1 introduction

-

2 Motivation

In science computer simulation has become an important tool. Simulation are becoming more and more advanced which increase the amount of data that are being generated. This data gets stored on harddrive and loaded again when its time to analyze the data. By doing in-situ real-time analysis where the data gets analyzed immediately after being generated. By doing computer simulation this way it may be possible to save time and hardware resources.

3 Hardware

The program have been tested on a server with the following hardware.
Motherboard: Supermicro X11DPU-Z+
CPU: Intel(R) Xeon(R) Gold 6130 CPU @ 2.10GHz, 32 core
DRAM: Samsung RDIMM, 2666 MT/s.
NVDIMM: Micron Technology NV-DIMM , 2933 MT/s

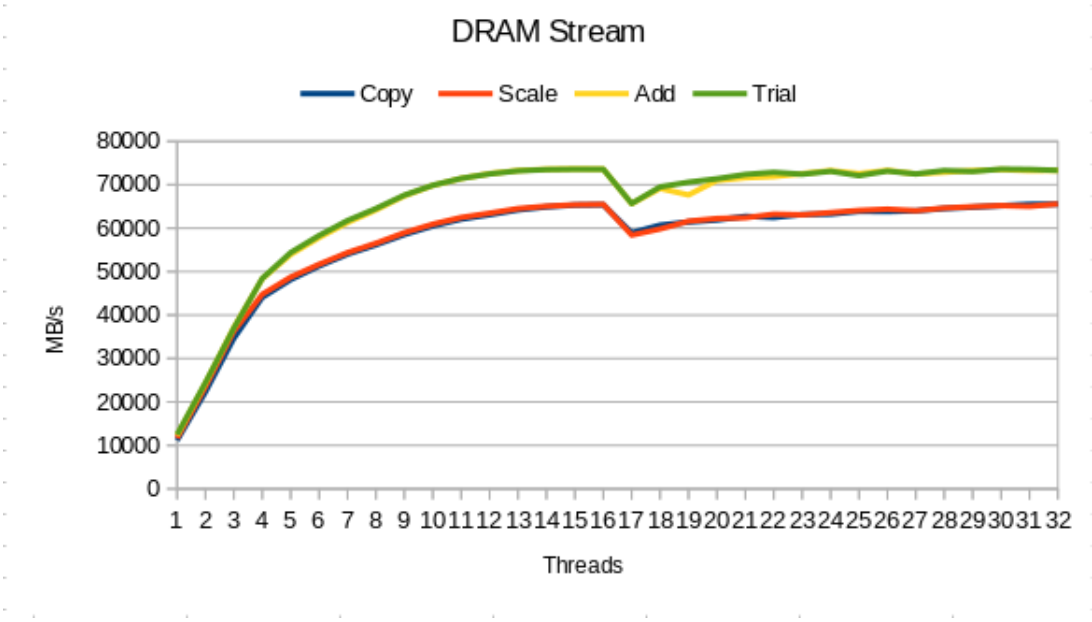
Both CPU have twelve memory slots each. Each CPU have six channels. There are one DRAM and one NVDIMM sharing one channel.

4 Benchmarks

There are four different benchmarks.

4.1 STREAM DRAM

The stream DRAM benchmark measure the memory speed of the DRAM. The benchmark uses the STREAM[1] benchmark without any changes in order to measure how fast the memory speed is. The benchmark tests all with all cores.



4.2 STREAM NVDIMM

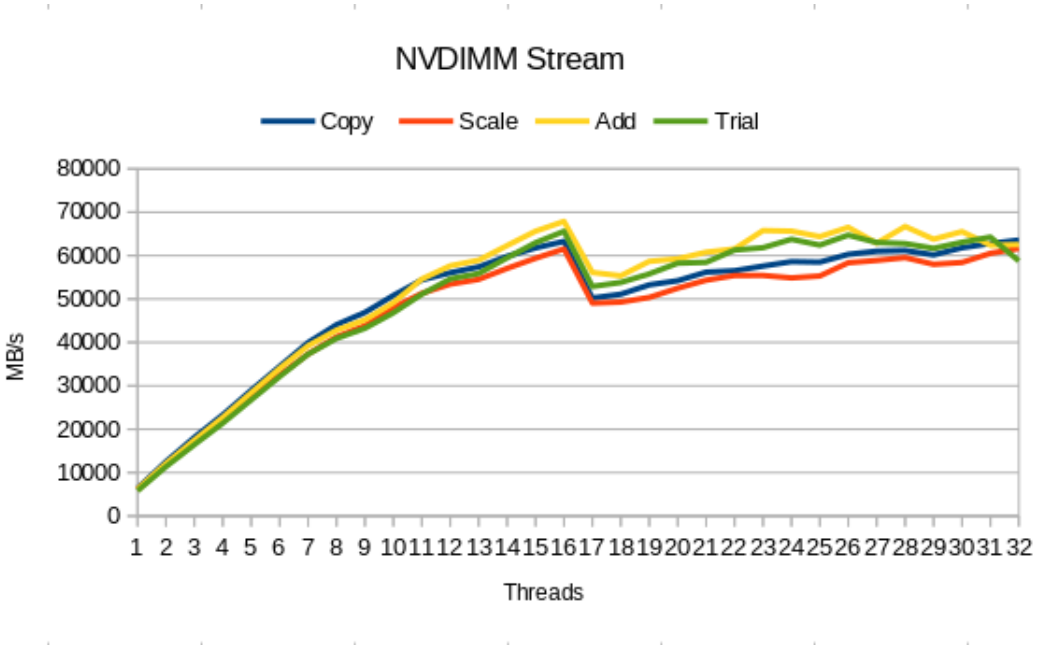
The stream NVDIMM benchmark measure the memory speed of the NVDIMM. This benchmark is the same as the STREAM DRAM benchmark mention above. The different is that the memory type have been changed from DRAM to NVDIMM. The original code looks like this.

```
#ifndef STREAM_TYPE
#define STREAM_TYPE double
#endif

static STREAM_TYPE  a[STREAM_ARRAY_SIZE+OFFSET],
                    b[STREAM_ARRAY_SIZE+OFFSET],
                    c[STREAM_ARRAY_SIZE+OFFSET];
```

It have been changed into this. In addition the PMEMobjpool must be initiated in main method.

```
PMEMobjpool *pop;
POBJ_LAYOUT_BEGIN(array);
POBJ_LAYOUT_TOID(array, double);
POBJ_LAYOUT_END(array);
TOID(double) a;
TOID(double) b;
TOID(double) c;
```

```
int main()
{
        const char path[] = "/mnt/pmem0-xfs/pool.obj";
        pop = pmemobj_create(path, LAYOUT_NAME, 10737418240, 0666);
        if (pop == NULL)
                pop = pmemobj_open(path, LAYOUT_NAME);

        if (pop == NULL) {
                perror(path);
                exit(1);
        }
}
```



4.3 benchmark 3

Graph below show the speed of a certain amount of NVDIMM threads while the rest of the threads are from DRAM to DRAM. The test have been conducted by transfer data simultaneously from DRAM-DRAM and NVM-NVM. All the threads are transferring the values of one array to another all the arrays have 100 million elements of type double. This transfer happens 1000 times and the graphs shows the average of the first 200 iterations. This is done to ensure that all the threads can't

finish early and make the remaining threads faster. The sum graphs shows the sum bandwidth of DRAM and NVM. Average graphs shows the average bandwidth of DRAM and NVM.

4.3.1 NVM-NVM

	NVM-NVM			
	DRAM			
Nvm-threads	1-200	201-400	401-600	601-800
1	61467.61	61386.09	61353.65	61336.69
2	58059.67	57953.79	57919.76	57919.85
3	54731.97	54315.45	54283.92	54283.78
4	51094.41	50585.73	50572.31	50575.95
5	47678.50	46871.58	46689.09	46685.24
6	44385.56	43187.26	43080.64	43078.06
7	40780.79	39226.49	39207.16	39218.66
8	37627.08	35387.07	35432.56	35417.49
9	34544.89	31523.73	31483.49	31488.33
10	31638.23	27458.24	27495.25	27492.96
11	28981.81	23557.12	23560.95	23564.00
12	25791.67	19559.22	19561.31	19593.83
13	20274.64	15354.39	15399.20	15408.11
14	13418.90	10169.24	10227.84	10270.53
15	6860.74	5335.82	5374.29	5384.83
	NVM			
Nvm-threads	1-200	201-400	401-600	601-800
1	3904.57	3880.07	3876.70	3875.89
2	7903.52	7874.97	7867.25	7859.85
3	11870.33	11837.86	11824.86	11830.10
4	16119.73	16071.52	16043.62	16037.98
5	19944.51	19857.52	19741.39	19756.41
6	23857.97	23754.71	23702.75	23680.78
7	27639.25	27440.73	27393.70	27407.13
8	31338.32	31201.13	31198.88	31189.02
9	34936.20	34731.46	34699.52	34677.29
10	38996.54	38777.84	38753.75	38753.99
11	42736.22	42539.06	42517.19	42468.53
12	46259.98	46000.67	45959.40	45925.94
13	50023.03	49826.89	49820.33	49831.49
14	54649.58	54261.89	54225.10	54224.66
15	57777.16	57428.98	57408.42	57410.92

Table 1: NVM-NVM part 1

	SUM			
	1-200	201-400	401-600	601-800
1	65372.18	65266.16	65230.36	65212.58
2	65963.19	65828.76	65787.01	65779.70
3	66602.30	66153.31	66108.78	66113.89
4	67214.14	66657.25	66615.93	66613.93
5	67623.01	66729.10	66430.48	66441.66
6	68243.54	66941.97	66783.39	66758.84
7	68420.04	66667.21	66600.86	66625.79
8	68965.40	66588.20	66631.44	66606.50
9	69481.09	66255.19	66183.01	66165.62
10	70634.77	66236.08	66249.00	66246.94
11	71718.03	66096.19	66078.15	66032.52
12	72051.66	65559.88	65520.72	65519.78
13	70297.67	65181.28	65219.53	65239.60
14	68068.48	64431.12	64452.94	64495.18
15	64637.90	62764.80	62782.72	62795.75

Table 2: NVM-NVM part 2

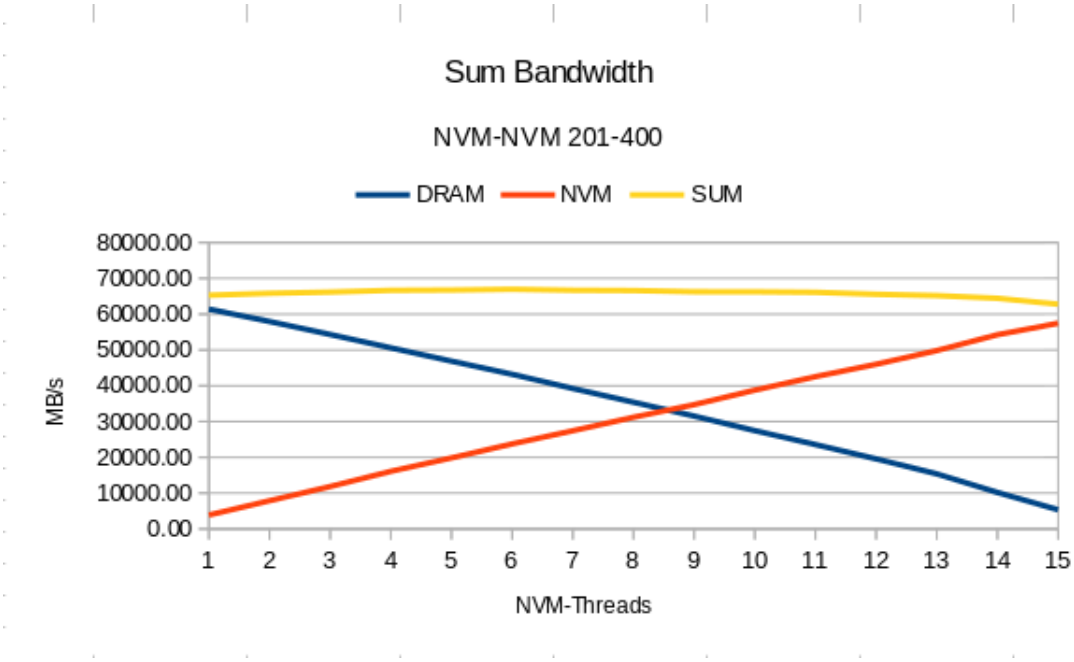


Figure 1: NVM-NVM graph

4.3.2 NVM-DRAM

	DRAM			
Nvm-threads	1-200	201-400	401-600	601-800
1	61065.68	60957.48	60913.51	60905.70
2	57290.21	57177.29	57128.94	57137.41
3	53444.16	53352.44	53312.60	53335.71
4	49534.28	49472.78	49453.22	49455.73
5	45564.56	45535.88	45505.91	45527.91
6	41582.52	41566.52	41554.04	41559.25
7	37553.81	37558.66	37563.31	37538.93
8	33507.32	33513.91	33502.82	33519.87
9	29447.94	29423.44	29454.19	29425.80
10	25319.44	25368.29	25316.75	25369.70
11	21191.87	21203.88	21137.97	21195.05
12	17046.99	16959.32	16906.76	16934.35
13	12794.79	12752.33	12760.52	12733.85
14	8540.70	8541.61	8554.70	8558.83
15	4302.48	4285.94	4269.80	4286.46
	NVM			
Nvm-threads	1-200	201-400	401-600	601-800
1	3860.86	3852.31	3850.79	3849.30
2	7808.10	7771.26	7768.53	7764.15
3	11784.02	11743.65	11747.15	11738.22
4	15864.13	15800.14	15775.34	15768.80
5	19966.21	19894.20	19879.03	19858.64
6	24116.88	24023.13	23969.54	24005.74
7	28224.59	28190.67	28122.59	28166.02
8	32432.20	32373.11	32312.86	32347.60
9	36621.83	36565.32	36542.64	36532.73
10	40840.12	40768.47	40744.65	40690.96
11	45108.04	44959.15	44812.09	44788.36
12	49339.47	49044.81	48915.19	48872.27
13	53678.86	53318.96	53167.97	53179.78
14	57999.37	57795.84	57718.35	57736.74
15	62353.42	61965.64	61975.85	61976.98

Table 3: NVM-DRAM part 1

	SUM			
	1-200	201-400	401-600	601-800
1	64926.54	64809.79	64764.30	64755.00
2	65098.31	64948.55	64897.47	64901.56
3	65228.18	65096.10	65059.74	65073.93
4	65398.41	65272.92	65228.56	65224.53
5	65530.77	65430.08	65384.94	65386.55
6	65699.40	65589.65	65523.58	65564.99
7	65778.40	65749.33	65685.90	65704.95
8	65939.52	65887.02	65815.69	65867.47
9	66069.77	65988.76	65996.83	65958.53
10	66159.56	66136.77	66061.41	66060.66
11	66299.90	66163.03	65950.07	65983.41
12	66386.46	66004.14	65821.95	65806.63
13	66473.65	66071.28	65928.49	65913.63
14	66540.07	66337.44	66273.05	66295.57
15	66655.90	66251.58	66245.64	66263.45

Table 4: NVM-DRAM part 2

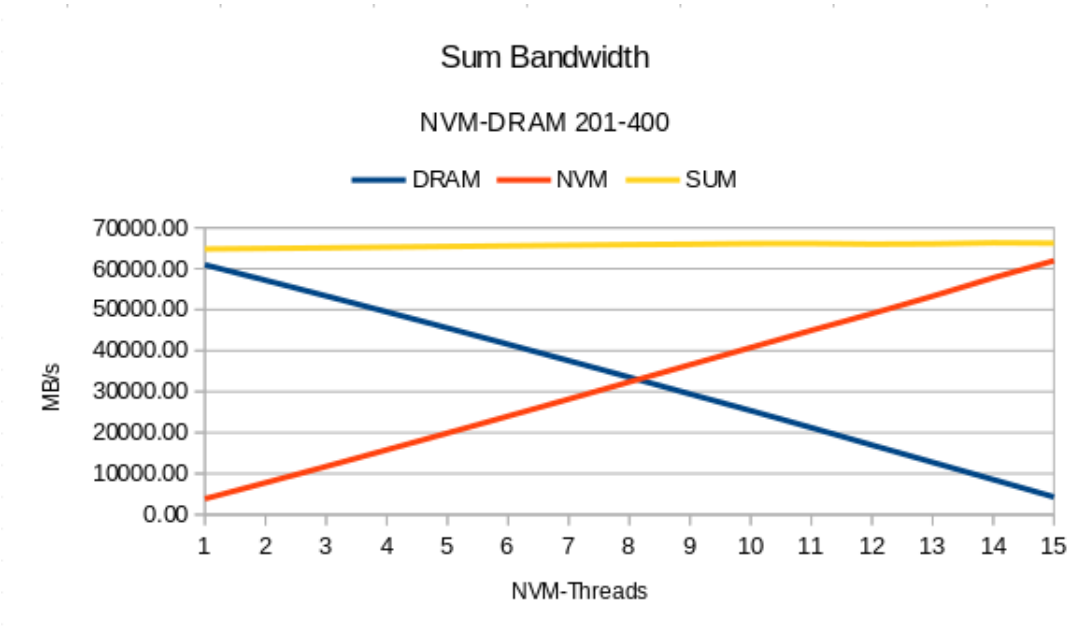


Figure 2: NVM-DRAM graph

4.3.3 DRAM-NVM

	DRAM			
Nvm-threads	1-200	201-400	401-600	601-800
1	61359.90	61224.11	61203.46	61178.06
2	57812.15	57669.46	57620.52	57616.77
3	54344.83	53999.49	53954.36	53968.20
4	50782.88	50237.42	50192.91	50189.49
5	47177.30	46351.21	46317.17	46350.45
6	43547.96	42421.95	42405.62	42400.92
7	39981.85	38479.77	38473.78	38445.10
8	36297.64	34422.79	34444.90	34412.94
9	33119.58	30307.57	30280.00	30260.69
10	30069.63	26093.92	26067.24	25995.86
11	27486.07	21798.82	21660.87	21662.13
12	24510.41	17494.34	17367.40	17366.20
13	19358.87	13186.08	13078.84	13145.12
14	13065.23	8813.79	8754.30	8793.96
15	6533.61	4359.86	4361.37	4331.11
	NVM			
Nvm-threads	1-200	201-400	401-600	601-800
1	3924.46	3912.83	3915.49	3911.43
2	7897.18	7900.96	7907.78	7893.54
3	11983.43	11965.44	11945.89	11953.59
4	16103.92	16104.71	16069.86	16100.76
5	20282.54	20291.57	20261.28	20291.88
6	24536.38	24546.42	24507.96	24528.71
7	28702.05	28757.57	28708.25	28747.34
8	32879.26	32949.28	32918.76	32925.15
9	37093.06	37145.98	37132.33	37074.03
10	41249.33	41346.61	41276.71	41215.13
11	45456.17	45345.26	45206.63	45188.12
12	49719.40	49568.72	49418.56	49423.70
13	53953.71	53929.63	53897.60	53911.93
14	58226.00	58138.83	58111.16	58128.00
15	62583.78	62115.70	62155.94	62130.10

Table 5: DRAM-NVM part 1

	SUM			
	1-200	201-400	401-600	601-800
1	65284.36	65136.94	65118.95	65089.49
2	65709.33	65570.42	65528.29	65510.31
3	66328.26	65964.93	65900.25	65921.79
4	66886.80	66342.13	66262.77	66290.25
5	67459.84	66642.78	66578.44	66642.33
6	68084.34	66968.36	66913.57	66929.63
7	68683.90	67237.33	67182.03	67192.44
8	69176.90	67372.07	67363.66	67338.09
9	70212.64	67453.55	67412.34	67334.71
10	71318.97	67440.53	67343.95	67210.99
11	72942.23	67144.08	66867.50	66850.25
12	74229.81	67063.07	66785.96	66789.90
13	73312.58	67115.70	66976.45	67057.05
14	71291.23	66952.62	66865.46	66921.95
15	69117.39	66475.55	66517.31	66461.21

Table 6: DRAM-NVM part 2

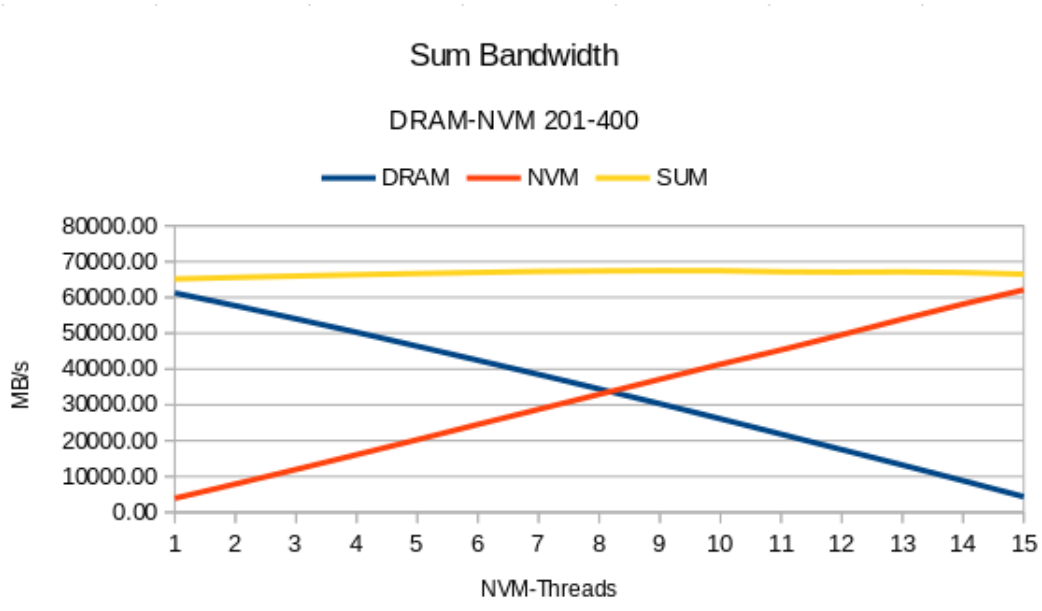


Figure 3: DRAM-NVM graph

4.4 benchmark 4

stuff

References

[1] John D. McCalpin. *STREAM source code*. URL: `https://www.cs.virginia.edu/stream/FTP/Code/stream.c` (visited on 12/20/2020).