The Goodness of Pure Storage FlashArray and VMware VAAI integration

You might be wondering what is the point of integration with any hypervisor when you have a fast storage. You should be able to do every operation faster, then why bother? Right? Not so fast, if you want to know the benefits of Pure Storage FlashArray with VMware VAAI block primitives integration keep reading.

What is VAAI?

VMware vSphere Storage APIs for Array Integration (VAAI) is a well-known feature set of vSphere 4.1 and later versions. You can find a great discussion on this topic from our VMware Storage guru, Cormac Hogan(@CormacJHogan) here.

In short, VAAI is a set of hardware offload APIs that help array vendors implement a set of storage primitives to perform common operations, such as cloning, data movement, metadata locking, and block zeroing. The benefit of these offload operations is to let the storage array handle the bulk of the tasks, freeing up the host CPU from mundane operations.

What are the VAAI Primitives?

Lets take a quick look at each of the VAAI block primitives.

1. Atomic Test and Set (ATS or Hardware accelerated locking): ATS is implemented in Purity (the operating system on Pure Storage FlashArray) as T10-compliant SCSI command Compare and Write (Operation Code 0x89). ATS enhances metadata operations including cluster heartbeat (datastore heartbeat), and avoids the serialization of LUN locking for every update. Furthermore, ATS enables multiple VMs to perform metadata updates with ease. This allows us to deploy thousands of VMs on a very large (64TB) datastore without incurring any performance penalty. We had demonstrated this benefit in our VMworld demo (see this blog post).
2. Block zero (Hardware accelerated Init): Block zero is implemented in Purity as T10-complaint SCSI command: Write same (Operation Code 0x93). The best part of of the block-zero feature is that the operation of zeroing a vmdk either during provisioning (EZT) or on-demand (LZT) is lightening fast because we just update our metadata instead of writing zeroes to the MLC drives (Writing zeroes to flash seems downright silly).
3. Extended copy or XCOPY (Hardware accelerated copy): XCOPY is implemented in Purity as T10-compliant SCSI command XCOPY (Operation Code 0x83). We have done a lot of work in this area to integrate our native ZeroSnap snapshot technology with the XCOPY operation to give you unbelievable efficiency through cloning or storage vMotion operations. For example, when a VM template is cloned, the data is not copied. Instead, a ZeroSnap snapshot that represents the VM is taken and the metadata is updated. This results in space savings and the reduction of time needed to perform the actual data copy. The following figure shows a 50GB VM template clone.

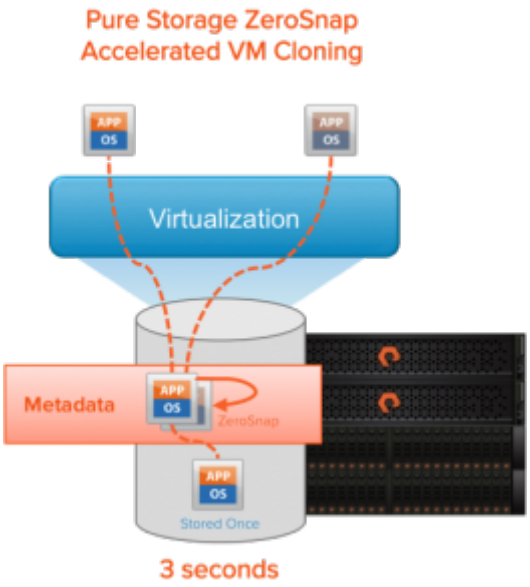


Figure: Pure Storage ZeroSnap and VAAI XCOPY primitive integration for accelerated VM cloning

4. Dead space reclamation (Block Delete or SCSI UNMAP) : The SCSI UNMAP command is supported on the Pure Storage FlashArray and we recommend to turn it on to reclaim freed up space. This is the only way any changes to the datastore are propagated to the LUN underneath. Most disk vendors' best practice is to turn this feature off as Block Delete will kill the compute engine (max out the CPU during block delete operations) and adversely affect I/O performance. But we strongly encourage you to turn it because we just need to update our metadata: a background job will collect the freed blocks. Our approach is much simpler and does not tax the CPU when freeing data blocks.

5. Thin provisioning Stun (TP threshold) : Pure Storage LUNs are thin provisioned by default and support for the STUN API will be released shortly. Stay tuned.

All these VAAI block primitives are compliant to T10 SCSI standard. From a Pure Storage FlashArray perspective, we support all the APIs. All these features works right out of the box – there are no special VIBs to install on the ESXi server.

OK – Enough theory for now. Let's take a look under the hood.

We can verify the support for VAAI primitives by running the command –

~ # esxcli storage core device vaai status get -d naa.624a937034be0128252023820001000f
naa.624a937034be0128252023820001000f
VAAI Plugin Name:
ATS Status: supported
Clone Status: supported
Zero Status: supported
Delete Status: supported

where the device naa.624a937034be0128252023820001000f is a Pure Storage LUN.

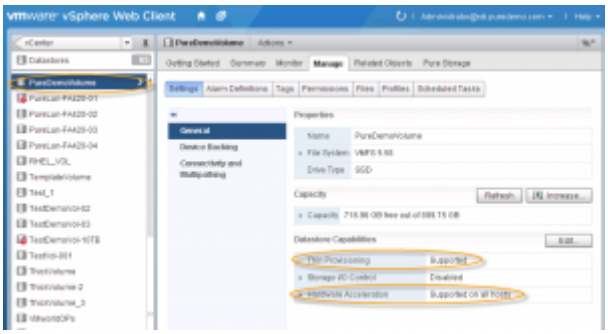
~ # esxcli storage nmp device list -d naa.624a937034be0128252023820001000f
naa.624a937034be0128252023820001000f
Device Display Name: PURE Fibre Channel Disk (naa.624a937034be0128252023820001000f)

Storage Array Type: VMW_SATP_ALUA
Storage Array Type Device Config: {implicit_support=on;explicit_support=off; explicit_allow=on;alua_followover=on;{TPG_id=0,TPG_state=AO}}
Path Selection Policy: VMW_PSP_RR
Path Selection Policy Device Config: {policy=rr,iops=1,bytes=10485760,useANO=0;lastPathIndex=2: NumIOsPending=0,numBytesPending=0}
Path Selection Policy Device Custom Config:
Working Paths: vmhba3:C0:T2:L21, vmhba2:C0:T1:L21, vmhba3:C0:T3:L21, vmhba2:C0:T2:L21, vmhba2:C0:T3:L21, vmhba3:C0:T0:L21, vmhba3:C0:T1:L21, vmhba2:C0:T0:L21
Is Local SAS Device: false
Is Boot USB Device: false

Notice the Pure Storage LUN automatically get claimed as an ALUA SATP and the only setting we do is to set the multi-path policy to Round Robin.

~ # esxcli storage core device list -d naa.624a937034be0128252023820001000f
naa.624a937034be0128252023820001000f
Display Name: PURE Fibre Channel Disk (naa.624a937034be0128252023820001000f)
Has Settable Display Name: true
Size: 10485760
Device Type: Direct-Access
Multipath Plugin: NMP
Devfs Path: /vmfs/devices/disks/naa.624a937034be0128252023820001000f
Vendor: PURE
Model: FlashArray
Revision: 332
SCSI Level: 6
Is Pseudo: false
Status: on
Is RDM Capable: true
Is Local: false
Is Removable: false
Is SSD: true
Is Offline: false
Is Perennially Reserved: false
Queue Full Sample Size: 0
Queue Full Threshold: 0
Thin Provisioning Status: yes
Attached Filters:
VAAI Status: supported
Other UIDs: vml.0200150000624a937034be0128252023820001000f466c61736841
Is Local SAS Device: false
Is Boot USB Device: false

From the vCenter perspective, we see Hardware acceleration is enabled for Pure LUN and the disk type is SSD (VMware Webclient View).



We can also verify the VAAI primitives is enabled either by checking the advanced settings in the VMware webclient or by running the following commands:

esxcli system settings advanced list -o /DataMover/HardwareAcceleratedMove
esxcli system settings advanced list -o /DataMover/HardwareAcceleratedInit
esxcli system settings advanced list -o /VMFS3/HardwareAcceleratedLocking
esxcli system settings advanced list -o /VMFS3/EnableBlockDelete
In order to set them you can run the following commands:

esxcli system settings advanced set –int-value 1 -o /DataMover/HardwareAcceleratedMove
esxcli system settings advanced set –int-value 1 -o /DataMover/HardwareAcceleratedInit
esxcli system settings advanced set –int-value 1 -o /VMFS3/HardwareAcceleratedLocking
esxcli system settings advanced set –int-value 1 -o /VMFS3/EnableBlockDelete
To speed up the XCOPY operation by four folds, we can set the value of MaxHWTransferSize to 16MB instead of the default 4 MB by doing –

esxcli system settings advanced list -o /DataMover/MaxHWTransferSize
esxcli system settings advanced set –int-value 16384 –option /DataMover/MaxHWTransferSize
Note if there was a way to set a higher value we could literally copy the entire VM in one pass.

Benefits of VAAI integration

The benefits of VAAI integration can be summed up as follows-

1. ATS helps in having bigger clusters (max nodes in vSphere 5.5 cluster is 32) with large number of VMs.

2. ATS also helps in hosting many more desktops in a cluster and not have to worry about VMs/datastore, as there is no lock serialization. Given the processors and memory in a host have increased so much this really helps to harness every CPU cycle and memory capacity, i.e consolidate more.

3. Block zero primitives help in faster EZT vmdk creation and also zeroing of block with LZT vmdk type.

4. Fault tolerance needs all the vmdk in the VM to be EZT and block zero helps in faster FT VM creation.

5. XCOPY helps in speeding up cloning VM from templates, and faster storage vMotion with the Pure Storage ZeroSnap integration.

6. For VMware vCAC deployment where 100s of VMs are created and deleted on a daily basis, the fast cloning becomes a very efficient way for deploying applications. We have customers creating 400 Windows/Linux VMs within an hour and running simulations on Pure Storage FlashArray in their vCAC environments.

7. Last but not the least, the block UNMAP primitive helps to reclaim dead space and help prevent vmdk bloating.

If this blog has piqued your interest in the various benefits of Pure Storage VMware VAAI integration, drop us a line, we would be happy to schedule a demo and go over a virtualization benefits of Pure Storage FlashArray.

Courteous comments and feedback are always welcome.