

Capacity Management Is Crucial in Virtualized Environments

Without the Right Tools, IT Service Optimization Is Tricky

Server virtualization for commodity servers is one of the most rapidly adopted technologies of all times. The traditional and less effective “one physical server, one application” approach is giving way to pools of resources being shared among virtual servers. Not only will you be able to save resource in terms of power, cooling and floor space in the data center, you can also save money by avoiding some of the over-provisioning that accompanies a one physical server, one application strategy. And virtualization offers a more flexible solution where resources can be redistributed in accordance with shifting demands.

So, the concept of server virtualization has some obvious advantages. But as with many things in life, there is more than one side to it. Virtualization of resources adds a new abstraction layer to the infrastructure stack. This layer is supposed to make virtualized solutions flexible and more efficient, but it also complicates IT Service Optimization. For demanding and performance-critical applications, this can lead to disruptions and violation of service level agreements. When implementing or refreshing a virtualization strategy, it's important to keep this in mind.

In this paper we will discuss how best to optimize IT services in complex virtual environments.

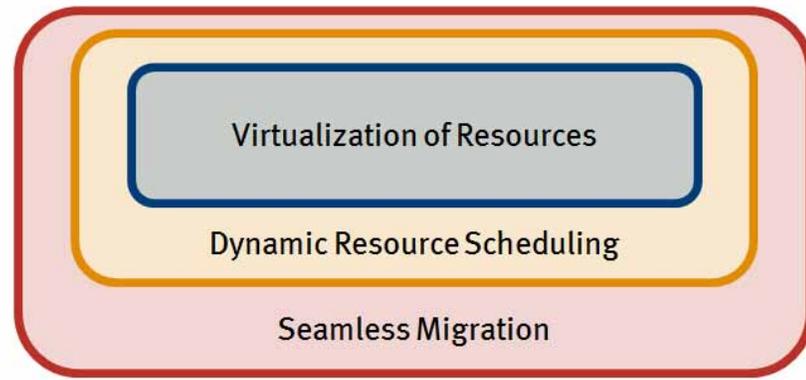


About the Author

Per Bauer is the Technical Account Manager at TeamQuest Corporation EMEA. Per supports key customers across Europe, Middle East and Africa with technical assessments and managing large implementation projects. Prior to joining TeamQuest, Per worked for a global ERP-vendor for 10 years, holding a position as manager for worldwide QA.

Concepts

When discussing the challenges that server virtualization imposes on capacity management and IT Service Optimization, we need a common understanding of the mechanisms that comprise a flexible virtualized environment. Conceptually, these are the three mechanisms used:



1. Virtualization of Resources — grouping resources into pools that are assigned to different virtual machines by a hypervisor.
2. Dynamic Resource Scheduling — the ability of the hypervisor to automatically assign and direct resources to virtual machines based on their current demand.
3. Seamless Migration — the ability to do resource scheduling across separate physical servers, enabling the implementation of clusters or farms of servers acting as one large resource pool.

By using these mechanisms, some of the traditional capacity management challenges in computing environments can be addressed. In the “one app, one physical server” era for commodity platforms, you normally had to provision for the peak demand of an application. If the normal level of activity was considerably lower than that of a peak, a majority of the resources sat unused most of the time. To make matters worse, the peaks could be weeks or months apart depending on the cyclicity of events. In a virtualized environment, the extra headroom needed to cover for peak usage can be shared among multiple different virtual machines and across several physical hosts, all this in a way that is totally transparent to the applications running inside the virtual machines. Combined with demand management activities aimed at optimizing the timing of the peaks, dynamic resource scheduling can be exceptionally effective at fighting excessive over-provisioning.

But are server virtualization and dynamic resources the silver bullet that will remove the need for capacity management altogether? Not really, in the next section we will discuss why.

The Challenge

A typical server virtualization project starts off by picking the low hanging fruits first. Commodity applications with a low level of sustained activity are migrated into a virtual environment. Since those commodity workloads typically run on older infrastructure offering less performance and because most of them are probably not particularly demanding in terms of resources anyway, an initial boost in performance and service quality is experienced. This affirms the decision to migrate workloads to virtual servers and focus is put on reducing the number of physical hosts more than anything else.

Eventually when all the low hanging fruits have been picked, it's natural to continue the process and involve more complex workloads. Even though server virtualization is supposed to save money in the long run, implementing a virtualization strategy normally requires a number of initial up-front investments in new infrastructure. In many cases the IT department is faced with a situation where in order to achieve sound and timely return on investment, those up-front investments need to be shared among a larger group of applications and services. This leads to more demanding workloads such as email, database and ERP systems being considered. They are more demanding in terms of resource consumption, and also in terms of criticality. Both the availability and the throughput of such applications are of high importance. They might be there to support core processes of the organization or be directly exposed to customers. This means that you need to be particularly careful with how you operate and manage these services. Making sure that you have sufficient capacity now and in the foreseeable future is implied.

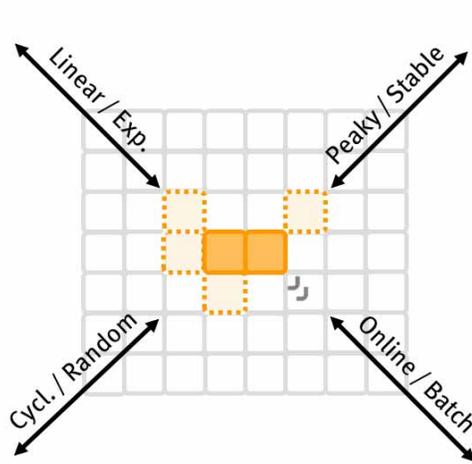
One important input to capacity management is proper characterization of each workload or application. When profiling a workload, some aspects are vitally important:

- Utilization of Resources — what resources will this workload use and to what extent?
- Online vs. Batch — is the workload used in an online scenario where the response time of individual transactions is crucial or is it rather a batch scenario where the total throughput is more important?
- Linear vs. Exponential — how will the workload perform when exposed to an increase in intensity?
- Peaky vs. Stable — will the workload always have the same level of activity or will there be occasional bursts of increased activity?
- Cyclical vs. Random — are those peaks of activity predictable or do they seem to happen randomly?

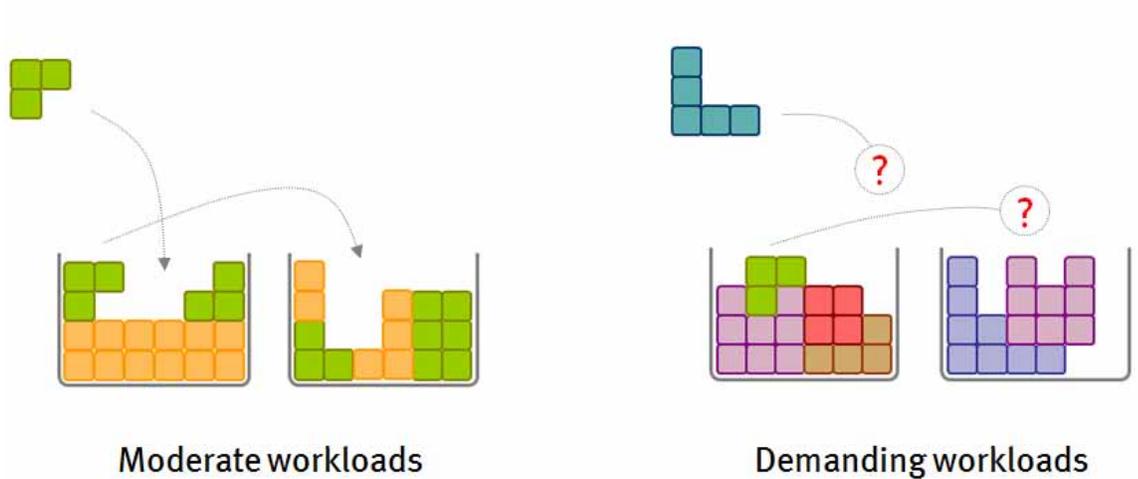
A profile combining all this information will give you a good idea about how you need to manage a specific workload and what is needed in terms of resources; the “personality” of a workload. When you start to mix multiple workloads on the same physical resources (which is probably one of the reasons why you virtualized in the first place), things get a bit more complicated. As mentioned before, dynamic resource scheduling is often used as the single mechanism to address shifts in demand for capacity, whether it's within one physical host or across multiple hosts. But the efficiency and usefulness of resource scheduling is very much dependent on the characteristics of the workloads. A group of workloads with moderate need for resources

are normally quite easy to balance and migrate across different hosts. But if you start to mix them with more demanding workloads, your options suddenly become severely restricted.

An analogy that I originally presented at the 2009 Gartner Data Center Summit in London seems to be gaining a lot of traction as a way to illustrate the problems that can arise when trying to mix and match various workloads. In that presentation I suggested using Tetris-like blocks to symbolize the irregularities among different workloads. A simple workload with moderate need for resources would be represented by a basic two-piece block. Higher needs for resources and higher complexity would cause the block to expand in various directions.



The larger and less symmetric the blocks get, the harder it is to combine them. Inability to combine the blocks (like in the illustration to the right below) translates into workloads that are starved for resources and yet can't be migrated to another host. All of a sudden, two of the key mechanisms that enable flexibility in virtualized environments become unusable. And even if you were able to combine them, large asymmetric blocks will most likely lead to white space fragmentation and lower resource utilization than you calculated and planned for.



A service optimization strategy for virtualized environments built on dynamic resource scheduling and seamless workload migration alone assumes complete mobility and freedom to mix workloads. In many situations, this turns out to be too much of a simplification. In addition to having different “workload personalities” standing in the way, a number of other constraints imposed by technologies or by the business environment will have an impact (specialized or proprietary hardware configurations, contractual obligations, security policies regarding data segmentation, change management procedures, etc.).

All in all, this demonstrates the need for a more thorough approach to implementing an enterprise class virtualization strategy. To allow for more demanding and important workloads to be virtualized, it’s not enough to single-handedly rely on reactive resource scheduling and workload migration mechanisms. To ensure that services are delivered optimally, the placement and migration policies for workloads need to be investigated and analyzed prior to deployment.

What are the options?

The different methods for predicting the capacity requirements of a workload and how well it will coexist with other workloads can roughly be divided into three categories.

Estimation

Methods in this category rely heavily on “common sense” and previous experience. The method of choice for server virtualization and consolidation scenarios is typically to stack workloads on top of each other until a predetermined threshold is reached. But in order to be successful at that, you need to fully understand aspects like:

- Which metrics are relevant for the assessment?
- What are the correct thresholds for those metrics?
- How do I normalize metrics across different platforms or platforms of different age?
- How can I account for non-linear performance changes as workloads are stacked?

If you get any of this information wrong, you will make incorrect predictions.

Using estimation, the quality of the prediction is based on opinions and gut feeling rather than hard facts. Tools in this category are focusing on making the procedure simpler, pretending that the Tetris-like blocks have fewer dimensions than they actually have. At the end of the day the quality of the prediction still relies on how well the questions above get answered. And in most cases some aspects, such as non-linear growth, are not even addressed.

Analytical Modeling

Another method for predicting how well mixed workloads will perform is through the use of an analytic queuing network solver. Queuing network models of the systems are described so that an analytic model solver can mathematically calculate where and how much queuing delay will occur. Before the model is used for predictive purposes, it is calibrated based on empirical studies of how servers have been used in reality.

All objects that have importance to the performance of a server are represented in the model that describes the system.

Once the model has been built, different scenarios can be evaluated by changing transaction intensity or moving workloads between different models. The predicted response time or throughput of the transactions tells you whether the scenario was successful or not. It's possible to focus on the relative difference between time spent queuing versus time spent working in the system in order to remove the need for explicit thresholds for every transaction type. A simple rule of thumb regarding that relationship will give you a good idea about the state of the configuration scenario.

This type of analytical modeling offers a predictive, rapid and repeatable process for optimizing mixed workload environments. With analytic modeling the quality of the result is less dependent on the individuals executing it. It also avoids the common mistake of assuming that performance will degrade linearly as workloads are stacked.

Synthetic Load Testing

The goal here is to produce synthetic transactions that mimic real life scenarios as close as possible. To get things right you need to closely examine the operating environment to find the right mix of transactions and their concurrency, develop repeatable test cases based on those transactions and run lengthy performance tests on equipment identical to the production environment (which might force you to invest in a parallel test environment). Plus you need to define success criteria in terms of response times for each transaction type.

Load testing offers a high level of accuracy if done right, but in most cases the cost and long test cycles can't be justified. It might be better suited for "once in a lifetime" quality assurance activities prior to going live with a critical service, not recurring IT Service Optimization exercises.

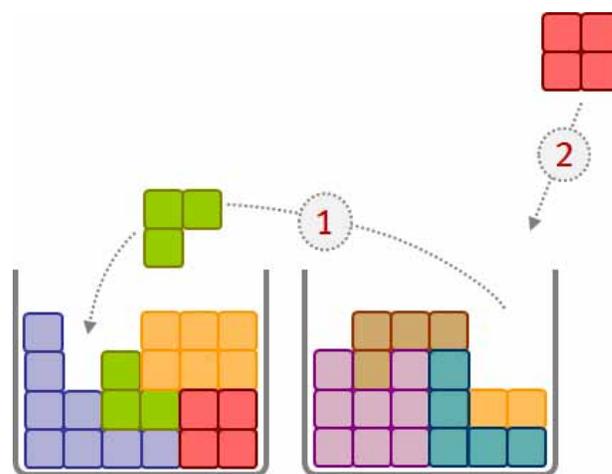
So which of the above methods should you pick? Most important of all is to at least have a strategy and not just rely on reactive mechanisms of the framework to take care of capacity management. After that has been established, the optimal choice is often a mix of different methods. You probably don't want to spend too much time analyzing less important utility applications, just as you can't afford the risk of a quick and oversimplified analysis of a business critical service. It's important to have a comprehensive toolbox that lets you choose the right method for different circumstances.

Conclusion

Commodity server virtualization vendors would like you to believe that the reactive performance management technologies built into their platforms is all you need to attain optimal performance. The truth is, reactive technologies such as dynamic resource scheduling and migration are helpful, but they are not a complete solution and they do not necessarily make IT Service Optimization easier. The added complexity from virtualized environments actually makes it more difficult to be sure that you are getting everything you can from your systems.

A “big enough” virtual pool of resources does not come without an effort. If you don’t pay attention to the characteristics of the workloads you’re hosting, there’s a big risk you’ll either over-provision or starve your applications.

To be sure that workloads will operate efficiently with each other (or that the analogous Tetris-like shapes will be capable of fitting snugly with one another) requires some serious analysis. You need to understand a lot about the resource requirements of the underlying applications in order to know whether one workload might interfere with another creating unnecessary resource contention and queuing delays. So, especially for critical workloads, it makes sense to do some careful up front analysis rather than simply tossing various workloads onto your systems hoping that dynamic resource scheduling and migration will be able to quickly and efficiently fit everything together for you on the fly.



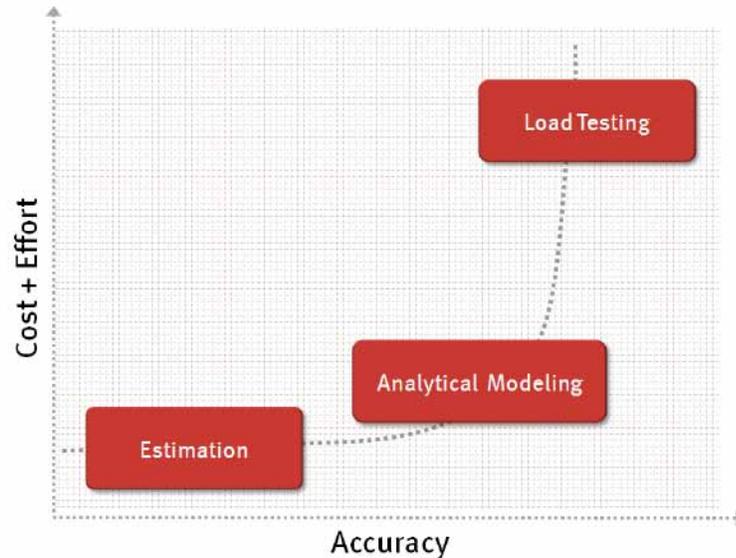
**Optimized workload placement
made possible by up front planning**

“Estimation,” as we mentioned earlier, typically entails stacking workloads on top of each other until a predetermined threshold is reached. It is the technique used by most of the tools available for surveying a data center and identifying virtualization possibilities. The results are cheap and easy to obtain, but represent rough estimates. By itself estimation will not come close to helping you reach optimal performance.

“Analytic modeling” is not as fast, easy, and inexpensive as estimation, but it comes close. (By “analytic modeling” we specifically mean modeling using an analytic queuing network solver. Less sophisticated analytic mathematical models are perhaps better considered as “estimation” tools.) Analytic modeling is much simpler and less time-consuming to set up than load testing (or simulation modeling), and it is much more accurate than estimation.

“Load testing” can be very accurate if the synthetic loads and the underlying system configurations are representative of the production environment. However, more time and expense is required the closer the loads and the infrastructure get to matching production. For most situations, it

just isn't practical to use this technique for optimizing IT services in a virtualized environments. It can be a useful compromise, however, to run smaller tests and then use analytic modeling to project what might occur with bigger loads on more impressive system configurations.



In summary, it's not enough to rely on reactive resource scheduling and workload migration mechanisms for IT Service Optimization. You need more for demanding and critical applications running in commodity-server virtualized environments. The smartest solution is to plan ahead, making sure that placement and migration policies are designed to ensure that workloads will run efficiently together. Estimation tools can be good for quick and dirty analysis of simple workloads. Load testing can be valuable especially when you are rolling out entirely new application workloads, but the best all-around technique for optimizing critical applications running in virtualized environments is analytic modeling using a queuing network solver.

TeamQuest Corporation

www.teamquest.com

Follow the TeamQuest Community at:

Americas

One TeamQuest Way
Clear Lake, IA 50428
USA
+1 641.357.2700
+1 800.551.8326
info@teamquest.com

Europe, Middle East and Africa

Box 1125
405 23 Gothenburg
Sweden
+46 (0)31 80 95 00
United Kingdom
+44 (0)1865 338031
Germany
+49 (0)69 6 77 33 466
emea@teamquest.com

Asia Pacific

Units 1001-4 10/F
China Merchants Bldg
152-155 Connaught Rd Central
Hong Kong, SAR
+852 3571-9950
asiapacific@teamquest.com

**Copyright ©2010 TeamQuest Corporation
All Rights Reserved**

TeamQuest and the TeamQuest logo are registered trademarks in the US, EU, and elsewhere. All other trademarks and service marks are the property of their respective owners. No use of a third-party mark is to be construed to mean such mark's owner endorses TeamQuest products or services.

The names, places and/or events used in this publication are purely fictitious and are not intended to correspond to any real individual, group, company or event. Any similarity or likeness to any real individual, company or event is purely coincidental and unintentional.

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THE DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a license agreement. The only warranties made, remedies given, and liability accepted by TeamQuest, if any, with respect to the products described in this document are set forth in such license agreement. TeamQuest cannot accept any financial or other responsibility that may be the result of your use of the information in this document or software material, including direct, indirect, special, or consequential damages.

You should be very careful to ensure that the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with respect to which it is used.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions. U.S. Government Rights. All documents, product and related material provided to the U.S. Government are provided and delivered subject to the commercial license rights and restrictions described in the governing license agreement. All rights not expressly granted therein are reserved.