Oracle Database 12c Release 2 and Parallel NFS – What's It Good for?

This is the second part of the article by Christian Pfundtner, DB Masters GmbH, the first part can be found in ORAWORLD #5 at www.oraworld.org.

Test Setup Oracle 12cR2 with dNFS with and without pNFS

- Oracle VM Server with one VM with Oracle 12cR2.
- NetApp FAS 3170 Cluster consisting of two storage heads with 51 x 300 GB hard disks each o one VServer and two LIFs (IP addresses – one per storage head)
- jumbo frames (MTU=9000) configured

This setup (old hardware, virtualized, ...) is not suitable for a statement on absolute performance values of current Intel servers and NetApp storages – these are significantly faster! We have already carried out PoCs with more than 140,000 IOps and more than 1,000 Mbps over 5 years ago. However, the hardware is sufficient to determine the differences in performance of the various NFS versions.

NetApp Storage Configuration

The two FAS3170 form a cluster of two nodes (fas3170a, fas3170b). The storage node has 10GBit LAN interfaces, each configured with a LIF (virtual interface) of the vServer vsdb122b.

Interface Name	Data Protocol Access	Management Access	IP Address/WWPN	Current Port
vsdb122b_if1	nfs	No	10.146.146.61	naclu1-3170a:a0b-146
vsdb122b_if2	nfs	No	10.146.146.161	naclu1-3170b:a0b-146

Even though there are two DATA volumes, the database files are only located on volume DATA1. The volumes are appropriately created on the disks of one of the storage nodes:

- DATA1 is on FAS3170; DATA1 is only utilized for database files
- DATA2 is on FAS3170b

In the course of the various tests, the DATA1 volume is moved from one storage node to another.

Oracle 12c2 VM Configuration

The Oracle 12c Release 2 database runs on an Oracle VM server with dual Intel Xeon X5450 (3GHz) and 8 cores in total. Storage connection is implemented with 10GBit network cards with IP address 10.146.2.61. The VM itself runs with OEL 7.3 and may use 4 cores.

The database with 8 GB is only located in the NetApp volume DATA1. This volume is mounted on /u01/app/oracle/oradata/DB122B/data1/ with NFS. The database is therefore small enough to fit in the cache of the FAS3170 storage node.

```
select name from v$datafile;

NAME
------------------------------------------------------------
/u01/app/oracle/oradata/DB122B/data1/system01.dbf
/u01/app/oracle/oradata/DB122B/data1/sysaux01.dbf
/u01/app/oracle/oradata/DB122B/data1/undotbs01.dbf
/u01/app/oracle/oradata/DB122B/data1/pdbseed/system01.dbf
/u01/app/oracle/oradata/DB122B/data1/pdbseed/sysaux01.dbf
/u01/app/oracle/oradata/DB122B/data1/users01.dbf
/u01/app/oracle/oradata/DB122B/data1/pdbseed/undotbs01.dbf
```

The Oracle database was configured for usage of dNFS. The successful configuration of dNFS can be verified, for example, in v$dnfs_servers:

```
select SVRNAME, DIRNAME from V$DNFS_SERVERS;

SVRNAME           DIRNAME                                    NFSVERSION
----------------- ------------------------------------------ ----------------
vsdb122b-lif1     /vol/db122b_ctrl1/db122b_ctrl1_qt          NFSv3.0
vsdb122b-lif1     /vol/db122b_data1/db122b_data1_qt          NFSv3.0
vsdb122b-lif1     /vol/db122b_log1/db122b_log1_qt            NFSv3.0
vsdb122b-lif2     /vol/db122b_ctrl2/db122b_ctrl2_qt          NFSv3.0
vsdb122b-lif2     /vol/db122b_data2/db122b_data2_qt          NFSv3.0
vsdb122b-lif2     /vol/db122b_log2/db122b_log2_qt            NFSv3.0
```

Besides this view, there are more views with information on dNFS:

- V$DNFS_CHANNELS … shows which NFS connections the database uses
- V$DNFS_FILES … shows which files are on dNFS
- V$DNFS_STATS ... performance statistics per file

These views are used for performance analysis in the scope of the performance tests.

The IP addresses of the LIFs are registered in /etc/hosts so that there are no name resolution issues:

```
/etc/hosts
10.146.146.61  vsdb122b-lif1.example.com vsdb122b-lif1
10.146.146.161 vsdb122b-lif2.example.com vsdb122b-lif2
```

Since there are several paths to the data, an oranfstab configuration is required. This oranfstab is also required to turn on pNFS. Parameter nfs_version must be set to pNFS for this purpose – examples for the oranfstab follow in the various performance tests.

References for the configuration of pNFS:

Oracle 12cR2 Database Installation Guide: Creating an oranfstab File for Direct NFS Client

https://docs.oracle.com/database/122/SSDBI/creating-an-oranfstab-file-for-direct-nfs-client.htm

Performance Test – Overview

As performance test, we use Oracle I/O Calibrate with a number of 100 disks (2 x 50) and a maximum latency of 20. I am well aware of the various reasons that are against this test (no SQL processing, no writes,...). Since the only reason for the tests is to determine the impact on performance with and without pNFS, a read-only access on the Oracle database files is sufficient for us. Only the RANDOM I/O parts of the I/O Calibrate are used for the CPU rating. The I/O Calibrate is always run two times in a row because the cache is potentially not filled during the first run.

The database size of 8 GB is chosen to make sure that the database can fit into the memory of the applicable storage node. This choice was made because the focus is on performance difference i.e. performance overhead – which is in the range of microseconds – and not on disk io performance which is in the range of milliseconds.

Additionally, we determine the CPU usage on both the database server and on the NetApp storage node. Unfortunately, the NetApp storages are not exclusively available for our tests, so that, without tests, the CPU usage varies between 3% and 10% and between 500 to 2,000 network packets. Since our tests generate a significantly higher load, it should still be possible to make a statement.

Before each test, the Oracle database instance is restarted to take over possible changed settings in the oranfstab on the one hand and to clean up the v$dnfs views on the other hand.

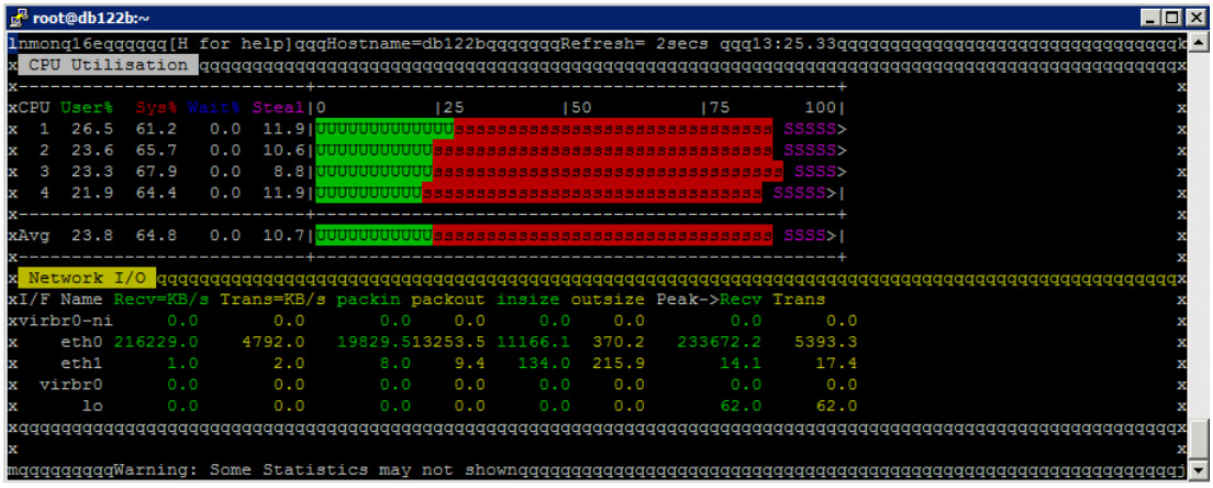
Performance Test #1: NFSv3, Access on DATA1 on FAS3170a via LIF on FAS3170a

nfs_version has not been specified in the oranfstab so that the database instance uses the default of NFSv3.

```
SVRNAME           DIRNAME                                    NFSVERSION
----------------- ------------------------------------------ ----------------
vsdb122b-lif1     /vol/db122b_ctrl1/db122b_ctrl1_qt          NFSv3.0
vsdb122b-lif1     /vol/db122b_data1/db122b_data1_qt          NFSv3.0
vsdb122b-lif1     /vol/db122b_log1/db122b_log1_qt            NFSv3.0
vsdb122b-lif2     /vol/db122b_ctrl2/db122b_ctrl2_qt          NFSv3.0
vsdb122b-lif2     /vol/db122b_data2/db122b_data2_qt          NFSv3.0
vsdb122b-lif2     /vol/db122b_log2/db122b_log2_qt            NFSv3.0
```

CPU Usage on the Database Server

CPU usage via NMON shows (10% steal) that the VM does not offer enough CPU resources and can therefore be regarded as limiting factor.



CPU Usage NetApp Storage

FAS3170a … DATA1 is located here

CPU	NFS	CIFS	HTTP	Total	Net in	Net out	Disk read	Disk write	Tape read	Tape write	Cache age	Cache hit	CP time	CP ty	Disk util	OTHER	FCP	iSCSI	FCP in	FCP out	iSCSI in	iSCSI out
30%	20411	0	0	20436	3874	135594	0	0	0	0	>60	100%	0%	-	0%	0	0	25	0	0	24	98
35%	25078	0	0	25107	5015	204491	0	0	0	0	>60	100%	0%	-	0%	0	0	29	0	0	57	196
37%	27408	0	0	27461	5522	210775	32	24	0	0	>60	100%	0%	-	2%	10	0	43	0	0	94	229
37%	23701	0	0	23866	4517	190456	5704	7440	0	0	>60	100%	36%	T	16%	144	0	21	0	0	25	66
35%	25524	0	0	25551	4839	187617	0	0	0	0	>60	100%	0%	-	0%	0	0	27	0	0	57	147
38%	27733	0	0	27772	5691	224473	40	24	0	0	>60	100%	0%	-	2%	0	0	39	0	0	93	195
38%	27575	0	0	27597	5577	227890	0	0	0	0	>60	100%	0%	-	0%	0	0	22	0	0	25	94
38%	28064	0	0	28103	5421	226463	8	0	0	0	>60	100%	0%	-	3%	10	0	29	0	0	57	217
37%	27020	0	0	27048	5813	220913	28	32	0	0	>60	100%	0%	-	4%	0	0	28	0	0	61	77
39%	29290	0	0	29327	5649	234077	4	0	0	0	>60	100%	0%	-	1%	0	0	37	0	0	57	270
36%	26670	0	0	26706	4942	208776	4	0	0	0	>60	100%	0%	-	1%	0	0	36	0	0	57	225

CPU usage fluctuates between 30% and, temporarily, up to 55%. When we subtract the typical base load, CPU usage by the benchmark should be in the range of 30% to 45%.

ANY1+	ANY2+	ANY3+	ANY4+	AVG	CPU0	CPU1	CPU2	CPU3
72%	49%	35%	21%	45%	40%	27%	54%	58%
68%	44%	32%	20%	42%	36%	27%	51%	53%
79%	53%	36%	24%	49%	42%	31%	60%	63%
74%	47%	31%	18%	43%	37%	24%	54%	58%
67%	42%	27%	15%	38%	32%	21%	49%	51%
69%	42%	27%	16%	39%	32%	22%	49%	53%
62%	39%	27%	16%	36%	31%	22%	45%	47%
64%	40%	29%	17%	38%	32%	24%	47%	49%
44%	25%	17%	10%	24%	21%	15%	31%	31%
69%	46%	32%	21%	43%	37%	26%	52%	55%
77%	51%	36%	22%	47%	40%	29%	60%	59%
66%	43%	31%	19%	40%	36%	24%	50%	52%
64%	37%	24%	13%	35%	28%	19%	46%	48%

The cores of the storages are almost equally used.

FAS3170b … no additional IO by the benchmark, as expected.

Oracle File - IO Statistics (v$iostat_file)

Oracle gives information on I/O requests of IO Calibrate in this view. Since we are particularly interested in random I/O latency, we only consider the "small read IOs".

FILE_NO	AVG_SMALL_READ_US	AVG_SMALL_SYNC_READ_US
1	355.507436	355.538691
3	355.995664	356.228999
4	356.920581	356.920581
5	356.409876	356.398215
6	355.770811	355.783105
7	.75	.75
8	356.28306	356.296397

Read latency across all database files is approx. 355 us (i.e. 0.355 ms).

Result Of IO Calibrate

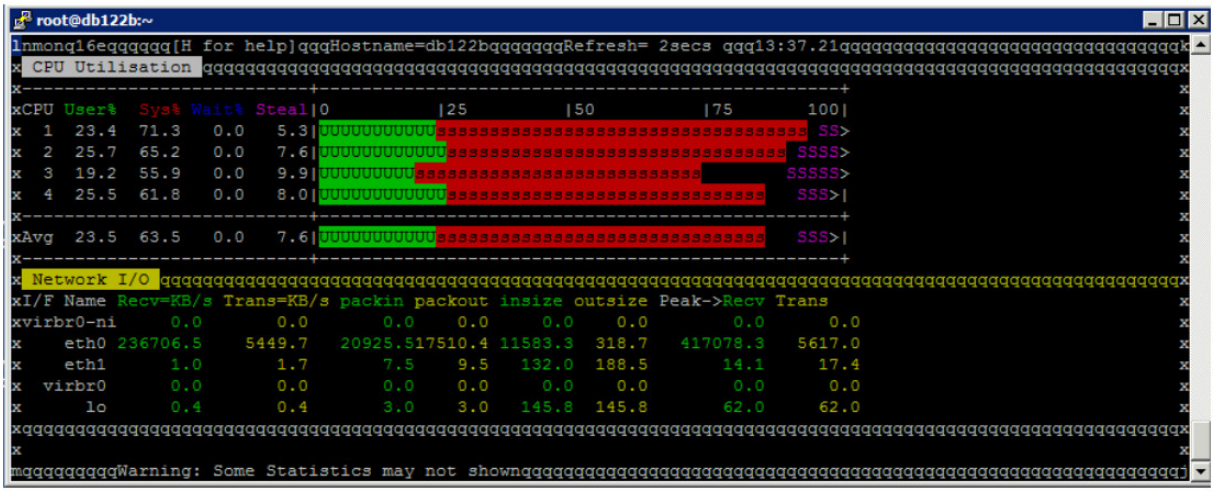
IOPS = 27066
Actual Latency = 1
MB/sec = 362

Performance Test #2: NFSv3, Access on DATA1 on FAS3170b via LIF on FAS3170a

Volume DATA1 is moved from the FAS3170a disks to the FAS3170b disks. The volume is still accessed via IP of FAS3170a. The move is carried out online and has no impact on the NFS mounts.

CPU Usage on the Database Server

Usage on the database server is comparable to the first test.



CPU Usage NetApp Storage

On the NetApp storage, there is a dramatically different picture.

FAS3170a ... the IP address used for access is located here

CPU	NFS	CIFS	HTTP	Total	Net in	Net out	Disk read	Disk write	Tape read	Tape write	Cache age	Cache hit	CP time	CP ty	Disk util	OTHER	FCP	iSCSI	FCP in	FCP out	iSCSI in	iSCSI out
18%	784	0	0	976	107896	111011	4	0	0	0	7	100%	0%	-	1%	145	0	47	0	0	102	229
27%	664	0	0	691	203384	198887	0	0	0	0	7	100%	0%	-	0%	0	0	27	0	0	57	164
29%	379	0	0	402	220757	207171	32	24	0	0	7	100%	0%	-	2%	0	0	23	0	0	25	66
28%	519	0	0	572	210769	202526	0	8	0	0	7	100%	0%	-	1%	9	0	44	0	0	94	262
28%	972	0	0	1004	201880	201141	0	0	0	0	7	100%	0%	-	0%	0	0	32	0	0	57	213
31%	1267	0	0	1288	213246	214907	24	24	0	0	7	100%	0%	-	2%	0	0	21	0	0	25	98
23%	3567	0	0	3603	97844	103090	5304	7972	0	0	7	100%	25%	T	12%	0	0	36	0	0	57	229
31%	1165	0	0	1206	217912	211595	8	0	0	0	7	100%	0%	-	1%	0	0	41	0	0	94	180
35%	1354	0	0	1384	240995	241761	24	32	0	0	7	100%	0%	-	2%	9	0	21	0	0	25	82

CPU usage fluctuates between 20% and almost 40% – only for bi-directional transfer of IO requests via Cluster Interconnect.

When we look at the CPU usage in detail – the head has 4 cores – we notice that two of these cores are significantly busier than the other two.

ANY1+	ANY2+	ANY3+	ANY4+	AVG	CPU0	CPU1	CPU2	CPU3
79%	43%	14%	2%	35%	24%	9%	51%	58%
76%	40%	12%	1%	33%	22%	7%	50%	53%
83%	48%	17%	3%	38%	29%	11%	55%	58%
85%	49%	16%	2%	39%	27%	9%	58%	60%
83%	47%	15%	2%	37%	27%	7%	55%	60%
77%	41%	14%	2%	34%	24%	8%	51%	55%
72%	39%	14%	3%	33%	22%	10%	48%	51%
86%	53%	18%	2%	41%	30%	11%	61%	61%
82%	50%	20%	5%	40%	28%	19%	53%	60%
84%	48%	14%	1%	37%	25%	8%	58%	59%
92%	60%	26%	7%	47%	39%	25%	61%	62%
73%	34%	12%	2%	31%	19%	10%	46%	49%
67%	29%	9%	2%	27%	16%	7%	41%	45%
78%	41%	13%	1%	34%	21%	7%	53%	53%
82%	47%	15%	2%	37%	26%	8%	55%	59%

```
naclu1-3170a*>
```

CPU usage fluctuates between 20% and almost 40% – only for bi-directional transfer of IO requests via Cluster Interconnect.

FAS3170b ... this is where the files are located on the disks

CPU	NFS	CIFS	HTTP	Total	Net in	kB/s out	Disk read	kB/s write	Tape read	kB/s write	Cache age	Cache hit	CP time	CP ty	Disk util	OTHER	FCP	iSCSI	FCP in	kB/s out	iSCSI in	kB/s out
30%	20332	0	0	20336	5090	168934	252	0	0	0	43s	100%	0%	-	2%	0	0	4	0	0	0	0
37%	24786	0	0	24797	6273	209242	260	24	0	0	43s	100%	0%	-	3%	2	0	9	0	0	0	0
22%	13481	0	0	13485	3460	113624	156	0	0	0	43s	100%	0%	-	3%	0	0	4	0	0	0	0
36%	22662	0	0	22664	5710	191144	7416	7448	0	0	43s	100%	35%	T	8%	0	0	2	0	0	0	0
39%	25709	0	0	25722	6500	217126	252	24	0	0	43s	100%	0%	-	2%	12	0	1	0	0	0	0
35%	23490	0	0	23492	5893	198291	188	0	0	0	43s	100%	0%	-	3%	0	0	2	0	0	0	0
34%	23112	0	0	23117	5813	194973	108	0	0	0	43s	100%	0%	-	2%	1	0	4	0	0	0	0
37%	25374	0	0	25377	6395	214141	180	24	0	0	43s	100%	0%	-	4%	0	0	3	0	0	0	0
19%	11523	0	0	11669	2886	97022	56	8	0	0	43s	100%	0%	-	2%	143	0	3	0	0	0	0
36%	23809	0	0	23825	6008	200954	100	0	0	0	43s	100%	0%	-	2%	12	0	4	0	0	0	0
41%	28339	0	0	28342	7156	239912	214	24	0	0	43s	100%	0%	-	4%	1	0	2	0	0	0	0

We see the same load that was previously on FAS3170a. CPU usage fluctuates between 20% and 50%, with values below 30% and over 40% being regarded as statistical outliers.

ANY1+	ANY2+	ANY3+	ANY4+	AVG	CPU0	CPU1	CPU2	CPU3
54%	44%	33%	24%	39%	34%	31%	44%	48%
36%	27%	20%	15%	25%	21%	20%	28%	30%
46%	39%	31%	24%	35%	30%	29%	39%	41%
53%	45%	37%	28%	41%	35%	34%	46%	49%
33%	27%	20%	14%	24%	20%	19%	28%	29%
54%	45%	36%	27%	41%	35%	34%	46%	48%
53%	44%	34%	24%	39%	32%	31%	45%	47%
36%	27%	19%	13%	24%	19%	18%	28%	31%
52%	40%	30%	22%	36%	31%	29%	41%	45%
46%	38%	30%	23%	35%	29%	29%	39%	42%
53%	44%	35%	26%	40%	34%	33%	44%	48%
40%	32%	23%	16%	28%	23%	22%	33%	34%
47%	41%	33%	26%	37%	32%	32%	41%	43%
47%	39%	31%	23%	35%	30%	29%	40%	42%
51%	42%	33%	25%	38%	32%	31%	43%	45%

All CPUs are used similarly heavy on this node.

If we consider the I/O times for small reads from v$iostat_file and calculate the average small reads I/O times, the result is as follows:

FILE_NO	AVG_SMALL_READ_US	AVG_SMALL_SYNC_READ_US
1	392.113305	392.128792
3	392.552918	392.81291
4	393.717501	393.717501
5	392.255744	392.247527
6	392.842502	392.844232
7	2.25	2.25
8	393.122799	393.134732

This means that the I/O via the "wrong" storage node takes approx. 35 us longer – that is not much – however, CPU usage on the storage node is dramatically higher!

Result Of IO Calibrate

I/O Ops/sec = 26416
Actual Latency = 0
MB/sec = 373

If we access data via network card on one storage node that are located on the other storage node, this generates approximately the same CPU load on both storage nodes – in other words, we duplicate CPU need on both storages by using the wrong IP address. The result of the I/O Calibrate is practically identical – there is no difference from a database point of view – the additional latency from running I/O via two storage heads is therefore in the range of significantly below 0.04 ms. Internal measuring accuracy of the I/O Calibrate is whole ms, which makes benchmarking via v$iostat_file (with microseconds) far more useful.

Performance Test #3: only NFSv4, Access on DATA1 on FAS3170a via LIF on FAS3170a

In the third test, we switch from NFSv3 to NFSv4 – but we still do not use pNFS functionality.

nfs_version in the oranfstab is now set to nfsv4 for all mounts:

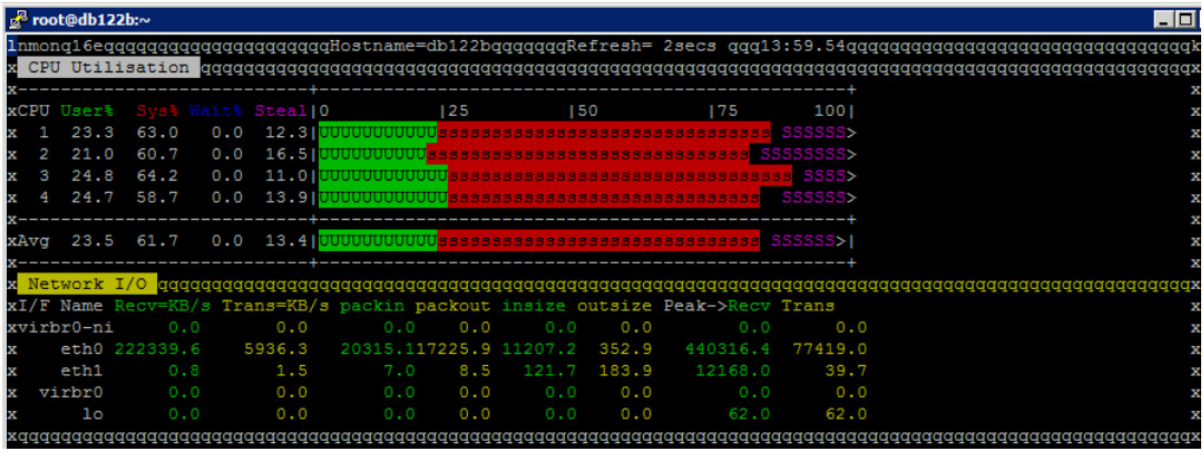
```
server: vsdb122b-lif1
local:  10.146.2.61
path:   10.146.146.61
donotroute
nfs_version: nfsv4
#nfs_version: pnfs
export: /vol/db122b_ctrl1/db122b_ctrl1_qt mount: /u01/app/oracle/oradata/DB122B/ctrl1
export: /vol/db122b_data1/db122b_data1_qt mount: /u01/app/oracle/oradata/DB122B/data1
export: /vol/db122b_log1/db122b_log1_qt   mount: /u01/app/oracle/oradata/DB122B/log1
server: vsdb122b-lif2
local:  10.146.2.61
path:   10.146.146.161
donotroute
nfs_version: nfsv4
#nfs_version: pnfs
export: /vol/db122b_ctrl2/db122b_ctrl2_qt mount: /u01/app/oracle/oradata/DB122B/ctrl2
export: /vol/db122b_data2/db122b_data2_qt mount: /u01/app/oracle/oradata/DB122B/data2
export: /vol/db122b_log2/db122b_log2_qt   mount: /u01/app/oracle/oradata/DB122B/log2
```

This is also correctly displayed in v$dnfs_servers:

SVRNAME	DIRNAME	NFSVERSION
vsdb122b-lif1	/vol/db122b_ctrl1/db122b_ctrl1_qt	NFSv4.0
vsdb122b-lif1	/vol/db122b_data1/db122b_data1_qt	NFSv4.0
vsdb122b-lif1	/vol/db122b_log1/db122b_log1_qt	NFSv4.0
vsdb122b-lif2	/vol/db122b_ctrl2/db122b_ctrl2_qt	NFSv4.0
vsdb122b-lif2	/vol/db122b_data2/db122b_data2_qt	NFSv4.0
vsdb122b-lif2	/vol/db122b_log2/db122b_log2_qt	NFSv4.0

CPU Usage on the Database Server

CPU usage on the database server is in the same range as with NFSv3.



CPU Usage NetApp Storage

There are differences on the storages compared to NFSv3 – metadata seem to be queried from both storage nodes.

FAS3170a

CPU	NFS	CIFS	HTTP	Total	Net in	kB/s out	Disk read	kB/s write	Tape read	kB/s write	Cache age	Cache hit	CP time	CP ty	Disk util	OTHER	FCP	iSCSI	FCP in	kB/s out	iSCSI in	kB/s out
60%	20572	0	0	20622	5428	181175	32	0	0	0	4	100%	0%	-	1%	5	0	45	0	0	162	162
59%	20080	0	0	20115	5149	172555	8	0	0	0	4	100%	0%	-	1%	0	0	35	0	0	24	106
71%	24720	0	0	24783	5969	209765	16	32	0	0	4	100%	0%	-	3%	7	0	56	0	0	126	292
47%	15996	0	0	16024	3932	134030	16	0	0	0	4	100%	0%	-	1%	6	0	22	0	0	25	82
70%	23044	0	0	23559	5809	197948	0	0	0	0	4	100%	0%	-	0%	494	0	21	0	0	24	64
72%	25234	0	0	25287	6002	207946	8	24	0	0	4	100%	0%	-	6%	1	0	52	0	0	126	391
74%	25176	0	0	25201	5835	203503	8966	9753	0	0	4	100%	36%	Tv	9%	3	0	22	0	0	24	98
69%	24615	0	0	24643	6113	205254	56	63	0	0	4	100%	3%	:	3%	6	0	22	0	0	24	97
76%	27549	0	0	27599	6476	220487	60	24	0	0	4	100%	0%	-	1%	5	0	45	0	0	126	277
72%	26584	0	0	26621	6375	214179	32	0	0	0	4	100%	0%	-	1%	15	0	22	0	0	24	65
69%	24599	0	0	24622	6178	200000	0	0	0	0	4	100%	0%	-	0%	0	0	23	0	0	24	65
75%	26730	0	0	26778	6609	220190	28	24	0	0	4	100%	0%	-	2%	0	0	48	0	0	131	294

CPU Usage in detail

ANY1+	ANY2+	ANY3+	ANY4+	AVG	CPU0	CPU1	CPU2	CPU3
100%	92%	66%	35%	73%	68%	84%	85%	57%
100%	91%	65%	34%	73%	72%	83%	83%	53%
98%	90%	65%	32%	71%	68%	82%	84%	51%
95%	74%	44%	20%	58%	48%	72%	70%	43%
96%	84%	61%	34%	69%	62%	79%	81%	53%
100%	91%	63%	31%	71%	67%	81%	84%	53%
91%	65%	29%	10%	49%	30%	64%	66%	37%
92%	76%	54%	27%	63%	53%	74%	75%	50%
99%	89%	65%	34%	72%	67%	82%	84%	56%
100%	93%	63%	34%	73%	67%	84%	84%	55%
100%	90%	60%	30%	70%	71%	80%	79%	50%
100%	92%	62%	33%	72%	69%	83%	82%	53%

Generally, CPU need on the storage is significantly higher when using NFSv4. The reason for this is the significantly higher complexity of the protocol due to additional functions. In case of our slightly old storages, CPU load increases from previous 30% and 45% (NFSv3) to 50% to slightly over 70% - an increase of approx. 20%

FAS3170b

As expected, the load is very low here.

CPU	NFS	CIFS	HTTP	Total	Net in	kB/s out	Disk read	kB/s write	Tape read	kB/s write	Cache age	Cache hit	CP time	CP ty	Disk util	OTHER	FCP	iSCSI	FCP in	kB/s out	iSCSI in	kB/s out
13%	47	0	0	47	143	315	0	0	0	0	>60	100%	0%	-	0%	0	0	0	0	0	0	0
13%	30	0	0	49	147	16	36	32	0	0	>60	100%	0%	-	3%	18	0	1	0	0	0	0
14%	21	0	0	171	145	82	12	0	0	0	>60	100%	0%	-	1%	149	0	1	0	0	0	0
13%	34	0	0	41	96	280	4	0	0	0	>60	100%	0%	-	1%	0	0	7	0	0	0	0
13%	19	0	0	20	101	29	64	24	0	0	>60	100%	0%	-	3%	0	0	1	0	0	0	0
12%	35	0	0	38	90	149	0	0	0	0	>60	100%	0%	-	0%	3	0	0	0	0	0	0
12%	28	0	0	128	97	147	8	8	0	0	>60	100%	0%	-	1%	94	0	6	0	0	0	0

ANY1+	ANY2+	ANY3+	ANY4+	AVG	CPU0	CPU1	CPU2	CPU3
22%	7%	2%	1%	8%	11%	8%	6%	9%
5%	1%	0%	0%	2%	2%	1%	1%	4%
9%	2%	0%	0%	3%	3%	2%	2%	5%
5%	1%	0%	0%	2%	2%	1%	1%	3%
6%	1%	0%	0%	2%	2%	2%	1%	4%
6%	1%	0%	0%	2%	2%	1%	1%	4%
6%	1%	0%	0%	2%	2%	1%	1%	4%
6%	1%	0%	0%	2%	2%	1%	1%	5%
8%	1%	0%	0%	3%	4%	1%	1%	5%
6%	1%	0%	0%	2%	3%	1%	1%	4%
10%	3%	1%	0%	4%	3%	2%	3%	7%
6%	1%	0%	0%	2%	2%	1%	1%	4%
5%	1%	0%	0%	2%	2%	1%	1%	4%
9%	2%	0%	0%	3%	2%	2%	3%	5%
18%	3%	1%	0%	6%	3%	4%	7%	8%
4%	1%	0%	0%	2%	2%	1%	1%	4%
5%	1%	0%	0%	2%	2%	1%	1%	3%
9%	2%	0%	0%	3%	3%	2%	1%	7%
5%	1%	0%	0%	2%	2%	1%	1%	4%

While the other node did not have anything to do in case of access via the correct storage node with NFSv3, we notice a – quite low – but measurable load in this test. Since the load fluctuates heavily (on start of IO Calibrate, the load was in the range of almost 10% for a short time) and became lower over time, querying metadata on the second storage node should only generate some % of load.

V$IOSTAT_FILE

If we consider the I/O times in v$iostat_file, we notice that these times are also significantly higher compared to usage of NFSv3.

FILE_NO	AVG_SMALL_READ_US	AVG_SMALL_SYNC_READ_US
1	430.709626	430.731171
3	431.11025	431.296142
4	433.162096	433.166678
5	431.648877	431.655014
6	430.952956	430.9747
7	2.33333333	2.33333333
8	431.838314	431.848624

Times are in the range of 430 us, 75 us (0.075 ms) higher than NFSv3.

Result of IO Calibrate

I/O Ops/sec = 23536
Actual Latency = 3
MB/sec = 378

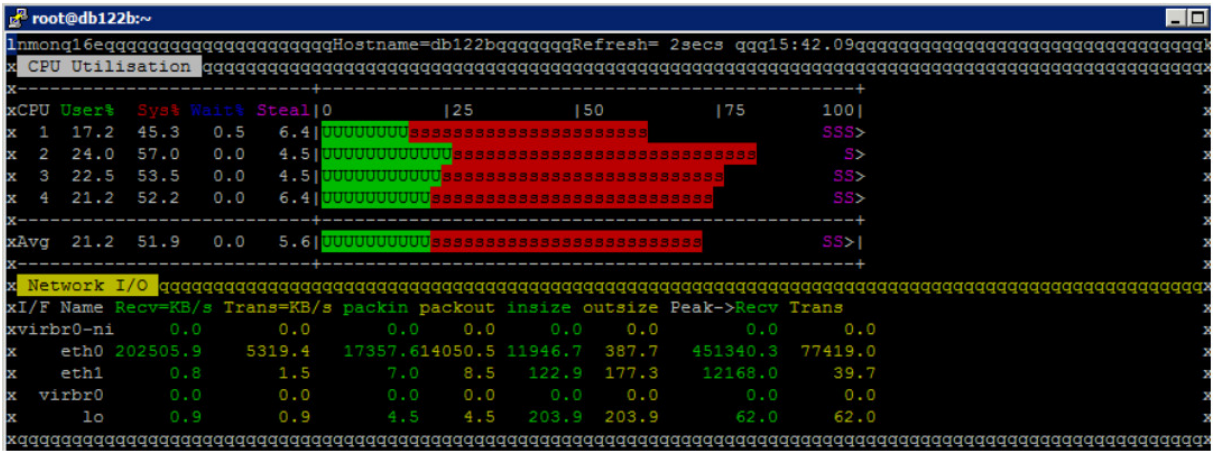
Regarding the relevant IOPS, the results of the I/O Calibrate are significantly poorer and there is also a higher latency. We think that the higher latency is due to the initial accesses on metadata since these accesses are averagely still below 0.45 ms in v$iostat_file.

Performance Test #4: NFSv4, Access on DATA1 on FAS3170b via LIF on FAS3170a

The oranfstab is the same as in the previous test. Only data is migrated to the disks of FAS3170b.

CPU Usage on the Database Server

CPU usage and network throughput on the database server clearly indicates that there is something wrong compared to the previous performance tests. CPU usage is significantly lower – which would be quite positive, but at the same time, I/O throughput is decisively poorer. The I/O Calibrate values will obviously reflect this.



CPU Usage NetApp Storage

FAS3170a

The CPU usage on the node responsible for the communication with the database server may be lower as in the previous test, but all I/O operations are forwarded to the other storage node.

CPU	NFS	CIFS	HTTP	Total	Net in	kB/s out	Disk read	kB/s write	Tape read	kB/s write	Cache age	Cache hit	CP time	CP ty	Disk util	OTHER	FCP	iSCSI	FCP in	kB/s out	iSCSI in	kB/s out
49%	629	0	0	666	158369	157128	0	0	0	0	21	100%	0%	-	0%	0	0	37	0	0	57	236
49%	393	0	0	451	158825	155752	24	24	0	0	21	100%	0%	-	2%	17	0	41	0	0	98	212
51%	822	0	0	988	160556	158018	8	8	0	0	21	100%	0%	-	0%	142	0	24	0	0	25	65
50%	847	0	0	902	156307	154725	0	0	0	0	21	100%	0%	-	0%	26	0	29	0	0	57	163
41%	483	0	0	525	128819	127640	24	24	0	0	21	100%	0%	-	2%	0	0	42	0	0	94	212
43%	366	0	0	391	141839	139766	0	0	0	0	21	100%	0%	-	0%	0	0	25	0	0	24	81
44%	362	0	0	405	147709	147179	0	0	0	0	21	100%	0%	-	0%	13	0	30	0	0	57	147
51%	337	0	0	382	159721	157076	6481	7206	0	0	21	100%	31%	T	10%	0	0	45	0	0	94	261
38%	330	0	0	357	127077	125967	16	0	0	0	21	100%	0%	-	1%	0	0	27	0	0	24	97
48%	404	0	0	439	163611	161703	4	0	0	0	21	100%	0%	-	1%	2	0	33	0	0	57	212
49%	356	0	0	405	163752	161685	24	24	0	0	21	100%	0%	-	2%	0	0	49	0	0	93	210

ANY1+	ANY2+	ANY3+	ANY4+	AVG	CPU0	CPU1	CPU2	CPU3
100%	77%	21%	5%	51%	31%	22%	74%	78%
100%	55%	11%	1%	42%	25%	14%	65%	64%
100%	66%	15%	2%	46%	29%	18%	69%	67%
100%	76%	15%	1%	48%	28%	19%	74%	73%
80%	48%	10%	1%	35%	21%	11%	54%	55%
100%	48%	10%	1%	40%	21%	13%	63%	64%
100%	74%	16%	1%	48%	29%	19%	74%	71%
100%	68%	19%	3%	48%	34%	23%	69%	66%
100%	80%	16%	2%	50%	30%	17%	76%	77%
100%	76%	14%	1%	48%	26%	17%	75%	75%

Again, two CPUs are heavily used, the load ranges between 40% and slightly over 50% with peaks of over 60%.

FAS3170b – this is where the data is currently located

CPU	NFS	CIFS	HTTP	Total	Net in	kB/s out	Disk read	kB/s write	Tape read	kB/s write	Cache age	Cache hit	CP time	CP ty	Disk util	OTHER	FCP	iSCSI	FCP in	kB/s out	iSCSI in	kB/s out
30%	18186	0	0	18199	5238	153512	1299	8	0	0	>60	100%	0%	-	10%	12	0	1	0	0	0	0
30%	18131	0	0	18158	5294	153249	1320	0	0	0	>60	100%	0%	-	6%	23	0	4	0	0	0	0
30%	14713	0	0	14719	4196	124231	5435	5798	0	0	>60	100%	28%	Tv	14%	0	0	6	0	0	0	0
25%	15037	0	0	15039	4328	126822	1056	36	0	0	>60	100%	4%	:	9%	0	0	2	0	0	0	0
29%	18384	0	0	18389	5229	155284	1135	0	0	0	>60	100%	0%	-	7%	0	0	5	0	0	0	0
27%	16807	0	0	16827	4803	141819	859	32	0	0	>60	100%	0%	-	5%	17	0	3	0	0	0	0
27%	16264	0	0	16481	4686	137125	754	0	0	0	>60	100%	0%	-	4%	213	0	4	0	0	0	0
30%	18608	0	0	18618	5417	157240	900	0	0	0	>60	100%	0%	-	3%	0	0	10	0	0	0	0
28%	17827	0	0	17833	5068	150442	960	24	0	0	>60	100%	0%	-	6%	0	0	6	0	0	0	0
24%	15027	0	0	15029	4338	126814	668	0	0	0	>60	100%	0%	-	4%	0	0	2	0	0	0	0

ANY1+	ANY2+	ANY3+	ANY4+	AVG	CPU0	CPU1	CPU2	CPU3
44%	32%	24%	15%	29%	24%	21%	35%	37%
28%	19%	13%	8%	18%	13%	12%	22%	23%
43%	29%	20%	12%	26%	22%	18%	29%	37%
45%	32%	24%	15%	29%	24%	23%	29%	41%
45%	32%	24%	15%	29%	24%	22%	30%	41%
46%	33%	25%	17%	31%	27%	24%	31%	43%
44%	32%	24%	16%	29%	25%	23%	30%	40%
38%	27%	19%	12%	24%	19%	18%	26%	33%
44%	33%	24%	15%	29%	22%	21%	35%	38%

The CPUs are more or less used equally with 25% to 40%.

What do we find in v$iostat_file?

FILE_NO	AVG_SMALL_READ_US	AVG_SMALL_SYNC_READ_US
1	492.882536	492.948468
3	495.428625	495.733918
4	498.66933	498.66933
5	495.088486	495.092377
6	495.803506	495.804537
7	6.28571429	6.28571429
8	496.395164	496.452218

As expected, I/Os take longer – with averagely almost 500 us (0.5 ms), this detour means approx. 70 us longer I/O times. This naturally has an impact on the results of I/O Calibrate.

Result of IO Calibrate

I/O Ops/sec = 20780
Actual Latency = 7
MB/sec = 344

The results collapse in the IOPS field from over 23,000 IOPS (Test#3) to below 21,000 IOPS – roughly 10%.

When we draw a comparison between NFSv3 and NFSv4 at this point (what many benchmarks unfortunately do), the result would be as follows:

Test	CPU DB Server	CPU FAS3170a	FAS3170b	IO Callibrate IOPS	v$iostat IO Zeit
#1 NFSv3, Zugriff auf DATA1 auf FAS3170a via LIF auf FAS3170a	90%	30 - 45%	0%	27000	0.355
#2 NFSv3, Zugriff auf DATA1 auf FAS3170b via LIF auf FAS3170a	90%	20 - 40%	20 - 50%	26400	0.395
#3 NFSv4, Zugriff auf DATA1 auf FAS3170a via LIF auf FAS3170a	85%	50 - 70%	einige %	23500	0.430
#4 NFSv4, Zugriff auf DATA1 auf FAS3170b via LIF auf FAS3170a	70%	40 - 50%	25 - 40%	20800	0.500

- NFSv4 is slower than NFSv3 – fewer IOPS, higher I/O times
- Access via the "wrong" storage head will in fact also lead to lower IOPS numbers and higher IO times, but the decisive factor is the drastically higher CPU usage on the storage nodes. This problem should be solved with pNFS.

Performance Test #5: pNFS, Access on DATA1 on FAS3170a via LIF on FAS3170a

As the first step, set oranfstab to pNFS.

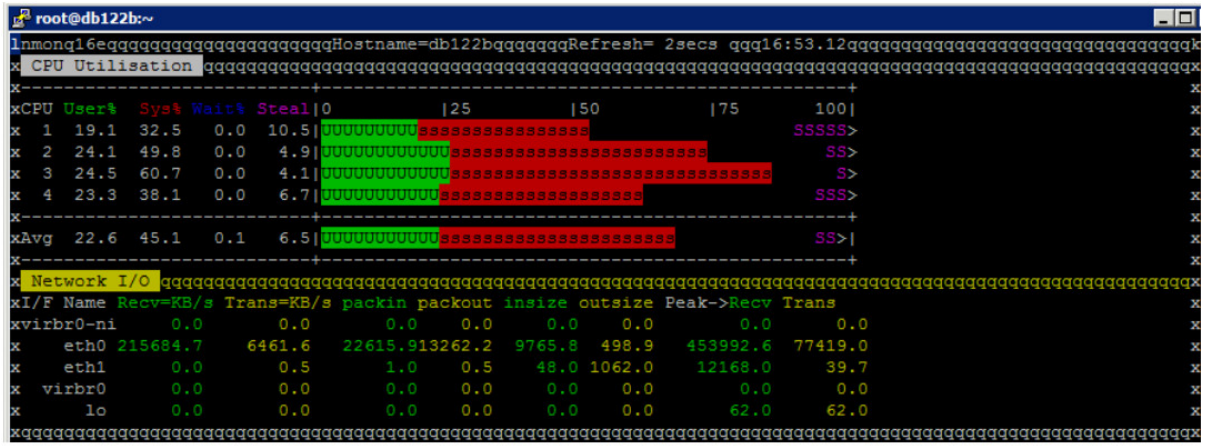
```
server: vsdb122b-lif1
local:  10.146.2.61
path:   10.146.146.61
donotroute
#nfs_version: nfsv4
nfs_version: pnfs
export: /vol/db122b_ctrl1/db122b_ctrl1_qt mount: /u01/app/oracle/oradata/DB122B/ctrl1
export: /vol/db122b_data1/db122b_data1_qt mount: /u01/app/oracle/oradata/DB122B/data1
export: /vol/db122b_log1/db122b_log1_qt   mount: /u01/app/oracle/oradata/DB122B/log1
server: vsdb122b-lif2
local:  10.146.2.61
path:   10.146.146.161
donotroute
#nfs_version: nfsv4
nfs_version: pnfs
export: /vol/db122b_ctrl2/db122b_ctrl2_qt mount: /u01/app/oracle/oradata/DB122B/ctrl2
export: /vol/db122b_data2/db122b_data2_qt mount: /u01/app/oracle/oradata/DB122B/data2
export: /vol/db122b_log2/db122b_log2_qt   mount: /u01/app/oracle/oradata/DB122B/log2
```

This is also correctly displayed in v$dnfs_servers:

```
SVRNAME          DIRNAME                                  NFSVERSION
---------------- ---------------------------------------- ----------------
vsdb122b-lif1    /vol/db122b_ctrl1/db122b_ctrl1_qt        Parallel NFS
vsdb122b-lif1    /vol/db122b_data1/db122b_data1_qt        Parallel NFS
vsdb122b-lif1    /vol/db122b_log1/db122b_log1_qt          Parallel NFS
vsdb122b-lif2    /vol/db122b_ctrl2/db122b_ctrl2_qt        Parallel NFS
vsdb122b-lif2    /vol/db122b_data2/db122b_data2_qt        Parallel NFS
vsdb122b-lif2    /vol/db122b_log2/db122b_log2_qt          Parallel NFS
```

CPU Usage on the Database Server

It is noticeable that the CPU usage of the database server is lower than expected (below 70% - slightly lower as in the previous test), but the network throughput is comparable to NFSv3 at the same time!



Storage CPU Usage

FAS3170a

The average load on the storage nodes is in the range of 60% - 80% with peaks of 90%.

CPU	NFS	CIFS	HTTP	Total	Net in	Net out	Disk read	Disk write	Tape read	Tape write	Cache age	Cache hit	CP time	CP ty	Disk util	OTHER	FCP	iSCSI	FCP in	FCP out	iSCSI in	iSCSI out
76%	28377	0	0	28422	7548	224917	2544	32	0	0	>60	99%	0%	-	9%	0	0	45	0	0	61	373
74%	25728	0	0	25756	6585	189065	1731	0	0	0	>60	100%	0%	-	9%	0	0	28	0	0	24	98
74%	27186	0	0	27227	6966	220222	1869	0	0	0	>60	100%	0%	-	6%	10	0	31	0	0	61	81
76%	28939	0	0	28984	7662	227135	2008	24	0	0	>60	100%	0%	-	7%	0	0	45	0	0	89	339
74%	28158	0	0	28181	7235	222673	1832	0	0	0	>60	100%	0%	-	9%	0	0	23	0	0	28	65
72%	27771	0	0	27816	7459	215016	1527	8	0	0	>60	100%	0%	-	14%	4	0	41	0	0	61	203
63%	25766	0	0	25804	6731	191485	1461	24	0	0	>60	100%	0%	-	8%	0	0	38	0	0	90	294
75%	28336	0	0	28367	7385	225277	1445	0	0	0	>60	100%	0%	-	7%	10	0	21	0	0	24	81
73%	29818	0	0	29959	7627	225543	1211	0	0	0	>60	100%	0%	-	6%	109	0	32	0	0	61	82
87%	26830	0	0	29836	7240	213174	4737	1967	0	0	>60	100%	11%	Tf	8%	2961	0	45	0	0	92	321
85%	26988	0	0	31456	6948	207666	4135	5525	0	0	>60	100%	20%	:	8%	4439	0	29	0	0	24	81
77%	21855	0	0	27464	5629	161926	607	0	0	0	>60	100%	0%	-	5%	5576	0	33	0	0	61	98
89%	27947	0	0	31655	7169	215881	939	24	0	0	>60	100%	0%	-	4%	3667	0	41	0	0	89	338
88%	27818	0	0	31754	7431	215789	879	0	0	0	>60	100%	0%	-	5%	3913	0	23	0	0	24	81
89%	27702	0	0	31253	7352	224506	966	0	0	0	>60	100%	0%	-	6%	3520	0	31	0	0	61	97

ANY1+	ANY2+	ANY3+	ANY4+	AVG	CPU0	CPU1	CPU2	CPU3
100%	96%	82%	58%	84%	80%	81%	80%	95%
100%	97%	84%	61%	86%	84%	82%	83%	94%
100%	90%	70%	47%	77%	70%	75%	76%	86%
96%	89%	69%	44%	75%	73%	73%	75%	79%
100%	91%	67%	44%	75%	71%	74%	74%	83%
99%	89%	70%	49%	77%	72%	74%	76%	85%
100%	88%	65%	40%	73%	71%	72%	70%	80%
97%	82%	56%	34%	68%	64%	66%	66%	75%
98%	87%	66%	41%	73%	64%	73%	74%	83%
85%	59%	35%	18%	50%	30%	53%	54%	62%
98%	87%	63%	38%	72%	56%	75%	74%	80%

The load on all CPU cores is more or less distributed equally.

FAS3170b

As expected, the load on the non-involved nodes is quite low again – there are some small occasional peaks that may be the result of metadata queries.

CPU	NFS	CIFS	HTTP	Total	Net in	Net out	Disk read	Disk write	Tape read	Tape write	Cache age	Cache hit	CP time	CP ty	Disk util	OTHER	FCP	iSCSI	FCP in	FCP out	iSCSI in	iSCSI out
2%	31	0	0	46	49	98	24	24	0	0	>60	100%	0%	-	5%	13	0	2	0	0	0	0
2%	40	0	0	50	267	233	0	0	0	0	>60	100%	0%	-	0%	0	0	10	0	0	0	0
4%	44	0	0	48	271	21	5752	6776	0	0	>60	100%	31%	T	9%	0	0	4	0	0	0	0
1%	25	0	0	32	249	14	24	24	0	0	>60	100%	0%	-	4%	0	0	7	0	0	0	0
1%	36	0	0	41	77	346	16	8	0	0	>60	100%	0%	-	2%	0	0	5	0	0	0	0
3%	17	0	0	30	140	14	0	0	0	0	>60	100%	0%	-	0%	11	0	2	0	0	0	0
4%	28	0	0	32	36	20	28	24	0	0	>60	100%	0%	-	2%	0	0	4	0	0	0	0
2%	35	0	0	35	239	279	0	0	0	0	>60	100%	0%	-	0%	0	0	0	0	0	0	0
3%	54	0	0	61	146	208	0	0	0	0	>60	100%	0%	-	0%	0	0	7	0	0	0	0
2%	15	0	0	163	29	12	32	32	0	0	>60	100%	0%	-	3%	143	0	5	0	0	0	0

ANY1+	ANY2+	ANY3+	ANY4+	AVG	CPU0	CPU1	CPU2	CPU3
22%	3%	1%	0%	7%	8%	7%	5%	6%
10%	3%	1%	0%	4%	5%	2%	1%	6%
6%	1%	0%	0%	2%	2%	1%	1%	5%
7%	2%	0%	0%	3%	3%	2%	1%	5%
8%	2%	0%	0%	3%	3%	1%	1%	6%
6%	1%	0%	0%	2%	2%	1%	1%	5%
5%	1%	0%	0%	2%	2%	1%	1%	4%
5%	1%	0%	0%	2%	2%	1%	1%	4%
7%	1%	0%	0%	3%	2%	2%	1%	5%

IO Times according to v$IOSTAT_FILE

The values are surprisingly low – below 20 us (0.02 ms) the question arises whether the values could be correct or if Oracle does not measure correctly with pNFS. The number of IOs are displayed correctly, only the I/O times are questionable!

FILE_NO	AVG_SMALL_READ_US	AVG_SMALL_SYNC_READ_US
1	18.8971259	18.898718
3	18.9023989	18.9072525
4	18.9125628	18.9125628
5	18.9641167	18.9654489
6	19.0164365	19.0173799
7	4	4
8	19.0554705	19.0565011

This means that the result of I/O Calibrate will be important. Will we also find these very positive data there?

Result of IO Calibrate

I/O Ops/sec = 25547
Actual Latency = 1
MB/sec = 346

And the answer is: Yes, absolutely! The values are a bit worse as with NFSv3 but still significantly better as with NFSv4.

Performance Test #6: pNFS, Access on DATA1 on FAS3170b via ???

And now for the most vital point: Does pNFS deliver as promised? The data on the storage were moved to FAS3170b with the database running. When pNFS lives up to its name, access via FAS3170b should be transparent (after a short period of time at least).

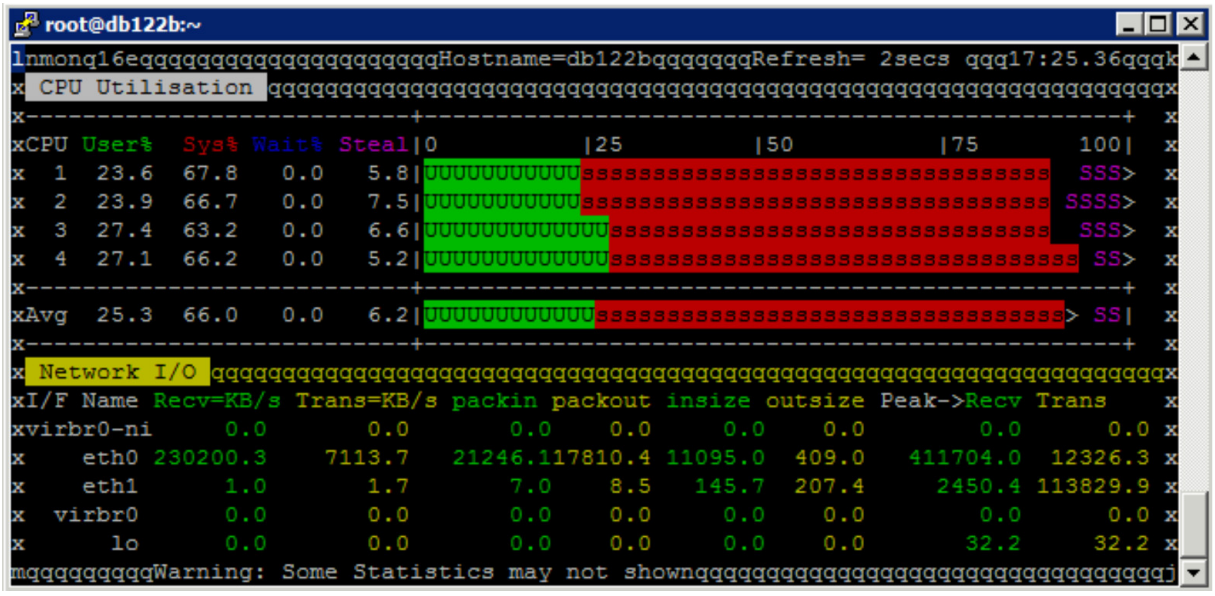
The result is disillusioning. It does not work. Both storage nodes are still under massive load. May the oranfstab be the problem?

Test #1) comment out donotroute
Test #2) define/permit multiple paths

Each test shows that pNFS always accesses via mounted path. There is either a special configuration for the oranfstab we have not found – the documentation does not have any information on that – or it simply does not work yet. We have opened two service requests at Oracle in that matter.

CPU Usage on the Database Server

CPU usage on the client is significantly higher again – compared to the tests with NFSv3.



Usage on the Storage

FAS3170a

Storage usage is similar to the other tests with NFSv4 – maybe slightly higher.

CPU	NFS	CIFS	HTTP	Total	Net in	Net out	Disk read	Disk write	Tape read	Tape write	Cache age	Cache hit	CP time	CP ty	Disk util	OTHER	FCP	iSCSI	FCP in	FCP out	iSCSI in	iSCSI out
63%	4044	0	0	4106	239564	236391	0	0	0	0	>60	100%	0%	-	0%	9	0	53	0	0	126	326
57%	2799	0	0	2822	222768	219828	0	0	0	0	>60	100%	0%	-	0%	0	0	23	0	0	24	98
64%	3018	0	0	3038	250500	247050	24	24	0	0	>60	100%	0%	-	2%	0	0	20	0	0	25	66
64%	3500	0	0	3547	248523	245725	8	0	0	0	>60	100%	0%	-	1%	0	0	47	0	0	127	278
63%	4771	0	0	4945	211963	208912	7880	7908	0	0	>60	100%	32%	T	8%	148	0	26	0	0	25	98
46%	5361	0	0	5453	141838	140327	24	24	0	0	>60	100%	0%	-	1%	70	0	22	0	0	24	57
67%	3886	0	0	4482	227211	224364	16	0	0	0	>60	100%	0%	-	0%	546	0	50	0	0	127	344
60%	3068	0	0	3092	229510	226724	0	0	0	0	>60	100%	0%	-	0%	3	0	21	0	0	25	82

ANY1+	ANY2+	ANY3+	ANY4+	AVG	CPU0	CPU1	CPU2	CPU3
100%	93%	65%	24%	71%	55%	72%	72%	84%
100%	96%	73%	31%	75%	59%	79%	76%	87%
100%	96%	68%	30%	74%	58%	74%	76%	87%
100%	97%	75%	37%	77%	63%	79%	80%	88%
100%	96%	78%	40%	79%	67%	78%	78%	92%
100%	95%	73%	32%	75%	60%	76%	80%	85%
100%	98%	82%	42%	81%	74%	79%	78%	92%

FAS3170b

The load is also in the range of the NFSv4 test when the wrong storage node is used for access.

CPU	NFS	CIFS	HTTP	Total	Net in	Net out	Disk read	Disk write	Tape read	Tape write	Cache age	Cache hit	CP time	CP ty	Disk util	OTHER	FCP	iSCSI	FCP in	FCP out	iSCSI in	iSCSI out
45%	27257	0	0	27262	7699	230497	8	24	0	0	6	100%	0%	-	2%	0	0	5	0	0	0	0
50%	28951	0	0	29070	8602	243486	36	0	0	0	6	100%	0%	-	1%	106	0	13	0	0	0	0
47%	26029	0	0	26240	7424	219893	7812	7900	0	0	6	100%	29%	T	8%	204	0	7	0	0	0	0
41%	24232	0	0	24351	6793	204677	12	24	0	0	6	100%	0%	-	2%	111	0	8	0	0	0	0
45%	26241	0	0	26344	7411	221736	24	8	0	0	6	100%	0%	-	1%	102	0	1	0	0	0	0
47%	27313	0	0	27520	7771	231047	8	8	0	0	6	100%	0%	-	1%	205	0	2	0	0	0	0
42%	25764	0	0	25764	7226	217594	12	24	0	0	6	100%	0%	-	3%	0	0	0	0	0	0	0
49%	28573	0	0	28580	8143	241602	16	0	0	0	6	100%	0%	-	1%	0	0	7	0	0	0	0

ANY1+	ANY2+	ANY3+	ANY4+	AVG	CPU0	CPU1	CPU2	CPU3
54%	41%	27%	15%	35%	28%	23%	42%	46%
63%	49%	33%	20%	42%	33%	29%	50%	55%
62%	48%	33%	20%	41%	31%	29%	49%	54%
57%	44%	30%	19%	38%	30%	27%	45%	49%
64%	48%	33%	21%	42%	33%	29%	49%	56%
63%	47%	30%	16%	40%	30%	26%	49%	53%
67%	50%	34%	22%	44%	35%	33%	52%	54%
66%	46%	30%	19%	40%	30%	28%	53%	50%
59%	46%	31%	19%	39%	30%	27%	47%	52%

Results of V$IOSTAT_FILE

As expected, V$IOSTAT_FILE also does not display correct values.

FILE_NO	AVG_SMALL_READ_US	AVG_SMALL_SYNC_READ_US
1	22.9905378	22.9887182
3	23.2977075	23.2995525
4	23.1889902	23.1880715
5	23.7003103	23.7031924
6	23.1372255	23.1385067
7	2	2
8	24.034558	24.0348808

Result of IO Calibrate

I/O Ops/sec = 24321
Actual Latency = 3
MB/sec = 237

The values of IO Calibrate are almost 10% below NFSv3.

Summary

Currently, the usage of ideal access paths with Oracle pNFS does not work yet and also the internal Oracle performance views display wrong data in the context of pNFS. It is interesting that the performance with pNFS is only some % lower compared to NFSv3, but the CPU usage on the database server is significantly lower.

Test	NFS	via IP	Daten	CPU DB Server	CPU FAS3170 A	FAS3170 B	IO Callibrate IOPS	v$iostat IO Zeit
#1	NFSv3	Node A	Node A	90%	30 - 45%	0%	27000	0.355 ms
#2	NFSv3	Node A	Node B	90%	20 - 40%	20 - 50%	26400	0.395 ms
#3	NFSv4	Node A	Node A	85%	50 - 70%	einige %	23500	0.430 ms
#4	NFSv4	Node A	Node B	70%	40 - 50%	25 - 40%	20800	0.500 ms
#5	pNFS	Node A	Node A	65%	50 - 85%	einige %	25500	0.020 ms ?
#6	pNFS	Node A	Node B	90%	45 - 80%	40 - 70%	24000	0.024 ms ?

On the storage side, pNFS is obviously significantly more complicated and not as sophisticated as NFSv3 – however, this will certainly be improved in the coming years. The reason may currently also be that Oracle ignores the "redirect" to the other storage node.

Should you already switch to pNFS?
A clear "yes" for test/dev systems and a clear "no" for all productive databases. As we have learned in past decades, a new Oracle feature should be used productively with the next release at the earliest (to make sure that production is not at stake due to certainly still existent bugs). However, you should start gathering experiences.

The found problems have been reported to Oracle – we hope for a quick resolution.