AGLT2 Site Report

Wenjing Wu/University of Michigan
Dan Hayden, Philippe Laurens, Shawn McKee

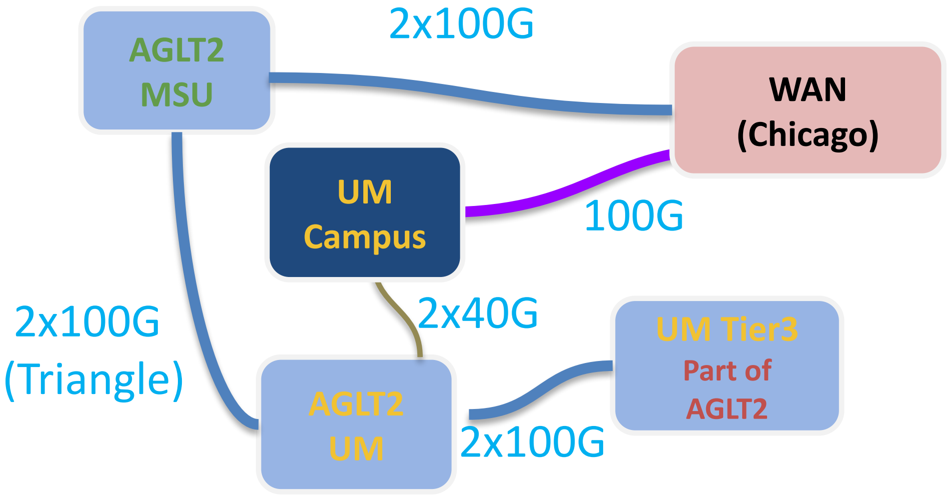
Oct 16, HEPix Fall 2023

ATLAS Great Lakes Tier 2
University of Michigan · AGLT2 · Michigan State University

AGLT2 Overview

- AGLT2 (ATLAS Great Lake Tier-2) is an LHC Tier-2 Computing Center for ATLAS, located at our UM site (University of Michigan) and MSU site (Michigan State University).

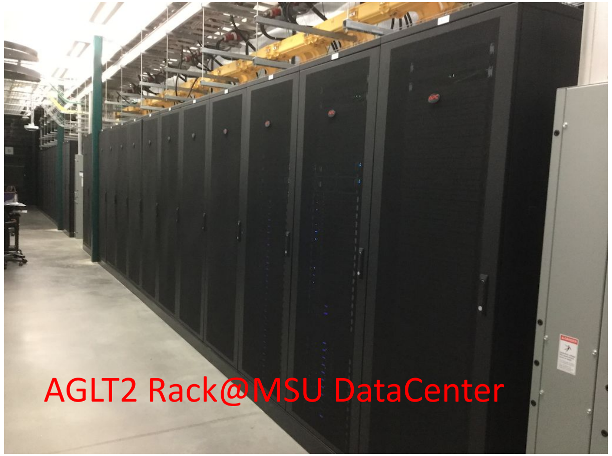
- What VO(s) we serve
  - ATLAS Tier2/Tier3
  - OSG (ligo, uscms, glow etc.)
- Resource overview
  - 17.5K cores/226 kHS CPU, 16 PB dCache Storage
- Resource Usage:
  - Over 92% are constantly used by ATLAS Tier-2 jobs
  - the rest is shared with UM ATLAS Tier3 and other VOs
- Redundant 100G paths between the 2 sites and to Chicago (ESnet)



AGLT2 Server Room (1)

- **Finished Networking upgrades at UM site in 2021**
  - Established separate resilient diverse fiber path 100G MSU-UM inter-site connectivity
  - replaced all network switches/cables/PDUs
- **MSU site migrated to the new MSU data center in 2021**
  - 12x33kW racks with dual redundant power (Utility&UPS), one management switch plus redundant (EVPN) data switches, 1/10/25/100 Gbps ports in each rack, room for expansion.
  - Deep and wide racks: room for 2 PDUs on each side, easy cable management, good airflow.
  - MSU data center provides the rack network devices, optics, cabling, switch configuration.
- **UM Server Room**
  - Located at LSA college building, 80KW APC UPS supports 8 Rack, plus one Rack connected to another 80KW UPS shared with LSA IT
  - Each Rack has separate data (100 Gbps/port) and mgmt switches (1 Gbps/port)
  - Use **AOC** breakout cables and QSFP->4*SFP+/QSFP28->4*SFP28 transceivers for data switches, and slim RJ45 for mgmt switches.
  - All racks installed with Smart PDUs with individual socket meter and control

AGLT2 Server Rooms (2)



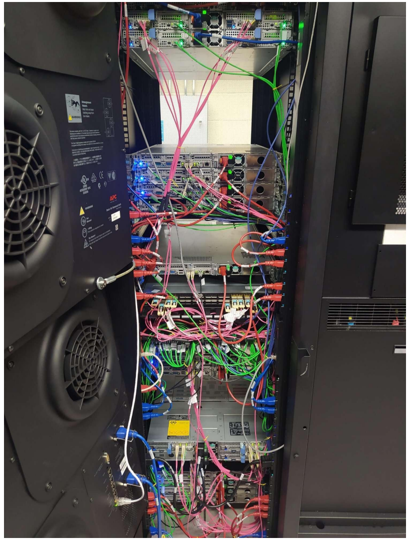
AGLT2 Rack@MSU DataCenter

Hot Aisle (back)

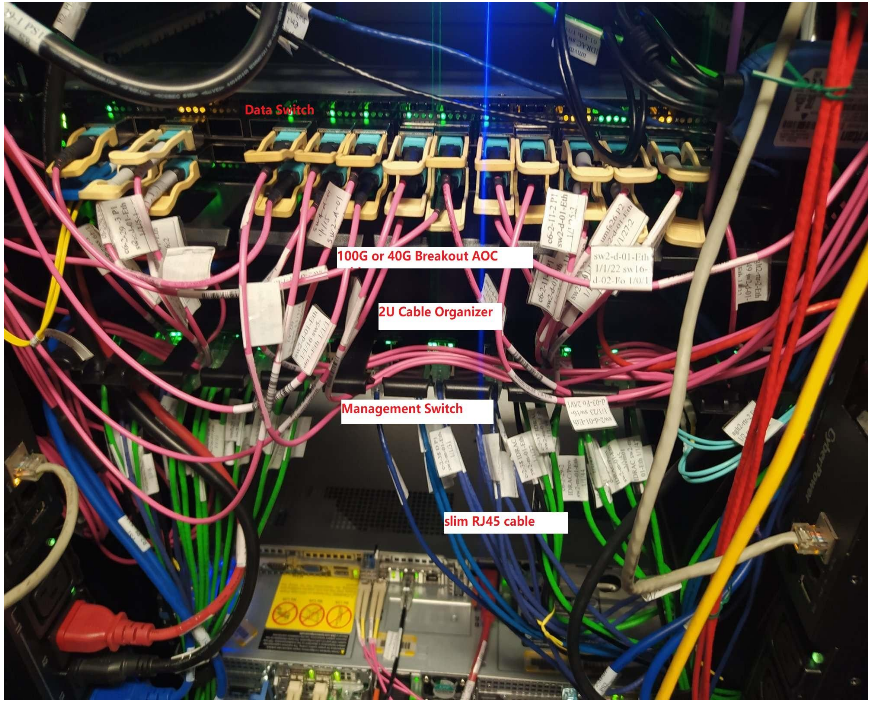
MSU Rack Front

MSU Rack Cable/PDU

AGLT2 Server Rooms (3)



UM Rack Back



UM Rack Switch / Cable

AGLT2 Storage

- **dCache (serving ATLAS Tier2)**
  - Total capacity **16PB**
  - 2 head nodes (@UM, each has postgresql database), with another 2 slave nodes for postgresql hot standby (all postgresql are on ZFS)
  - 6 dCache door nodes (3@UM, 3@MSU)
  - 47 pool nodes (17 @MSU, 30 @UM)
  - 2x25G bonded Ethernet for pool nodes
  - Performance evaluation between hardware RAID6 and ZFS over HBA.
- **Lustre (serving UM ATLAS Tier3, mounted on all the UM cluster nodes)**
  - Total capacity **2.3PB**
  - Version: Server 2.12.3, Client 2.12.5
  - ZFS 0.7.11 on the OSS, ldiskfs on MGS/MGT
  - 2x25G bonded Ethernet on each OSS
  - 85% storage capacity are on new hardware (R740xD2)
  - New testbed on CentOS Stream 8, 2.15.1

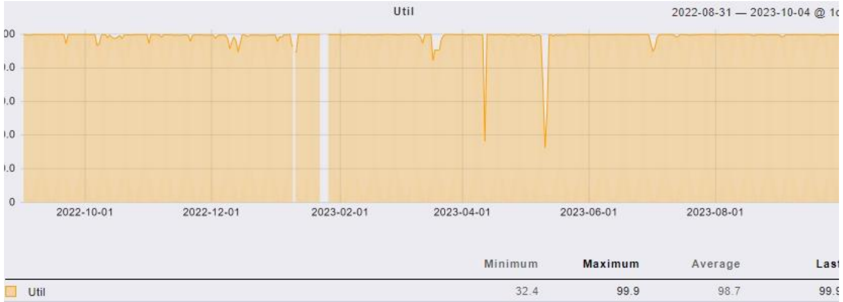
RAID 6 VS. ZFS

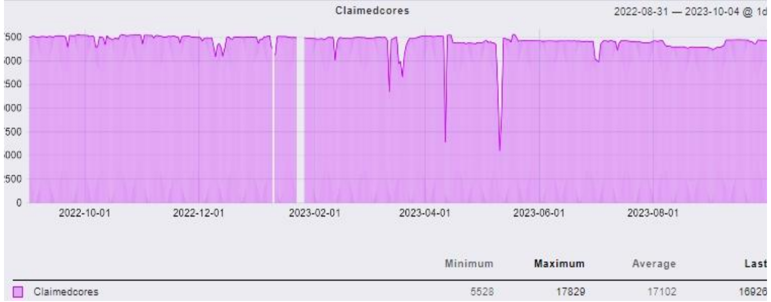
We want to understand if we can replace hardware RAID-6 with ZFS

- **Configuration**
  - PERC H730P, RAID6 over 12 disks
  - Dell HBA330 Mini, ZFS raidz2 over 12 disks, tuned with zfs recordsize=1MB compression=off atime=off xattr=sa ashift=12
  - zfs-0.8.5-1(EL7) vs. zfs-2.1.13-1(EL9)
- **Test with root files from ramdisk (file system in memory)**
  - /dev/zero does not give accurate result for zfs with compression due to high compression rate
  - choose typical root files from ATLAS storage, compression rate is 1.103
- **Conclusion (details)**
  - ZFS 2.1.13(LZ4 compression) vs RAID 6: gains 15.6% in Write and loses 20.8% in Read, lose 1.82% in space with lz4 compression for typical ATLAS files
  - ZFS 2.1.13 vs ZFS 0.8.5: 3-25% improvements in IO Performance.

AGLT2 HTCondor Cluster

- 3 gatekeepers for ATLAS to increase resiliency
- 308 Physical nodes, with cores range in (24,32,40,56,64,96), RAM per core ranges in (2,3,5,6)GB
  - Generations of work nodes: Dell R630->C6420->R6525
- **17.5K** logical cores, total of **226 kHS06**, average 13.02 HEPSCORE/core
- 2GB to 6.3 GB RAM/core, with 1000 job slots for High Memory Queue (6 GB/core)
- 14GB to 52 GB Disk/core, supports Merge Queue with higher disk requirement.
- All slots dynamical partitioning, and Cluster Utilization rate over 98.7% (including downtime for hardware and software upgrade)
- Continue to run ATLAS@home/BOINC backfilling jobs from 2019 (oversubscribe the CPUs)





Cluster backfilling: BOINC

- Motivation
  - HTCondor cluster isn't always fully used
    - site downtime
    - storage outage
    - site draining (HTCondor/firmware update, Grid Central Service downtime)
    - not enough single core jobs to fill the dynamically partitioned job slots etc
  - CPU utilization is not high (<90%) due to low CPU Efficiency jobs, i.e, analysis jobs, which means free CPU cycles to exploit)
- How backfilling jobs work and are configured?
- Harvest
  - Biggest site contributing to ATLAS@home, Scavenged CPU time from AGLT2 Cluster (2313 CPU days per day in the recent 100 days) is equivalent to a site with 3K Cores.
  - Per Cluster, CPU Utilization increase by **7%** in long term(83% without BOINC vs. 90% with BOINC), but CPU Efficiency for Grid jobs lose **5%** (details)

Software and Technology Details

- **AGLT2 runs a number of software packages required for an ATLAS site:**
  - OSG 3.6/HTCondor-CE 6.0.0-1/HTCondor10.0.9 (do firmware and HTCondor updates ~every quarter)
  - storage uses dCache 8.2.27 (do firmware and dCache update ~every quarter)
- **VMware cluster**
  - VMware 7, Two TrueNAS servers (iSCSI for VMs)
- **Other Storage**
  - Lustre(2.12.8 on CentOS 7 and 2.15.1 on CentOS 8 stream)
  - NFS(0.5PB), AFS(1.8.7) and have collaborative access to Ceph on OSiRIS (12 PB).
- **Site Monitoring**
  - CheckMK(2.0.0-p6), Elasticsearch(7.17), Zeek(4.2.0), Elastiflow(5.3.4) and NetDisco, PerfSONAR
  - Recently added dell OME monitoring for all hardware.
- **Tape Backup**: Amanda 3.5.1 on CentOS 7 (most recent version: 3.5.2 on CentOS 7)
- **Provisioning:**
  - CentOS 7: Cobbler 2.8.5 on CentOS 7 (most recent version: 3.3.3 on CentOS Stream 8)
  - RHEL9: Satellite 6.12.5.1+Capsule Server 3.3.0 (future focus)
- **Configuration management**: CFEngine 3.7.2 on CentOS 7 and 3.12.4 on CentOS 8 Stream (most recent version : 3.20.0 on CentOS 8 stream)

Transition to RHEL9

EOL for RHEL7 and clones in June 2024 means we need a new OS

- **Choice of OS after CentOS 7-> RHEL9.2**
  - Both **MSU** and **UM** have licenses for RHEL through the University and satellite servers are hosted by IT
  - RHEL9+ gives a modern kernel, compilers, and improvements for data transfers and long lifetime
  - Skipping RHEL8 and clones for longer term solution
- **Challenges from transitioning from cobbler to the UM Satellite Server (version 6.12 on EL8)**
  - The AGLT2 network is not routed to access the UM College Satellite server
  - The UM College Satellite server is not set up support PXE booting
- **Solution**
  - We deployed a capsule server (~Foreman SmartProxy) as a proxy between the AGLT2 network and the UM Satellite server
  - Settled on using UEFI HTTPboot as the bootstrap mechanism (lots of little bugs and quirks to deal with)
    - Requires BIOS change to UEFI and enabling HTTPboot on the correct NIC but works well
- Lots to do to improve the build and leverage the various Satellite features
  - Ansible or Puppet are potential options for us, if we choose to move off CFengine
  - Lots of security, auditing, monitoring and version control tools to explore in Satellite

Network Security

AGLT2 has been working with the WLCG SOC effort to help secure our networks while maintaining performance and participated in the WLCG SOC Hackathon, Aug 2023.

Our previous network had a Zeek+MISP+Elasticsearch on CentOS 7 setup for dual 40G. Cost to set up was about $2K plus repurposing an R630

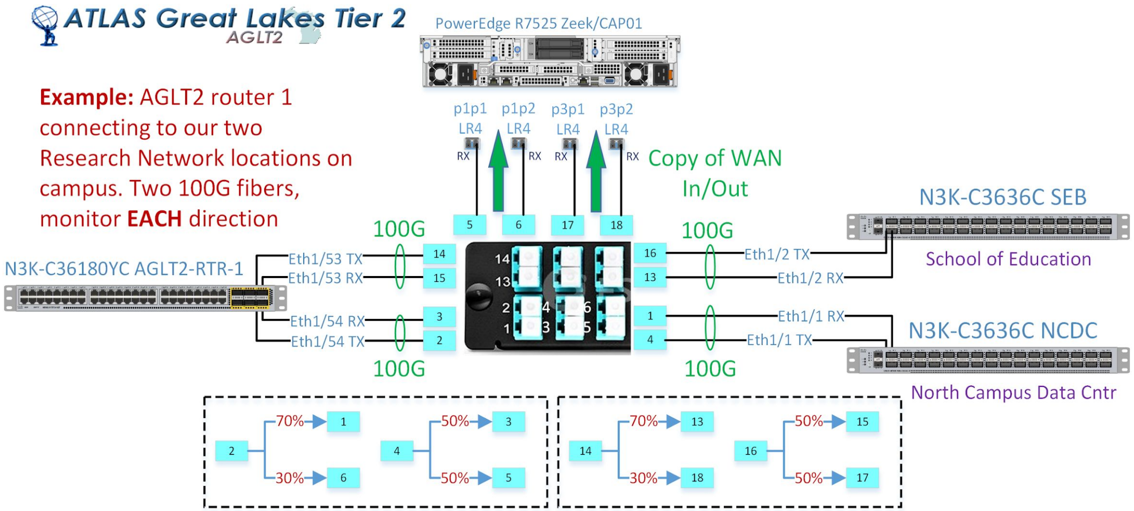
Our new UM network has 2 capture nodes (MSU will have 1):

- Dell R7525, each with two Bluefield-2 NICs (2x100G)
- Installed Zeek on RHEL9

ElasticSearch:

- Migration to OpenSearch on RHEL9 in progress.

Need better integration/config with MISP/Elastialert



CAP01/RTR-1 diagram (similar for CAP02/RTR-2)

PTP at AGLT2

For about $1500, AGLT2 added dual GPS clocks to enable PTP

Roof antenna added in Fall 2022 (now have excellent lock for GPS)

- PTP provides < 1 microsecond time accuracy, improving the NTP timing of our site services.
- **GOAL**: make perfSONAR latency much more powerful **BUT** needs pS mods

  To do: PTP Clients (NTP ~20 u-secs)

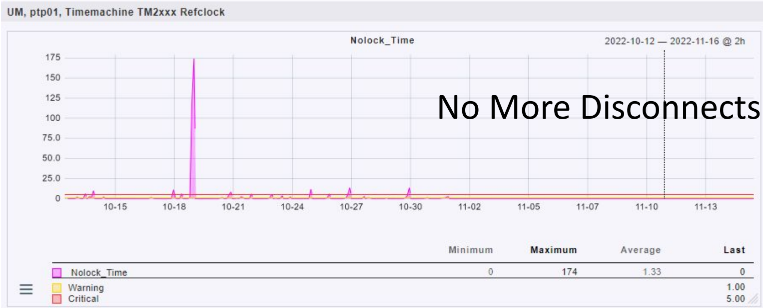
Summary

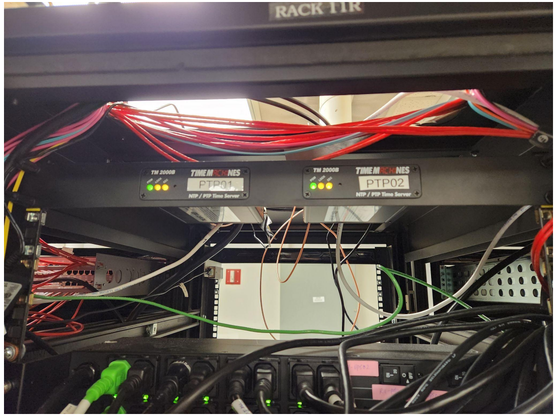
- Updates of OS, software, firmware and security patches are applied in a timely way to keep AGLT2 updated
- Both Data centers had big upgrades to a next generation infrastructure.
- **FUTURE:** Migrate site towards RHEL 9, WLCG SOC on EL9

Questions ?

Improving PTP Lock via Antenna

Start of Nov 2022: our antenna installed on the roof of Physics Bldg





How backfilling jobs run

- Backfilling jobs come from ATLAS@home(initially started as a volunteer computing project) and appear as a separate ATLAS PanDA Queue AGLT2_BOINC
- Backfilling means running 2 set of jobs on the cluster simultaneously, and they are controlled by 2 batch systems (HTCondor/BOINC) and different user accounts.
- HTCondor with NICE number 0, BOINC with NICE number 19 (NICE decides process Priority in CPU allocation)
- BOINC jobs are only run when the HTCondor jobs are not using the CPU cycles.
- **cgroup** is also used to control the CPU usage by each set of jobs. Both HTCondor and BOINC jobs are under the system slice, with different cpu.share value (100 vs 2)
- On Work node, if HTCondor is ‘Busy’, BOINC only instantiates jobs for 50% of the cores, if HTCondor is ‘Retiring’, BOINC uses 100%

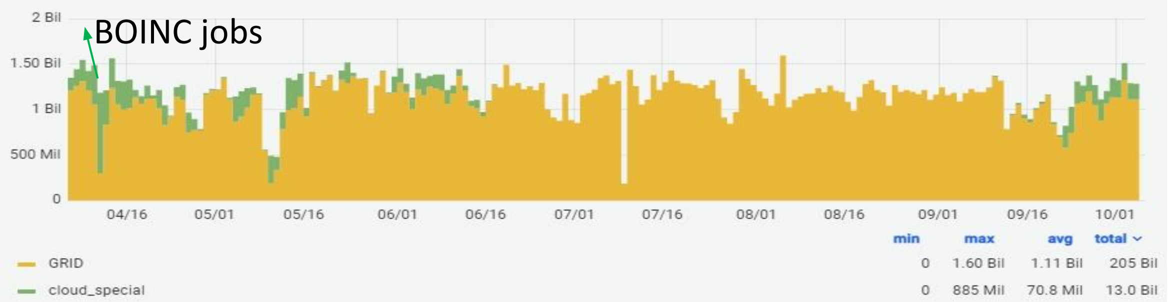
backfilling increases the CPU Utilization

age of the clusters. CPU efficiency (ϵ_{CPU}) is used to measure the efficiency of the jobs, and wall time utilization (u_{wall}) and CPU time utilization (u_{cpu}) measure how fully these clusters are being utilized. Assuming that in a given period M days, the total wall time (in seconds) of all jobs is T_{wall}, the total CPU time (in seconds) of all jobs is T_{CPU}, and the total number of available cores of the site is N_{core}, then:

$$u_{\text{wall}} = \frac{T_{\text{wall}}}{3600 \times 24 \times M \times N_{\text{core}}} \quad (1)$$

$$u_{\text{cpu}} = \frac{T_{\text{cpu}}}{3600 \times 24 \times M \times N_{\text{core}}} \quad (2)$$

$$\epsilon_{\text{CPU}} = \frac{T_{\text{cpu}}}{T_{\text{wall}}} \quad (3)$$



Compare the CPU Utilization with vs without backfilling (backfilling jobs are cloud_speical)

- between 6/25-9/7/2023, there were no backfilling jobs running due to central service down, the total CPU cores fo ATLAS stays the same from 06/2022-10/2023 (16000 Cores)
- CPU Utilization increase by **7%** (83% without BOINC vs. 90% with BOINC)
- CPU Efficiency for Grid jobs lose **5%**

AGLT2 Site CPU Utilization(16000 CPU cores in 100 d
Sun Feb 26 23:00:00 2023 to Wed Jun 7 00:00:00 202

	cpu_eff	cpu_util	wall_util
BOINC	0.36	0.12	0.34
Grid	0.91	0.78	0.87
All	0.75	0.90	1.21

AGLT2 Site CPU Utilization(16000 CPU cores in 74 d
Sun Jun 25 00:00:00 2023 to Thu Sep 7 00:00:00 20

	cpu_eff	cpu_util	wall_util
BOINC	0.56	0.00	0.00
Grid	0.96	0.83	0.88
All	0.96	0.83	0.88

ZFS vs RAID Testing Results

	compression= 1.05		compression =1.09	
	zfs-0.8.5-1	zfs-0.8.5-1	zfs-2.1.13-1	zfs-2.1.13-1
	Read	Write	Read	Write
ZFS no compression	2200	1600	2400	1650
ZFS lz4 compression	1600	1800	2000	1850
RAID6	2525	1600	2525	1600
lz4 vs. no compression	-27.27%	12.50%	-16.67%	12.12%
ZFS lz4 vs. RAID	-36.63%	12.50%	-20.79%	15.63%
ZFS no compression vs. RAID	-12.87%	0.00%	-4.95%	3.13%

ZFS vs. RAID6 IO Performance

mount -o size=80G -t tmpfs none /mnt/ramdisk/
dd if=/data/bigrootfile of=/mnt/ramdisk/file
Read: dd if=/dcache/file of=/dev/null bs=1M
Write: dd if=/mnt/ramdisk/file of=/dcache/file bs=1M
zfs tuning: recordsize=1MB atime=off xattr=sa ashift=12

Testing Commands

ZFS	2.1.3 vs. 0.8.5	2.1.3 vs. 0.8.5	2.1.3 vs. 0.8.5
	Read	Write	compression rate
ZFS no compression	9.09%	3.13%	
ZFS lz4 compression	25.00%	2.78%	3.81%

ZFS 2.1.3 vs. ZFS 0.8.5

	usable space (TB)	compression rate	real usable TB	zfs space loss
RAID	200	1	200	
ZFS	178	1.103	196.365	-1.82%

ZFS 2.1.3 vs. ZFS 0.8.5 space usage