Behavioral Model for Cloud-aware Load and Power Management

Kiril Schröder
C.v.O. University of Oldenburg
schroeder@informatik.uni-oldenburg.de

Wolfgang Nebel
C.v.O. University of Oldenburg
nebel@informatik.uni-oldenburg.de

ABSTRACT

Within the last few years, the development of data centers has been moving into high-grade flexible architectures that adapt to the needs (by means of virtualization). This flexibility can be used by load management methods to minimize the energy demand. Depending on quality of service and the hardware used, the application of a load and power management (LPM) results in a big dynamic range of the number of servers currently required. Previous energy models for data centers did not take into account this dynamic sufficiently and thus are not suitable for cloud data centers. Therefore, we present two contributions in this paper. First, we enhance an existing LPM for virtual machines, which has been designed for single data centers, enabling it to interact in flexible environments, for example in inter cloud LPM systems. Second, we develop a model which abstracts the behavior of the LPM concerning the server allocation. This model can be consulted for forecasts and obtains an average precision of 93 %.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Modeling techniques; D.0 [**Software**]: General; F.2.0 [**Analysis of Algorithms and Problem Complexity**]: General; G.1.2 [**Numerical Analysis**]: Approximation—*Linear approximation*

General Terms

Algorithms, Management, Theory

Keywords

Resource Allocation; Virtualization; Modeling; Cloud Computing; Energy Efficiency

1. INTRODUCTION

Within the last few years, the development of data centers (DC) has been moving into high-grade flexible architectures that adapt to the needs. The use of so-called hybrid clouds or inter cloud architectures [31, 12] enables particularly small and medium enterprises to run and offer IT services with less hardware equipment on their own. The high flexibility is beneficial to the use case disaster recovery as well [3].

The virtualization technology is a base for these architectures [8]. The services are no longer operated directly on dedicated servers but encapsulated in virtual machines (VM). In principle, VMs can be operated together with other ones on a single physical server without influencing each other. Hereby, the number of required servers can be reduced, which is referred to as consolidation.

By using the live migration technique [20], load management methods can minimize the energy demand of data centers dynamically. In this so-called dynamic virtualization, only as many servers as currently needed are provided. A large part of business services are run on a single server and not distributed on several servers, due to their low workload [29]. For these services, the dynamic virtualization obtains savings of 40 % to 80 % on average [36, 23, 22] – depending on running services, servers and the specific load and power management (LPM) employed. The resulting energy demand varies considerably [13] in contrast to old and also partly still existing static structures. These exhibit an energy demand that is almost constantly on a peak level, independent of the workload [5].

In literature, different energy models for data centers were introduced [26, 1, 24, 27]. These are generally simplified, but give a good clue to be able to assess changes of the architecture. However, the energy demand dynamic addressed above is only insufficiently taken into account. For this reason, in this paper, we model the typical consolidation behavior of a LPM. By integrating this behavioral model into a data center energy model, the energy demand dynamic caused by dynamic virtualization can be considered.

Previous methods using dynamic virtualization for energy minimization have no behavioral model, and thus the information about the effects of migrations is missing. Therefore, these methods are not able to satisfy local target loads. But this is a prerequisite to enable an ecologically and economically motivated LPM for distributed data centers as, for example, it is described in [25] by the exclusive use of renewable energies.

The scenario, we address, consists of a group of cooperating DCs, which are fully virtualized and run services that can be operated in VMs on a single server. As DCs are energy bulk consumers, the actual electricity demand is highly relevant for the electric energy grid. Shifting the data pro-

HotTopiCS'13, April 20–21, 2013, Prague, Czech Republic.

cessing either temporally or locally also means shifting the electricity demand. Therefore, not only for distributed load optimizations minimizing the costs, but also to stabilize and even to improve the electric energy grid management (examined in [30]), the dynamic of virtualized environments should be known in advance.

The rest of this paper is organized as follows: In Section 2 we present the related work. We point out which data center models and algorithms for dynamic load management of virtual machines exist. Afterwards, we give a brief explanation of the base algorithm, state the problem and introduce several heuristics to handle it. After that, we present our approach to model the typical behavior of the enhanced algorithm. In Section 5, we evaluate our proposed extensions and the developed behavioral model. At the end, we summarize the outcomes of this paper and give an outlook into future work.

2. RELATED WORK

For virtualized server environments, dynamic distribution methods already exist. One distinguishes between reactive and proactive methods. Reactive methods [21, 39, 35] execute migrations when reaching threshold values. For guaranteeing quality of service (QoS), these values have to be quite pessimistic. Here, proactive methods [6, 16] are better suited since they already avoid upcoming bottlenecks beforehand. A problem which can nonetheless appear in the worst case is the so-called deadlock state (see [19]). If no additional server is (temporarily) used to resolve the deadlock, performance losses can occur. In [19], a method is introduced handling both cases, guaranteeing QoS and preventing deadlocks without using additional servers.

The work in [14] introduces how the arising workload should be scheduled for using as much renewable energy as possible. Therefore, batch jobs are scheduled according to the forecasts for renewable energies. These jobs are non critical, i.e. they are allowed to be scheduled at different times. In [2] also critical jobs are included. Besides renewable energies the scheduling can also be executed according to the electricity price (effects operational expense) and the ambient temperature (effects cooling system). These appertain to the location parameters. Inter-site LPMs which optimize loads based on location parameters are introduced in [28, 10, 17, 41]. Although the servers are virtualized in [17], all of these methods refer to services operated in clusters, which allow a more fine granular load management compared to complete services encapsulated within a single VM.

VMware vCloud Director [37] is a commercial product for migrating VMs between different data center sites. In connection with VMware vCloud Connector[1], which handles the update of the network addresses, any services are inter-site movable. Unlike data center internal migrations, higher downtimes of several seconds occur here at present [33, 15, 38]. However, by suitable optimization techniques, such as the delta compression, both the downtimes and the complete migration time can be decimated [32, 40, 4]. With VMware Capacity Planner[2] one can determine how many resources are needed for the operation of certain VMs. Though, this program does not consider any LPM (not even VMware DRS[3]) and thus can only be consulted for the energy demand if no LPM is used.

For an ecologically or economically optimized data center internal as well as inter-site management, it is necessary to figure out how the energy demand and with it the operational costs arise from the workloads. In latest energy models for data centers [26, 1, 24, 27] the hardware components are modeled. However, the consequences of dynamic load management employed in virtualized environments are either missing or handled only insufficiently. In [13], the consequences of different LPM policies are shown by simulations with historical data. The authors in [24] go into the effect of the job scheduler analytically. but they only consider HPC data centers solving the problem to manage tasks within computing clusters. As mentioned before, this can be done more fine granular than with VMs. In the data center model of [27], the consequences of the scheduler are taken into account in an abstracted way. No specific scheduler is used, but it is assumed that the scheduler achieves a certain consolidation rate. It is not discussed if such an abstraction suffices.

The primary use of a LPM behavioral model is to give an estimate of future constellations concerning the VMs to be operated and the future workloads. The model can thus be considered as a forecast method. A good survey of prediction models is given in [7]. In grid computing, linear models are used in general for load balancing due to their simplicity and fast calculation [11]. Here, the forecast periods are usually small since, otherwise, the forecast error rises too strongly. To enable longer forecast periods with acceptable quality, linear regression models can be used, for example for network traffic [34].

3. DYNAMIZATION EXTENSIONS

The core problem addressed in this paper is to develop a behavioral model of the LPM. As mentioned, different LPMs already exist and according to the reasons, delineated in Section 2, we chose the LPM introduced in [19] to be modeled. By reference to the concepts presented there, we reimplemented the LPM. However, the method has to be extended to be able to work in dynamic cloud environments.

In this section, we summarize the major characteristics of the LPM first, and afterwards, we point out the necessary extensions.

3.1 Basic Load and Power Management

The LPM makes a dynamic consolidation possible which (statistically) guarantees at any time that sufficient resources are available. In a first step, a forecast algorithm is consulted, which identifies the periodical behavior of services and which can give workload forecasts. With the help of the forecast data, the LPM can minimize the number of active servers dynamically. Thereto, a so-called safe distribution is formed initially which corresponds to the distribution of the static operation of VMs. In this distribution, each VM is statically allocated to a specific server. It is assumed that the safe distribution provides sufficient resources at any time. Starting from this distribution, additional consolidations based on the dynamic workloads can be made

[1] `http://www.vmware.com/products/datacenter-virtualization/vcloudconnector/`

[2] `http://www.vmware.com/products/capacity-planner/`

[3] `http://www.vmware.com/files/pdf/VMware-Distributed-Resource-Scheduler-DRS-DS-EN.pdf`

at runtime. This leads to dynamic distributions, which are unsafe.

The method comes to the migration decisions in an iterative process: In each timing step the best possible single action is determined, in contrast to a window wise optimization, in which the best sequence of actions in the regarded interval is determined. For that reason, it must be ensured at any time that the safe distribution can be reached just in time again to provide sufficient resources – even in the worst case when all VMs are operated with their predicted maximum workload. A safe way back must always exist, which avoids the deadlock states already mentioned. This is guaranteed by preventing cyclical migration references[4] at the decision-making.

Hence, each server has two sets of allocated VMs (see Figure 3): the statically allocated VMs, which can be operated without resource shortages at any time, and the dynamically allocated ones, which are actually allocated at the moment. For further information, please see [19]. The server, which a VM is allocated on in the safe distribution, is called home server. Referencing the example in Figure 3, the home server of VM1 is server S1 and the home server of VM4 is server S2.

3.2 Problem Definition

At present, the concept does not support dynamic changes of the VM workload profiles and of the set of managed VMs during runtime. When the profile changes over time, there are two issues to be handled: First, the predicted values should be updated, which can be done by performing the prediction method again. Second, the safe distribution should be updated as well because the packing rate might get worse or the assured qualities might be violated. This can also occur when adding or removing VMs to the managed pool.

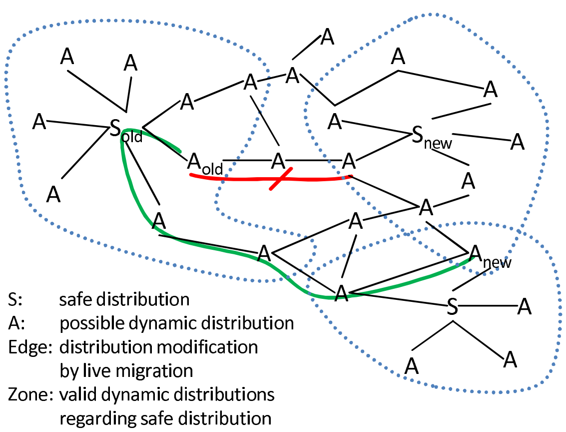

Figure 1: Schematic representation of the transformation of the system into a new safe distribution.

So, the challenge is to change the safe distribution considering the current dynamic distribution. Constraints about timing and the migration references have also to be taken into account. Unlike the dynamic migration procedure, one can assign new static allocations for several VMs at the same time since no data must be moved, but just a (new) logical relationship must be defined.

The relationship of safe (S) and dynamic (A) distributions with the restriction of the safe return is outlined in Figure 1. At any time, several safe distributions are possible. Starting from these, a certain number of changes (migrations) could have been executed, which lead to a dynamic distribution. The dynamic distributions at which the safe way back is guaranteed at the current time are encircled. If another safe distribution is aimed at, in each step to it, a safe way back must be guaranteed. In the given example a valid migration path is shown, which leads across another safe distribution. Since it will normally be impossible to reach the target safe distribution within a single timing step, it is necessary to take into account the temporal behavior. Time passes with every migration, during which the system state changes. Therefore, the path in the example could be no longer valid after executing some migrations.

3.3 Repacking Heuristics

It is a complex problem to transfer the operation from a safe distribution to an arbitrary different one, since temporally changing dependencies might have to be considered. This can quickly increase the computational effort to find an optimal solution (by use of linear programming, for example).

To prevent a worsening of the safe distribution over time, the heuristics introduced in this section can be used. To this, the problem is simplified: an arbitrary safe distribution is no longer provided as a target but a new, "better" safe distribution which also corresponds to the current dynamic distribution. Referencing the example in Figure 1, for a given distribution A, a better safe distribution S_{new} is searched which also contains A in its surrounded region at the moment. That way, it is conceptually made sure that the new distribution is always reachable since it is already valid at the current time.

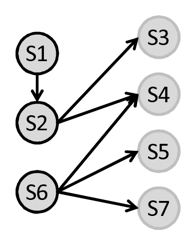

Figure 2: There are no migration references starting from servers which are currently off or in stand-by (grey encircled).

Background Repacking.

During operation, the system will be in a dynamic distribution that differs from the corresponding safe distribution. Because of the dynamic consolidation, several servers will be usually in stand-by. Since no VMs are actively allocated on these servers, no migration references exist between them or to the active servers (see Figure 2, grey encircled). So, the static allocations on the inactive servers can be changed at discretion without cycles arising. To optimize these allocations, an arbitrary offline bin packing method can be employed.

[4]Migration references between servers mean that for reaching the safe distribution one or several VMs must be migrated in the direction of the reference.

3.3.1 Adding of VMs

The problem of adding new VMs is to integrate these into the already existing safe distribution in a way that as few additional servers as possible will be needed. While for the initial computation of the safe distribution an offline bin packing algorithm is used, its online variant can be used here. If several VMs are added at the same time, then an offline algorithm can be applied for this new set as well.

3.3.2 Removing of VMs

Removing a VM leads to free (static) capacities on a single server. If it is an inactive server, then these free capacities can be reallocated by the background repacking. If it is an active server, this heuristic is not applicable. The following heuristic can be used for it.

Direct Repacking.

From the currently dynamically allocated VMs which are not statically allocated to the concerning server, one or more VMs can be determined (for example via the "best fit" algorithm). These VMs can be newly statically allocated to it. As these VMs are currently allocated to the server, no new migration references can arise – only existing references have to be canceled. As a result, it is not necessary to consider possible cycles.

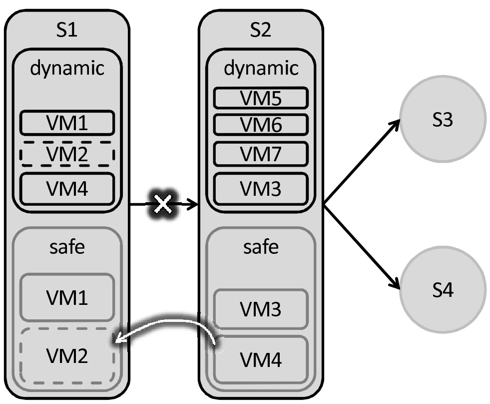

Figure 3: By removing a VM (VM2), free capacities arise for the static allocations (safe distribution).

An example of this procedure is given in Figure 3: VM2 will be removed from the VM pool. Thereby, free capacities for static allocations arise on its home server. VM4 is currently not allocated on its home server S2 but on server S1. As enough static capacities are now on S1, VM4 can be statically allocated on S1. The migration reference from S1 to S2 will diminish within this process.

Recursive Repacking.

If a VM, newly to be assigned, is currently statically allocated on another active server, the problem is shifted to this server since free capacities arise here in turn. In this case, the direct repacking can recursively be applied to this server and to all following ones.

To avoid the effect of the recursive occurrence of further free capacities, such VMs which are statically allocated on an inactive server at the moment should be selected preferably. The resulting free capacities on the inactive servers can be reallocated by the background repacking after all desired VMs are removed from the VM pool.

3.3.3 Change of VM Workload Profiles

If the workload profile of a VM changes over time, then the capacity reserved for the static allocation changes as well. If the reserved capacity decreases, then this corresponds just to the process that a VM was removed from the pool. If it increases and the capacity of the statically allocated server is exceeded, several VMs must be assigned to another server. After the reassignment, free capacities will arise on the current server. This also corresponds to removing VMs. Therefore in both cases, the approach mentioned in section 3.3.2 is applicable here as well.

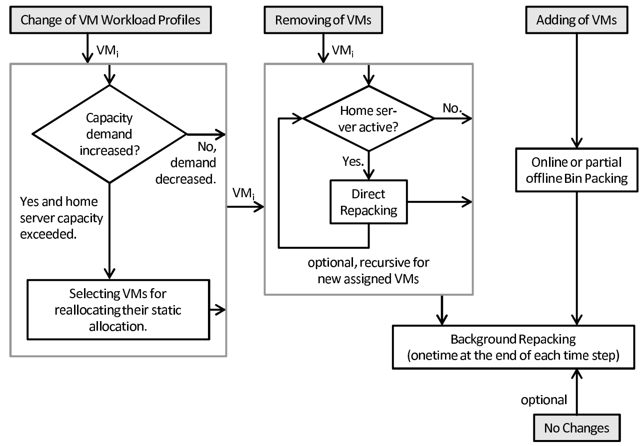

Figure 4: Flow chart for the application of the repacking heuristics for the adjustment and optimization of the safe distribution.

To get a better overview when which heuristic should be used, a corresponding flow chart is given in Figure 4. In principle, the background repacking should be executed after completing the change processes because this has a more global view of the situation than the special heuristics.

4. MODEL OF BEHAVIOR

On account of the high energy demand dynamic, described at the beginning, a future-oriented data center energy model should also consider the consolidation behavior of the LPM. The behavior can be captured by performing simulations and can thus be used for the planning. However, the LPM considered here has a polynomial runtime behavior concerning the number of VMs [18]. This may be practicable for some requests, but not for distributed load optimizations. To receive data about the LPM behavior faster, a model of its behavior has to be implemented.

A solution meeting the requirements represents a linear regression model. This is based on the condition that the chosen variables can also be computed with linear complexity. An autoregressive approach is not followed since using autoregressive models, the occurring error rises significantly with an increasing forecast period [11]. For the regression, the least squared error method is employed.

On account of simplicity, only the CPU value is considered for determining the workload and within the data center, a homogeneous server environment is assumed. In this case, it is not necessary to consider the mapping of loads to the specific servers since, in total, the effects are the same. Therefore, only two energy states for servers must be taken into account: on and off/stand-by. The number of active servers can thus be taken as the dependent variable.

As independent variables, both the data of each single VM and the data which gets aggregated from the complete VM pool are possible. The powers and temporally following values of the variables are viewed in addition. Historical values are not considered since the LPM makes its decisions based on future values.

4.1 Modeling and Analysis

In this section, several variables are examined concerning their influence. Based on this, the regression model will be determined. In the following examinations, the boundary conditions, which must be taken into account to be able to use the model for forecast purposes, are pointed out.

For the executed simulations, a pool of 10000 different VMs, which base on real VM workload data[5], were available. Most of the VM workload profiles are shaped like the ones, shown in Figure 5: periodic with load peaks. A period of one week can often be observed. The data has a time resolution of 5 minutes and the utilization value is scaled to a normalized server, which is used for the entire evaluation.

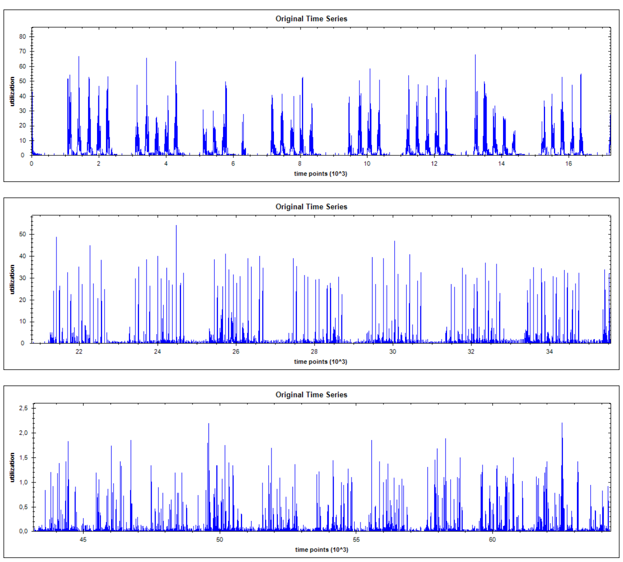

Figure 5: Exemplary VM workload profiles used for evaluations.

Altogether, 100 different scenarios were generated, which were simulated over 10 days. These scenarios differ in the number of used VMs and the specific selection of VMs: 10 different numbers (10 to 200 VMs) x 10 different VM selections. The values of the maximum (relative) difference, the average (relative) (absolute) difference, and the standard deviation were evaluated as error metrics.

4.1.1 *Selection of Variables*

The first examination shows which influence the variables have on the regression quality. Each evaluated regression model was created with the complete data set which had been generated by the simulations.

Using only the workload values or the number of VMs already delivers expedient results (average error less than 10 %). By the combination of the variables, better results can be achieved up to an average error of 7 %. It has to be noticed that this error refers to single points in time. The energy demand is normally viewed for a time period and thus the average error should be built over several points in time. However, this metric would not be meaningful here, since the average error for the values used in the regression process has an expectation value of zero.

In a next step, the temporally following values of the variables were consulted for the regression. To this, both the number of additional values and the temporal distance (step width) were varied. Rising the number and the step width respectively leads to better results. But the more values are taken into account, the more complex the modeling and the later computational effort will become. And rising the step width entails that appropriately long workload forecasts have to be valid.

Third, we examined the impact of using powers of the variables. Up to a power of third degree, considerable improvements can be obtained. However, following evaluations have shown that this leads to an over-fitting of the model. This means that although the supporting values, which were used for the regression process, can be modeled better, the generated model is less suitable for generating predictive values.

According to the results, the following two regression models are defined:

$$\#SRV_a(t) = \alpha_0 + \sum^{s} \left(\begin{array}{l} \alpha_{1,s} \cdot SoL(t+s) + \\ \alpha_{2,s} \cdot \#VM(t+s) + \\ \sum_{i=0}^{9} (\alpha_{3+i,s} \cdot \#VMC_i(t+s)) \end{array} \right)$$

$$s \in \{0\} \qquad (1)$$

$$s \in \{0, 5, 10, 15, 20, 25\} \qquad (2)$$

- $[\#SRV_a]$ the number of active servers,
- $[SoL]$ the (added up) overall workload / utilization of all VMs, scaled to the capacity of a normalized server,
- $[\#VM]$ the number of VMs in the VM pool,
- $[\#VMC_i]$ the number of VMs which belong to a certain workload class at the moment (ten classes with 10 % increment) – this value represents a tradeoff in the detail level of the variables and it shall represent the variability of the loads –,
- $[\alpha_i]$ regression parameters determined by the regression process.

In model (1), only the values for the respective time t are used. In model (2), for each of these values, there are additionally five temporally following values used with a time distance of up to 25 min., indicated by the index s.

[5]NOWIS Nordwest-Informationssysteme GmbH & Co. KG

4.1.2 *Training Length*

In this analysis we examined how many supporting points are necessary to get a certain forecast period with adequate quality. Best results were achieved with a training length of 24 hours or more. As long as the forecast period lasts only few hours, also short training lengths suffice to get good results (average error less than 7 %).

Comparing the forecast quality of the models (1) and (2), it can be noticed that the regression model (2) provides worse results at training lengths below one day, in range of some hours even considerably worse. The additional variables lead to an over-fitting at these short time periods. Only if the training length is at least one day, its forecast quality is superior.

4.1.3 *Effect of Changes*

Next, it should be pointed out how large the training length must be in order to get good predictive results at changes (adding or removing VMs). It was also investigated which error is caused by a specific level of change.

The results for the average relative error show: the more VMs are added or removed, the bigger the error gets. This also holds for rising training lengths, but the increase of error diminishes. As in the case of the training length examination (4.1.2), the regression model (2) proves only at longer time periods to be the better one. To restrain the error, particularly at shorter forecast and training lengths, no VM pool changes of over 50 % should be executed.

4.2 Alternative Regression Model

In [27], the following formula is introduced for calculating the required number of servers:

$$\widetilde{\#SRV_a} = U + (1 - U) \cdot l, \quad U, l \in [0, 1]$$

The detail of the server number is scaled to the server number required at most, at a maximum overall utilization U. Parameter l is a measure for the concentration or spread of the loads over the servers in the entire data center.

In the LPM used here, the maximum (added up) utilization of all VMs is identified by the forecast procedure. Deriving from that, the number of servers needed for the safe distribution, which indicates the maximum predicted total utilization, can be calculated. But this calculation can only be done after completing changes, which has a non linear computing complexity. Thus, a capacity limit according to the VM number can be used instead. The corresponding linear regression model is:

$$\#SRV_a = \alpha_0 + \alpha_1 \cdot SoL + \alpha_2 \cdot (\#VM - SoL) \qquad (3)$$

5. EVALUATION

In the last section, we determined the behavioral model of the LPM using linear regression. To prevent increased error rates because of several changes and to consider the actual situation, this model should be generated anew or should be updated[6] whenever needed.

[6]for example by "boosting" [9]

In this evaluation, two issues will be assessed. First, which quality does the regression model have in the (simulated) live operation, under the boundary conditions defined above? Second, which impact causes the application of the introduced repacking heuristics on the safe and the dynamic distributions? The evaluation was done based on simulations. Therefore, a pool of 10000 different VMs, which base on real VM data (see Section 4.1), were available.

5.1 Forecast Quality

For determining the forecast quality, a total of 100 different simulation runs were executed. Each simulated 10 days and contained a VM pool of 300 VMs (initial 150 VMs, at most 300 VMs). To simulate a dynamic in the VM set, an event list of operations was defined. The operations were scheduled every 4 to 8 hours. Each operation added or removed VMs. According to the above mentioned boundary condition, each change was up to 50 % of the VMs contained at a single moment. The simulation runs were different in the choice of the VM pool and in the event lists (ten different ones respectively). The allocation estimate always referred to the interval until the next change, which corresponds to a 4 to 8 hour forecast.

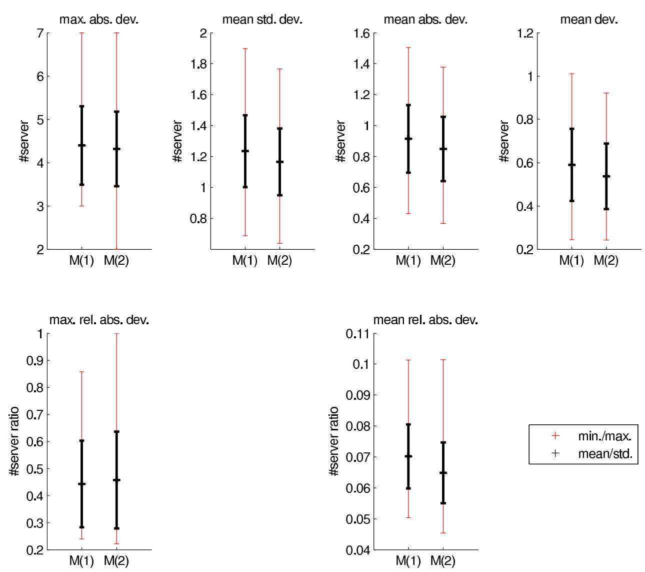

Figure 6: Forecast quality of the regression models in the simulated live operation.

The results for the models (1) and (2) are illustrated in Figure 6 by six error metrics. For each metric and model the mean, standard deviation, minimum, and maximum values for all simulation runs are given. In principle, it should be noted that adding temporally following values (model (2)) causes only low improvements. The most meaningful metric is the "compensating" average error (mean dev.) since this describes the difference in the entire viewed prediction interval. The average error of 0.6 servers shown here corresponds to a precision quality of 95 %. At a point by point consideration, the precision quality decreases to 93 %.

Model (3) was evaluated in the same way. We used the server number of the safe distribution as well as the maximum VM number as capacity limit. With both values, the model performs an average 10 % worse in all metrics. A combination with the introduced models does not lead to any improvements.

5.2 Impact of the Repacking Heuristics

To evaluate the impact of the repacking heuristics on the safe and the dynamic distributions, the simulation scenarios for the evaluation of the forecast quality became a reference. The following six heuristics and heuristic combinations, which were used respectively at these scenarios, are compared to:

- [*b*] background repacking,
- [*cb*] continuous background repacking (execution at any time, whether changes occur or not),
- [*d*] direct repacking,
- [*db*] direct and background repacking,
- [*dr*] direct and recursive repacking,
- [*drb*] direct, recursive, and background repacking.

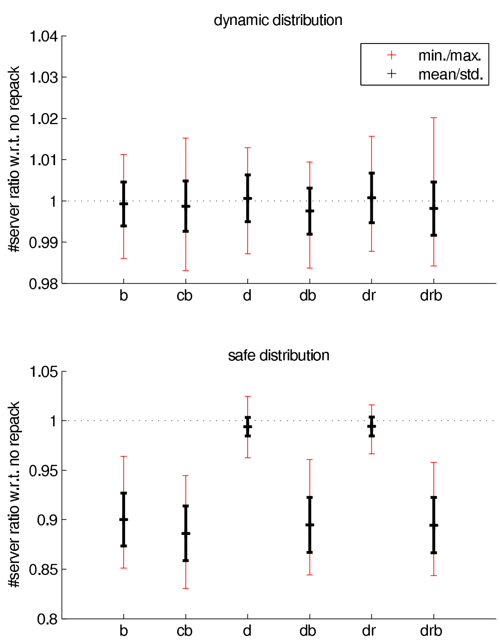

Figure 7: Impact of the different repacking heuristics on the number of servers required in the dynamic and safe distribution.

The results of this examination are illustrated in Figure 7. For each heuristic (combination) the mean, standard, minimum, and maximum deviation with respect to no repacking are given. The values refer to entire prediction intervals. Using a point to point consideration would have led to falsifying outliers. That is because the LPM usually has different migration decisions when different repacking heuristics are applied.

The number of servers reserved for the safe distribution can be reduced up to an average of 10 %. The number of servers required dynamically is almost equal. So, an important result is: Using different safe distributions, has nearly no influence on the efficiency of the dynamic process.

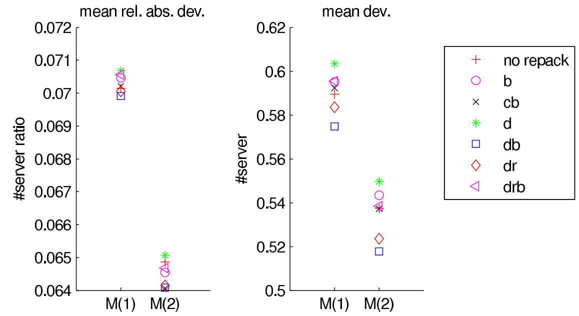

Figure 8: Impact of the heuristics on the predictability of the LPM behavior: marginal.

We also examined the impact of the heuristics on the predictability of the LPM behavior, shown in Figure 8. As in Figure 6, the results for estimating the number of servers required in the dynamic distribution are illustrated, but only by two error metrics and only the mean value is shown. The error values differ only a little. Thus, the predictability is not perceptibly influenced using different heuristics.

Although the heuristics achieve improvements for the safe distribution, nearly no improvements can be achieved for the dynamic process. Therefore, even a method which optimally solves the addressed repacking problem will not be able to obtain considerable improvements here.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented several heuristics to handle some drawbacks of an existing LPM. Now, the regarded VM pool can change at runtime what is a requirement to be able to work in a dynamic cloud environment. Those changes can also occur if the LPM is consulted as a part of a superordinate load management where loads are moved between data centers. The effects of such moves have to be known beforehand. Therefore, as main contribution, we have presented a LPM behavioral model that can be used to give an estimate about the number of needed servers and, in conjunction with a DC power model, the total energy demand can be estimated as well. In an extensive, simulation based evaluation performed, it has been shown that the model attains an average precision of 93 %.

For simplification reasons, only the CPU utilization was used for the dynamic resource allocation. This shows the greatest possible volatility concerning the actively required servers. Adding more resource types in the allocation process, like memory utilization and network IO, leads to lower change rates. First examinations show that this improves the quality of the behavioral model on some sets of VMs. We want to analyze this phenomenon more deeply, conceivably with characterizing VM profiles.

At present, the model includes only the allocation effects of the LPM. In future, the LPM behavioral model shall be embedded into a data center energy model, which considers heterogeneous servers as well as cooling and infrastructure, to point out the actual energy demand dynamic caused by the LPM.

7. REFERENCES

[1] Z. Abbasi et al. Thermal aware server provisioning and workload distribution for internet data centers. In *HPDC*, 2010.

[2] B. Aksanli et al. Utilizing green energy prediction to schedule mixed batch and service jobs in data centers. *SIGOPS Oper. Syst. Rev.*, 45(3), Jan. 2012.

[3] T. Aoyama and H. Sakai. Inter-Cloud Computing. *BISE*, 3(3), 2011.

[4] M. Beevor et al. Disaster Recovery White Paper: Reducing the Bandwidth to Keep Remote Sites Constantly Up-to-date. Whitepaper, DataCore, Riverbed, waterstons, 2008.

[5] R. Bianchini and R. Rajamony. Power and Energy Management for Server Systems. *Computer*, 37(11), Nov. 2004.

[6] N. Bobroff et al. Dynamic Placement of Virtual Machines for Managing SLA Violations. In *Integrated Network Management*, 2007.

[7] G. E. P. Box et al. *Time Series Analysis: Forecasting and Control*. fourth edition, 2008.

[8] R. Buyya et al. Market-oriented cloud computing: Vision, hype, and reality for delivering IT services as computing utilities, in. In *CSSE*, 2008.

[9] D.-S. Cao et al. The boosting: A new idea of building models. *Chemometrics and Intelligent Laboratory Systems*, 100(1), Jan. 2010.

[10] K. Church et al. On Delivering Embarrassingly Distributed Cloud Services. In *HotNets*, 2008.

[11] M. Dobber et al. A prediction method for job runtimes on shared processors: Survey, statistical analysis and new avenues. *Perform. Eval.*, 64(7-8), Aug. 2007.

[12] J. Erbes et al. The future of enterprise it in the cloud. *Computer*, 45, 2012.

[13] D. Gmach et al. Capacity planning and power management to exploit sustainable energy. In *CNSM*, 2010.

[14] I. n. Goiri et al. GreenSlot: scheduling energy consumption in green datacenters. In *SC*, 2011.

[15] E. Harney et al. The efficacy of live virtual machine migrations over the internet. In *VTDC*, 2007.

[16] F. Hermenier et al. Entropy: a consolidation manager for clusters. In *VEE*, 2009.

[17] T. Hirofuchi et al. A multi-site virtual cluster system for wide area networks. In *LASCO*, 2008.

[18] M. Hoyer. *Resource Management in Virtualized Data Centers Regarding Performance and Energy Aspects*. PhD thesis, C.v.O. University of Oldenburg, 2011.

[19] M. Hoyer et al. Proactive dynamic resource management in virtualized data centers. In *e-Energy*, 2011.

[20] C. C. Keir et al. Live Migration of Virtual Machines. In *NSDI*, 2005.

[21] G. Khanna et al. Application Performance Management in Virtualized Server Environments. In *NOMS*, 2006.

[22] D. Kusic et al. Power and performance management of virtualized computing environments via lookahead control. *Cluster Computing*, 12(1), Mar. 2009.

[23] D. Meisner et al. PowerNap: eliminating server idle power. In *ASPLOS*, 2009.

[24] T. Mukherjee et al. Spatio-temporal thermal-aware job scheduling to minimize energy consumption in virtualized heterogeneous data centers. *Comput. Netw.*, 53(17), Dec. 2009.

[25] K. K. Nguyen et al. Renewable energy provisioning for ICT services in a future internet. In *The future internet*. 2011.

[26] E. Pakbaznia and M. Pedram. Minimizing data center cooling and server power costs. In *ISLPED*, 2009.

[27] S. Pelley et al. Understanding and Abstracting Total Data Center Power. In *WEED*, 2009.

[28] A. Qureshi et al. Cutting the Electric Bill for Internet-Scale Systems. In *ACM SIGCOMM*, 2009.

[29] J. Rolia et al. Automating Enterprise Application Placement in Resource Utilities. In *DSOM*, 2003.

[30] D. Schlitt et al. Analysis of Attainable Energy Consumption Reduction in ICT by Using Data Center Comprehensive Load Management. In *The Economics of Green IT, Workshop*, 2010.

[31] B. Sotomayor et al. Virtual Infrastructure Management in Private and Hybrid Clouds. *IEEE Internet Computing*, 13, 2009.

[32] P. Svärd et al. Evaluation of delta compression techniques for efficient live migration of large virtual machines. In *VEE*, 2011.

[33] F. Travostino et al. Seamless live migration of virtual machines over the MAN/WAN. *Future Gener. Comput. Syst.*, 22(8), Oct. 2006.

[34] S. Vazhkudai and J. M. Schopf. Using Regression Techniques to Predict Large Data Transfers. *Int. J. High Perform. Comput. Appl.*, 17(3), Aug. 2003.

[35] A. Verma et al. pMapper: power and migration cost aware application placement in virtualized systems. In *Middleware*, 2008.

[36] A. Verma et al. Server workload analysis for power minimization using consolidation. In *USENIX*, 2009.

[37] VMware. VMware vCloud Architecting a vCloud. Whitepaper, VMware, 2012.

[38] W. Voorsluys et al. Cost of Virtual Machine Live Migration in Clouds: A Performance Evaluation. In *CloudCom*, 2009.

[39] T. Wood et al. Black-box and gray-box strategies for virtual machine migration. In *NSDI*, 2007.

[40] T. Wood et al. CloudNet: dynamic pooling of cloud resources by live WAN migration of virtual machines. *SIGPLAN Not.*, 46(7), Mar. 2011.

[41] Y. Zhang et al. GreenWare: greening cloud-scale data centers to maximize the use of renewable energy. In *Middleware*, 2011.