Wirtschaftsinformatik Proceedings 2007 Wirtschaftsinformatik

February 2007

Sizing Considerations for Enterprise Applications in Dynamic Data Centre Environments

Heiko Thimm
University of Applied Sciences Kiel, heiko.thimm@fh-kiel.de

In: Oberweis, Andreas, u.a. (Hg.) 2007. *eOrganisation: Service-, Prozess-, Market-Engineering*; 8. Internationale Tagung Wirtschaftsinformatik 2007. Karlsruhe: Universitätsverlag Karlsruhe

ISBN: 978-3-86644-094-4 (Band 1)
ISBN: 978-3-86644-095-1 (Band 2)
ISBN: 978-3-86644-093-7 (set)

Sizing Considerations for Enterprise Applications in Dynamic Data Centre Environments

Heiko Thimm

Institute of Business Information Technology
University of Applied Sciences Kiel
24149 Kiel
heiko.thimm@fh-kiel.de

Abstract

Dynamic data centres are regarded as promising approach to achieve a high degree of resource utilization. In such data centres, applications are dynamically allocated to servers under consideration of the actual resource demand and the actual load states of the servers. However, as a result of this flexible deployment model, it is difficult to size the initial servers and other hardware equipment. We present considerations for this specific problem and a corresponding sizing method that is based on a heuristic algorithm. For a given set of applications, our algorithm determines the number of required servers, the subset of applications to be allocated to each server, and the corresponding runtime characteristics of each server. The maximum load that needs to be considered for each of these servers is optimized from a global data centre perspective. This is achieved through a smart rule based orchestration of the individual applications' load profiles.

1 Introduction

The performance of an enterprise application depends substantially on the performance capacity of the underlying IT infrastructure. Henceforth, application performance requirements of the business, such as the application response time, can only be satisfied if the application is deployed on a well sized IT infrastructure.

As the experience with enterprise computing during the last decade shows, the backend servers of enterprise applications, such as database servers, have to be regarded as single most critical

IT infrastructure component from a hardware sizing point of view. Normally, the corresponding hosts on which these backend servers are being deployed belong to the most expensive data centre components. Therefore, whenever an existing application landscape is modified, e.g. extended by new applications, a corresponding sizing project needs to be completed. Such sizing projects usually follow the general approach to determine the required performance capacity of the hosts mainly based on workload assumptions. It is common business practise for traditional data centre environments to derive the required performance capacity from the expected peak load of the application(s). The resulting performance capacity usually includes a safety charge to compensate the fuzziness of this sizing method.

It is well known that this sizing practise results into poor server utilization, a fact that conflicts with the recent trend of increasingly shrinking corporate IT budgets. In the search for solutions to this problem of underutilized hosts, new technologies such as virtualization and capacity management techniques have been developed. These new technologies enable the implementation of data centres which are not based anymore on a persistent 1:1-allocation of applications to hosts. Under the notion of *dynamic data centres* we broadly subsume this new type of data centre. In dynamic data centres, applications may be dynamically allocated to hosts that belong to a pool of shared servers. This approach enables very flexible deployment options under consideration of the applications' actual resource demand and the actual server performance capacity at runtime. Therefore, available capacity management solutions for dynamic data centres monitor the system load and gather corresponding load data. Through an analysis of this load data, it is possible to obtain insights that are helpful to achieve a high degree of server utilization by dynamic resource allocation actions.

In our opinion, specialized sizing methods for dynamic data centres are required that already in the hardware planning phase take the "built-in" deployment dynamics of dynamic data centres into consideration. We could not find existing work that is geared at such sizing methods. In our research, we focus on this gap of knowledge. We strive on the investigation of effective sizing approaches and, in the long run, on the development of corresponding sizing tools for dynamic data centres. In this paper, we present our initial considerations for this research agenda that are largely based on our experience with sizing traditional data centre servers. In addition to that, we propose a sizing method that makes use of a heuristic algorithm. First, this algorithm obtains a base allocation of applications to hosts by grouping applications with similar characteristics together. An own dedicated host is allocated to each of these groups of applications and also to those applications that definitely need to be deployed on a dedicated server. Each host's runtime properties are determined from the characteristics of the applications being allocated to the host. Then, in a next step, performed are optimization operations across the host-specific application sets. These operations re-orchestrate the sets of applications to make use of complementary load patterns among the applications. The performance capacity being required for a server may be minimized through such operations. In the near future, we will study the effectiveness of our algorithm through a simulation study.

The rest of the paper is organized as follows. Section 2 presents main aspects of traditional hardware server sizing for enterprise applications. Section 3 contains key characteristics of dynamic data centres from a sizing point of view. In Section 4.1, we first discuss considerations for the problem of server sizing for dynamic data centres. Then, in Section 4.2, we present various methods to predict the required performance capacity that are specialized to the different deployment modes found in dynamic data centres. Section 4.3 contains an informal description of our sizing method. Related work is discussed in Section 5 and our conclusions are given in Section 6.

2 Traditional Hardware Server Sizing for Enterprise Applications

Observations about traditional sizing practice. In several years of experience in sizing many different enterprise applications such as SAP R/3, Siebel CRM, and Oracle Applications, we observed that, in principle, sizing methods are usually composed of three steps.

First, the load profile of the targeted production system is determined. Typically, this is done through an assessment of the various activities to be performed by the system such as user activities, background jobs, and the system's own bookkeeping activities. In rare cases only, the load profile is derived through performance tests with a real system because of the large amount of efforts required for such experiments. Given the load profile, the peak load of the production system is determined and further considered in the next step.

In the second step, the performance capacity required by the hardware server on which the application will be deployed is predicted from the peak workload of the system. This predicted performance capacity is expressed either in application specific terms such as the number of SAPS in case of SAP R/3 [LoMa03] or in other terms, that are related to standard performance benchmarks. In case of OLTP applications, it is often referred to TPC-C which is the OLTP benchmark of the *Transaction Processing Performance Council (TPC).*

In the third step, the performance relevant server configuration is obtained from the performance capacity resulting from the previous step. Usually system engineers make use of hardware specific performance specification data and corresponding configuration recommendations. For example, such a guideline may describe that for a specific application the RAM size should be 2 GB per CPU of a particular CPU type.

Sizing risk. Traditional sizing methods as described above may be regarded as "fuzzy methods" because they mainly work with assumptions. By the concept of *sizing risk* we address this "fuzzy nature" of these methods. We use the notion of sizing risk to express the probability that the server may be overloaded at some point in time in the future during the production usage phase. In overload situations, the actual performance capacity of a server is lower than the performance capacity needed to satisfy the given application specific service levels such as response time for interactive users. It is possible to lower the sizing risk by extensive studies of the future deployment characteristics so that very accurate sizing assumptions are made available. However, such extensive studies are usually limited by cost and time constraints, respectively. In practice, often it is dealt with the sizing risk by the consideration of a safety charge that is added to the required performance capacity. It is assumed that this extra safety charge will compensate load generating activities ignored by the sizing method otherwise. Obviously, also the consideration of a safety charge is limited by given cost constraints.

Types of traditional sizing projects. It is possible to classify sizing projects into different types depending on the broader project context.

By *initial sizing project*, we refer to the case where the targeted application is deployed for the *first time at all* within the organization (hence the term *initial*). That is, there does not exist any experience in the deployment of the new application so that the sizing assumptions are relatively wage. As a consequence, a relatively high sizing risk needs to be considered for this type of sizing projects. For risk mitigation, it has been recommended to complete performance experiments with a corresponding test system.

In some cases, it is possible to leverage pre-existing deployment experience in initial sizing projects. For example, consider a hardware platform switch for an existing application landscape. Due to more accurate sizing assumptions, usually, such sizing projects need to deal with only a minor sizing risk as compared to sizing projects without pre-existing deployment experience.

In so-called *upgrade sizing projects*, it is necessary to determine the additional performance capacity needed by an application system already running in production mode. Typical causes for such projects include an increase of the number of application users, the implementation of functional extensions, and release upgrades. Such projects are typically a subject of capacity management for which dedicated tools are available. For example, such tools allow to monitor, simulate, and analyse the work load based on actual load data and to predict the extra performance capacity needed. Henceforth, usually only a low sizing risk is to be considered in upgrade sizing projects.

Discussion. As a result of the above described sizing practice, traditional data centres suffer from a low degree of server utilization. For the decision makers this situation presents a dilemma because they only have a choice between a high sizing risk and a low degree of server utilization. Increasingly more attention to this problem has been paid for the last several years due to the cost pressure that IT departments need to deal with. In the search for solutions to this problem, approaches have been developed that different technology providers call “dynamic IT”, “dynamic infrastructure”, “adaptive infrastructure”, “dynamic data centre”, or “adaptive computing”. These initiatives share all the same idea to enable a flexible dynamic resource management and some self-management capabilities in order to provide a cost-effective and adaptable IT infrastructure. In this work, we broadly subsume these approaches under the notion of *dynamic data centre*. In the next section, we describe the general principles of dynamic data centres from a sizing point of view.

3 Key Characteristics of Dynamic Data Centres from a Sizing Perspective

In traditional data centres, enterprise applications are deployed typically on exclusive hosts. That is, the hardware servers are considered as exclusive computing resources for only a single application. They are not regarded as shared resources that may run multiple enterprise applications at the same time in a shared mode as it has been the case in mainframe computing environments. The consideration of an exclusive host usually leads to an installation procedure where the application software is combined with the server in a relatively radical way. For example, often the IP address of the host is hard-coded in the configuration files of an application. As a consequence, a separation of the application software from the server at a later point in time is very hard to accomplish. Thus, usually it requires a lot of efforts to move such an application from one server to a different one.

Through the use of virtualization techniques, dynamic data centres are capable to work without such a persistent assignment of application software to underlying hardware servers. The servers are presented to the applications as pooled resources that may be deployed flexibly by the applications according to different deployment models. This enables to allocate applications to a given server only for a certain period of time and to re-allocate the application later to a different server. Furthermore, some applications may even be forced into a planned downtime mode. For example, during a high load phase, the application may be deployed on a server with a high performance capacity. From this server, the application may be moved to a less powerful server for a different time period where the application load is only low.

This flexibility allows for dynamic data centres to allocate applications to available hardware servers dynamically (hence the term *dynamic* data centre) under consideration of the actual resource demands of the applications and the actual load states of the servers. In several initiatives (e.g. [GSWK05]) concepts are investigated for a central management instance that is capable to automatically schedule and manage such dynamic re-allocation actions.

The re-allocation of applications, however, leads to some negative effects such as extra costs, an increasing risk for system failures, and application down time. Therefore, re-allocation actions should not occur with a too high frequency.

From a hardware server sizing view it is necessary to differ between different kinds of resource sharing models that may occur in dynamic data centres. In the following we present three different models.

Exclusive resource sharing deployment mode. If applications are deployed according to this mode, only one application may run on a given host at a time. That is, the complete performance capacity of the host is available for exclusive usage by only one application at a given point in time. However, it may occur that the application is moved to another server or put on hold in order to allow another application to be (exclusively) deployed on the same server.

Non-exclusive resource sharing deployment mode. In the non-exclusive sharing mode, the data centre servers are shared by multiple applications at a time. Each of these applications consumes a certain share of the servers total performance capacity.

Mixed resource sharing deployment mode. This type of deployment mode presents a combination of the previously two mentioned modes. For some periods of time, the server is deployed exclusively by only a single application, while during other periods of time multiple applications are deployed in parallel. Note that this may include the case where an application that has been deployed exclusively on a server from some point in time on will be accompanied (at the same server) by further applications. That is, at this mentioned point in time, it is switched from an exclusive deployment mode into a non-exclusive deployment mode.

4 Sizing Considerations for Dynamic Data Centre Environments

It is envisioned that dynamic data centres may be capable to allocate and re-allocate applications to computing resources autonomously without any participation of human system administrators, in the future. However, in today's available solutions, the allocation task is still controlled by the data centre personal. These solutions mainly build on the existence of load data gathered in the production usage phase. This load data is analyzed and the results are used to derive allocation plans.

For the task of sizing initial data centre servers, however, such load data obviously is not available. Among other reasons, this has lead to the fact that today it is still searched for an effective approach for sizing initial servers for dynamic data centre environments. The inherent property of such environments, that the allocation of applications to servers are dynamically changing over the time, presents one of the crucial problems for this effort.

Our research strives on the investigation of such sizing approaches and on the development of corresponding sizing tools in the long run. As starting point for the development of a first approach, we identified the general considerations presented in Section 4.1. Given this basis, we devised a set of methods to predict the required performance capacity and a first heuristic sizing method presented in Section 4.2 and 4.3, respectively.

4.1 General Considerations and Requirements

Optimization towards a high server utilization. It needs to be addressed that dynamic data centres are designed specifically to allow for a high utilization of the computing resources. Therefore, it is required to reflect the different deployment modes of Section 3. For example, consider a given number of applications that are to be deployed on a single server in non-exclusive mode. The demanded performance capacity should not be obtained by simply adding together the peak loads of the individual applications' load profiles. In reality, the resulting total peak load will only occur in the very rare worst case situation where all individual applications' peak loads occur at the same point in time in parallel. Usually, relatively simple optimizations, e.g. by orchestrating the individual load profiles on a common time scale in an interlocking mode will lead to better sizing results. In order to achieve this optimization, it is necessary to explore thoroughly the expected load patterns of all applications.

Frequency of re-allocation operations. It is no question that the dynamic allocation of applications to servers also provides some drawbacks which has also been described in [GSWK05]. Each time when an application is re-allocated from one server to another one or put on hold, respectively, some performance capacity is bound to these extra operations. The extra load of these operations may lead to distortions and, possibly, even into an instable state of the data centre. Furthermore, each dynamic re-allocation involves the risk of application service failures, even if the same operation was completed successfully many times in the past[1]. In order to prevent these drawbacks, it is necessary to limit the re-allocation frequency. A thorough analysis of the available re-allocation options is necessary which will include a careful prediction of the short term and long term effects of the re-allocation operations. For sizing projects this calls for a starting allocation that does not need to be revised through re-allocation operations in an early stage (i.e. shortly after production start).

Cross application specific aspects. As presented in Section 2, for sizing projects, a sizing risk needs to be considered. For traditional data centres, this sizing risk may be viewed separately for each single server. Due to the fact that in dynamic data centres applications are flexibly deployed on different servers, it is recommended to also look at the data centre as a whole from a risk investigation point of view. That is, for dynamic data centres, the sizing risk needs to include the single-server specific risks but also the cross-servers specific risks. For example, consider the fact that if a server is not sized properly, a high re-allocation frequency is likely to occur. For reasons described above, such a high re-allocation frequency will affect the inappropriately sized server, but the other servers, too.

[1] Consider in this context one of the system administrators' golden rule "Never change a running system".

4.2 Predicting the Required Performance Capacity Based on Load Schedules

The required performance capacity usually presents the key constraint for sizing IT infrastructure components. For dynamic data centres, it is necessary to predict this required performance capacity for multiple interdependent servers. In the following, we present straightforward prediction methods for each of the deployment modes described in Section 3. In our description, by A, we denote the set of applications that are to be deployed on a single host with $A=\{A_1, A_2,\ldots, A_n\}$ and A_k denoting a single application with $A_k\in A$ and $k\le n$. By $S_j(A,t)$, we refer to the aggregated total load profile resulting from a particular orchestration j of individual application load profiles. Considered in an individual orchestration are the load profiles of the applications in A for the time interval t with $t=[t_s, t_e]$. In the following, we refer to such an orchestration by the notion of *schedule*. By $S_{A,t}$, we denote the set of $j=1,\ldots,m$ alternative schedules $S_j(A,t)$ that may be orchestrated with respect to A for time interval t. We assume that this orchestration problem may be solved by function $SCHED(A,t)$ that takes as input the set of corresponding applications A and the time interval t, respectively, and yields the corresponding alternative schedules $S_{A,t}$. In addition to that, we define a function $OPT-SCHED(S_{A,t})$ that finds within the set of alternative schedules $S_{A,t}$ that schedule ${}^{opt}S_{A,t}\in S_{A,t}$ which leads to the lowest total peak load. The further functions considered in our framework are as follows:

- $L_{\max\_app}(A_k,t)$: function that computes the peak load of application A_k wrt. t
- $L_{glob\_\max\_app}(A,t)$: function that finds the max. peak load among the set of applications A and wrt. t
- $L_{\max\_sched}(S_j(A,t))$: function that computes the max. peak load of schedule $S_j(A,t)$
- $L_{glob\_\max\_sched}(S_{A,t})$: function that computes the max. total peak load wrt. set of schedules $S_{A,t}$
- $P(L)$: function that computes the performance capacity required to satisfy load L

The three prediction methods described below share a common initial step where a time interval $t=[t_s, t_e]$ is determined. The size of interval t is defined so that the load profiles of all applications given by A are included in t.

Exclusive sharing deployment mode. Recall that in this mode, the set of applications given by A are deployed on the same server but the server is running only one of these applications at a time. It is possible to predict the required performance capacity for the given server in two steps. First, with respect to the $i=1,...,m$ applications, the maximum peak load is obtained by $L_{glob\_\max\_app}(A,t)$. For this maximum peak load, the corresponding required performance capacity is determined through $P(L_{glob\_\max\_app}(A,t))$.

Non-exclusive sharing deployment mode. In this mode, the server is shared by several applications at a time. The different long term load profiles of the applications may be orchestrated together so that the peak load of the resulting schedule will be minimal. Based on this general idea, we propose a prediction method that consists of the following steps. First, the set of alternative schedules $S_{A,t}$ is obtained by $SCHED(A,t)$. Then, the schedule ${}^{opt}S_{A,t}$ is found that leads to the lowest peak load through $OPT-SCHED(S_{A,t})$. In turn, the maximum peak load of schedule ${}^{opt}S_{A,t}$ is determined by $L_{glob\_\max\_sched}({}^{opt}S_{A,t})$ and the corresponding required performance capacity is obtained by $P(L_{glob\_\max\_sched}({}^{opt}S_{A,t}))$.

Mixed-mode deployment mode. A server that runs applications in mixed-mode deployment mode, at predefined points in time, will switch from exclusive sharing into non-exclusive sharing and vice versa, respectively. Therefore, for the sizing task both of these deployment modes need to be addressed, for example as follows. First, the set of all applications A is divided into two subsets A_{ex} and A_{ne}, respectively. By A_{ex}, we denote that subset of applications that are to be deployed *exclusively* during a set of time intervals t_{ex}. By A_{ne}, we refer to those applications that are to be deployed *non-exclusively* during a set of time intervals t_{ne} with $t_{ne}=t-t_{ex}$. Then, the prediction method for the exclusive deployment mode is applied to A_{ex} and the prediction method for the non-exclusive deployment mode is applied to A_{ne}. From the resulting two numbers that each express required performance capacity, the larger value is to be considered as the final performance capacity necessary for the deployment of all mixed-mode applications.

4.3 Towards an Algorithm for Initial Sizing Projects for Dynamic Data Centres

Based on the above described considerations, we devised a first pragmatic sizing approach for dynamic data centres. To present this approach, the definitions given in Section 4.2 are extended as follows:

- α : set of all applications to be deployed with $\alpha = \bigcup_{i=1}^{l} \alpha_{e_i} \cup \alpha_{se} \cup \alpha_{ne} \cup \alpha_m$ and α_{e_i} the $i = 1,...,l$ applications that need to be deployed definitely in *exclusive mode,* α_{se} the set of applications that may be deployed in a *special exclusive mode,* α_{ne} the set of applications that may be deployed *non-exclusively*, and α_m the set of applications that may be deployed in *mixed-mode*
- B: base allocation with $B = \{\langle H_{e_i}, \alpha_{e_i}\rangle, \langle H_{se}, \alpha_{se}\rangle, \langle H_{ne}, \alpha_{ne}\rangle, \langle H_m, \alpha_m\rangle\}$ and $\langle H_{e_i}, \alpha_{e_i}\rangle$ the $i = 1,...,l$ hosts H_{e_i} on which the $i = 1,...,l$ applications α_{e_i} are deployed in *exclusive mode,* $\langle H_{se}, \alpha_{se}\rangle$ the single host H_{se} on which the set of applications α_{se} are deployed in a special *"semi-exclusive mode",* $\langle \mathrm{H}_{ne}, \alpha_{ne}\rangle$ the single host H_{ne} on which the set of applications α_{ne} are deployed in *non-exclusive mode,* $\langle H_m, \alpha_m\rangle$ the single host H_m on which the set of applications α_m are deployed in *mixed-mode*
- $S_{\alpha_{se},t}, S_{\alpha_{ne},t}, S_{\alpha_m,t}$: sets of alternative schedules for each of the application sets $\alpha_{se}, \alpha_{ne}, \alpha_m$, respectively, computed by function $SCHED(A,t)$
- ${}^{opt}S_{\alpha_{se},t} \in S_{\alpha_{se},t}, {}^{opt}S_{\alpha_{ne},t} \in S_{\alpha_{ne},t}, {}^{opt}S_{\alpha_m,t} \in S_{\alpha_m,t}$: single schedules - computed by function $OPT-SCHED(S_{A,t})$ - where the peak load is minimal

Our approach, that takes the interdependencies between the applications into account, consists of the following steps:

1. The short and long term deployment characteristics of all applications given by α are explored and described in a sizing information repository which will include per

application the corresponding load profile, deployment constraints, and service level requirements.

2. From the sizing information repository, the base allocation B is determined under consideration of all relevant constraints.

3. The servers given in B are sized separately.

For steps 2 and 3, we developed a first version for a heuristic algorithm that looks as follows:

1. Select for each single application in α a proper deployment mode by an evaluation of the sizing information repository under consideration of the following rules. *R1.1:* Consider the *exclusive deployment mode* as a pre-selection for applications that need to run on an own dedicated server. If for such an application the peak load occurs frequently such as shown in the example of Figure 1 (left diagram) select the (strict) exclusive deployment mode. For the converse case, where the peak load occurs only rarely and where substantial periods of idle time exist such as in the other example of Figure 1, select the special exclusive deployment mode. *R1.2:* Select the *mixed-mode deployment mode* for applications that only at some specific points in time need to run on an own dedicated server. *R1.3:* Select the *non-exclusive deployment mode* for applications which may all the time run on a host together with multiple other applications.

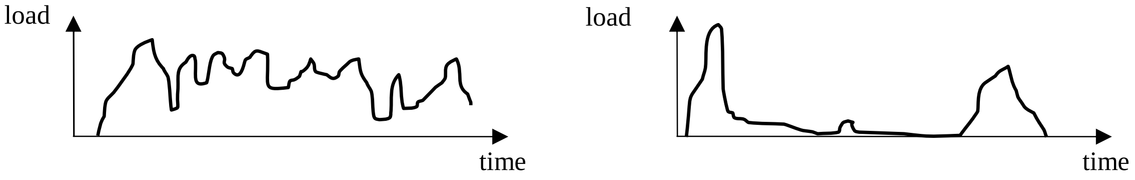

Figure 1: Load profile of an application to be considered for strict exclusive deployment (left side) and special exclusive deployment (right side).

2. Obtain base allocation $B' = \{\langle H_{e_i}, \alpha_{e_i}\rangle, \langle H_{se}, \alpha_{se}\rangle, \langle H_{ne}, \alpha_{ne}\rangle, \langle H_m, \alpha_m\rangle\}$ being an initial allocation according to the following rules. *R2.1:* Consider a separate dedicated host H_{e_i} for each of the $i = 1,...,l$ applications that are to be deployed in (strict) exclusive mode. *R2.2:* Consider a single common host H_{se} for all applications together that are

to be deployed in the special exclusive mode. *R2.3:* Choose a single common host H_{ne} for all those applications together that may be deployed non-exclusively. *R2.4:* Choose a single common sever H_m for all those applications together that may be deployed in mixed-mode.

3. Select a time interval t, s.t. the load profiles of all applications given by α_{se}, α_{ne}, and α_m in B' are included in $t = [t_s, t_e]$.

4. Compute the sets of alternative schedules $S_{se,t}, S_{ne,t}, S_{m,t}$ for the application sets α_{se}, α_{ne}, and α_m through the use of function $SCHED(A,t)$. Next, obtain the schedules ${}^{opt}S_{se,t} \in S_{se,t}, {}^{opt}S_{ne,t} \in S_{ne,t}, {}^{opt}S_{m,t} \in S_{m,t}$ through the optimization function $OPT-SCHED(S_{A,t})$.

5. Obtain optimized base allocation B from B' by modifying the application sets and schedules. The principles of this optimization approach are shown in Figure 2. It is attempted to close potential gaps in the schedule ${}^{opt}S_{ne,t}$ and ${}^{opt}S_{se,t}$, respectively. By "filling such gaps" with fitting load profiles that belong to applications in ${}^{opt}S_{m,t}$ the peak load of schedule ${}^{opt}S_{m,t}$ may be reduced. To formulate this optimization principle, we introduce the notion of *transfer operation* denoted by T_i with $T_i = T_i(A_k, \alpha_m, \alpha_{ne}, \alpha_{se})$. We define a transfer operation to move an application $A_k \in \alpha_m$ from its current source host H_m into either α_{ne} or α_{se} of the destination host H_{ne} or H_{se} if the above described optimization criterion is met. This criterion may be formulated as two post conditions for transfer operations as follows:
(1) $L_{glob\_\max\_sched}\left(SCHED\left((\alpha_m - A_k), t\right)\right) < L_{glob\_\max\_sched}\left(SCHED(\alpha_m, t)\right)$
(2) $L_{glob\_\max\_sched}\left(SCHED\left((\alpha_{ne} \cup A_k), t\right)\right) \le L_{glob\_\max\_sched}\left(SCHED(\alpha_{ne}, t)\right)$
or $L_{glob\_\max\_sched}\left(SCHED\left((\alpha_{se} \cup A_k), t\right)\right) \le L_{glob\_\max\_sched}\left(SCHED(\alpha_{se}, t)\right)$.

   It is possible that several alternative transfer operations T_i exist, i.e. $i = 1,...,q$. For these cases, we propose to select that T_i among the q different alternatives where the resulting peak load reduction of schedule ${}^{opt}S_{m,t}$ in relation to the peak load of the application A_k reaches the maximum value. That is, that T_i is chosen where $A_k \in \alpha_m$ yields the maximum value among all the alternatives for:

$$\frac{L_{glob\_\max\_sched}\left(SCHED(\alpha_m ,t)\right) - L_{glob\_\max\_sched}\left(SCHED((\alpha_m - A_k),t)\right)}{L_{\max\_app}\left(A_k ,t\right)}$$

Note that after a transfer operation is completed, update operations are required for the application sets that have been modified. That is, it is necessary to re-compute the sets of alternative schedules and the schedule that leads to the minimal peak load for α_m and also either α_{ne} or α_{se} .

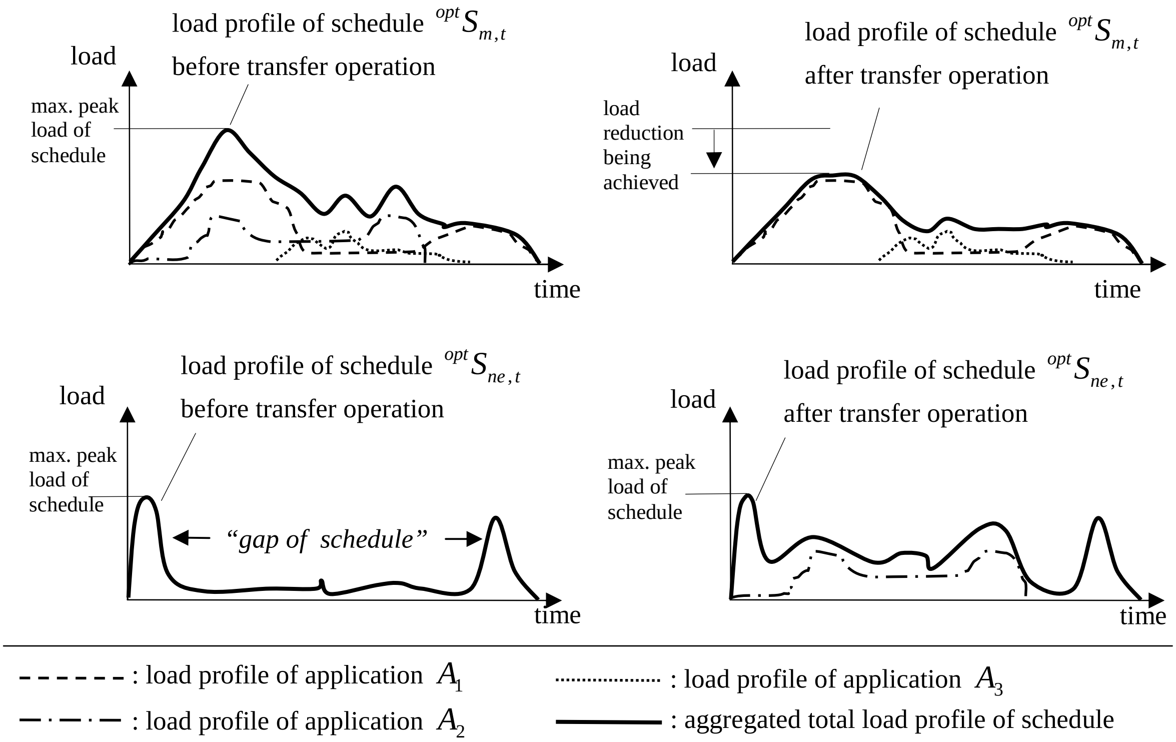

Figure 2: Optimization by transfer operations. The left side shows the global load schedule ${}^{opt}S_{m,t}$ (upper corner) and ${}^{opt}S_{ne,t}$ (lower corner), respectively, prior to a transfer operation. The transfer operation will move application A_2 into the gap of the global schedule ${}^{opt}S_{ne,t}$. As a result, the maximum global load of ${}^{opt}S_{m,t}$ will be reduced while the maximum load of ${}^{opt}S_{ne,t}$ will remain unchanged.

6. Predict the required performance capacity of all servers given by the base allocation B through the methods presented in Section 4.2. For the i=1,...,l servers H_{e_i} use the

prediction method of the exclusive sharing deployment model. For the servers H_{se} and H_m, respectively, use the method of the mixed-mode deployment model. For the server H_{ne} use the method of the non-exclusive sharing deployment model.

7. Configure each server according to the predicted performance capacity.

5 Related Work

The problem of how to deal from a resource management point of view with the dynamics of a set of applications has been addressed previously. In the AutoGlobe Project of the Technical University Munich [GSWK05], concepts for the static and dynamic allocation of computing services are studied. This project aims at an adaptive computing infrastructure that includes advanced self-management services. In the AutoGlobe approach, a static allocation optimization is proposed that makes use of aggregated historic load data. Similar to our approach, in this optimization it is attempted to allocate services with complementary resource requirements on a common server. However, most research on the problem of resource allocation that can be found in the literature is focused on dynamic allocation techniques, e.g. to deal with overload situations or system errors. In the AutoGlobe Project, a fuzzy controller is proposed that handles such situations by corrective actions that are deduced through a rule based approach under consideration of the actual load situation. Online load measurements for dynamic resource allocation are also considered in [ChGS02]. The load measurements are combined with different prediction and resource allocation techniques in order to dynamically vary the resource shares in shared data centres to the changing workloads of applications. A so called predictive controller and various prediction algorithms for dynamic resource allocation in enterprise data centres are proposed in [XZSW06].

In [ThiKl96] an adaptation mechanism is proposed for distributed Multimedia Database Systems that may dynamically adapt concurrent multimedia presentations to fluctuating network bandwidth. This mechanism makes use of the simplex method to globally optimize the adaptations so that the maximum presentation Quality of Service (QoS) is achieved under consideration of the individual user QoS. The difference of our project to these research projects is that we look at the allocation issue for multiple interdependent servers from a sizing perspective.

Our work is also related to system configuration and performance modelling research. In [AbRW01], a systematic method to find a satisfactory hardware and software configuration of a distributed message converter system is presented. Using layered queuing network models, a solution is described that distributes different jobs to different hosts and also configures the processes on the hosts. In addition to this general task that is related to the initial sizing task in our work, we also need to consider the aspect of dynamic resource allocations which was not necessary in this project. A mathematically based method for configuring distributed workflow management systems is proposed in [GWWK00]. This method is targeted at meeting the application's demands in terms of performance and availability while aiming to minimize the total system costs. Similar to our work, it is considered that it may be necessary to adapt the configuration over time due to changes of the workflows. The mathematical core of the proposed method consists of Markov-chain models that are derived from the application's workflow specifications. From these models the overall system's performance is derived. In contrast, we predict the required system performance mainly from the load profiles of the applications. The proposal for a large-scale network parameter configuration method presented in [YeTK02] shares with our approach that efficient parameter state space search techniques are required in order to optimize the allocation of applications to servers. In this related work and also in [XLRX04], finding an optimal configuration is formulated as a black-box optimization problem. For our long term research goal, which is the development of innovative sizing tools for dynamic data centres, we will also evaluate if techniques may be applied to our sizing problem that have been originally developed for configuring mechanical and electronic products [KrHG02].

6 Conclusions

In this paper, we have presented the current status of our research on effective sizing methods for dynamic data centres. We are investigating such approaches in order to develop effective and reliable sizing tools for such data centres in the long run. For the near future, we expect a growing demand for such sizing tools because dynamic data centres are becoming more and more popular.

We presented the main result of our current work status, which is a method for effective initial server sizing. This method leads to a set of hosts with specific deployment characteristics and a corresponding set of applications that fit to these characteristics. The maximum load that may occur at each of these servers is optimized from a global data centre perspective. Due to the fact that our method makes use of a heuristic algorithm there is no guarantee that our method will yield the global optimum.

In a next step, we will evaluate our algorithm through a simulation study. Based on the simulation results, we will further develop and refine our sizing method. This will include more concrete definitions for the rule-based selection of proper deployment modes. These definitions will also address the concept of service levels qualities. Through the simulation study, we also expect to get insights about the proper dimension of the time interval considered in our algorithm. Our future work will also include a concrete specification of the scheduling functions applied in our algorithm and the functions for the various search tasks such as the identification of gaps in load schedules. We expect that for these issues standard algorithms are readily available or may be adapted to our specific purpose. Furthermore, in our future research, we will extend our sizing method to allow users to guide and to influence the sizing proposal generation. This will include the concept of costs, e.g. for performance capacity. This will also include lower and upper bounds for the number of servers per application class and the performance capacity of the servers. For example, one may use this option to guide the sizing proposal generation towards specific needs and preferences, respectively, defined for the data centre equipment. Moreover, we want to allow that users may influence the destination server that is considered in transfer operations.

References

[AbRW01] Risse, T., Aberer, K., Wombacher, A., Surridge, M., Taylor, S.: *Configuration of Distributed Message Converter Systems*, Performance Evaluation, Vol. 58, Issue 1, Oct. 2004, Elsevier, pp. 43-80

[ChGS02] Chandra, A., Gong, W., Shenoy, P.: *Dynamic Resource Allocation for Shared Data Centres Using Online Measurements*, Proc. Quality of Service - IWQoS 2003, 11th Int. Workshop, Berkeley, CA, USA, in Springer LNCS 2707, pp. 381-400

[GWWK00] Gillmann, M., Weissenfels, J., Weikum, G., Kraiss, A.: *Performance and Availability Assessment for the Configuration of Distributed Workflow Management Systems*, Proc. of the 7th Int. Conf. on Extending Database Technology 2000, Konstanz, Germany, in Springer LNCS 1777, pp. 183-201

[GSWK05] Gmach, D., Seltzsam, S., Wimmer, M., Kemper, A.: *AutoGlobe: Automatische Administration von dienstbasierten Datenbankanwendungen*, 17th Int. GI Conf. on Database Systems for Business, Technology, and Web, Karlsruhe, Germany, February 2005

[KrHG02] Krebs, T., Hotz, L., Günter, A.: *Knowledge-based Configuration of Configuring Combined Hardware/Software Systems*, Proc. 16 Workshop Planen, Scheduling und Konfigurieren (PuK 2002), Freiburg

[LoMa03] Lober, B., Marquard, U.: *Anwendungs- und Datenbank-Benchmarking im Hochleistungsbereich von ERT-Systemen am Beispiel von SAP*, Datenbank-Spektrum 7/2003, dpunkt.verlag Heidelberg, S. 6-12

[ThiKl96] Thimm, H., Klas, W.: *Delta-Sets for Optimized Reactive Adaptive Playout Management in Distributed Multimedia Database Systems*, Proc. 12th Int. IEEE Conf. on Data Engineering, March 1996, New Orleans, LO, USA, IEEE Computer Society Press, pp. 584-592

[XLRX04] Xi, B., Liu, Z., Raghavachari, M., Xia, C., Zhang, L. : *A Smart Hill-Climbing Algorithm for Application Server Configuration*, Proc. ACM WWW 2004, May 2004, New York, NY, USA, ACM Press, pp. 287-296

[XZSW06] Xu, W., Zhu, X., Singhal, S., Wang, Z.: *Predictive Control for Dynamic Resource Allocation in Enterprise Data Centres*, Proc. of the 10th IEEE/IFIP Network Operations & Management Symp. (NOMS 2006), April, 2006, Vancouver, Canada

[YeTK02] Ye, T., Kaur, H., Kalyanaramann, S.: *Large-scale network parameter configuration using online simulation*. Tech. Rep., Rensselaer Polytechnic Institute, 2002, currently also under review in IEEE Transactions on Networking

Einführung in den Track

Modellierung als Innovationsmotor

Prof. Dr. Ulrich Frank
Universität Duisburg-Essen

Prof. Dr. Robert Winter
Universität St. Gallen

Die Gestaltung und Verwaltung komplexer Systeme erfordert geeignete Abstraktionen. In den Ingenieurwissenschaften ist dies seit langem bekannt. Aber auch in der Betriebswirtschaftslehre werden vielfältige Modelle von Unternehmen eingesetzt, um Gestaltungs- bzw. Veränderungsentscheidungen zu unterstützen. In der Wirtschaftsinformatik kommt Modellen insofern eine besondere Bedeutung zu, als sie nicht nur eine Grundlage für den Entwurf betrieblicher Informationssysteme darstellen, sondern darüber hinaus ein Medium schaffen, um eine zielgerichtete Zusammenarbeit zwischen IT-Experten, Domänenexperten und Anwendern zu unterstützen. Modelle der Unternehmensstrategie, der Geschäfts- und Produktionsprozesse sowie des unterstützenden Informationssystems sind damit wesentliche Voraussetzung für ein effektives IT-Management sowie für die Planung und Realisierung innovativer Formen des IT-Einsatzes.

Der Track ist darauf gerichtet, die zentrale Rolle der Modellierung zu verdeutlichen und ihren angemessenen Einsatz in der Praxis zu fördern. Dazu sollen nicht nur Modellierungskonzepte berücksichtig werden, sondern auch kritische Erfolgsfaktoren für deren wirtschaftliche Anwendung in der Praxis.

Programmkomitee:

Prof. Dr. Hans-Jürgen Appelrath, Universität Oldenburg

Prof. Dr. Jörg Becker, Universität Münster

Prof. Dr. Werner Esswein, Technische Universität Dresden

Prof. Dr. Ulrich Frank, Universität Duisburg-Essen

Prof. Dr. Norbert Gronau, Universität Potsdam

Prof. Dr. Dimitris Karagiannis, Universität Wien

Prof. Dr. Gerhard Knolmayer, Universität Bern

Prof. Dr. Susanne Leist, Universität Regensburg

Prof. Dr. Heinrich C. Mayr, Universität Klagenfurt

Prof. Dr. Markus Nüttgens, Universität Hamburg

Prof. Dr. Erich Ortner, Technische Universität Darmstadt

Prof. Dr. Michael Rebstock, Fachhochschule Darmstadt

Prof. Dr. Elmar J. Sinz, Universität Bamberg

Prof. Dr. Klaus Turowski, Universität Augsburg

Prof. Dr. Robert Winter, Universität St. Gallen