1 Conclusion

1.1 Introduction

The conclusion will start with a summary of what has happened in the previous sections. Then it will move on to answer the research questions that was introduced in the beginning of the thesis. After the research questions have been answered there will be a section with reflections of what could have been done better or different. The thesis will end with a section that consider what further work could be.

1.2 Summary

This thesis started in section 1 with explaining what persistent memory is. What are the advantages and disadvantages of persistent memory such as cost and persistent memory leak. Four research questions where introduced in the end of the section that will be answered later in this section.

The thesis continued with explaining the basic when programming with NVDIMM in section 2. The section explained the difference between the biggest libraries and what library was chosen to be part of the different benchmarks and tests in this thesis. There was also a short description of all the methods used in the benchmarks and tests. The section ended with a short example of how to create a serial code that finds the average of all the elements in an array where the array is stored on NVDIMM.

Section 3 are about the benchmarks that have created for this thesis and a Stream benchmark that has been created by someone else. The Stream benchmark tested the speed of DRAM when only DRAM was working. There was also created a modified version of the Stream benchmark that measured the speed of NVDIMM when only NVDIMM was working. The section moved on to test DRAM and NVDIMM when they were competing for resources. There are three benchmarks. While a group of threads transferred from DRAM-DRAM another group of thread are transferring data from either NVDIMM-NVDIMM, DRAM-NVDIMM or NVDIMM-DRAM. All possible combinations of threads were tested and there was only one thread per core.

The next section are about DRAM and NVIDMM working together. How much data should be allocated to NVDIMM and how many threads should be working on the data. There are two version of this program and the difference between versions are about how NVDIMM and

DRAM access data that belong in the other group.

The last section are about NVDIMM and DRAM doing two different types of jobs. DRAM are working on generating data while the NVDIMM are transferring the last set of generated data to from DRAM to NVDIMM and analyzing the data afterward.

1.3 Research questions

The research questions will be answered in this section. The first paragraph ineach sub-section will contain the research question and it will followed up with the answer.

1.3.1 Question 1

What is the data transfer speed of NVDIMM compared to DRAM?

The expectation before starting this thesis was that NVDIMM would be slower than DRAM, but faster than a normal hard drive. What interesting is how fast the speed of the NVDIMM is compared to DRAM. This question is possible to answer by comparing the results Stream benchmark int section **??** and the result from modified Stream benchmark in section **??**.

Table 1 show the copy speed of the two Stream benchmarks mentioned above. Table 1 end at sixteen threads while the Stream benchmarks test up to 32 threads. This is because the NVDIMM fluctuate too much to find a pattern. When the Stream is testing speed of only one thread the NVDIMM have a performance of 43% of the speed of DRAM. When the number of threads increases the difference remain at around 40% up until six threads. The exception to this is the test with two threads where the NVDIMM performance is 52% of the DRAM performance. The performance of NVDIMM at seven threads is at 51% and when the number of threads increases the performance increases and end at 75% when there is sixteen threads.

The reason the performance increases as more threads are added is because DRAM at 65,000 MB/s have reached the maximum capacity of the memory bandwidth. This allows the NVDIMM threads to catch up with the DRAM performance, but NVDIMM is only able to reach a performance 75% of DRAM.

Threads	DRAM	NVDIMM	Difference in %
1	11673.5	5036.7	43.1
2	22995.1	11970.2	52.1
3	33554.9	12715.1	37.9
4	42917.3	16349.6	38.1
5	50260.9	20935.0	41.7
6	53612.5	23119.7	43.1
7	56100.8	28694.5	51.1
8	58554.6	32104.6	54.8
9	60491.7	37491.8	62.0
10	62242.2	41394.8	66.5
11	64257.1	44856.8	69.8
12	64890.3	44695.1	68.9
13	65648.8	47377.9	72.2
14	65606.5	48853.3	74.5
15	65665.5	51273.9	78.1
16	65509.8	48704.8	74.3

Table 1: Difference in STREAM performance between DRAM and NVDIMM.

1.3.2 Question 2

In an competitive environment, in what way will NVDIMM and DRAM affect each other?

1.3.3 Question 3

When the size of the data is higher than the capacity of the DRAM. How much data should be transferred to NVDIMM? How many threads should be allocated to work on the data on NVDIMM?

1.3.4 Question 4

While DRAM is working on a task, is it possible for NVDIMM to be working on a different type of task?

1.4 Reflections

1.5 Further work