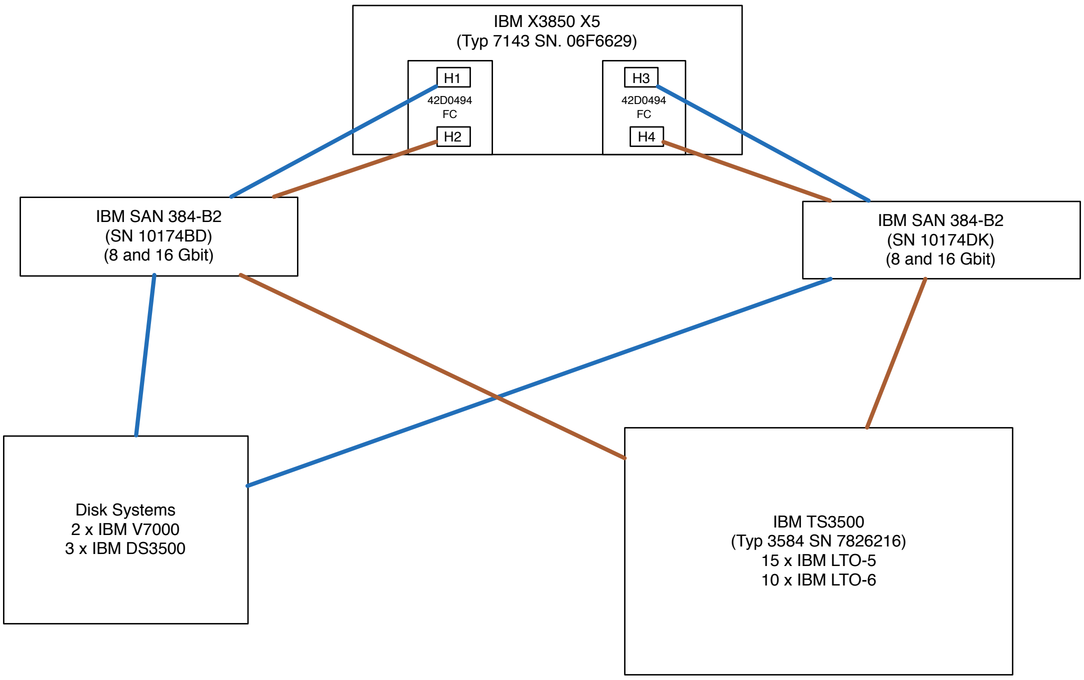
Performance Problems with TSM Tape to Tape Workloads on IBM LTO Drives

1. Test Setup

We use an IBM X3850 X5 with a single small TSM Instance on it. So impact of production TSM Workload is minimized. The Server is attached via two IBM branded Emulex LPE120002 8 Gbit Dual Port Adapters to two IBM SAN385-B2 switches with 8 and 16 Gbit Ports. From there we connect to a IBM TS3500 with IBM LTO-5 and LT0-6 drives. All involved ports run at 8 Gbit speed. The Server is also attached to a disk storage pool striped across 3 IBM DS3500 Systems.



2. Method of Measurement

For measuring performance we use:

1. TSM Performance Instrumentation at a 1 minute resolution. So we call “instrumentation begin” wait 60 seconds, call “instrumentation end” and loop to the beginning
2. Reading lpfc driver counters /sys/class/fc_host/hostX/statistics/[rx|tx]_words. Thereby we can see the data throughput on the HBA Port basis at a 1 minute resolution.
3. Graphing Values collected by the Splunk UNIX Plugin. Thereby we can see serveral things like CPU load, Disk Throughput, Tape Throughput, Disk Usage, etc. at a 10 minute resolution

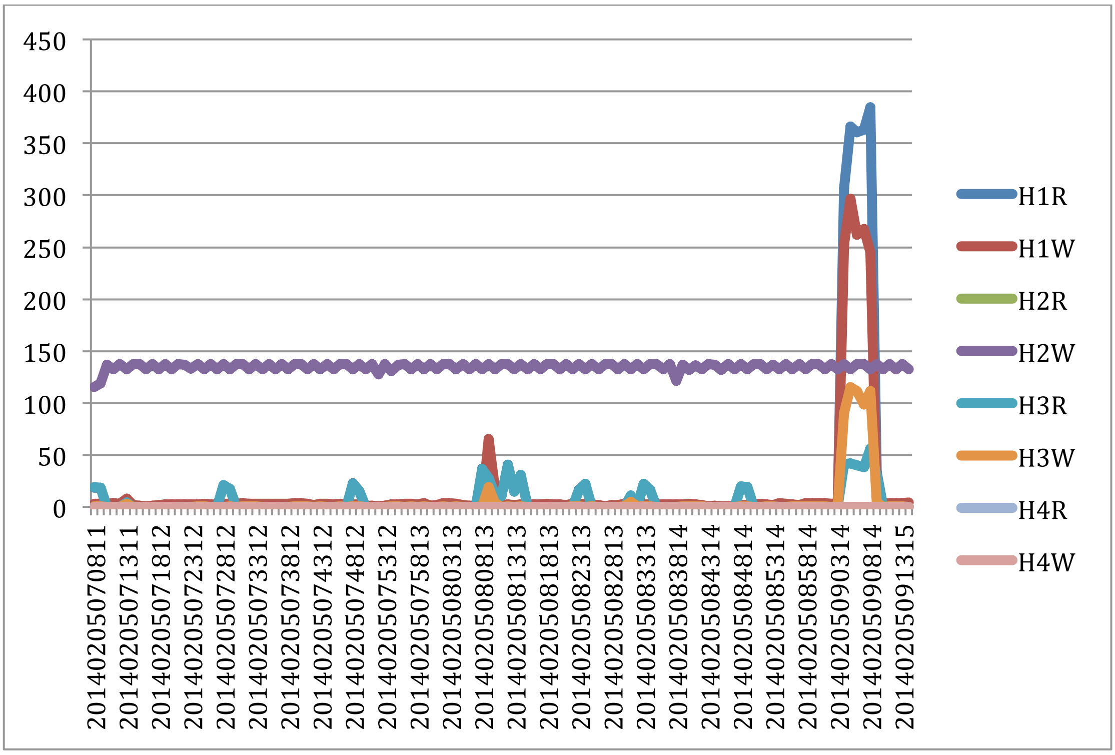
4. Tests Performed

- At first we created a 1 TB large file with uncompressible data (read from /dev/urandom) on a TSM client of the server. Then we archived the file via network directly to tape.
- In the second test, we did a move data between tape to tape on the server. Whereby both tapes where connected to the same HBA port.
- In the third test, we did a move data between tape and disk storagepool on the server.
- In the fourth test, we did a migration between disk and tape.
- In the fifth thest, we did again a move data between tape to tape but this time the tapes where connected to different HBA ports.
- In the sixth test, we put TSM out of the equation and tried a disk to tape data move with the dd command
- In the seventh test, we moved data from tape to tape with the dd command
- In the eight test, we moved data from tape to tape using itdt's tapephcp command and lin_tape devices
- In the ninth test, we moved data from tape to tape using itdt's tapephcp command and generic devices
- In the tenth test, we again moved data via TSM from tape to tape but with the testing Firmware provided by IBM

5. Results

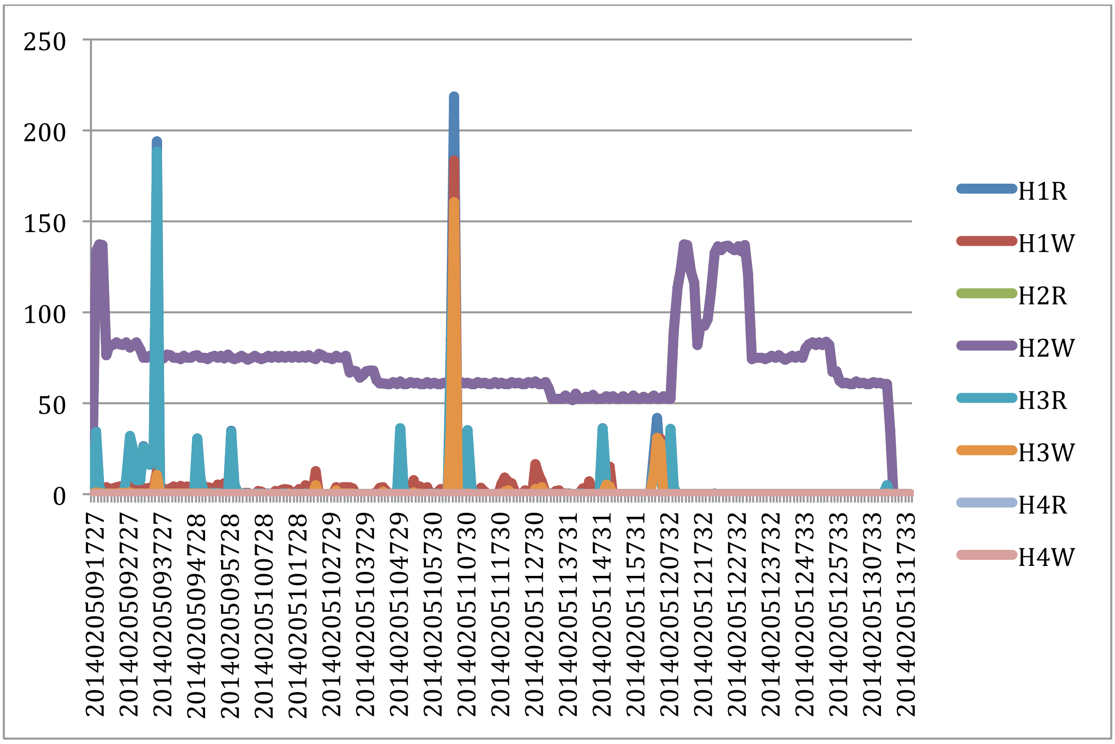
5.1 Archive Operation

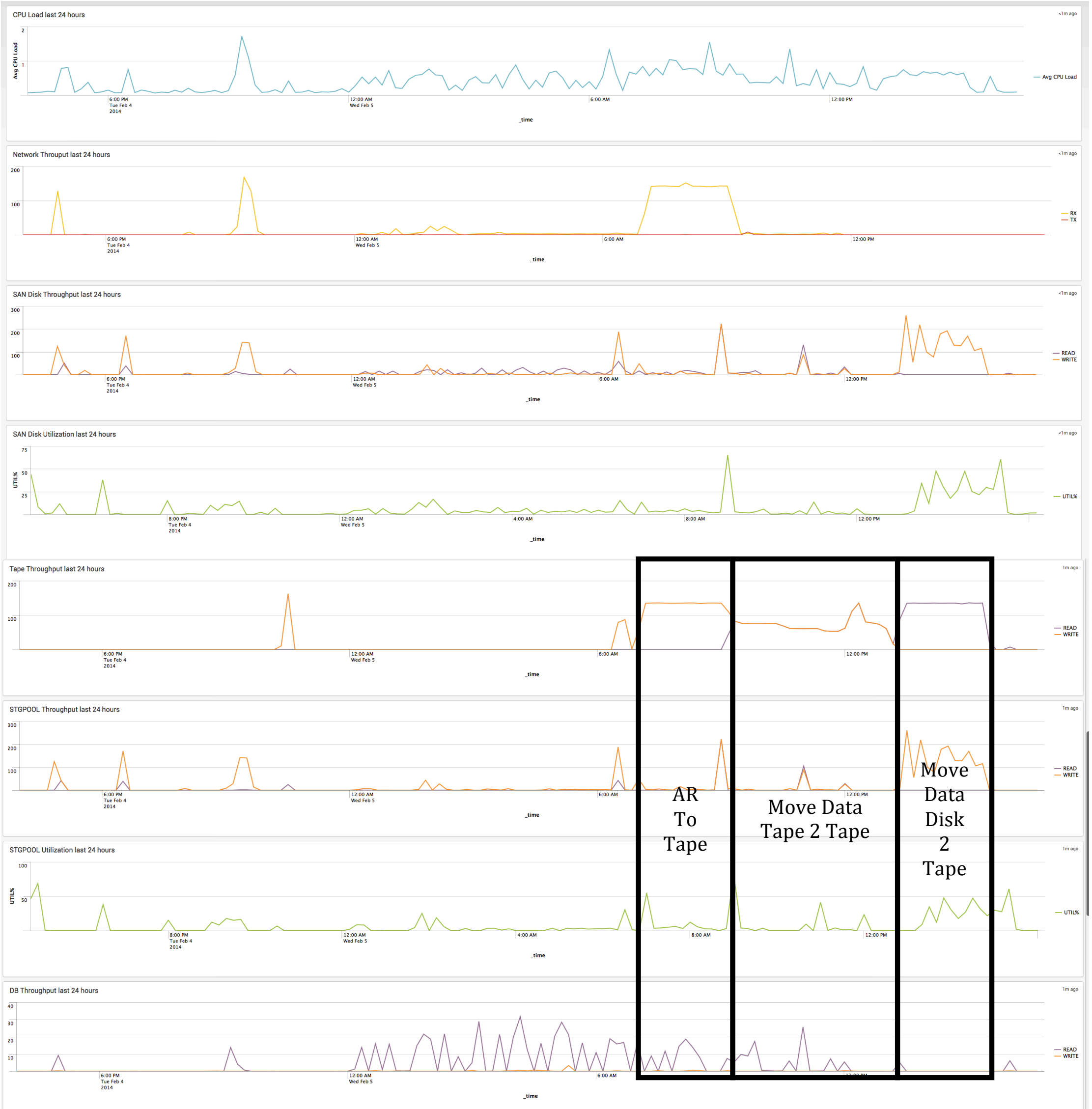
As can be seen on the chart (H2W). The tape we wrote to via HBA Port 2 showed a relatively constant write rate of 130 MB/s. So this goes as fast as expected from an LTO-5 Drive.



5.2 Move Data Tape 2 Tape same HBA Port

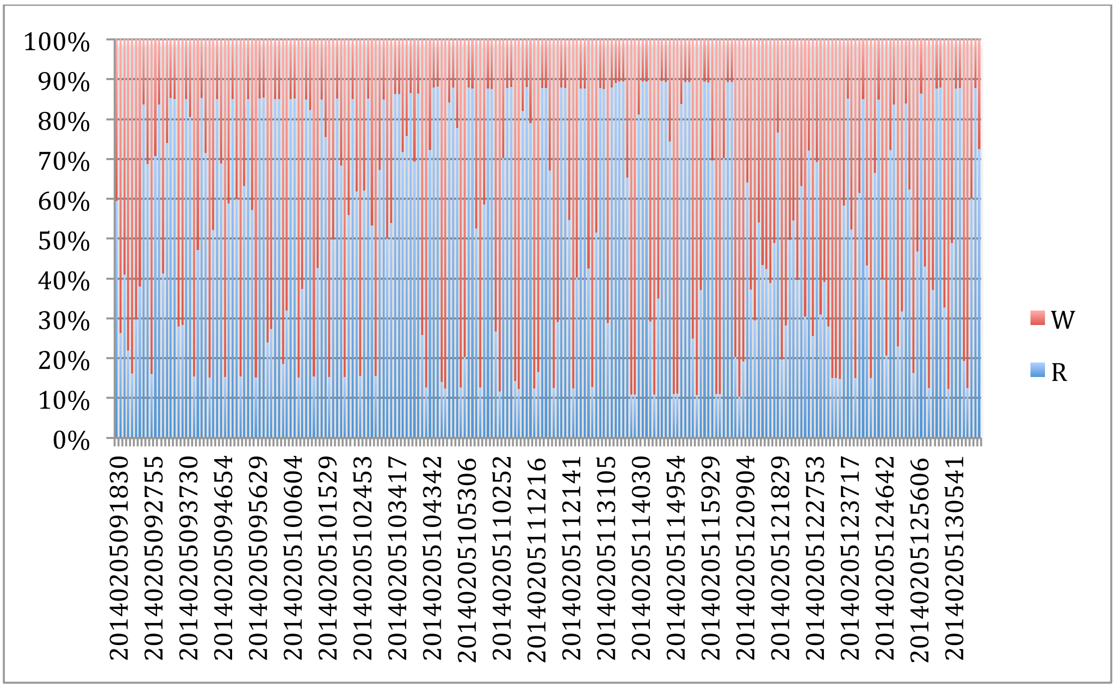
As can be seen, the tape to tape operation starts at the expected rate of 130 MB/s but soon falls to a value of 75 MB/s which gets even as low as 50 MB/s. However in between it recovers to 130 MB/s for a short amount of time an then falls again. The system thereby was idle the whole time, as can be seen by the splunk graphs between 9:20 and 13:20 o'clock.





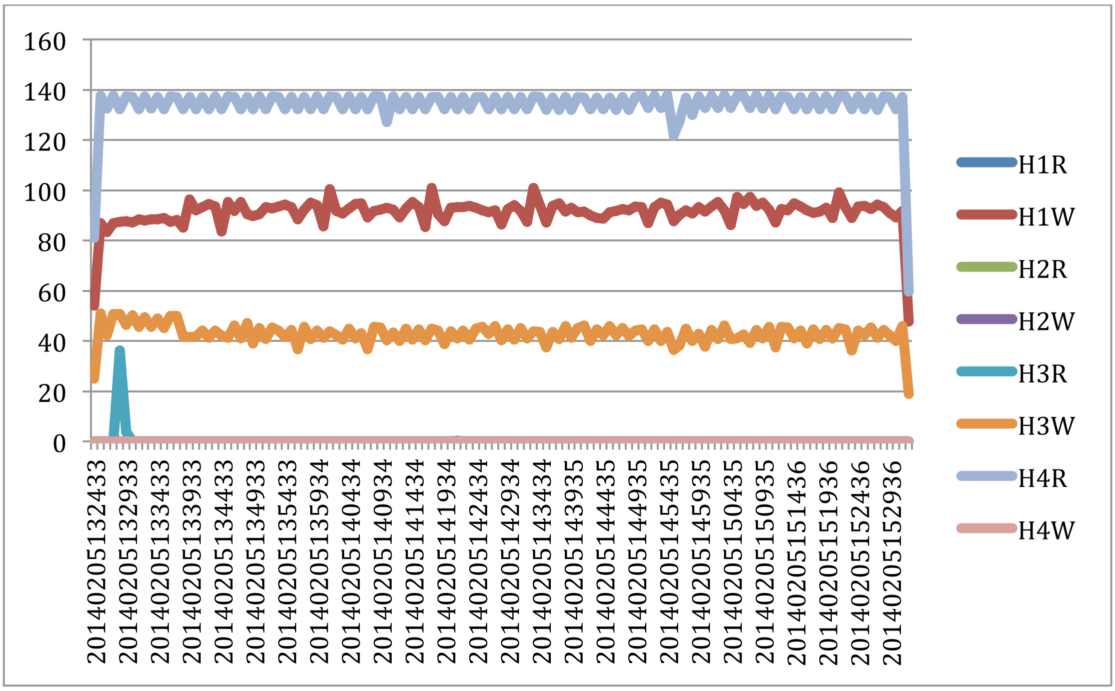
If we look at the tape operations (read/write) via the TSM Instrumentation, we see that sometimes the read takes long and sometimes the write takes long. So there is no clear identification of side of the bottleneck possible because there are times when the read waits for the write, times where the write waits for the read and times where read and write are both fast.

The chart below shows the performance of the individual read and write calls of TSM. Meaning they calculate the amount of data read or written divided by the time spent in the read and write calls thereby ignoring any processing or idle times outside of read and write functions of the threads. The longer the bar, the better the operation performed in contrast to the other. So if we have a long read bar and a short write bar, this means that the write was much slower than the read (eg. Write bottleneck). Ideally would be if read and write bars are equally long. As can be seen in the chart the side of the bottleneck alternates but this time it was more often on the write side.



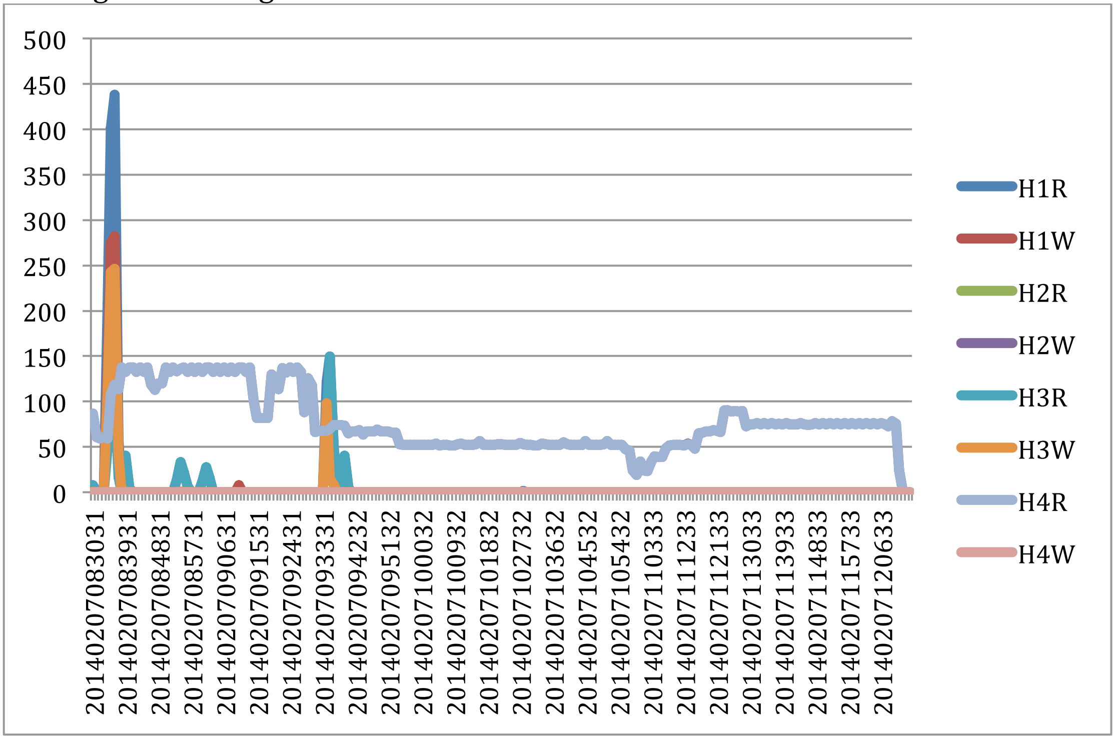
5.3 Move Data Tape 2 Disk and Disk 2 Tape

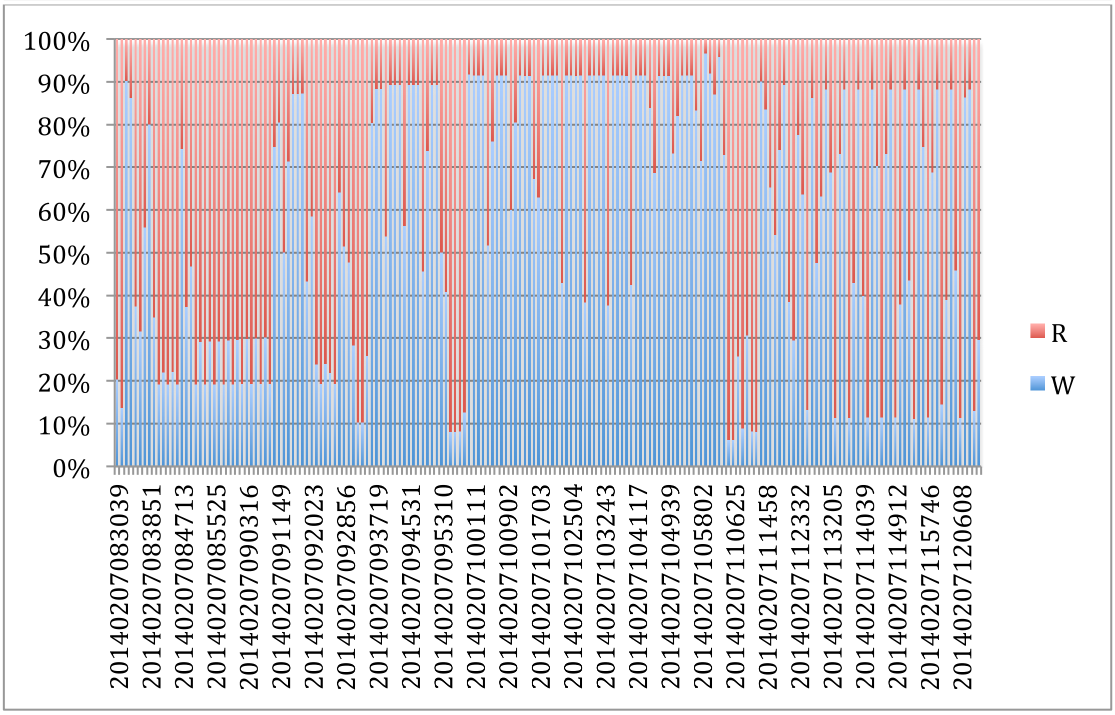
Moving Data from tape to the TSM disk cache is again fast. As can be seen in the diagram, reading from tape is about 130 MB/s constantly (H4R). The same holds true for the other direction, meaning moving the data from TSM disk cache to tape.



5.4 Move Data Tape 2 Tape different HBA Ports

Moving data from tape to tape while using different HBA Ports for reading and writing, does not show any significant difference in the behavior than when reading and writing over the same HBA Port.





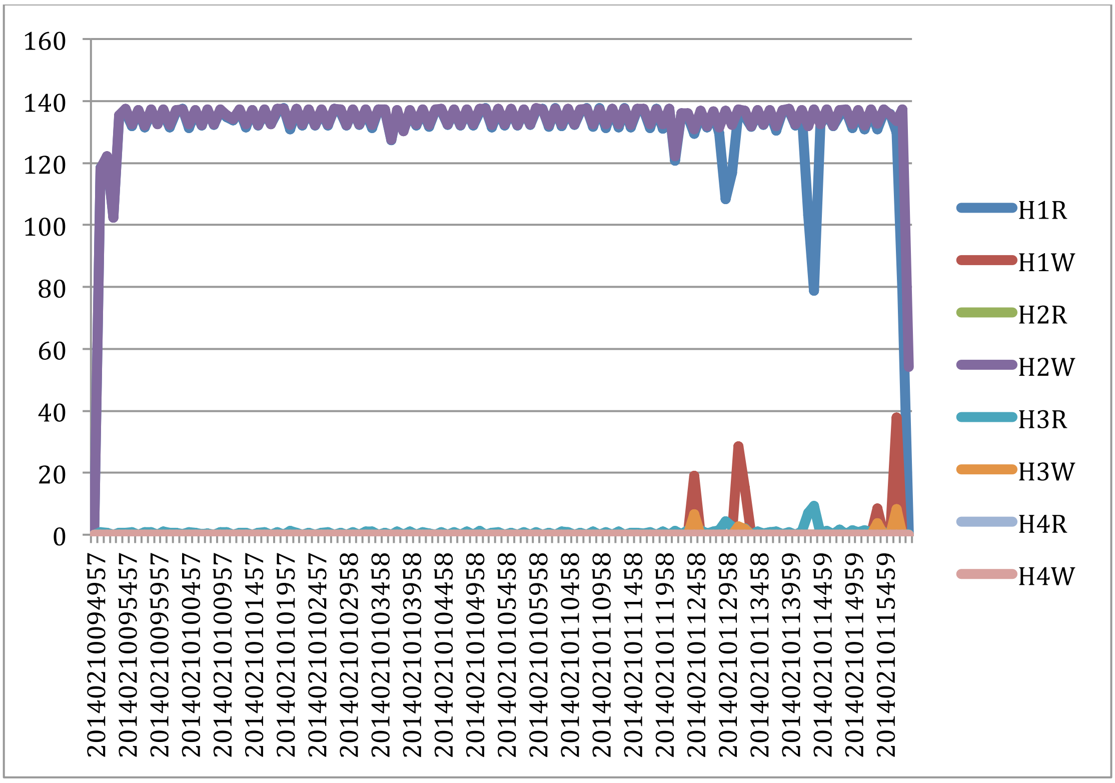
5.5 Moving Data Disk to Tape with dd

This time again, we took our 1 TB non-compressible file and wrote it from disk to tape via the following dd command-line:

```
dd if=/dsm/stg/0/uncompressable.bin2 of=/dev/IBMtape2
bs=262144 count=4096000
```

We have chosen a block size of 256K since this is the same block size TSM uses for data transfers.

The result was the same as for moving data disk to tape via TSM. Constant performance of 130 MB/s.

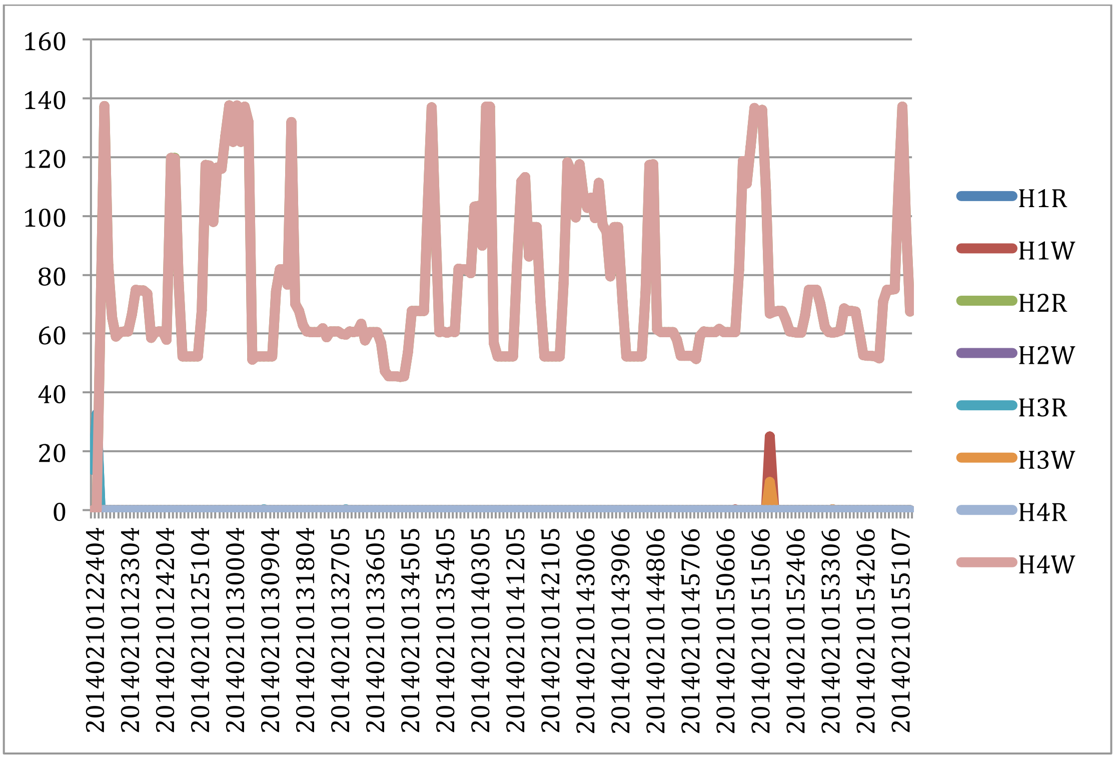


5.6 Moving Data Tape to Tape with dd

Last but not least we tried do copy data between two drives using the following dd command-line:

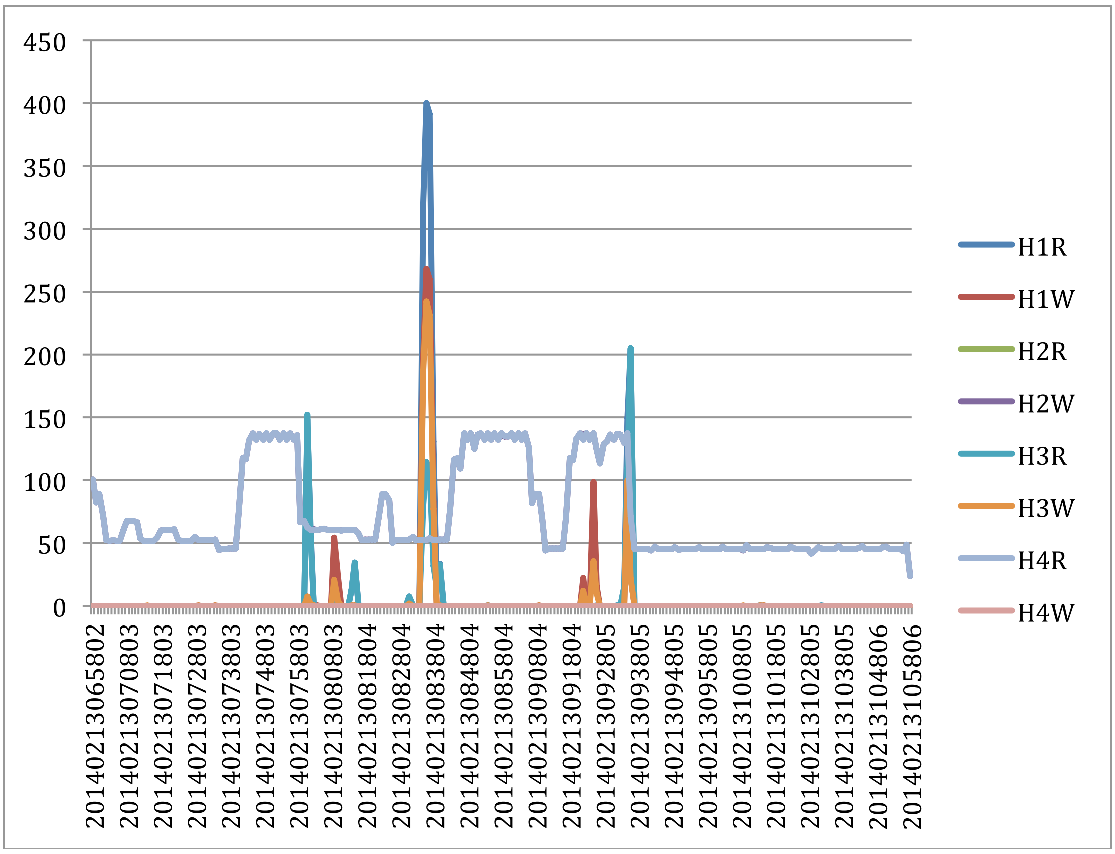
```
dd if=/dev/IBMtape2 of=/dev/IBMtape7 bs=262144
count=4096000
```

In this case the performance was bad again as expected. However it seems that this time the performance curve was not as smooth as with TSM but had more spikes where – for a short amount of time – performance was OK. However we think it shows that the problem of bad tape to tape copy performance is not directly TSM related.



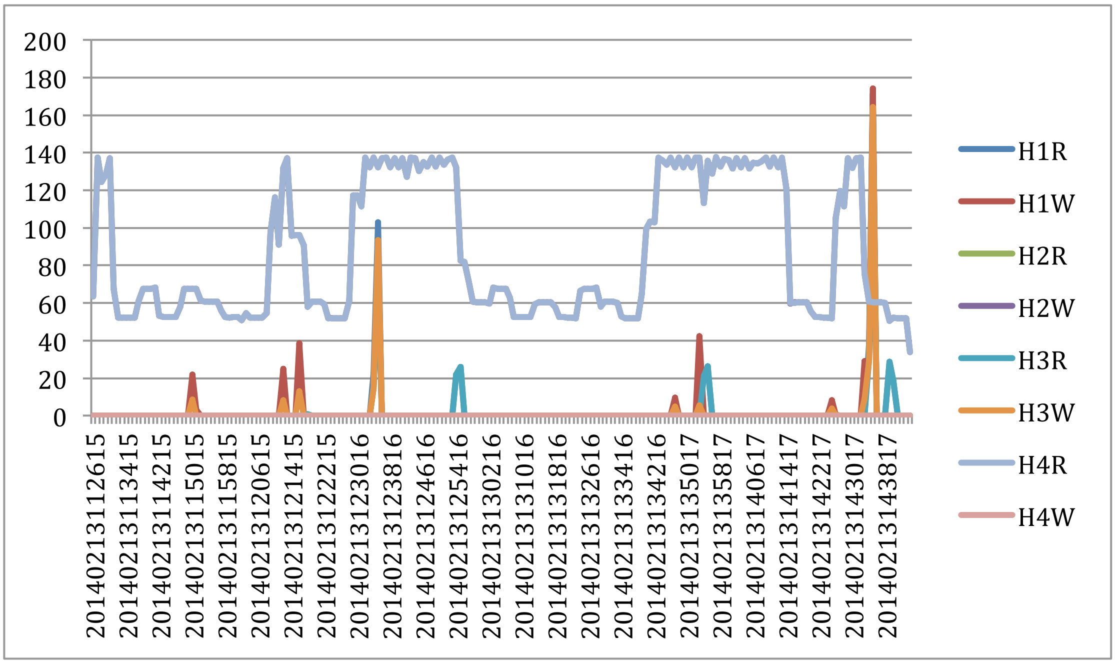
5.7 Moving data Tape to Tape with itdt and lin_tape

On advice of IBM Support we used itdt's tapephcp command to copy data between two tapes. As can be seen, there exists the same problem.



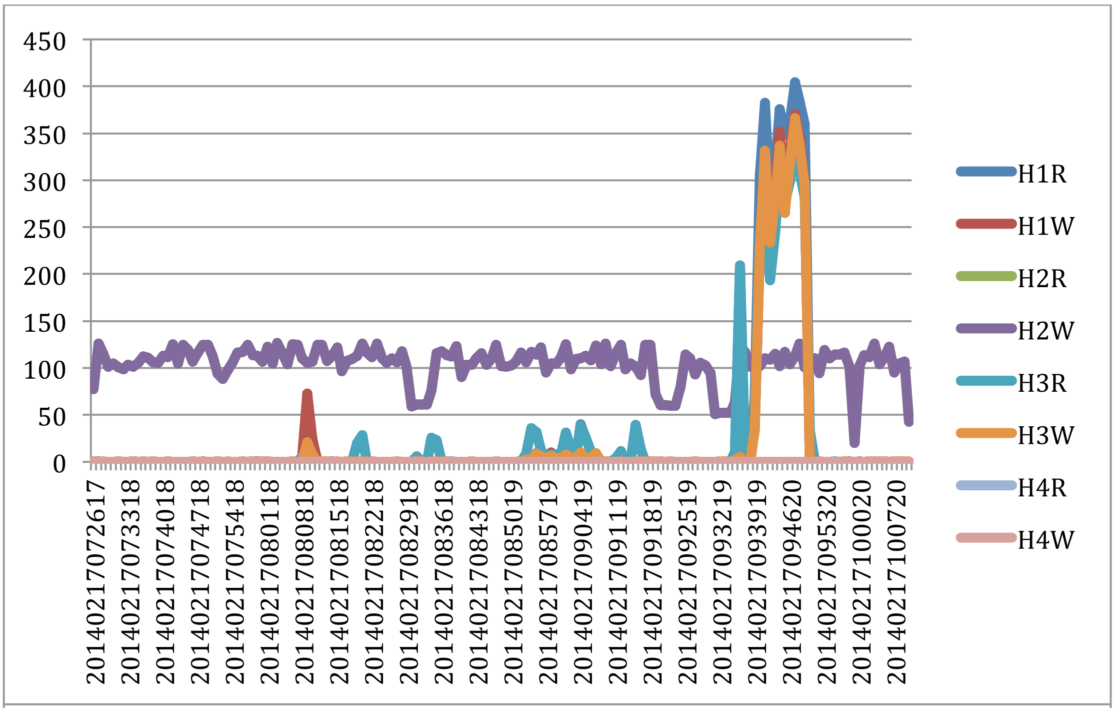
5.8 Moving data Tape to Tape with itdt and scsi generic

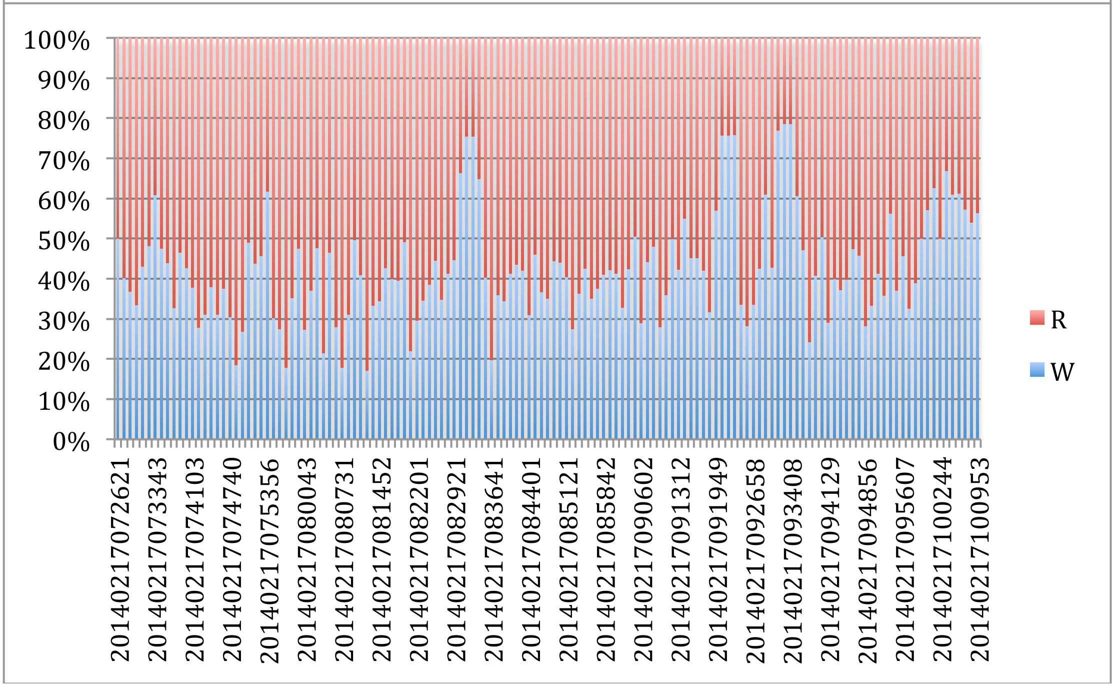
After using itdt tapephcp with lin_tape devices, we used it with the linux scsi generic interface. Note that we used exact the same drives and volumes in this test as we did with the test in 5.7. Again we see no difference in the behavior. So it seems that lin_tape is eliminated from the list of suspects.



5.9 Moving data LTO-5 to T10K with TSM move data

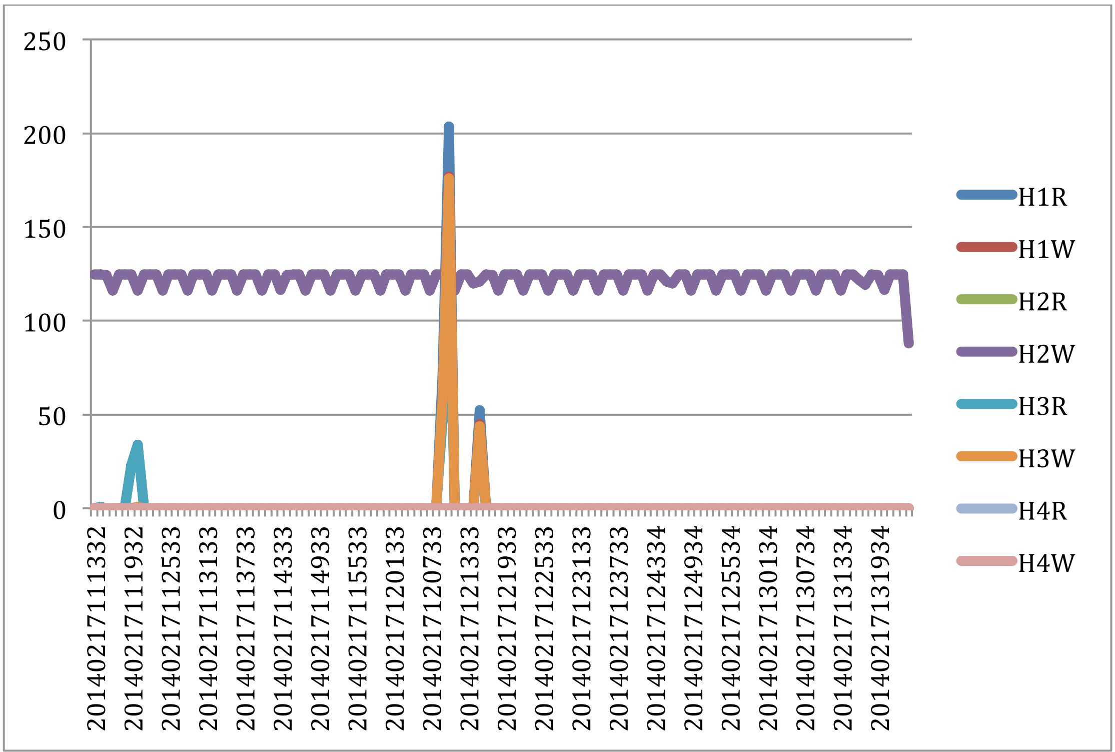
When we moved data from LTO-5 to T10K-B drives. We did see much more better results than moving data between LTO-5 and LTO-5. Although there where short periods of time where the transfer-rate broke down to approx.. 60 MB/s theses times were much shorter than with LTO-5 to LTO-5. When we compare the performance values for reading and writing from TSM instrumentation we can see that this performance drops where caused by the LTO side.

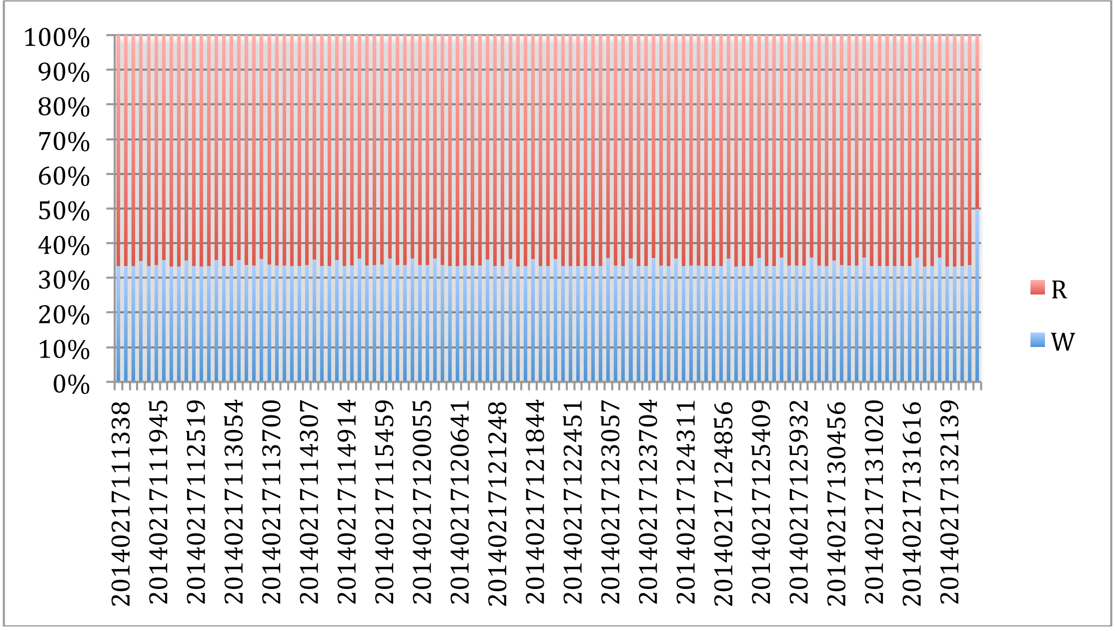




5.10 Moving data T10K to T10K with TSM move data

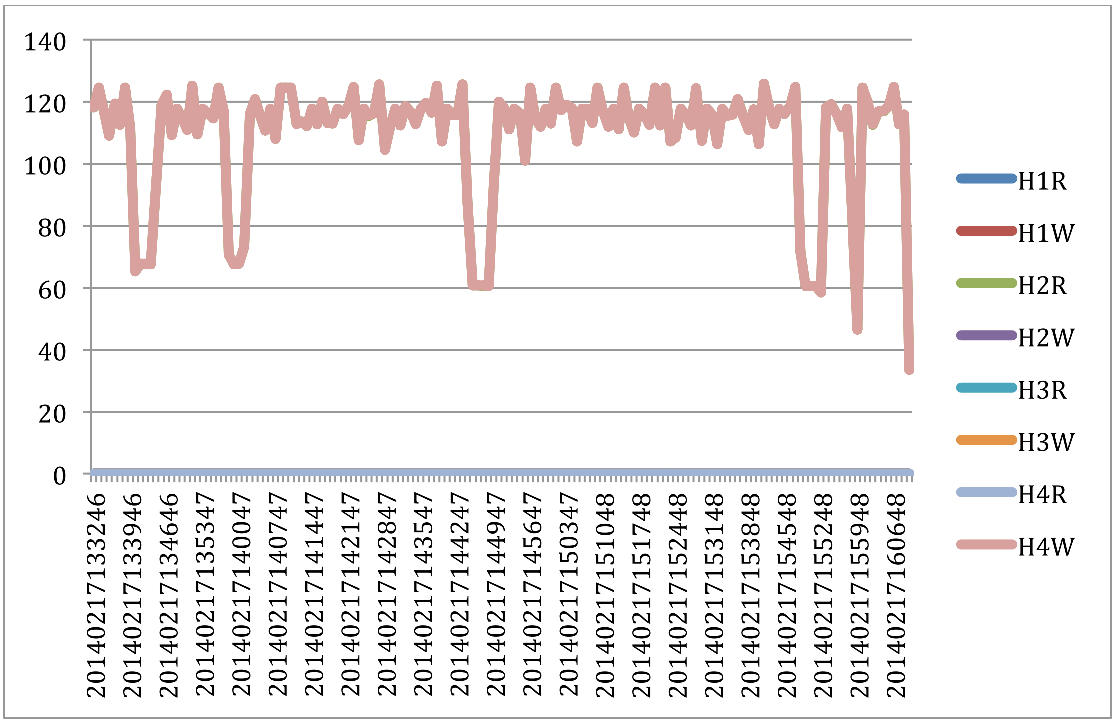
When we moved data between two T10K-B drives the move data run with the highest possible performance evenly over the whole time. So even though T10K-B has only two speeds (120 and 70 MB/s) and only a Buffer of 256 MB per drive it did much better than LTO.

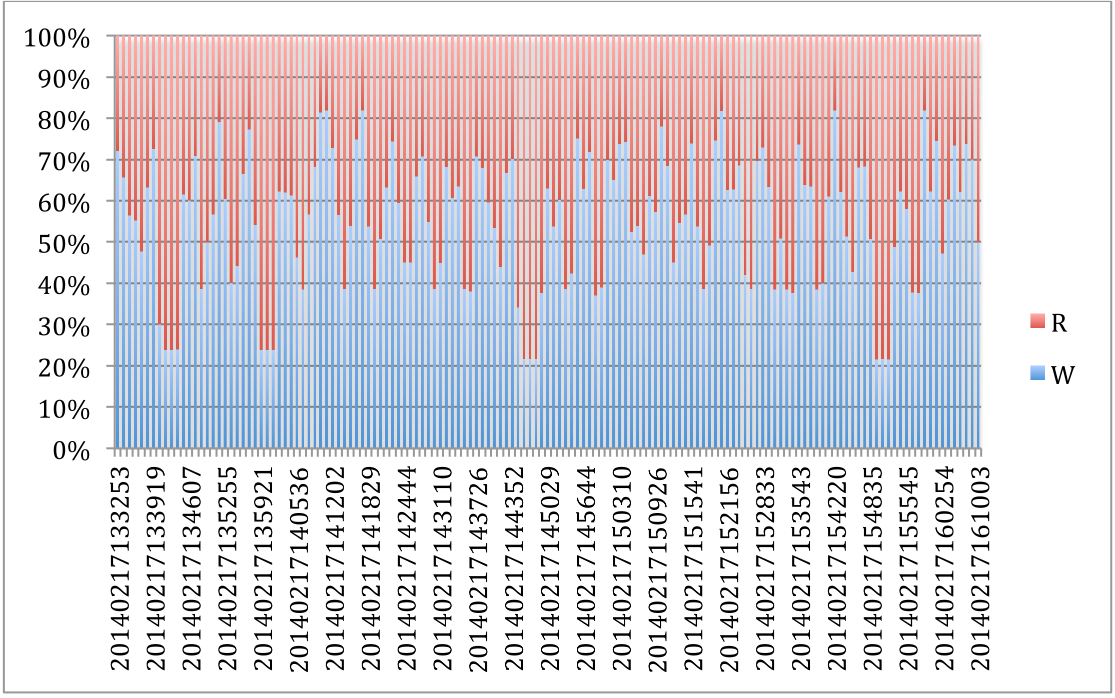




5.11 Moving data T10K to LTO-5 with TSM move data

When we moved data from T10K-B to LTO-5 back again, we could see the same pattern as we saw when we moved it from LTO to T10K. Most of the time performance was good with several short occasions where the transfer-rate broke down. When we compare the performance values for reading and writing from TSM instrumentation we can see that this performance drops where caused by the LTO side.



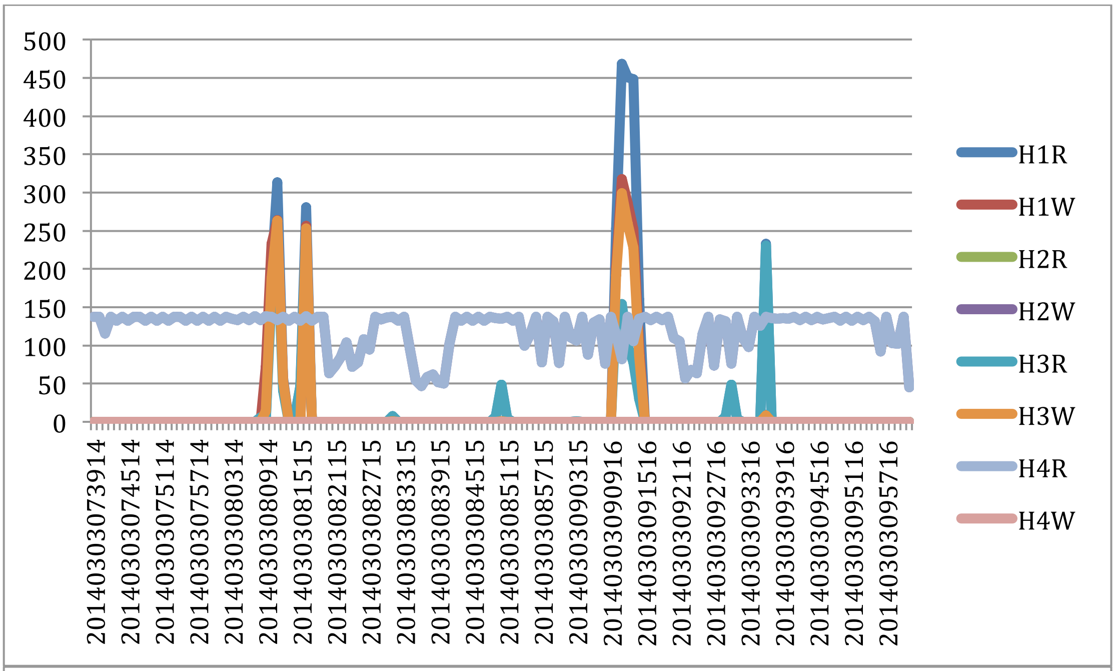


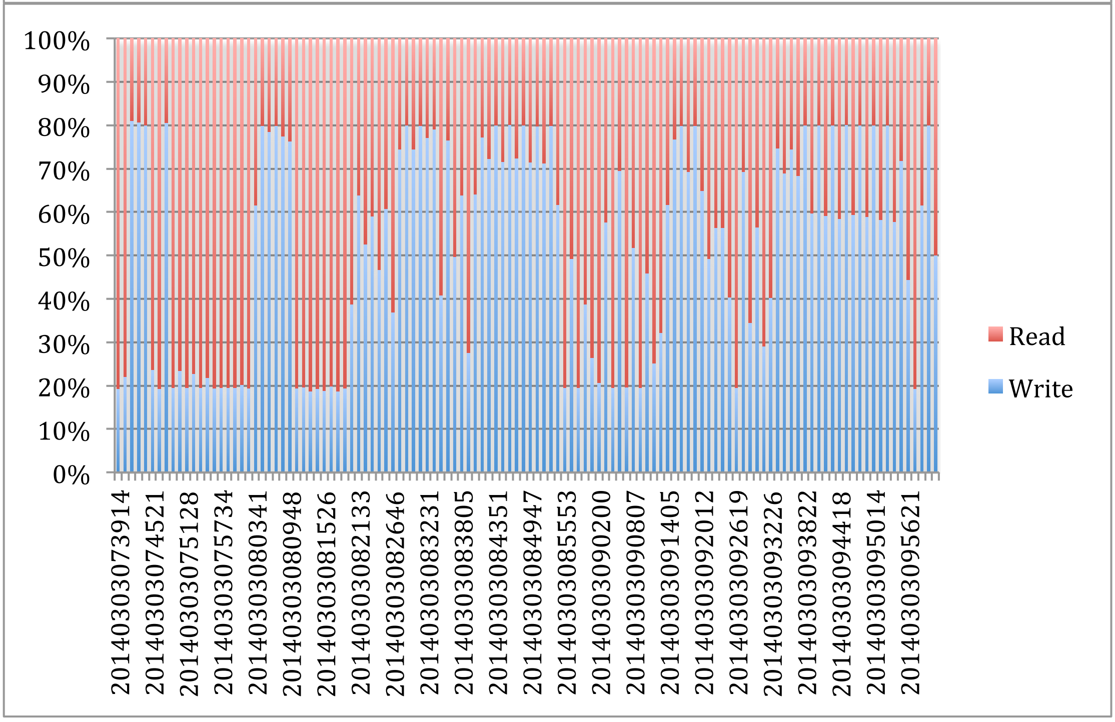
5.12 Moving data LTO-5 to LTO-5 with FW E24F

We moved the 1TB file between LTO-5 drives two times using drives running the E24F testing Firmware from IBM.

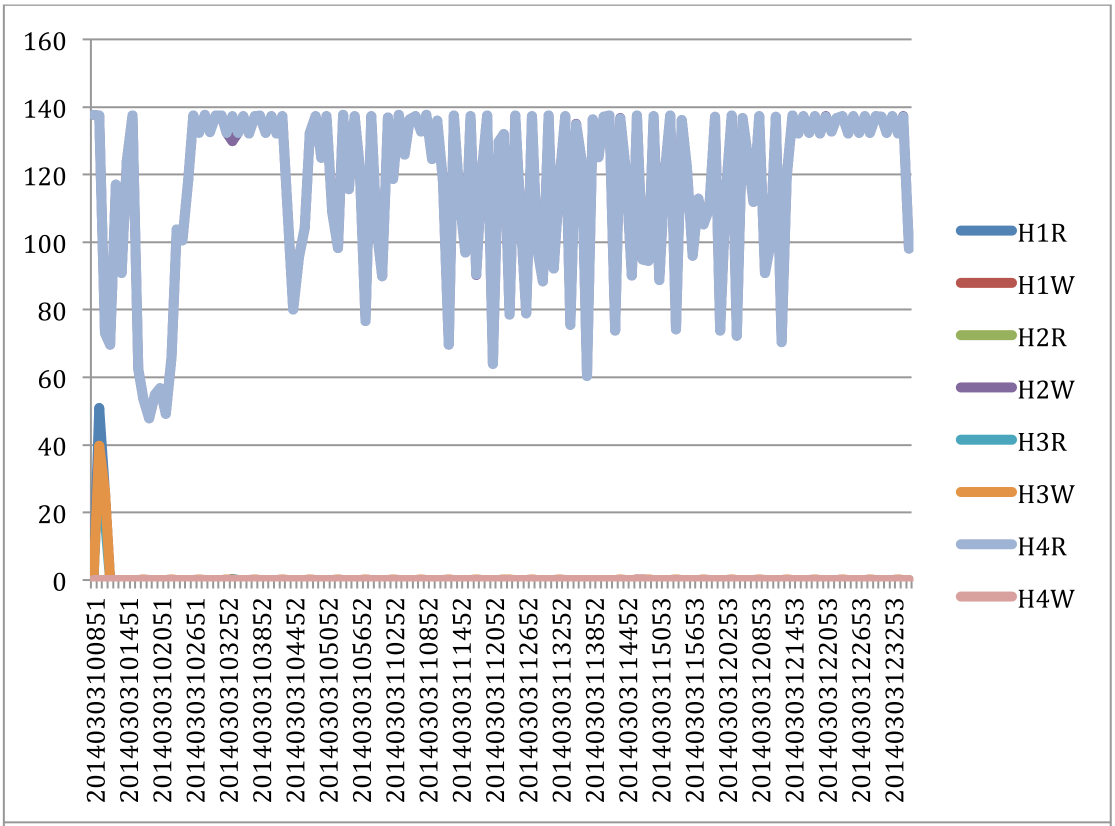
Moving data with TSM using LTO-5 FW E24F showed a decent improvement compared to the D8D4 FW. However results are still not optimal. What we see is that once speed matching goes down, it recovers more quickly from the lower speeds. However there are still too much and two high jitters of the tape to tape performance. So we recognize this as a step in the right direction but not as an ultimate solution to the problem.

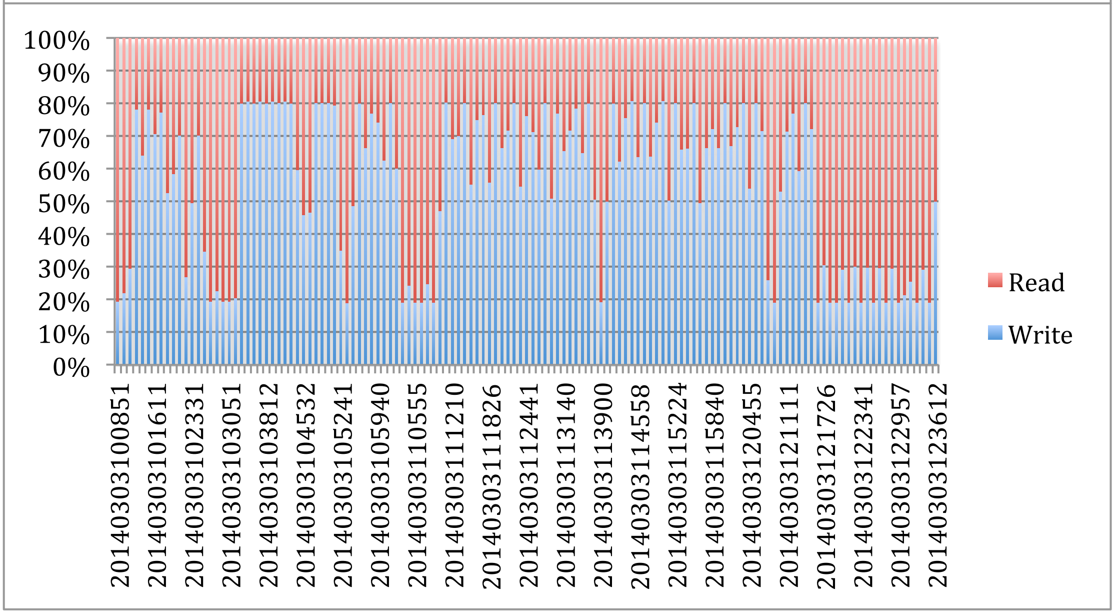
1st Run:





2nd Run:





6. Already taken actions

In order to solve the performance problem we already tried to switch off Intel C-Sates of the CPUs by disabling it via UEFI and setting intel_idle.max_cstate=0 and processor.max_cstate=0 in the Linux kernel. Since this was mentioned as a possible cause for certain I/O performance problems. (http://www.novell.com/support/kb/doc.php?id=7011982)

Also we switched off Intel Enhanced Speed Step Tech to run the server at full speed all the time.

However those settings had no or at least no significant impact on the problem.

Also we switched on bottleneckmon on the FC Switches but it did not indicate any problem either.

We tried to recreate the problem with Oracle T10K drives on similar server hardware attached to the same SAN Fabric, which are driven via TSM SCSI-Generic passthru driver – meaning TSM sends plain SCSI commands to the tape issuing ioctls to a Linux sg device - instead of lin_tape. In this constellation we could not observe the problem.

We also eliminated TSM from the equation by showing that the problem exists also outside of TSM by using dd for tape to tape copy operations.

7. Conclusion

As only tape to tape workloads are affected and as those jobs are not slow all the time - as seen above - and it can be observed on all LTO tape drives, we don't think there is a direct problem with individual drives.

We could imagine that there is some kind of speed matching problem between the drives, as the bottlenecked side alternates between reading and writing.

As this seems to be a complex issue, we could also involve IBM System-X support together with IBM Linux support. The X3850 X5 machine mentioned above has support contracts for both.

We also have a single Brocade DP 16Gbit FC Adapter. If desired we could install this card on the server and see if the problem also persists with a completely different HBA hardware and driver stack.

We guess that the problem exists already for a longer time. However the reason why we now stumbled across it is, that we currently started migrating from LTO-5 to LTO-6 and also have switched our disaster recovery strategy from TSM Server-to-Server backups to direct STG-Pool Backups via SAN. Therefore we currently have to move about 10 PB from tape-to-tape and since it does not run with the excepted speed we noticed and reported this problem.

8. Next Steps

We rely on the help and expertise of IBM-Support in this case, since this seems to be a complex issue. Therefore we would like to get suggestions from IBM support on how to proceed with this.