HIAG DATA

Successfully Running CloudStack with High-Performance Workloads Using Managed Primary Storage

Berlin, 13.06.2018 / Andrija Panić HIAG Data / Mike Tutkowski NetApp SolidFire

Storage history within HIAG Data

PURE KVM, ACS 4.3+ (→ 4.4 → 4.5 → 4.8.x.x)

- CEPH
  - Great scalability, great redundancy, zero SPOF
  - Low TCO
  - Already available integration in CloudStack
- NFS HYBRID (HDD+SSD)
  - Limited scalability (DIY), not SPOF-free
  - Acceptable TCO
  - Native (best) available integration in CloudStack (less bugs)
  - Somewhat better performance (relative thing), no QoS
- NFS SSD
  - Limited scalability (DIY), not SPOF-free
  - Not so low TCO
  - Native (best) available integration in CloudStack (less bugs)
  - Good performance, still no QoS

Storage challenges (CEPH and NFS)

- STABILITY
  - MOSTLY FIXED IN THE MEANTIME !
- PERFORMANCE
  - MAY VARY IN YOUR SETUP
- USABILITY AND MANAGEMENT
  - SOME FIXED, SOME STAY AROUND

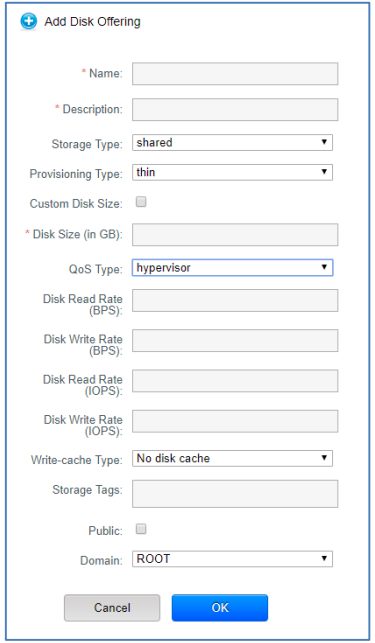
Storage challenges

- STABILITY
  - Exceptions in librbd (RBD client library) - agent crash (VM HA sometimes kicks in)
    - i.e. resize DATA volume
    - i.e. delete volume with more than 16 snapshots on
    - i.e. delete multi-TB volume (timeout, libvirt unresponsive, agent down, VM downtime)
    - **mostly fixed in the meantime!**
  - CEPH rebalancing and deep scrubs can impact client IO
  - Flapping MON and OSD due to network issues
    - “slow requests, oldest blocked for > 30sec”
  - Volume snapshotting can take long time to complete and can often block job queue for other items for the KVM agent (“executeInSequence” issue)
  - NFS = ZFS-based kernel panic = all Agents rebooted due to kvmheartbeat.sh script
- PERFORMANCE
- USABILITY AND MANAGEMENT

Storage challenges

- STABILITY
- PERFORMANCE
  - CEPH rebalance, deep scrubs and flapping OSD can sometimes cause "slow requests, oldest blocked for > 30sec" = support tickets
    - throttle rebalance process (takes days, but has low IO impact)
    - disable deep scrubs (any better solution !?)
    - replace OSD (or network devices!)
  - No configurable (CEPH) stripe size (in librados, yet) – i.e. stripe_size = object_size = 4MB = can take penalty on performance during smaller IO
  - Non-existing CloudStack CEPH snapshots removal code caused hundreds of snaps to exist for some volumes, causing extreme performance penalty
    - Desperate man partial relief = disable ACS hourly snapshots
    - Implement proper CEPH snapshot removal code (our humble contribution)
- USABILITY AND MANAGEMENT

Storage challenges



- STABILITY
- PERFORMANCE (cont.)
  - Noisy neighborhood if no QoS used
  - Basic “QoS” via "Hypervisor QoS" on Disk Offerings
    - Can just limit read/write rate in either B/s or IO/s per volume
    - Not applied to hot-plugged volumes – full throttle ahead ! ☺
    - Not real QoS, more like “damage control” of a kind
  - Theory vs. practice exercise:
    - Theory: imagine 100 x pure sequential IO streams (100 clients), any storage excels at sequential IO, right ? (i.e. rbd caching on KVM, 100 VMs)
    - In practice this becomes 100% random IO on storage backend, because all IO streams are eventually interleaved !!!
  - HDD solutions eventually become useless (even with all magical caching!)
  - Raw random IO power is needed (aka SSD)
- USABILITY AND MANAGEMENT

Storage challenges

- STABILITY
- PERFORMANCE
- USABILITY AND MANAGEMENT
  - Snapshot life-cycle issues
    - 50+ snapshots on CEPH = serious performance issues (i.e. RBD layering issue)
      - temp relief via custom snapshot deletion script
      - permanent fix with proper ACS code implemented
    - Having more than 16 snapshots on CEPH (hourly snapshots?), can cause agent down (during volume deletion), VM HA actions = downtime
      - fixed with rados-java 0.2.0 - many thanks to Wido!
  - Snapshots in practice
    - For NFS and CEPH, making snapshot can take forever (2TB volume ?) due to copying it from Primary Storage to Secondary Storage
      - failure can cause broken snapshot scheduler!

Storage challenges

- STABILITY
- PERFORMANCE
- USABILITY AND MANAGEMENT
  - Snapshots in practice (cont.)
    - For NFS and CEPH, it's not possible to directly revert back volumes from snapshot
      - workaround to restore ROOT volume from snapshot - convert snapshot to template(ROOT) then creating new VM from snapshot – takes time
      - workaround to restore DATA volume from snapshot – convert snapshot to volume, then attach new DATA volume – takes time
    - Downloading CEPH-based volume from ACS = RAW file extension although image is really QCOW2 format & Deploying VM via ISO will incorrectly mark ROOT volume type in DB (QCOW2 instead of RAW) causing different snapshot/template creation issues
      - Locally fixed, and should also be fixed upstream in the meantime
  - Volume migration between different Primary Storages
    - For KVM, so far there was no such thing as online storage migration, like Storage XenMotion for XenServer and Storage vMotion for VMware

Storage challenges

- STABILITY
- PERFORMANCE
- USABILITY AND MANAGEMENT
  - Volume migration between different Primary Storages (cont.)
    - For CEPH and NFS (KVM) only, offline storage migration is supported (stop VM, migrate volumes, start VM) - long downtime for customer
  - “Special” case - regular VM Live Migration never completes on busy VMs
    - High RAM change rate - phase 2 live migration never completes, phase 3 never reached
      - due to above, can't put ACS host in maintenance mode, which means cloud-wide management issues in some cases (blocked Kernel/Qemu upgrades, etc.)
    - Can be reproduced easily on Windows (Prime95) or Linux (stressapptest), i.e.
      - i.e. Ubuntu 14.04 VM, 4 vCPU (2Ghz), 32GB RAM, “apt-get install stressapptest”, 10G networking
      - live migrate while **idle** ~ **7 sec** (10sec as observed via Apache CloudStack GUI)
      - live migrate **while "stressapptest** -M 1024 -s 1000**"** - **never finishes** (no auto-convergence)
        - observe: “*watch –n 0.5 “virsh domjobinfo i-XX-YYY-VM”*” – phase 2 repeats forever…
    - Dynamic Autoconvergence helps
      - live migrate while "stressapptest -M 1024 -s 1000" ~ **45 sec** (dynamic auto-convergence)
      - stressapptest RAM write rate: **12 GB/s**!!! (so dynamic auto-convergence obviously works ☺)

HOW DO WE SOLVE ALL THESE CHALLENGES ?

HOW DO WE SOLVE ALL THESE CHALLENGES ?

Stop expecting high-IO performance

HOW DO WE SOLVE ALL THESE CHALLENGES ?

Stop expecting high-IO performance

Delete “QoS” from dictionary

HOW DO WE SOLVE ALL THESE CHALLENGES ?

Stop expecting high-IO performance

Delete “QoS” from dictionary

Snapshots are thing of a past

HOW DO WE SOLVE ALL THESE CHALLENGES ?

Stop expecting high-IO performance

Delete “QoS” from dictionary

Snapshots are thing of a past

Anyone still using live migration ?

HOW DO WE SOLVE ALL THESE CHALLENGES ?

Stop expecting high-IO performance

Delete “QoS” from dictionary

Snapshots are thing of a past

Anyone still using live migration ?

OR …

SOLIDFIRE - General Info

STABILITY / PERFORMANCE / USABILITY & MANAGEMENT

- SolidFire QoS is defined by 3 values: min_iops, max_iops, burst_iops
  - Min IOPS == guarantied IOPS all the time (i.e. 500 IOPS)
  - Max IOPS is what volume can get 99% of the time (i.e. 2K IOPS)
  - Burst IOPS is allowed for short periods of time (i.e.10K IOPS; VM reboots, short file transfer)
- 100% guaranteed IOPS (min_iops) in both normal and in most fault conditions
  - During cluster expansion, repair and maintenance (i.e. node reboot/upgrade, cluster shrinking/growing, replacing dead SSD, etc.), during SSD health checks
- SF snapshots stay on SF - creation and restore from snapshots are instant !
  - No more blocked job queue, no unneeded network traffic and IO to Secondary Storage NFS
  - No issues with large number of snapshots
- Mastering SF management skills is very easy - all underlying work is done by SF support team, including cluster upgrade, monitoring, alerting, etc.
  (think of this as MANAGED STORAGE in every possible way)
- Last but not least - no external libvirt library – no libvirt issues (agent disconnections)

SOLIDFIRE – Recent Improvements

Some of the improvements/features that were done recently

- Creating a template directly from a volume (i.e. no need to go via snapshot)
- Reinstall VM from template (was missing)
- Download volume from ACS (qcow2) (was missing)
- Offline storage migration (was missing)
- Online storage migration (was missing, new in ACS@KVM world)
- Proper KVM auto-convergence implemented (was missing)
- Cleanup of iSCSI target on VM shutdown (general iSCSI improvements)
- Set min/max IOPS on volume upload (improvement)
- Etc...

Feature-wise, we feel 1:1 in comparison to non-managed storage (NFS/CEPH) – i.e. no major features missing

SOLIDFIRE – Snapshot Management

PROBLEM WITH NON-MANAGED STORAGE

- Customers heavily rely on snapshots, but...
- Lack of real snapshots (can take hours to create)
- Lack of ability to restore easily (can just create volume/VM copies)
- Internally, we even renamed this feature to “Backup”, as more appropriate term

SOLUTION

- Leverage SolidFire snapshots, which stays on SolidFire storage
  - Real snapshots - instantly create, instantly restore from
  - Virtually no space consumption on SF side (simply replicas of volume metadata)
  - No need to move hundreds of GB over network
  - No need to write hundreds of GB to Secondary Storage
  - No blocked job queue in CloudStack !

Demo: SolidFire Snapshot Create and Restore

Short demo/video: (3min):

- We have a running VM with some "important" data
- Make sure we are on the safe side – make volume snapshot
- We demonstrate data loss – delete this “important” data
- Let’s restore the volume from snapshot – restore data
- Back in business ☺

Demo1: SolidFire Snapshot Create and Restore: https://youtu.be/HRpRvvX4diw

SOLIDFIRE– QoS and Guaranteed performance

PROBLEM WITH NON-MANAGED STORAGE

- No really usable QoS with non-managed storage
- Can only throttle read/write via Hypervisor – b/s or IOPS, but in reality not good enough
  - Hot-plugged volumes have no QoS applied, so single customer VM can potentially kill storage performance
  - No guaranteed performance, just trying to prevent "noisy neighborhood" scenario
  - Need to magically monitor and voodoo-understand if your storage needs IO boost ☺
  - Demanding clients are unhappy most of the time (lack of consistent performance)

SOLUTION

- Leverage QoS and guaranteed performance per volume on SolidFire
- Zero possibility for "noisy neighborhood" scenario
- Clients get much more than they pay for, most of the time (MAX IOPS, BURST IOPS)
- Perfect monitoring environment which alerts you when more horsepower is needed

Demo: QoS on KVM vs. QoS on SF

Short demo/video: (4 min)

- We have running VM (ROOT disk on NFS), proper KVM QoS applied during VM start
- We attach (hot-plug) while VM running: 1 NFS drive, 1 CEPH drive and 1 SolidFire drive
- We observe lack of QoS for hot-plugged NFS and CEPH drives (and SF drive - by design)
- We observe QoS on backend SolidFire cluster for the SolidFire drive
- We stop VM, start VM – so KVM QoS is applied for NFS and CEPH drives
- We then explain applied QoS for NFS and CEPH DATA drives (and still no KVM QoS for SF drive - as expected, by design)

Alternate way to examine KVM QoS (iotune) parameters:

virsh blkdeviotune i-2-14-VM vda | grep -E "(read_bytes_sec|write_bytes_sec|read_iops_sec|write_iops_sec)"| grep -v max

Output:

- read_bytes_sec : 125829120
- write_bytes_sec: 125829120
- read_iops_sec : 1000
- write_iops_sec : 1000

Demo2: QoS on KVM vs. QoS on SF: https://youtu.be/ZAZps5QU17c

SOLIDFIRE – Migration from non-managed storage

PROBLEM TO SOLVE

- SolidFire cluster ready and accepting new volumes but ...
  - Old/existing volumes stay on non-managed storage
  - Migrate tens of hundreds of volumes to SolidFire?
- Offline storage migration
  - Logistically impossible (negotiate downtime slots with customer, migrate... repeat)

SOLUTION

- Online storage migration (new in 4.11, but we are using it in 4.8)
  - Watch out for libvirt 1.3.1 bug and set “vm.migrate.speed=2147483647” (max integer value) in agent.properties (otherwise migration speed = 1.3 MB/s)
  - VM’s volumes are migrated (mirrored) in parallel (qemu 2.5+) and at the end VM is live migrated
- But busy VMs never finish live migration !
  - Make use of Qemu auto-convergence (code + global config parameter) - thanks to Mike T.
  - Newer "dynamic auto-convergence” implementation (Qemu 2.5+) is a must
  - “Regular" auto-convergence (Qemu 1.6+) is useless in our case
  - Throttling 24 vCPUs enough to actually migrate heavily written (to) 60GB RAM - takes time

Demo: Qemu Dynamic Auto-Convergence vs. Extremely Busy VM

Short demo/video: (25 min > 3 min)

- We have a 24 vCPU VM with 60GB RAM
- We will use stressapptest utility to write to 59.5GB of RAM
  - “apt-get install stressapptest”
  - “stressapptest -M 59500 -s 10000 --pause_delay 10000”
- CPU usage in VM goes to 100% across all 24 CPUs (htop, top)
  - on host level == initially 75% of the total CPU utilization (32 Cores) via virt-top

- We will start VM live migration and observe low level details with “virsh domjobinfo VM”
  - “Memory processed” # so far migrated RAM to another host
  - “Memory remaining” # dirty pages (RAM) to be migrated/processed
  - “Memory total” # 60GB (total VM RAM)
  - “Dirty rate” # amount dirty pages per second (calculated from previous iteration)
  - “Iterations” # number of dirty page (RAM) migration iteration (effectively phase2)
  - After 27 iteration, 900GB RAM moved (24 minutes)– VM is migrated ☺

Demo3: Qemu Dynamic Autoconvergence vs. Extremely Busy VM:
https://youtu.be/sB0IOrCkzYE

Demo: Online Storage Migration (NFS/CEPH to SolidFire)

Short demo/video: (8 min > 4 min)

- Just functionality demo – i.e. idle VM, no workload inside VM
  - ROOT on SF, DATA on NFS, DATA on CEPH
- Online storage migration possible only via API/CloudMonkey (for now)
- https://cloudstack.apache.org/api/apidocs-4.11/apis/migrateVirtualMachineWithVolume.html
- Example command:
  - migrateVirtualMachineWithVolume
    virtualmachineid=88479b2d-57fe-427b-917d-0f3a62de2e50 # VM UUID
    hostid=0314ed18-bba8-480b-b6eb-e512d37c9f9c # destination host for live migration
    migrateto[0].pool=447820f9-157e-4e03-8365-0af19380c4e7 # SF UUID
    migrateto[0].volume=f67bc860-cc5d-45ac-82e4-1eb9cb04c047 # NFS volume UUID
    migrateto[1].pool=447820f9-157e-4e03-8365-0af19380c4e7 # SF UUID
    migrateto[1].volume=14067bf5-5b6b-42bf-aaf4-8764cdf16b3d # CEPH volume UUID
- Interesting moment:
  - 5'16'' VM live migration starts, after both volumes mirrored

Demo4: Online Storage Migration (NFS/CEPH to SolidFire):
https://youtu.be/Eo8BuHBnVgg

Demo: Online Storage Migration of Busy VM (RAM + IO Heavy)

Short demo/video: (46 min > 6 min)

- Demo showing joint effort of Qemu autoconvergence and online volume migration
  - VM with 24 vCPU, 60 GB RAM
  - ROOT on SF, 2 x DATA volumes on NFS
  - 2 x NFS = RAID0 (mdraid)
  - IO stress: FIO heavy write to RAID0 (limited to 150MB/s due to slow DEV NFS storage)
  - RAM stress: stressapptest with 59.5GB workload
  - Before we start migration, we show no “block jobs” on the source NFS volume
- During migration we show progress on the NFS mirroring/migration to SolidFire
- Once all volumes’ mirroring progress have reached 99-100%, VM Live migration starts
- We observe over 23 iterations of RAM copy (phase 2)
- Finally VM is live migrated (after ~1.5h)
- Interesting moments:
  - 27’35’’ VM live migration starts, after both volumes mirrored
  - 46’05’’ VM live migration finished, VM destroyed on source host, running on destination host

Demo5: Online Storage Migration of Busy VM (RAM + IO Heavy):
https://youtu.be/tR0VWz7Vl_s

INTERESTING RESOURCES

Code and issues links

- Proper CEPH snapshot removal code: https://goo.gl/XP3uVR
- RBD layering issue: https://goo.gl/SZvEzp
- 16 snapshots fix with rados-java 0.2.0: https://goo.gl/56UAWB - many thanks to Wido!
- "executeInSequence" issue: https://goo.gl/euZssQ

Direct demo links

- Demo1: SolidFire Snapshot Create and Restore: https://youtu.be/HRpRvvX4diw
- Demo2: QoS on KVM vs. QoS on SF: https://youtu.be/ZAZps5QU17c
- Demo3: Qemu Dynamic Autoconvergence vs. Extremely Busy VM: https://youtu.be/sB0lOrCkzYE
- Demo4: Online Storage Migration (NFS/CEPH to SolidFire): https://youtu.be/Eo8BuHBnVgg
- Demo5: Online Storage Migration of Busy VM (RAM + IO Heavy): https://youtu.be/tR0VWz7Vl_s

Other links

- Online Storage Migration by Mike.T. : https://youtu.be/VbnMwYYkRnQ
- SolidFire with Cloudstack, 46 videos youtube playlist: https://goo.gl/XjK41s
- Low level live migration details: https://goo.gl/bx3ZrL, https://goo.gl/Evm1j5
- Excellent Mirantis' "what to expect" with CEPH: https://goo.gl/XqGuuH
- Red Hat CEPH Architecture (including data striping): https://goo.gl/cJGr65

Small bonus:

- Online storage migration script: https://github.com/HiagData/CloudStack.OnlineVmMigration/

QUESTIONS ?

THANK YOU !