Memory Management Fundamentals

Why Today's Virtual & Cloud Environments Demand a New Understanding of the Data Center

VMTURBO

Memory

A lot of the time, memory is the most constrained resource in a virtualized environment. There are a number of techniques that hypervisors employ to manage memory.

Virtualization allows us to over commit memory, which improves utilization, but can lead to other problems if not managed properly.

So, let's start at the very beginning.

Memory Over-Commitment

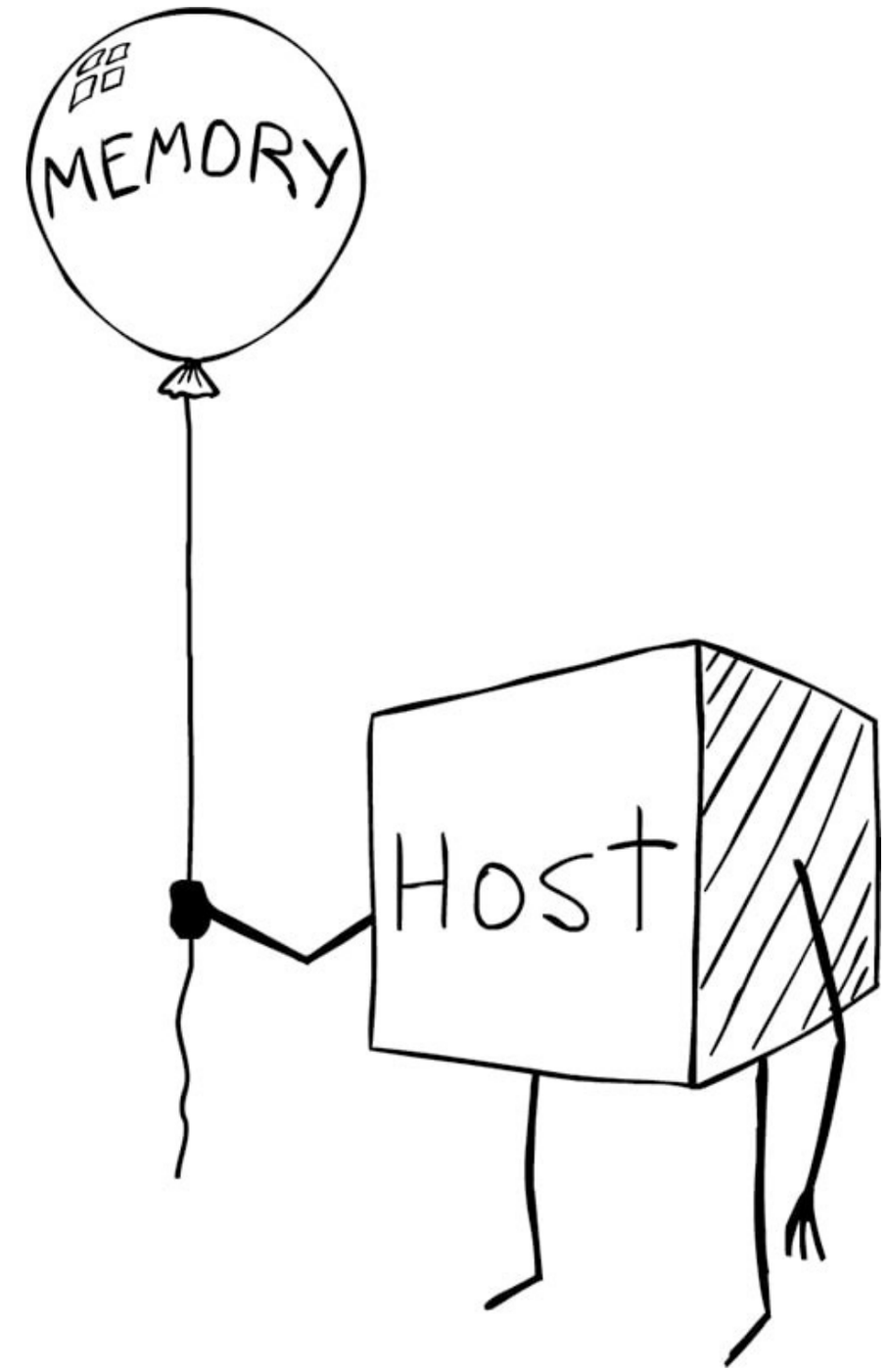
Memory Over-Commitment is a feature that allows a hypervisor to allocate more memory than the physical host actually has available. For example, if the host server has 2 GB of physical memory available, you can provision 4 virtual guest machines, each with 1 GB of memory allocated.

Usually, this is harmless, as most virtual machines only use a portion of their allocated memory. So in the example above, a Virtual Machine with 1 GB of memory allocated might only use 400 MB, so if all four machines have similar usage patterns, you will still have 400 MB of memory left over on the physical host.

Still, some VM's might consume all (or even more) of their allocated memory. Alternatively, memory on the physical host might start to run out. In these cases, the hypervisor can identify and reallocate unused memory from other VM's, using a technique called memory ballooning.

Memory Ballooning

Memory Ballooning occurs when a host is running low on available physical memory. It involves the use of a driver – called the balloon driver – installed on the guest Operating System (OS).

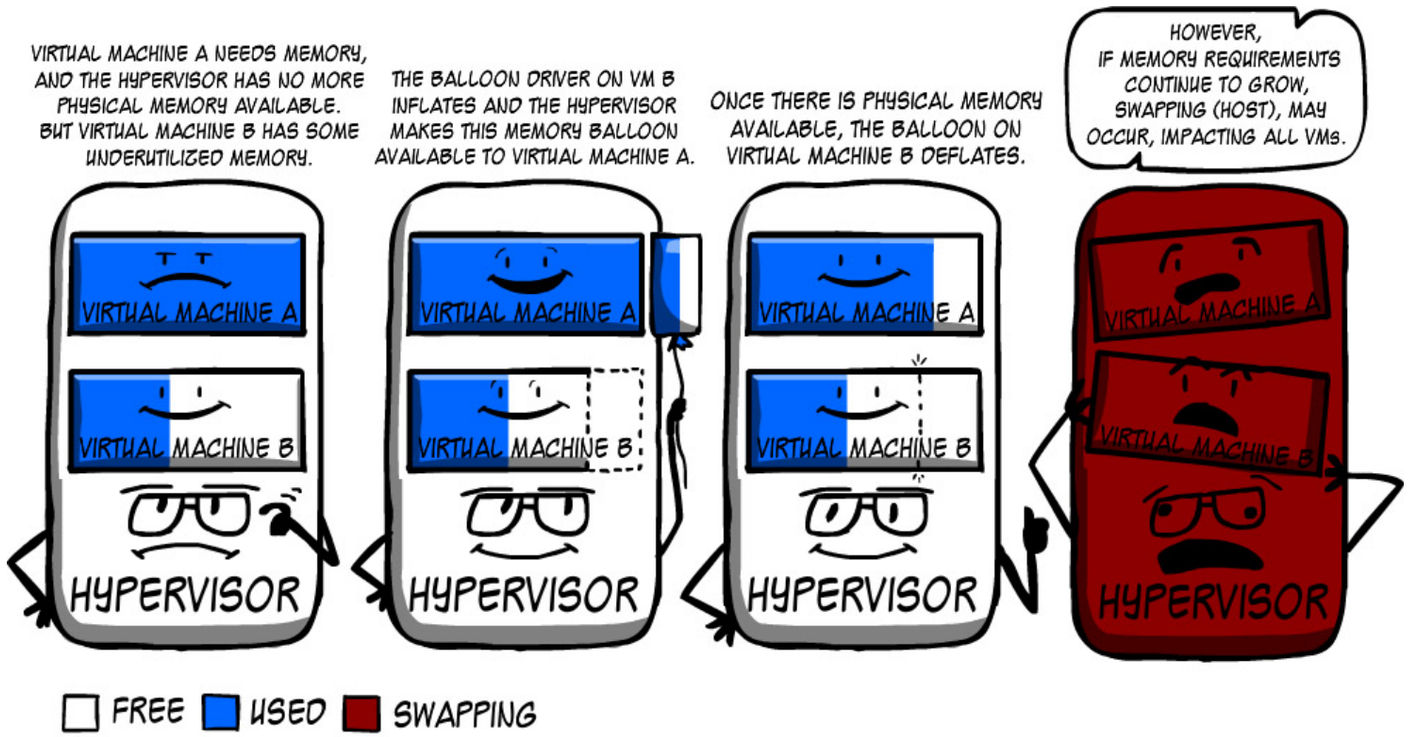


Memory Ballooning

How Does It Happen?

1. Virtual Machine A needs memory, and the hypervisor has no more physical memory available.
2. Virtual Machine B has some underutilized memory.
3. The Balloon driver on VM B ‘inflates’ and this memory is now available to the hypervisor.
4. The Hypervisor makes this memory balloon available to VM A.
5. Once there is more physical memory available, the balloon on VM B ‘deflates’.

There is a huge advantage here – physical memory utilization is more efficient, at the expense of potential performance problems. So far, so good.



This entire process is invisible to the guest operating system, but it can potentially affect performance on the virtual machine. Excessive ballooning on the hypervisor (inflating and deflating) can impair performance of any applications running. An administrator troubleshooting and monitoring just the VM and the guest OS will

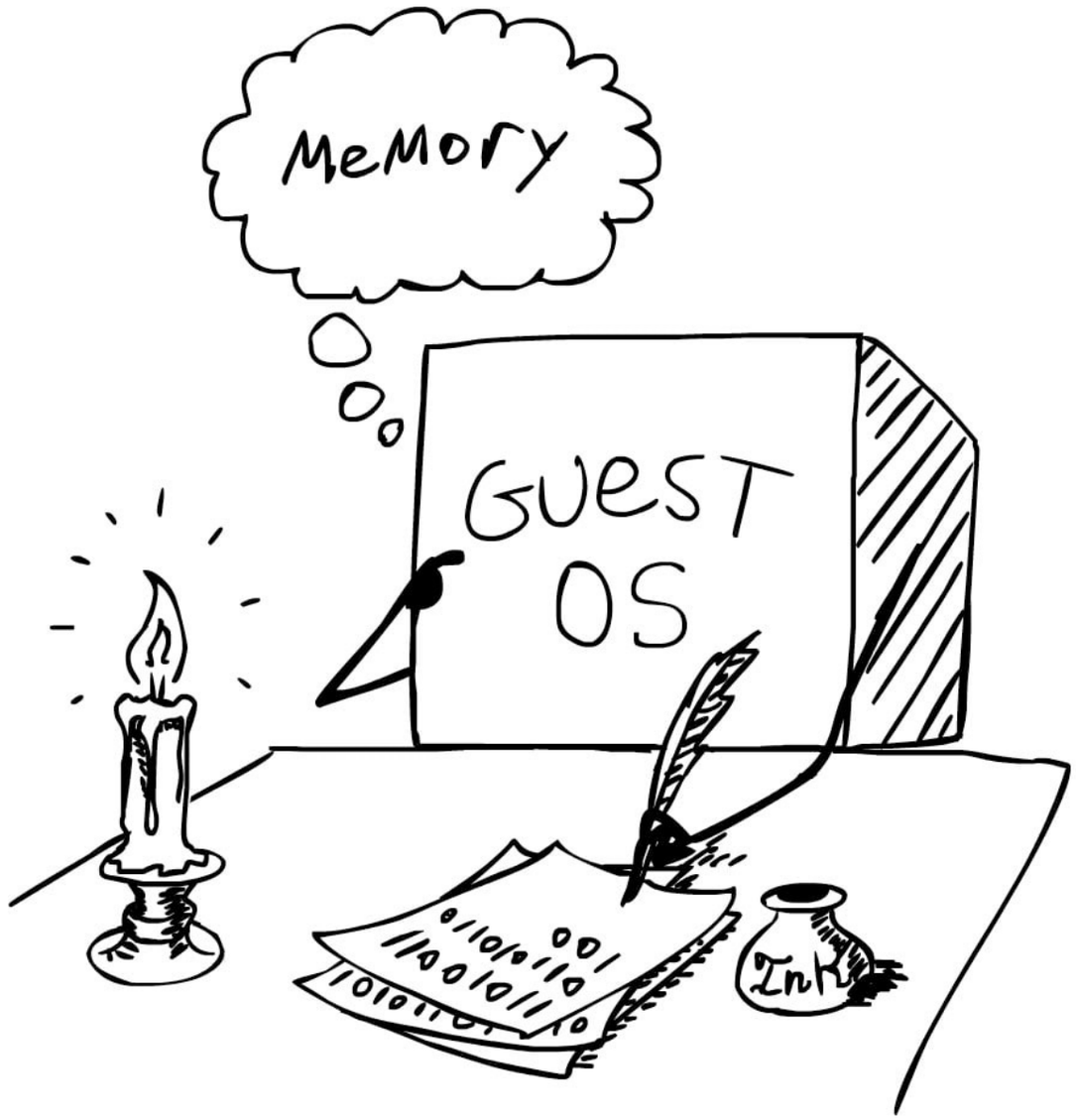
not be able to pinpoint the root cause of the problem, much less fix it. Ballooning might manifest as high Disk I/O or latency. However, as you can see, the root cause is at the hypervisor level.

To avoid ballooning, you can create a ‘memory reservation’ for the virtual machine, guaranteeing an amount of physical memory.

Ballooning can lead to swapping, another memory management technique.

Memory Swapping

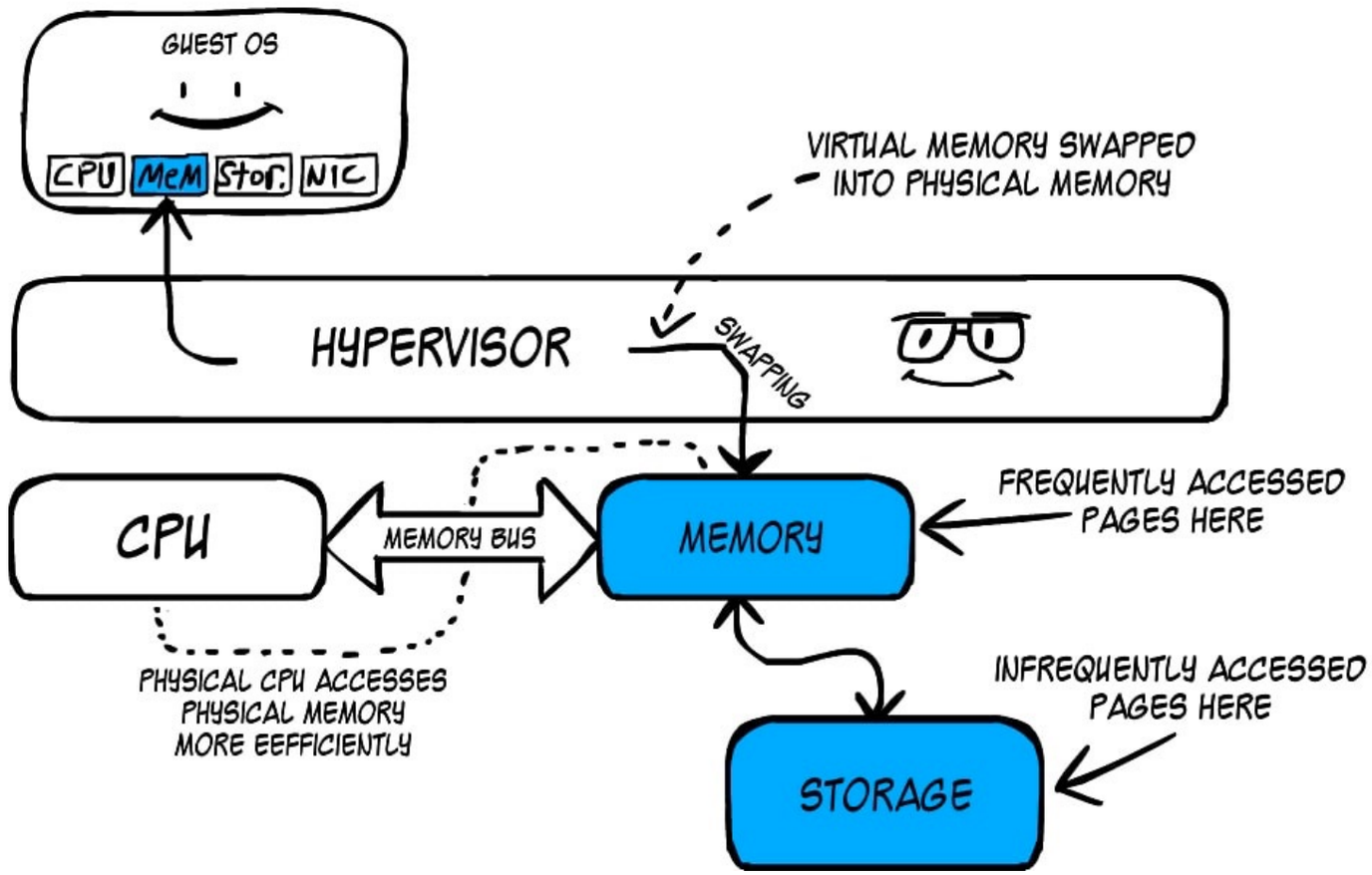
Ballooning can lead to swapping at the hypervisor layer, however, swapping also occurs at the Guest OS level.



Guest OS Swapping

Memory Swapping is not unique to virtual environments.

It comes from a time when physical memory (RAM) was extremely expensive.



Virtual memory is created on an attached disk, which emulates physical memory.

When the CPU is trying to access virtual memory, the memory page is swapped into physical memory, where the CPU can access and manipulate it efficiently. Applications often access a small number of pages -- the WORKING SET --

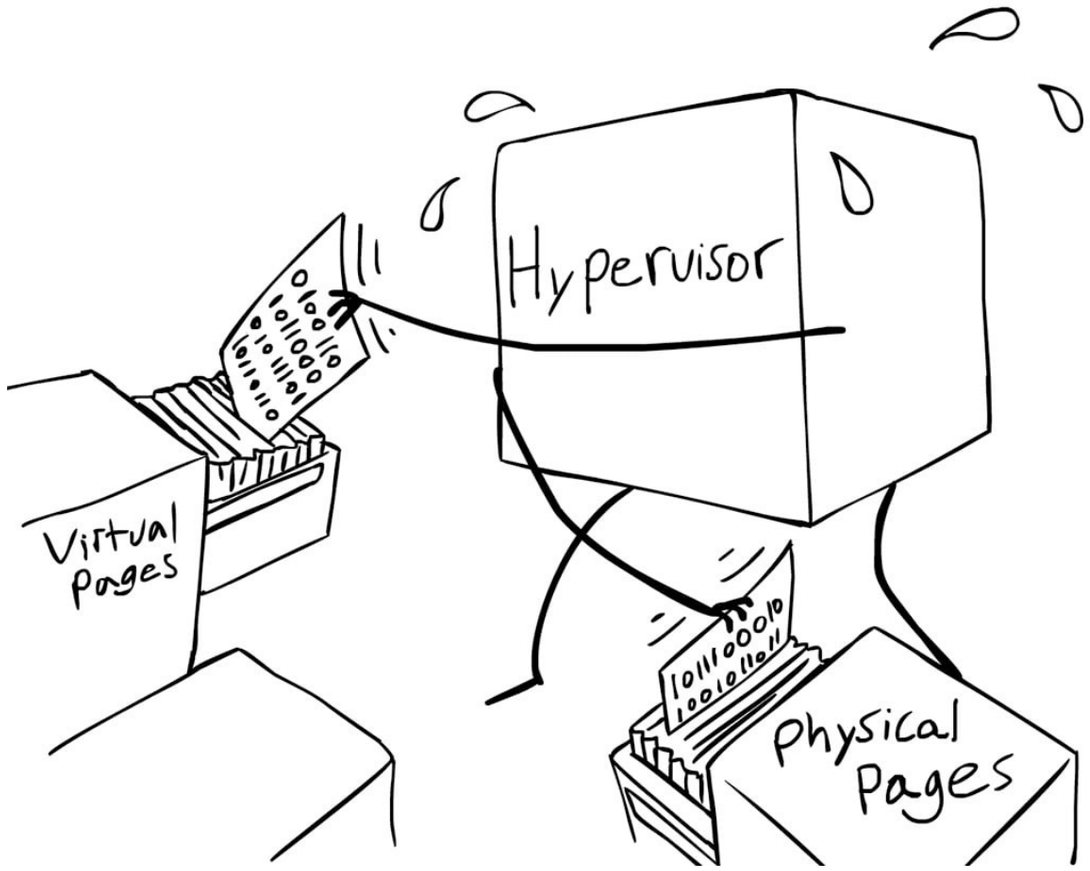
frequently while using other pages sporadically or rarely. Swapping mechanisms typically keep the working-set in physical memory while other pages may be swapped to storage. Swapping increases the utilization of physical memory: frequently accessed pages reside mostly in physical memory, while infrequently accessed pages reside mostly in storage, not wasting physical storage.

This increase in utilization comes at a cost – swapping slows down computations, as pages are read from/written or swapped between storage and/or physical memory. Storage is much slower than physical memory. This leads to performance degradation.

In order to assure performance, the physical memory has to be tuned, so that it matches the working set. Where the physical memory exceeds the working set, memory will be underutilized. However, if the working set is more than the physical memory available, excessive swapping occurs, resulting in inevitable performance problems.

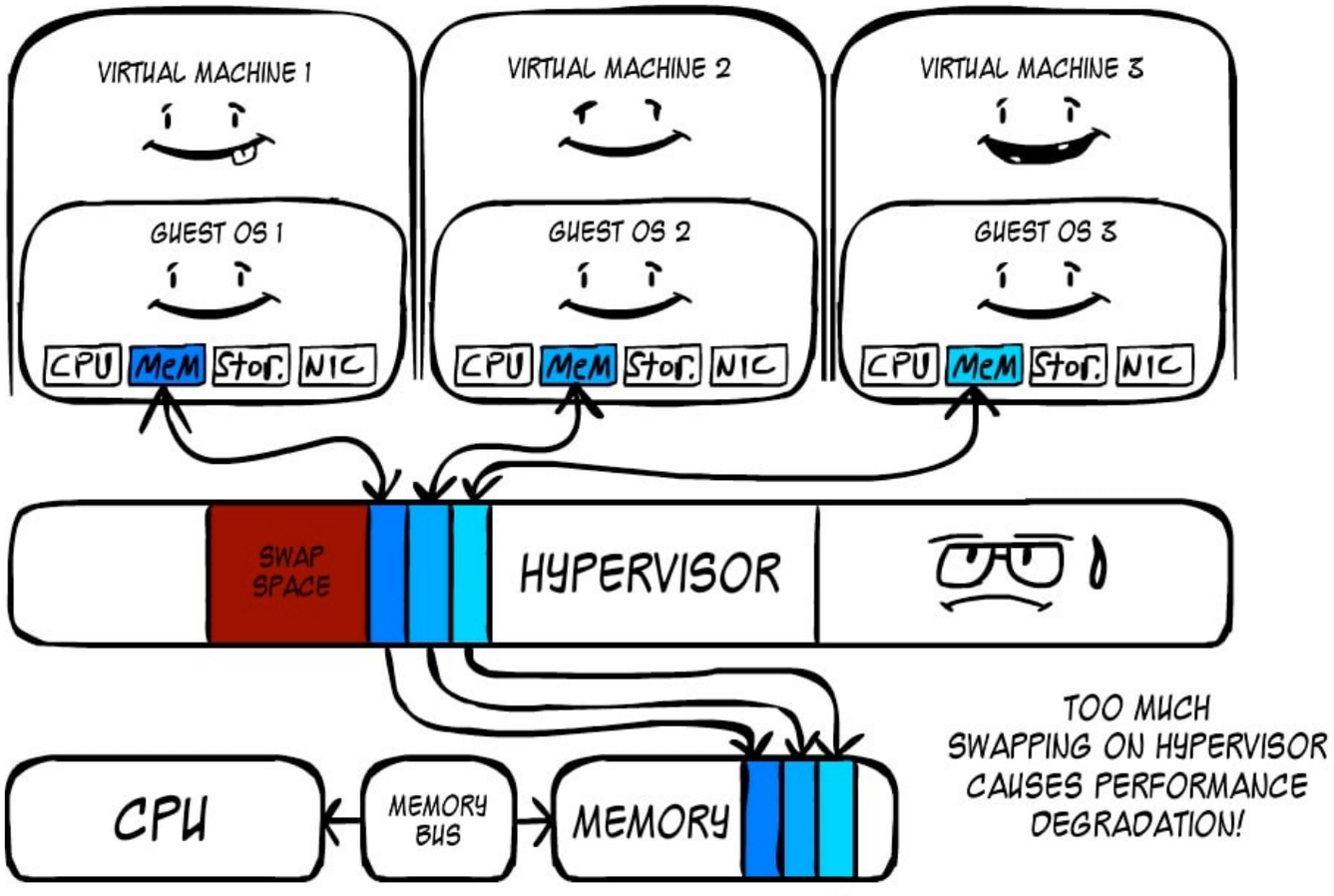
Hypervisor Swapping

When the Guest OS is accessing memory, the hypervisor has to swap the virtual pages into real physical pages.



Hypervisor Swapping

Guest OS swapping is preferable to hypervisor swapping, as the Guest OS is at least aware of the process.



The hypervisor is able to create swap space and keep memory pages there. Again, these memory pages emulate physical memory as far as the virtual machines are concerned. So when the Guest OS is accessing this memory, the hypervisor has to swap the virtual pages into real physical pages. The complexity in this scenario occurs because the Guest OS is completely unaware of the entire process. All the

VM’s residing on the hypervisor will start to experience performance problems, even those that do not perform a single swap. An administrator monitoring the Virtual Machine is oblivious to the root cause, unless he correlates the metrics from the guest with the hypervisor metrics.

Over-commitment, Ballooning and Swapping help to maximize efficiency of physical memory available to the hypervisor, but have to be carefully tuned and controlled, in order to prevent performance problems.

Traditional Techniques Have Limits

Over-commitment, Ballooning and Swapping help to maximize efficiency of physical memory available to the hypervisor, but have to be carefully tuned and controlled, in order to prevent performance problems.

Traditional Techniques Have Limits

Manual (monitoring) approaches rely on humans to make sense of the complexity.

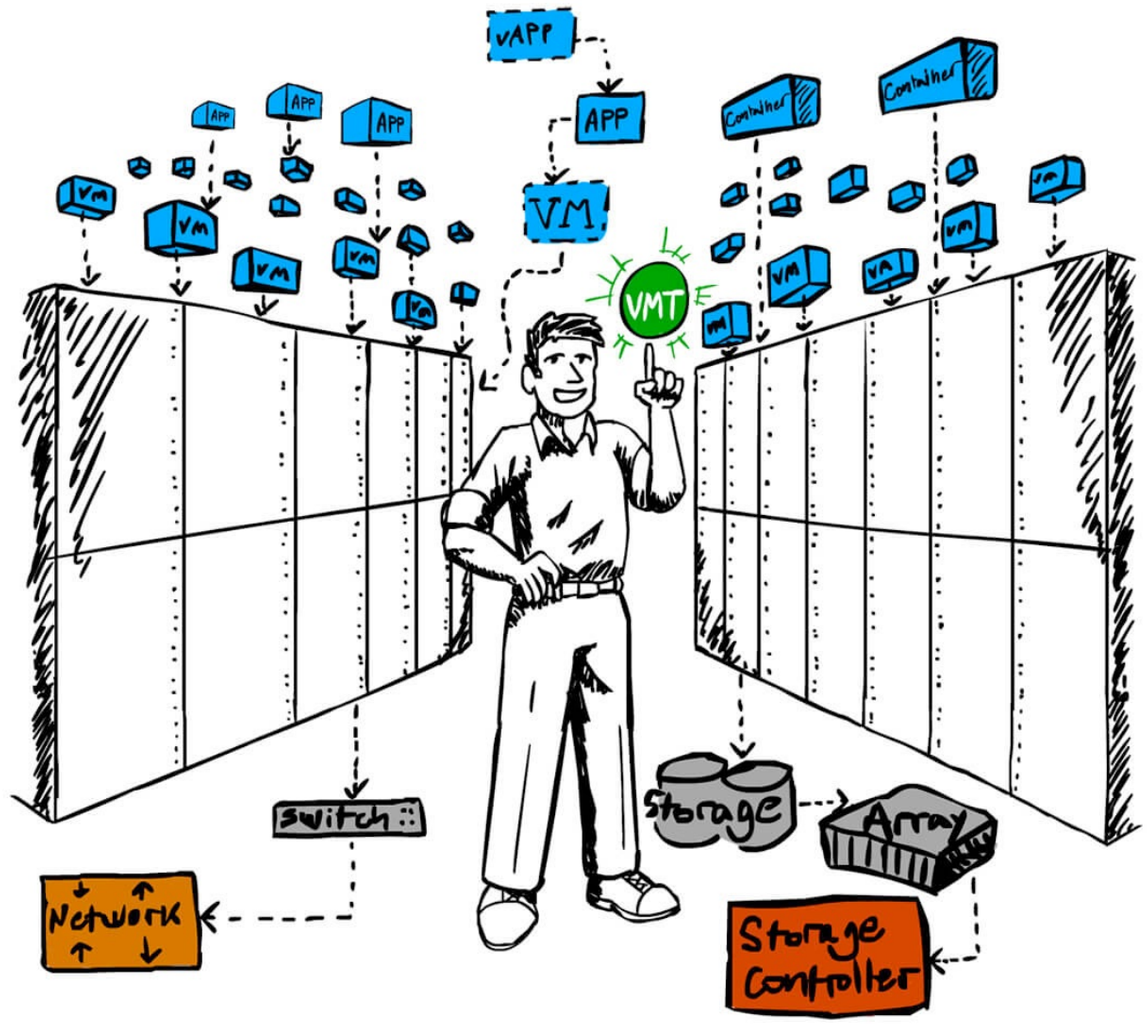
These techniques do not scale.

Traditionally, techniques such as memory reservation are used to guarantee performance of the most important workloads. The environment is then closely monitored, and metrics from the Guest OS and Hypervisor are correlated to identify the root cause of any problems once they occur. Ballooning and host swapping occur at the Hypervisor level, however, might manifest in problems at the Virtual Machine level.

However, most environments are NOT static. So as things change, Virtual Machines move around, as application demand spikes and dips, the environment has to be constantly tuned to adequately guarantee performance. Manual management becomes very cumbersome, and as the environment grows and additional complexity is introduced, this becomes impossible for a human to manage.

Economic Abstractions Enable Scaling

Solving the memory management challenge demands a new understanding of the data center and its dynamics.

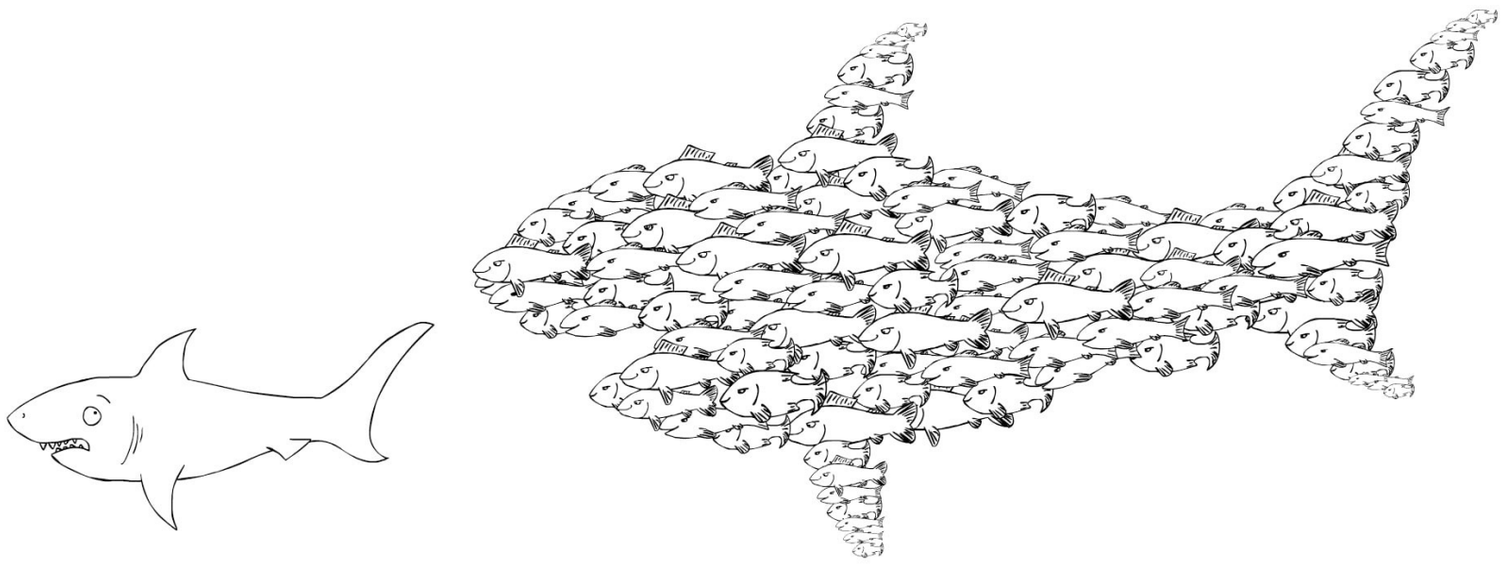


Economic Abstractions Enable Scaling

Break the big complex problem into a series of decentralized small problems that are easily managed by software.

Finally, a solution that scales.

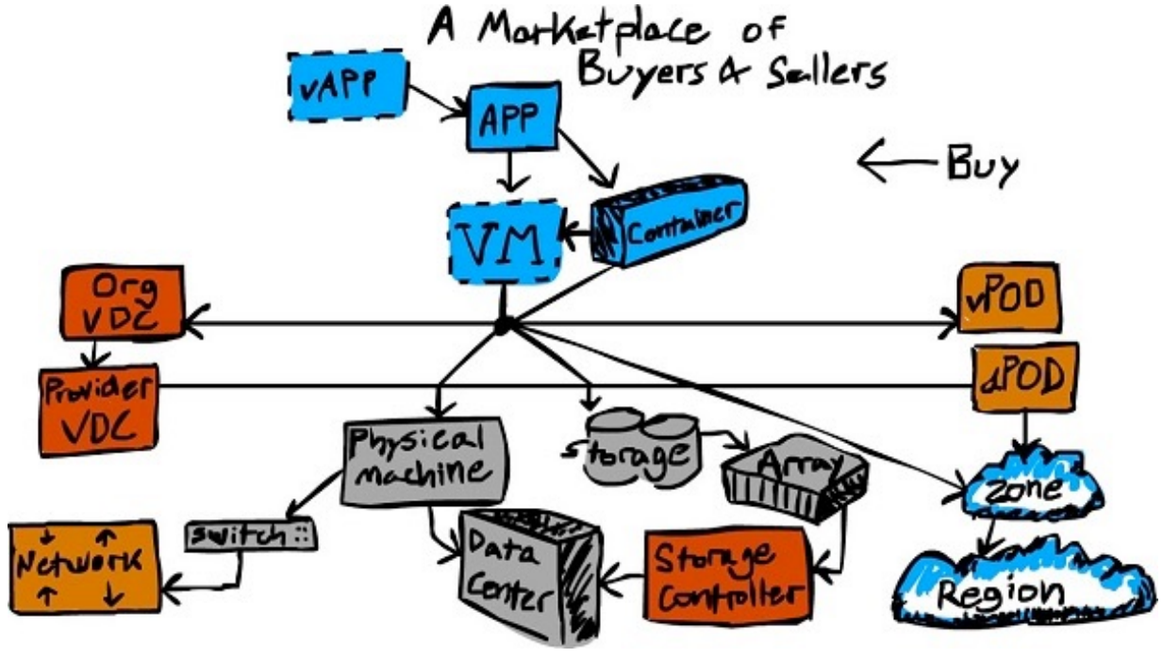
VMTurbo deploys as a VM in the environment and models the data center after an economic market—workload (applications, VMs, containers) are “buyers” and compute, storage, fabric, etc. are “sellers.” The software’s market-based algorithms are intrinsically decentralized and modular, breaking the complexity down to a continuous series of exchanges between any two entities (buyer and seller). However infinite the exchanges, these entities simply follow the supply and demand principles of economics to work out placement, sizing, and stop/start decisions among themselves. VMTurbo essentially creates an “Invisible Hand” in the data center.



VMTurbo's economic model involves two sets of abstractions:

1. Modeling the virtualized IT stack as a service supply chain, where components (e.g., VMs) consume services of other components (e.g., physical hosts) and offer services to their consumers (e.g., guest OSes and applications).

2. Using pricing mechanisms to match demand of services with supply along this supply chain. Resource services are priced to reflect imbalances between supply and demand, and drive resource allocation decisions. For example, a bottleneck, reflecting excess demand over supply, will result in raising prices of the respective resource. Applications competing over the resource will shift their workloads to alternate resources to lower their costs, resolving the bottleneck.



How Do Economic Abstractions Address Memory Management?

Using its patented economic scheduling engine, VMTurbo looks at this problem using the following principles:

1. The goal is to keep the entire environment in a healthy state. This means that excessive swapping is avoided, either at guest or hypervisor level.

2. Resizing is employed to increase the margins of physical to virtual memory. Virtual Machines are allocated memory based on their needs, eliminating guess work. By allocating memory based on demand, this means that the available infrastructure supply is matched with application demand to ensure that all the Virtual machines make use of physical memory in the most efficient way.

3. Resource usage is kept away from unhealthy states by providing actionable recommendations (which can be automated), as the environment approaches this 'forbidden region'.

    - For example, if a Guest OS experiences increasing levels of swapping, it will either be allocated more physical memory from the hypervisor, or migrate to another hypervisor – making the best decision based on the current state of the whole environment.

    - Where Hypervisor swapping starts to increase, the hypervisor increases the 'price' of memory, which leads to Virtual Machines migrating to another Hypervisor, and reduces swapping

    - Ballooning also increases memory prices, triggering either a migration of Virtual Machines, or tuning of balloon parameters.

4. The pricing mechanism co-ordinates swapping at both guest and hypervisor level. Any specific business policies and higher priority workloads can be added to the logic of VMTurbo, providing the best possible decisions at any point in time.

No need to monitor, analyze or mater memory management techniques, interactions and problems. Using an automated control mechanism, performance degradation is avoided altogether. At the same time, efficiency is maximized by identify underutilized resources.

Memory is only one of the many aspects of the environment you have to keep your eye on. Throw in CPU, storage (IOPS and latency), network and this slowly becomes an n-dimensional problem. Watch: The Application Performance Challenge.

About VMTurbo

VMTurbo's Demand-Driven Control platform enables customers to manage cloud and enterprise virtualization environments to assure application performance while maximizing resource utilization. VMTurbo's patented technology continuously analyzes application demand and adjusts configuration, resource allocation and workload placement to meet service levels and business goals. With this unique understanding into the dynamic interaction of demand and supply, VMTurbo is the only technology capable of controlling and maintaining an environment in a healthy state.

The VMTurbo platform first launched in August 2010 and now has more than 30,000 users, including many of the world's leading money center banks, financial institutions, social and e-commerce sites, carriers and service providers. Using VMTurbo, our customers, including JP Morgan Chase, Salesforce.com and Thomson Reuters, ensure that applications get the resources they need to operate reliably, while utilizing their most valuable infrastructure and human resources most efficiently.