WHITE PAPER

CiRBA

Data Center Intelligence

Dynamic Capacity Management In Virtual Environments

Andrew Hillier, Co-Founder & CTO, CiRBA

Contents

Capacity planning is undergoing a transformation as virtualization technologies gain adoption in IT environments. This paper describes the nuances of capacity supply in virtual environments, how to model the demands that workloads place on this supply, and how to effectively align supply and demand in a way that maximizes efficiency and minimizes operational risk. By applying these principles to the optimization, governance and planning of workload placements in virtual environments, IT organizations can shift away from traditional capacity planning and move toward true *Dynamic Capacity Management*.

Introduction

Capacity planning is changing. The introduction of virtualization is shifting the focus away from the traditional measurement / analysis / trending activities that are used to estimate when workloads will outgrow their servers. Instead, we are being faced with the discrete-time management of finite-element workloads running on flexible pools of capacity. In other words, capacity planning in virtual environments looks more like a rapid-fire dating service than it does a long, slow process of applications and servers growing old together.

This "new school" capacity planning, often referred to as Dynamic Capacity Management, is more focused on management in aggregate, where resource supply and demand is pooled to leverage economies of scale. In these pools, it is the unused capacity, or whitespace that becomes the indicator of the environment's ability to absorb short-term shocks and service long term growth. And based on the personalities of the workloads, the architecture of the underlying servers, service level requirements, availability requirements, technical consideration, business considerations and a raft of other factors, this is no simple task.

As this paper will show, there are a number of technical, business and workload "constraints" that impact the operation of IT environments, and these play a much larger role in capacity planning in virtual environments than they did in the physical world. Also, understanding what a workload will actually look like when running in a virtual environment becomes important, which in turn requires an understanding of the types of virtualization available as well as the types of servers they can run on. And finally, capacity planning in virtual environments requires an understanding of how many individual workloads can be safely stacked together on a single physical host before unacceptable operational risks are introduced. As in many aspects of life (including dating), there is a near continuous risk-reward spectrum that allows potentially higher rewards to be found by those who are willing to take higher risks when matching supply and demand in their IT environments.

Capacity Supply: Types of Virtual Infrastructure

Although not intended to be a survey of specific virtualization technologies, it is useful to consider the types of virtualization that exist today, and to use these general classifications as a framework for discussing how to model capacity supply and demand in virtualized IT environments.

Types of Virtualization

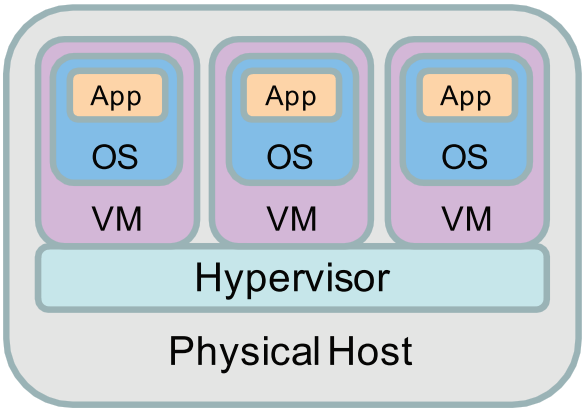
There are as many different types of virtualization as there are types of hardware platforms, if not more. In fact, it seems that the number of technologies whose names end in "PAR" alone is quite extensive. Fortunately, many of these technologies share common traits, and can be broken down by the way they generally work and the features they provide.

Virtualization Models

In general, virtualization technologies are designed to create an abstraction layer that isolates resource demand from supply. Because this paper is focused on server virtualization technologies, this abstraction generally manifests self in one of two ways:

Below Kernel

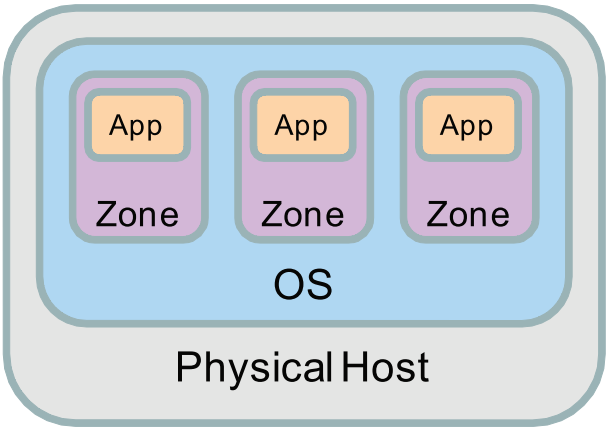
This type of virtualization happens "below" the operating system, which may not even be aware that virtualization is happening. This is the modus operandi of hypervisor-based virtualization technologies and is undoubtedly the most common type of virtualization. This approach is very advantageous from a *transparency of strategy* perspective, as the applications running in VMs each get their own copy of the operating system, thus allowing it to "look and smell" just like it did before virtualization was introduced.



This approach has drawbacks, however, in that the post-virtualization world has all of the same operating systems, middleware and applications as it did before the transition. Little is gained from a software footprint perspective. Also, the virtualization layer can create additional CPU overhead when it "intercepts" operations intended for hardware components and transforms them to comply with a virtual server model. For below kernel technologies, operating systems and device drivers are often "para-virtualized" to reduce this CPU overhead. This means that the OS or device drivers are aware of the underlying virtualization layer and are optimized to avoid unnecessary overhead.

Above Kernel

Above kernel virtualization techniques perform the isolation of applications from within the OS, segregating and containing them from one another while still executing them within the same operating system kernel. The resulting operational environments, referred to as "Containers", "Jails", "Zones", "workload partitions", etc., allow several applications to safely reside on a single server within a single operating system.



One advantage of this approach is the efficiency gained through economies of scale. Such constructs typically support "sparse" operational models that allow them to share resources, such as installed software, between multiple instances. This approach also allows the applications to communicate directly with device drivers, thus avoiding the overhead associated with virtual device drivers.

These same advantages also create several drawbacks. While having the installation of a single patch affect many running applications may seem convenient at first, the common mode complexity this creates when managing production applications can often outweigh this. Also, the level of separation between applications may or may not be strong enough to satisfy the requirements of every environment.

Virtualization Features

Independent of the way the technology works, most virtualization technologies seek to provide a set of features that leverage the advantages of the virtual paradigm.

Clusters

Different technologies support different notions of clustering, but the general concept is that the nodes of the cluster be interchangeable from an operational perspective, thus allowing applications to "move" between them. This movement, often called "migration" or "motioning", allows the rebalancing of workloads between the discrete elements of the cluster, thus fostering operational efficiency and optimizing performance. The nodes of a cluster can also boot up VMs that had previously been running on other nodes, thus providing the context for a reboot-based form of high availability that is similar to that used in many Windows environments.

Resource Pools

In some technologies, the term resource pool is used to refer to cluster-like capabilities, while in others it is used to refer to logical pools of resources that are managed and tracked separately on top of the underlying infrastructure. In the latter definition, resource pools provide a way to reserve capacity, to limit utilization, and to set the priority of different applications and/or application systems within a virtual environment. This provides a finer level of control that would otherwise be lost when the "one box per app" pattern is no longer followed.

High Availability

Some virtualization technologies provide the ability to detect when a physical host fails and automatically restart the affected VMs on other servers in the same environment. Because this requires visibility to the same storage that the failed server was using, this facility is often limited to the clusters described above.

It should be noted that this form of high availability is not suitable for all applications, and should not be confused with resiliency-based HA techniques. For relatively stateless applications a "reboot" approach may be suitable, but for transactional or mission critical applications the goal is often to host them on resilient infrastructure that is designed to not fail in the first place (or to have sufficient internal redundancy so as to isolate failure from the application). The HA capabilities, or lack thereof, of the various virtualization technologies should be considered with this in mind.

Types of Host Systems

Along with the type of virtualization being employed, another variable is the type of physical server it will be deployed on. The maturation of x86-based virtualization has popularized the use of small, inexpensive "commodity" servers to house many workloads. This stands in contrast to the "vertically scaled" virtualization techniques commonly used on midrange and mainframe platforms, which employ less "quantized" pools of resources to achieve a similar outcome.

Commodity Servers

The use of large pools of relatively small servers is common in most x86-based virtualization technologies. The use of inexpensive off-the-shelf servers, including blades, is appealing to organizations that value the flexibility and low initial capital expenditure of this approach. It also allows the repurposing of existing gear, of which there is no lack in IT environments that routinely upgrade their servers. Together, the implication is that there are potentially massive amounts of compute power in today's IT environments. In large Windows-based installations that have used the "one app per box" architectural pattern, this mentality has resulted in lightly utilized servers where the average utilization is 15% or lower.

The drawback of this approach is that it is like building a swimming pool out of buckets. It is great for applications that fit in a bucket, but does not create a true contiguous "pool" (i.e. a whale would not fit in such a pool). While this is not an issue for most applications, it is a worthwhile distinction to make. A related, and probably more important concern, is that the unused portion of each bucket is difficult to make use of, and an application that takes up most of a bucket may cause the extra space to be wasted. This whitespace fragmentation can cause unused capacity to add up, lowering the overall utilization of the compute resources.

Vertically Scaled Servers

Employing fewer, larger servers to host virtual environments is an approach that is possible in most virtualization techniques and is the norm on midrange and mainframe systems. In x86 environments, the use of so called "fat nodes" instead of smaller servers (e.g. using 8-way quad-core servers instead of 2-way systems) has recently been the topic of much discussion, and has both advantages and drawbacks.

Because this approach is analogous to building a full swimming pool, it has advantages over the more "quantized" approach. Should an application require more resources than a single "bucket" can provide, a "fat node" accommodates this by simply flexing the boundaries to fit the CPU requirements (picture a movable barrier in a public pool). It also fosters higher utilization, as the economies of scale are better and the consolidated "whitespace" can now be used by applications that might suddenly require the extra CPU time.

The last, and perhaps most interesting effect of this lack of quantization is the requirement for migration and/or motioning. In the bucket analogy, it is useful to move things between buckets to balance out the individual workloads within the finite resources of each bucket. When considering a true swimming pool, there is no reason to move anything, as the applications in the pool expand and contract without hitting "artificial" limitations.

Given this, one can argue that motioning is only required because of the finite size of the individual resources that make up a pool, and that most vertically scaled solutions do not require such a concept, thus simplifying the solution. In the same line of thinking, "bucket-based" HA solutions are unnecessary if the underlying platform design has sufficient resiliency so as to not fail in the first place.

The drawback, of course, is that vertically scaled platforms are typically much more expensive than commodity servers, and that resiliency comes at an even higher price. The purpose of this paper is not to make arguments for which technology is best suited for which application, but rather to outline the parameters and constraints that influence the planning and design of virtual environments.

Capacity Demand: Modeling Workloads in Virtual Environments

Determining the optimal virtualization plan requires getting a clear picture of what the application workloads will look like in the target virtual environment. There are several factors to consider when modeling virtual workloads: the "personality" of the workload, how well the target platform performs for the particular type of workload, and how much overhead is introduced by the virtualization technology being targeted.

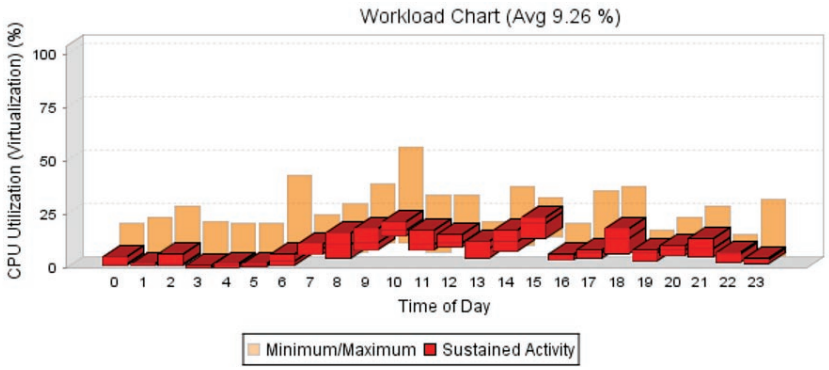
Workload Characteristics

Few workloads run at a constant utilization level, and the variations in demand for resources over time sculpt out an operational pattern that must be taken into account when modeling workloads. Because virtualization allows multiple applications to share the resources of a single host system, the size and shapes of these patterns, and how they interplay, is critical in the design of properly functioning and efficient virtual environments.

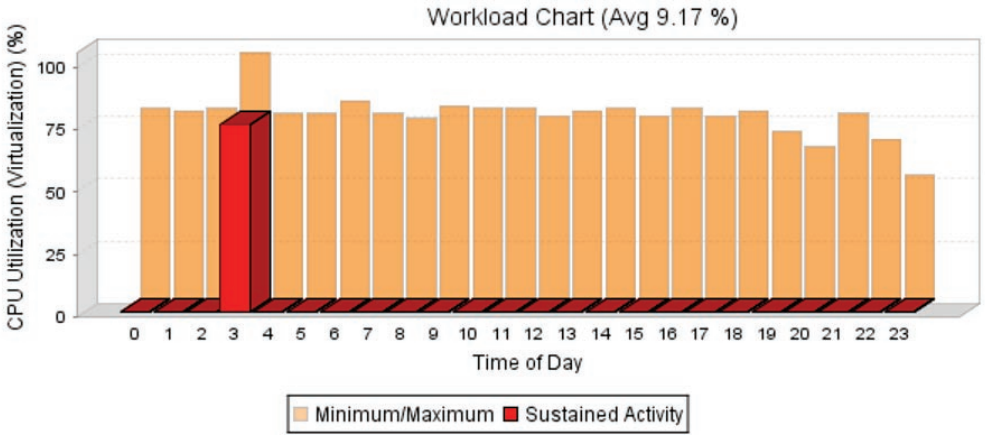
Peak vs Sustained Activity

Many applications have a high variability in their demand for system resources. This may result from increased user activity (e.g. web sites at lunch time), variations in the business day (trading start of day) or simply because that is when work was scheduled (batch jobs). In any case, these peak utilization levels are important to the analysis of workloads, as they may dictate how much peak compute capacity the application will require.

Low variability, sustained activity on the system is equally important in the determination of capacity requirements. Many applications tend to have high transient spikes in utilization but spend the vast majority of their time at lower utilization levels. This must also be weighed into the analysis, and for some types of workloads, such as batch jobs, the sustained activity may be the deciding factor.



CPU Utilization chart above shows the activity over a 24 hour period. The hourly utilization levels are expressed as quartiles, with each vertical section representing 25% of the operational time. In simple terms, this means that in any given hour the system spent one-quarter of the time in the upper orange band (with the top of that band being the maximum), half of the time in the thicker red area, and one quarter of the time in the lower orange band (with the bottom being the minimum). This helps provide an intuitive picture of the operational pattern of the server, which in this case has an overall average of 9% utilization but peaks at approximately 50% mid-day and has a sustained utilization of around 15-20% throughout most of the business day.



This CPU Utilization chart shows a completely different workload "personality". This server, like the previous example, has an average utilization of 9%, but the pattern is extremely "peaky", with the server hitting 80% peak utilization every hour and having virtually no sustained activity (as is evident by the red zone being at the "floor"). There is a burst of sustained activity between 3 and 4am, which is common when backups or batch jobs are initiated overnight.

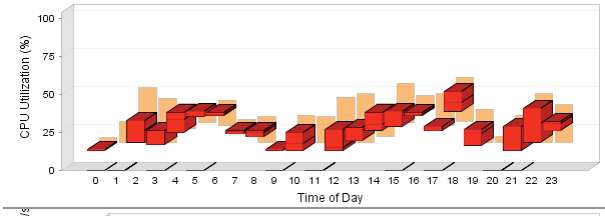
The two examples shown highlight the difference in variability between two workloads, which serves to illustrate how these metrics impact virtualization plans. The first example shows a relatively sustained pattern, and such workloads tend to stack up to higher levels of utilization than the second example, which has high transient demand but little sustained activity. For the latter case, the analysis must attempt to dovetail these demands in order to get reasonable levels of utilization without incurring excessive "contention risk".

Workload Personalities

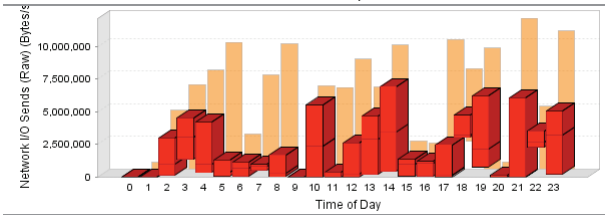
Extending this concept to the broader set of utilization metrics, including disk and network I/O rates, provides a view into a system's overall workload personality. Systems that host CPU intensive applications start to look quite different than pass-through queue managers, databases supporting OLTP, OLAP, etc.. The amount of I/O relative to CPU activity, the ratios of reads to writes, and even the shapes of the utilization curves over a 24 hour period contribute to the overall profile of the workload on the system.

This information serves two purposes. Firstly, certain workloads can be consolidated better than others. In trying to achieve high overall utilization, a certain amount of diversity is also useful to make sure the target system is not overly-stressed in any one area.

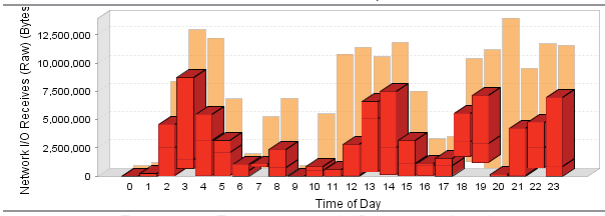
The second use of workload personalities is to drive the benchmarking strategy used when analyzing source and target systems. In transformations involving similar source and target architectures (e.g. x86 to x86) this is less critical, as the personality-based variation is minimal. For cross-platform analyses, however, the personality of the workload plays a bigger role, as certain platforms are much better at hosting certain workload personalities than others. For example, when migrating x86 workloads onto mainframes, the personality plays a major role in the outcome; CPU intensive workloads do not favor the mainframe architecture, whereas mixed and/or OLTP type workloads do.



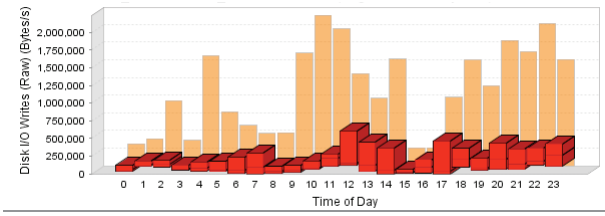
Moderate, concentrated CPU activity



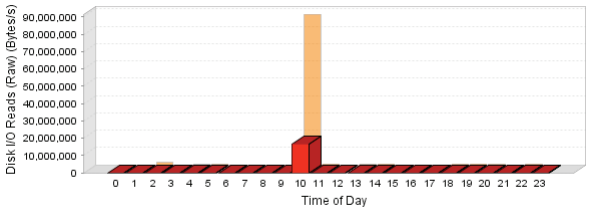
Variable, somewhat sporadic network send activity



Variable network receive activity that reciprocates sends



Somewhat peaky disk writes with low sustained activity



Virtually no disk read activity except for a single very high burst

These graphics show the operational profile of a database server running an OLTP-style workload. Note that the network I/O and the disk write activity reflect a balanced, sustained transaction rate and the CPU activity is generally in response to this I/O activity. This particular example is interesting in that the disk read activity is virtually nonexistent, which either indicates that there are no queries being made to this database or that there is very effective read caching in place.

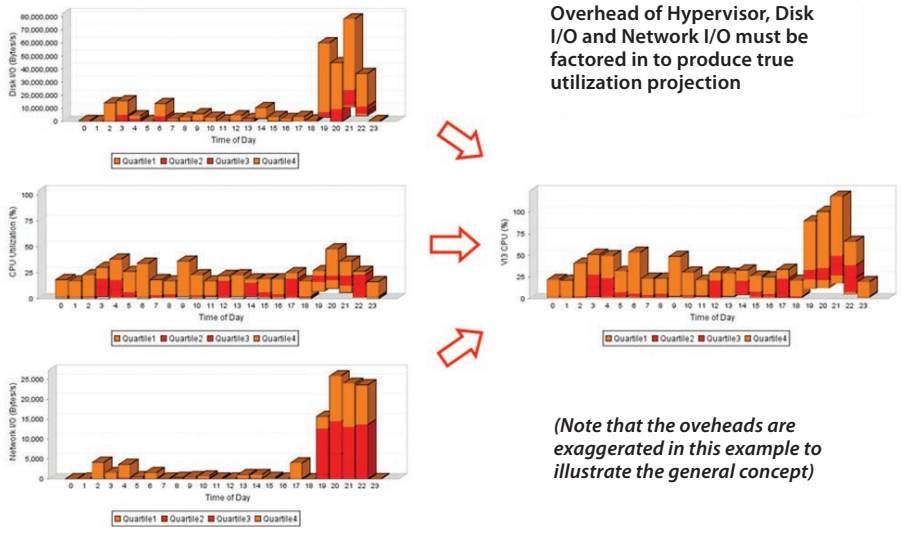
To account for these personality differences in virtual environments, each VM must be assigned a benchmarking strategy based on the personality of the workload it is running. This will cause the utilization information to be normalized onto the target systems using the best possible normalization factors, providing an accurate view of what the workloads will look like when projected onto the target environment.

Virtualization Overhead Models

Many virtualization technologies employ virtual device drivers to abstract the running applications from the physical hardware they require. While enabling very powerful features, such as portability and live migration, this also incurs load on the CPU as the host system converts the virtual calls into real I/O operations on the underlying hardware.

A simple method to account for overhead is to simply set conservative limits on utilization levels during the analysis, but this may not provide an accurate picture. Because this load is skewed toward periods of heavy I/O, the operational patterns that result may not accurately reflect what will actually happen, and the simplicity of a "flat" overhead model may cause underprovisioning of target hardware.

A more accurate way to estimate utilization levels in virtual environments is to employ an overhead model that takes fixed scheduler overhead, disk I/O overhead and network I/O overhead into account. In this way, a more accurate view of true utilization levels can be derived.



These graphics illustrate of the effect of I/O activity on the CPU utilization of virtual environment. In this case, the CPU utilization of a physical server (middle curve on left) will incur a corresponding CPU load in the virtual environment, but the servicing of the I/O activity (top and bottom curves on the left) will also load the underlying virtualization platform and cause the true impact to resemble the curve on the right. Because the I/O activity is concentrated in the evening hours, the virtualization overhead will also be more pronounced during this period.

It should be noted that the overhead of a virtualization solution may not manifest itself at the VM level, and instead will be associated with the underlying hypervisor or OS image. This may cause the utilization pattern of the VM itself to not accurately reflect its contribution to the overall utilization of the system.

Capacity Planning: Matching Supply and Demand

Once a clear picture has been established with respect to what the workloads will truly look like in the target virtual environment, the next step is to determine the optimal strategy to combine them on the target servers. Again, there are two factors that influence this: whether the applications should be stacked to peak or sustained activity, and how much contention risk is tolerable in the long term operation of the environment.

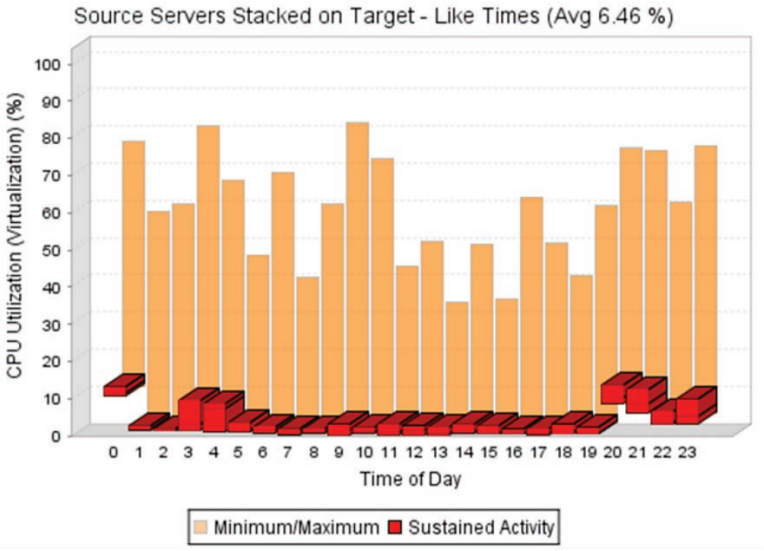
Transactional vs Batch Workloads

In order to determine how many applications can be safely virtualized onto each physical host, it is useful to classify workloads into two distinct types:

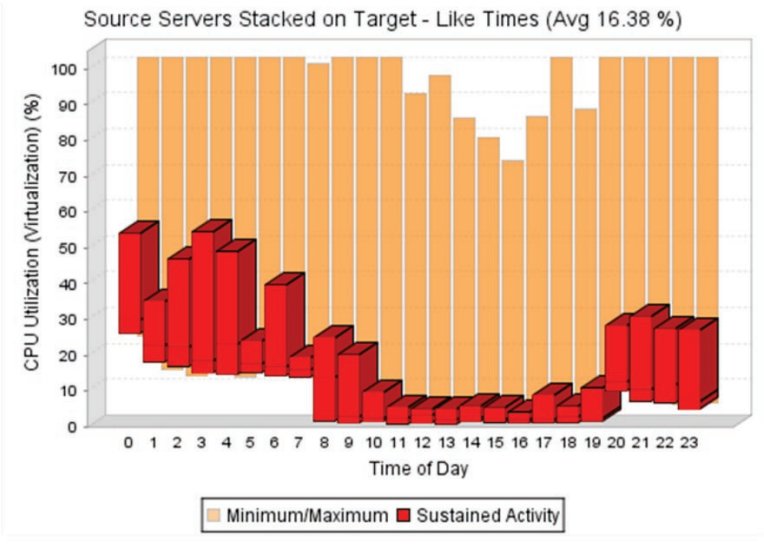
Transactional – Workloads that have end-to-end service level objectives. During periods of high CPU utilization, the CPU queueing will cause the end user to experience "slowdowns". The term transactional is loosely applied here, and these workloads include any application where service level agreements or user experience will be adversely affected if it contends for resources with other applications. This type of application includes production transaction processing systems, critical end-user applications (e.g. trading systems) and certain types of VDI applications.

Batch – Workloads that are job-oriented, not performance-oriented, and therefore are not sensitive to transient degradation in resource supply. The focus with this kind of workload is on getting a certain amount of work done in a reasonable timeframe without overloading the host infrastructure. Production batch jobs, compilers, automated test suites and other non-critical and/or non-production applications fall under this definition.

By categorizing workloads into these definitions it is possible to construct tiered virtual environments that are designed to provide a specific level of service to each application in such a way that overall environment utilization is balanced against individual application performance.



Sample analysis of transactional workloads using a "peak weighted" scoring algorithm. In this approach, workloads are combined until the peak utilization of each physical host system reaches a pre-defined threshold (in this case 80%). As a result, two blade servers are required to virtualize the source workloads, and drilling down on one of the host systems shows relatively modest sustained activity and an average utilization of 6.46%. This conservative sizing is necessary to ensure that applications do not contend for resources, but comes at a cost of lower virtualization ratios.



This graphic shows a sample analysis of batch workloads using a "sustained weighted" scoring algorithm. In this approach, workloads are combined until the sustained utilization of each physical host system reaches a pre-defined threshold (in this case 80%). As a result, only one blade server is required to virtualize the source workloads, and drilling down on the host system shows relatively high sustained activity and an average utilization of 16.38%, almost triple the peak-weighted analysis. This aggressive approach provides higher virtualization rates but will cause the individual applications to contend for resources, adversely affecting performance.

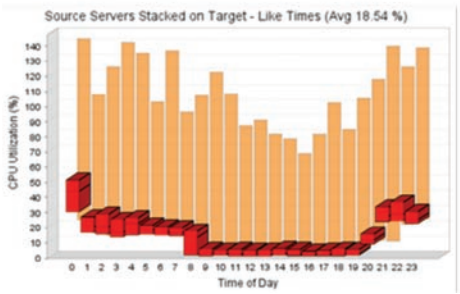
Workload Contention Probability

When virtualizing transactional workloads that are sensitive to service level objectives, it is possible to drive higher utilization levels if the peaks "dovetail" with one another over time. Of course, this is a risky game, and placing applications with high peak demands on the same server increases the chances of contention between workloads, causing slowdowns that may affect users and/or service levels.

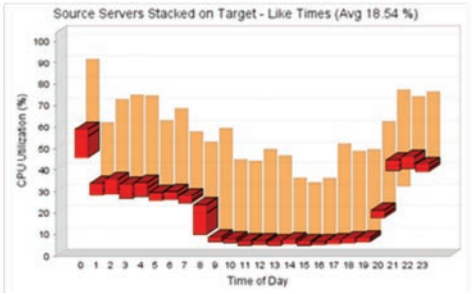
Keeping this in mind, optimizing the placement of workloads requires analysis based on specific operational risk tolerances (or, conversely, to a specific confidence levels). For example, if there is absolutely no tolerance for workloads "colliding" in a particular environment then there would be zero risk tolerance. This translates into a confidence level of 100%, meaning there is high confidence that applications will play nicely together.

If, on the other hand, it is deemed acceptable that at the busiest time of day there is a one in one hundred thousand chance of two applications contending for resources, the analysis would be configured to target 0.001% risk, or 99.999% confidence. Although seemingly minor, this slight difference can have a profound impact on results, as the chances of all workloads peaking at the same time is astronomically unlikely, and even a slight "relaxing" of the tolerable risk has sizeable consequences. At 20:1 stacking ratios, a 99.999% confidence will typically produce 33% better stacking ratios than 100% confidence.

100% Confidence (Tolerate No Risk)



99% Confidence (Tolerate 1% Risk)



This example shows the analysis of two different risk levels. The left hand curve shows the 100% confidence case, and can be interpreted as saying that there is a 100% chance that the combined workloads of all VMs will fall within this range. The right hand curve shows the 99% case, meaning that there is a 99% chance that the workloads will fall within the range shown (or, conversely, there is a 1% risk that they will exceed this range). At 99% confidence it appears that all the VMs will fit on a single server, whereas the 100% case will cause the VMs to be split across two physical servers. Such analysis is useful for setting up different virtual compute pools designed to run workloads at different service levels.

The Impact of Business and Technical Constraints

Just as utilization in physical environments is limited by the fact that each workload is constrained to its host system, utilization in virtual environments is similarly constrained by numerous business and technical considerations. This means that the "ideal" capacity plan, which assumes complete mobility of workloads and complete freedom to intermix them as necessary, can rarely be achieved. Instead, environments must be constructed to meet certain design criteria, imposed either by the technologies in use or by the business environments within which they operate.

Technical Constraints

The nature and impact of technical constraints varies based on the type of virtualization being used. For below kernel (e.g. hypervisor-based) technologies these constraints are largely related to hardware variance, and include affinities between existing servers (e.g. ones that use the same storage) as well as "gotchas" such as unusual hardware or networking requirements. For above kernel technologies (i.e. OS-level containment) these considerations are valid, but so too is the configuration of the operating system itself. Because applications share many aspects of the operating system in these technologies, extra diligence is required.

In below kernel virtualization, each application has its own copy of the operating system, rendering OS-level constraints somewhat irrelevant (except for optimizations such as memory sharing). Given this, the specific considerations that are necessary to analyze for these types of technology include:

- Hypervisor compatibility
- Motioning compatibility
- Specialized hardware
- Unusual communication and/or protocols
- Memory sharing optimization

For above kernel technologies this list is augmented with a number of OS-level considerations that will determine how well applications can co-reside, including;

- OS version compatibility
- Kernel patch levels and service packs
- Installed software versions
- Installed patches/hotfixes
- Time zones

These are just a sampling of the types of considerations that can affect the design and operation of virtual environments.

Non-Technical Constraints

The consideration of non-technical constraints is imperative if an organization is to arrive at an optimal end state that confoms to operational governance, internal policy and regulatory compliance. Equally as important, it allows the individual contribution of these constraints to be assessed in the overall context of the virtualization outcome, thus allowing organizations to make intelligent decisions regarding the kinds of business and process-level changes that need to accompany the technical changes associated with virtualization.

The impact of these constraints on capacity planning is significant, as they create additional barriers to the mobility of VMs and they "fragment" capacity into smaller pools. For example, if there are only 3 production servers in a certain location then the maximum VM-to-host ratio is 3:1, which will almost certainly leave the host system underutilized. This is a common effect, and in many cases systems will not be utilized to their full capacity if prudent capacity planning is performed.

Business Constraints

The true business constraints on an organization are those that arise from organizational, financial, regulatory or political considerations. Examples of business constraints that may be necessary to include in an analysis are:

Business Services - it is often desirable to host different business services in different pools, especially if distinct SLAs must be maintained and tracked. In heavily regulated environments it may be necessary to maintain separations due to regulatory requirements (e.g. equity traders versus research groups)

Customers – environments hosting the applications of different customers may be required to host them on separate infrastructure, either due to contractual obligations or to simplify the systems and financial management of those servers.

Departments – keeping departments apart is necessary if the financial management elements, such as chargeback, are not in place. It is also necessary in cases where different groups have unique requirements and need dedicated infrastructure.

Locations – if two servers currently reside in different physical locations, and there is no business mandate to change this (e.g. data center consolidation), then it does not make sense to virtualize them onto the same physical host. As obvious as this seems, some workload-only analysis approaches do not have visibility into the physical location of a server and can therefore recommend just such a design.

Process Constraints

In recent years IT environments have seen considerable advancement in the maturation of IT service delivery and management processes. Not taking these into account when virtualizing can serve to undo years of progress in one fell swoop. Common process-level constraints include:

Maintenance Windows – combining applications that do not have overlapping maintenance windows on a single physical host can create a situation where it is impossible to perform hardware maintenance, as there is never a time when all applications

can be brought down. This is particularly true in non-motioning technologies, where the VMs cannot be shifted to other servers to facilitate maintenance.

Change Freezes – if a VM is subject to a change freeze during critical times of the business cycle then this may have a ripple effect across other applications in the same pool or cluster. It is often necessary to freeze other VMs on the same server (for example, to prevent changes that may inadvertently affect memory sharing), and it is also prudent to suspend all motioning activity to eliminate any operational volatility. In some technologies this can seriously impair the ability of the pool to balance its workloads, making it imperative that these freezes be taken into account in the planning process.

Operational Environments – combining production and non-production environments in the same virtual clusters can create complexity and require clever resource pool configurations in order to make it safe. In some environments this complexity is outweighed by the benefits in utilization, while in others this effect is less pronounced (larger environments tend to have the economies of scale to keep them separate).

HA and DR Strategies – existing HA and DR strategies that are based on data replication or other application-level availability features must be accounted for in the virtualization process, as they may have requirements that are beyond the capability of the virtualization platform. Dual-hosted SCSI, database-level clustering and other advanced technologies may complicate the transformation process and may even rule systems out as candidates for virtualization.

Vendor Service Plans – sometimes the simplest things cannot be ignored, and an application running on a server that has a specific break/fix response time cannot simply be moved to another that does not satisfy this requirement. The HA features of the target technology may mitigate this, but if not then it is necessary to consider this during the analysis phase.

Security Constraints

The topic of security in virtual environments is a very interesting one, and many organizations have yet to fully rationalize their virtualization plans against their security policies. Rather than enter into a lengthy discussion on the topic, this section will instead highlight the obvious constraints that security places on the virtualization process:

Security Zones – while mixing production and non-production applications in the same virtual pool may or may not be a useful thing to do, mixing security zones is almost never a good idea. There are few virtualization technologies with the technical credentials to pull this off, and serious vulnerabilities can result.

Export Restrictions – in environments where sensitive technologies or algorithms are in use it may be necessary to separate the servers where export licenses are required from those where they are not.

General Data Segmentation – regulatory requirements, such as HIPAA, often make it necessary to ensure that certain bodies of information can never be viewed by a single person at the same time (think names and social insurance numbers). While this is commonly implemented as a storage constraint, the virtualization paradigm often makes the storage of multiple VMs visible at the physical host level, thus turning it into a server constraint.

Ongoing Analysis: Dynamic Capacity Management

One interesting aspect of virtualization is that all of the constraints that come into play in planning virtual environments also come into play on continuous basis when managing them. This stems from the fluidity that these environments are capable of, which places them in a constant state of transition, thus requiring the ongoing scrutiny of the various constraints outlined previously.

The act of continuously matching supply and demand in virtual environments in a way that minimizes operational risk is referred to as Dynamic Capacity Management. To fully balance efficiency and risk on an ongoing basis, this management discipline consists of three key components:

Placement Optimization

Ensuring that VM placements are optimal given the resource, business, and technical constraints is essential in virtual environments. Matching supply and demand on a regular basis ensures that the right workloads are running on the right servers at the right times. This involves "VM Rebalancing" analysis that constantly provides guidance with respect to the best VM placements to service workloads in an efficient and safe manner. The proper management of "whitespace", or spare capacity, is also critical to ensure that transient demands are met in a way that meets SLAs and that unexpected or infrequent workload fluctuations have a "cushion" of capacity to absorb them.

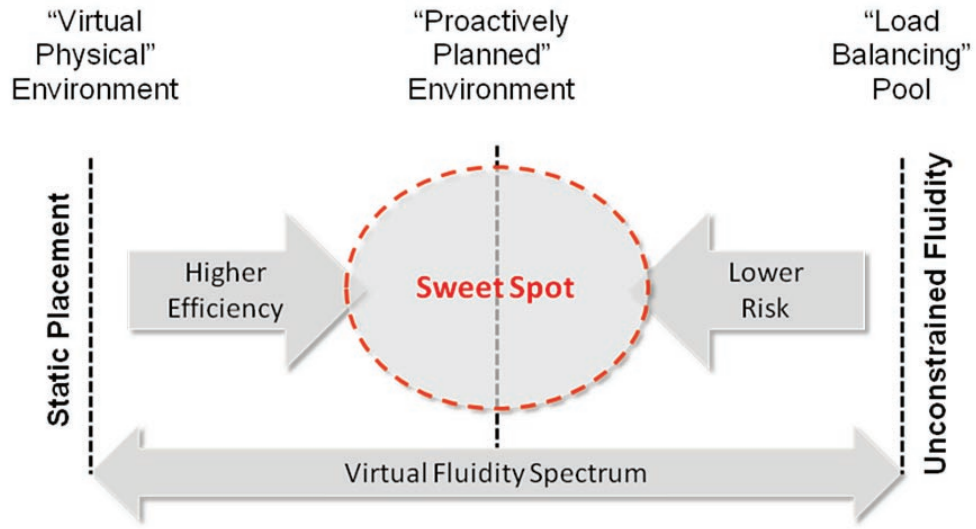
Placement Governance

One advantage of the dynamic and fluid nature of virtual environments is that they can reconfigure themselves on the fly to react to demands. The downside of this is that they can easily go offside from a security, risk and compliance perspective if these considerations aren't "embedded" in the decision-making process. This is a concern that did not exist before the introduction of virtualization, as physical environments lacked the ability to change themselves in ways that can invalidate the fundamental business, process and technical rules that govern them. As such, governance of the placement of virtual workloads is an essential element of Dynamic Capacity Management that allows the efficiency benefits to be realized without the potential downsides that can accompany them.

Placement Planning

Spontaneity is not a good characteristic when it comes to data centers. Although the ability of virtual environments to react to changing conditions is powerful, operating a data center in a purely reactive mode is not optimal. Capacity planning has evolved over decades as a forward-looking discipline designed to match future supply and demand, and it is a mistake to think that the inherent load balancing capabilities of virtual environments can completely replace this. Instead, capacity planning must further evolve to include proactive VM placement planning and validation, where utilization patterns and operational history are leveraged to generate a forward-looking view of where workloads should be hosted.

This approach brings a degree of determinism to data center operation, as it allows VM placements to be known ahead of time, and application demands to be planned for before they happen, not after. This not only reduces the need for "intraday rebalancing", but allows separation of change/capacity management from incident management, creating a more process-centric approach that prevents virtual environments from descending into a "fog" of reactive motioning.



Ultimately, the ability to proactively plan workload placements reduces or even eliminates the need for live motioning of VMs, and instead replaces this with well planned, less frequent migration of workloads into the right places before the operational cycles or application demands begin. This allows virtual environments to stay in the operational "sweet spot" that combines both high efficiency and low risk.

Conclusion

The ability of virtual environments to drive efficiency hinges upon the ability to strike the right balance between efficiency and risk. If utilization levels are too low an environment may be overprovisioned, making inefficient use of resouces. If utilization levels are too high then an environment may be underprovisioned, creating unnecessary operational risk.

The need to balance these competing requirements is not new, and has always been one of the goals of capacity planning. What has changed with the mainstream adoption of virtualization is the ability to decouple resource supply and application demand, allowing applications and their underlying resources to be combined in more efficient ways.

The goal of Dynamic Capacity Management is to match this supply and demand by putting the right workloads on the right servers at the right times. When done properly, this leverages all technical, businesss, and workload nuances in order to make the right decision. When done broadly, this allows many of the true benefits of virtualization to be realized. When done proactively, this allows these benefits to be realized in a way that prevents falling into "reactive" operational models and brings true control to the data center.

About the Author

Andrew Hillier, Co-founder & CTO, CiRBA, Inc.

Andrew Hillier has over 15 years of experience in the creation and implementation of mission-critical software for the world's largest financial institutions and utilities. A co-founder of CiRBA, he leads product strategy and defines the overall technology roadmap for the company.

Prior to CiRBA, Mr. Hillier pioneered a state of the art systems management solution which was acquired by Sun Microsystems and now serves as the foundation of their flagship systems management product, Sun Management Center. Mr. Hillier has also led the development of solutions for major financial institutions, including fixed income, equity, futures & options and interest rate derivatives trading systems, as well as in the fields of covert military surveillance, advanced traffic and train control, and the robotic inspection and repair of nuclear reactors.

Mr. Hillier holds a Bachelor of Science degree in computer engineering from The University of New Brunswick.

Other Publications and White Papers

Download the White Papers

Transformational Analytics: Virtualizing IT Environments
April 2008

Advanced Workload Analysis Techniques
April 2008

Consolidating Workloads onto Mainframes
April 2008

How to Choose the Right Virtualization Technology For Your Environment
November 2007

Virtualization Analysis for VMware
September 2007

About CiRBA

CiRBA's *Data Center Intelligence*™ enables IT organizations to operate the most cost effective virtualized data center possible. Only CiRBA's **Placement Intelligence Technology** continually captures and analyzes technical, business, and resource constraints to safely guide workloads to the right physical or virtual infrastructure.

For more information, visit www.cirba.com.

CiRBA

1595 16th Avenue, Suite 400
Richmond Hill, Ontario
Canada, L4B 3N9

Toll Free: +1.866.731.0090
Telephone: +1.905.731.0090
Fax: +1.905.731.0092
Online: www.cirba.com

Dynamic Capacity Management
In Virtual Environments