

# ACCELERATING VMWARE DEPLOYMENTS AND VM PERFORMANCE THROUGH ITRENEW STORAGE ARRAYS

## *Abstract*

The IT world has changed exponentially over the past 10 years through server and storage virtualization. Though the concepts have been sound solutions from functional perspective, there have been struggles with respects to performance of virtualized vs. un-virtualized environments. Virtualizing 5 servers will consume upwards of around 35% of additional CPU and upwards of around 50% more memory than independent physical servers running the same applications and workloads. The VMware based infrastructures of today continue to scale out live production deployments thus causing a detrimental effect on running applications whenever there is a configuration change or addition made to the VMware datacenter. Cloning VMs can consume significant CPU cycles and memory that can increase the I/O latency of the storage or handicap the database's operations per minute.



## **Introduction**

Performance on virtualized servers should not be impacted by the storage subsystems. Traditional Storage Arrays do not interact with the host operating environments for the applications. Without the link of communication such as copying blocks, zeroing out blocks, or locking blocks we see performance of the hypervisor is not efficient.

Using these Storage Arrays lead administrators to schedule configuration changes during off hours and sometimes having to have non-production equipment to model the impact on the production equipment. This all usually amounts to additional overtime and resource expenses.

## **Solutions to the Performance and Scaling Problem**

### ***vStorage APIs for Array Integration (VAAI)***

VMware has developed a unique API for hardware storage vendors to implement into their storage systems. The VAAI allows specific storage operations to be offloaded from the host ESX servers down to the Storage Array itself, which results in less CPU, memory, and traffic on the iSCSI or FC network.

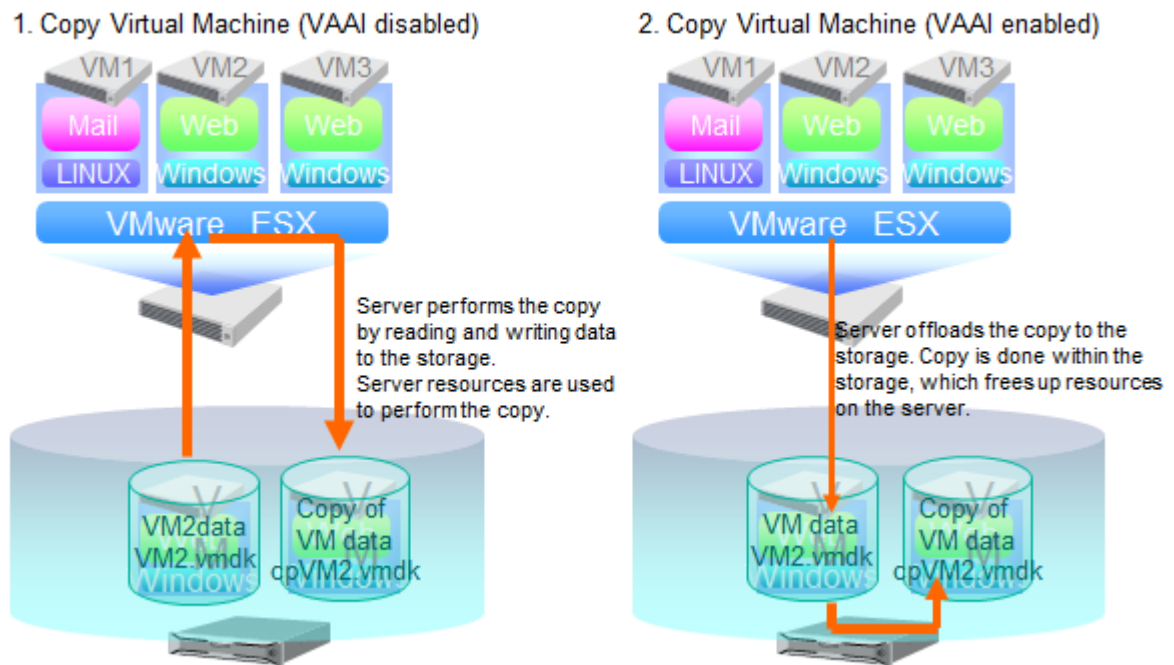
The ITRENEW Storage Arrays currently implement 3 of the VAAI primitives:

- Fast/Full Copy
- Write Same/Zero
- Hardware Offload Locking

### ***Fast and Full Copy Primitive***

When a Virtual Machine is created from a template, the source template data is read from the Storage Array for where the datastore resides and is then written to the new datastore where the new VM will be located. This process is rather simple but will cause an enormous burden on the ESX host CPU and memory that should be addressing incoming VM requests.

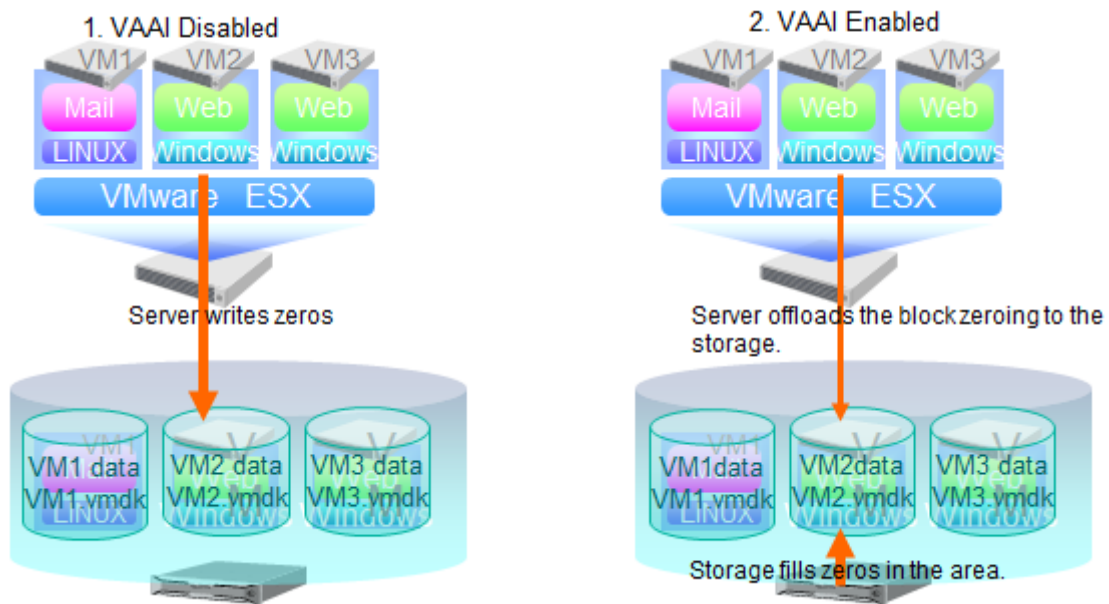
With the Fast/Full Copy primitive, VMware uses the VAAI to simply offload the task to the Storage Array without consuming valuable CPU cycles. The ITRENEW Storage Array will block copy the VM from either one block range of the LUN to another block range, or from one LUN to another LUN which essentially bypasses much of the induced overhead of the virtualized infrastructure and goes at the maximum speed the Storage Arrays can deliver. The overall VM creation process is 10 to 15x faster with VAAI than without and there is no impact to the production VMs.



### **Write Same and Zeroing Primitive**

When a Virtual Machine is cloned, the process will include cloning the empty space of the virtual disk as well as the data within the virtual disk. The VM will generally be created with a zeroing function indicating all of the available disk space created on the fat disk will be zeros. The cloning process traditionally clones every block that will be maintained by the ESX host and will generate an enormous amount of SCSI commands to complete the task.

With VAAI, the process of the clone is radically improved by offloading much of the process to the Storage Array by issuing the WRITE\_SAME SCSI Block Command which improves the overall efficiency of the storage layer.



### ***Hardware Offload Locking***

Since the initial release of VMware's VMFS clustered file system, it has been based on SCSI Reservations. As VMware has evolved and it has revolutionized the overall compute architecture to be scalable in terms of CPUs and memory through additional ESX hosts. These hosts will access common storage volumes based on how the VM's resources are provisioned. It is more and more common to have many VMs residing on a common data store LUN with ESX server ownership for each VM's being separate.

To prevent data corruption from occurring, an ESX node will issue a RESERVATION lock for the LUN during the write operations. During this time, other ESX nodes will have READ ONLY access to the LUN until the RESERVATION lock has been released. Needless to say, it was not good practice to have too many Virtual Machines residing on the same LUN when using clustered ESX nodes. As the LUNs are getting larger these days and more and more storage is becoming virtualized, it is hard not to consolidate VMs into common data stores.

The VAAI offers a more efficient model for accessing LUNs that are clustered with ESX nodes. Rather than issuing the RESERVATION locks on the per LUN basis, it is now being issues down to the block level. Multiple ESX nodes can have concurrent write access to the same LUN.

With the VAAI hardware offload support, clustered VM performance will be dramatically improved as well as the practice of reducing the number of VMs per data store can be eliminated by actually increasing the number of VMs per data store. Simple operations introduced by vMotion, VM power on/off, and normal VMFS metadata I/O are improved.