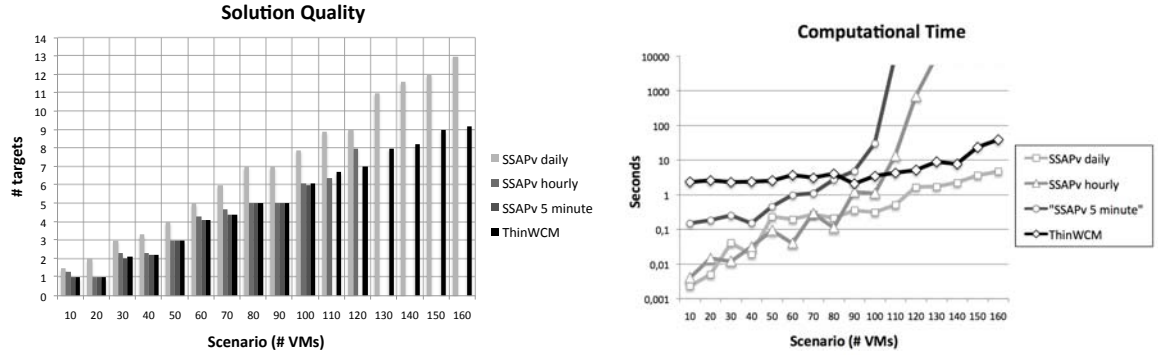

Figure 3: Aggregated Experimental Results

In the diagram on the left-hand side, for each model variant the average number of required targets per scenario is displayed as bar height. Missing bars indicate that no solution was computable within four hours. In most experiments, *ThinWCM* derived the optimal solution and dominated *SSAPv hourly* (and obviously *SSAPv daily*). The graph on the right-hand side of Figure 3 shows, on a logarithmic scale, the average computational time per scenario required to solve a model (for ThinWCM, time to compute the SVD is included). The exact model *SSAPv 5 minute* could be solved for up to 100 VMs, while *SSAPv hourly* could be solved for up to 120 VMs. *SSAP daily* and *ThinWCM* could be solved within a minute even for 160 VMs, with a much higher solution quality when applying *ThinWCM* instead of *SSAPv daily*.

5. Related Work

While there has been a lot of work on capacity planning in IS, little work has focused on efficient server consolidation. Closest in spirit to our work is the work by Bichler et al. (Bichler et al. 2008) and by Rolia et al. (Rolia et al. 2003), both use integer programs to exploit workload complementarities and statistically multiplex resources over time to minimize the amount of targets while ensuring sufficient

capacity in each time interval. They apply approximations such as time-slot coarsening and meta-heuristics such as Genetic Algorithms (GA) to make their solutions computationally tractable. (Rolia et al. 2005) and (Cherkasova et al. 2006) describe an approach based on statistical multiplexing using GA that penalize low target utilizations and target overload to minimize the number of targets. (Seltzsam et al. 2006) also forecast workload profiles to multiplex server resources. (Urgaonkar et al. 2002) analyse best-fit and worst-fit heuristics to bundle complementary services on common servers.

In contrast to our work, we did not find approaches in the literature that apply multivariate statistics like SVD to reduce the dimensionality of the consolidation problem in order to transfer and solve the problem in a low-dimensional space.

6. Conclusion and Outlook

In this paper we introduced *ThinWCM*, a server consolidation model based on truncated SVD of workload data to derive indicators for VM workload levels and complementarities. In first experiments with industry data, *ThinWCM* found the optimal solution in most cases and solved much larger problems than decision models with a comparable solution quality.

To the best of our knowledge, there is now previous work on how to apply multivariate statistics in order to solve IT problems such as server consolidation more efficiently.

In our future research we plan to evaluate larger sets of workload traces and we will explore additional heuristics for server consolidation. Furthermore, as today IT service management suffers from the complexity of handling vast amounts of high-dimensional data, we plan to apply multivariate statistics to dynamically control and visualize data center workload with a few indicators only. In particular, we plan to predict trends and detect workload anomalies that require intervention like moving a VM to another target before an anticipated overload situations occurs.

References

Bichler, M., Setzer, T., and Speitkamp, B. 2006. „Capacity Management for virtualized Servers. In: Proc. Workshop on Information Technologies and Systems (WITS). Milwaukee, 2006.

Bichler, M., Setzer, T., and Speitkamp, B. „Provisioning of Resources in a Data Processing System for Services Requested“, Int. Patent No. PCT/EP2007/063361. Published 06/2008.

Cherkasova, L., and Rolia, J. "R-Opus: A Composite Framework for Application Performability and QoS in Shared Resource Pools". HP Labs. Palo Alto. 2006.

Filani, D., He, J., and Gao, S. 2008. “Technology with the Environment in Mind”, Intel Technology Journal, vol. 12, no. 1.

Garey, M. R., and Johnson, D. S. Computers and Intractability: A Guide to the Theory of NP-Completeness. New York. W.H. Freeman and Company. 1979.

Rolia, J., Andrzejak, A., and Arlitt, M. "Automating Enterprise Application Placement in Resource Utilities". Proc. Workshop on Distributed Systems. Heidelberg. 2003.

Rolia, J., Cherkasova, L., and Arlitt, M. "A Capacity Management Service for Resource Pools", Proc. Int. Workshop on Software and Performance. Palma. 2005 pp. 229-237.

Seltzsam, S., Gmach, D., Krompass, S., and Kemper, A "AutoGlobe: An Automatic Administration Concept for Service-Oriented Database Applications". Proc. 22nd Int. Conference on Data Engineering (ICDE 2006). Atlanta. 2006.

Urgaonkar, B., Shenoy, P., and Rescoe, T. "Resource Overbooking and Application Profiling in Shared Hosting Platforms". Proc. Usenix OSDI. 2002.