documentation of Submission

Report Submission 29.10.2023

Justus Dreßler: all members contributed equally

Julius Halank: all members contributed equally

Thorsten Kröhl: all members contributed equally

4.1 memory benchmarking tools

STREAM Benchmark:
A synthetic benchmark program that measures the sustainable memory bandwidth and the corresponding CPU Speed, to check for bottlenecks. Counts how many bytes the user asked to be read plus how many bytes the user asked to be written. It is primarily designed to work with much larger datasets than the available cache-size of the system, therefor results are more indicative of the performance of very large, vector style applications. It performs four simple vector operations (Copy, Scale, Add, and Triad) on large arrays of data to measure memory transfer rates. The best perfomamce of typically 10 repetitions is chosen.
https://www.cs.virginia.edu/stream/ref.html

AIDA64:
A system diagnostics and benchmarking program, that evaluate memory bandwidth and latency through Read, Write, and Copy operations. It offers command line parameters that can be combined and automated for network testing, automated reporting, remote connection maintenance and security features. This program is optimised for most of the popular AMD, Intel and VIA Processors. The benchmarks are synthetic, so the results show only the theoretical achievable performance.
https://www.aida64.com/products/features/benchmarking

SiSoftware Sandra:
A program for system diagnosis as well as memory benchmarking. Utilizes various synthetic benchmark tests e.g. arithmetic operations and resulting throughput for integer and floating-point types, as well as for operations specific to multimedia memory bandwidth. Doesn't use a fixed amount of RAM, but about 40-60% of system RAM, therefor results may be slower than e.g. with the STREAM benchmark. Usage of "[...]aggressive use of sheduling/overlapping of instructions in order to maximise memory throughput even on “slower” processors."
https://www.sisoftware.co.uk/q-a-memory-benchmark/

Most commonly used memory benchmarks:

Sequential Memory Operations:

- assessment of memory bandwidth through sequential operations on large data arrays

Integer and Floating-Point Arithmetic:

- assessment for a combination of integer and floating-point operations to stress different aspects of memory subsystems

Multi-Threaded Tests:

- assessment of memory bandwidth under multi-threaded conditions, simulating real-world usage

Cache and Memory Latency Tests:

- assessment of latency of cache and latency of main memory, to find out data access speed

Random Access Patterns:

- assessment for the speed of random access, offering a different perspective on memory subsystem performance

Buffered and Unbuffered Memory Tests:

- assessment of different memory types, such as buffered (registered) and unbuffered (unregistered), to provide a more nuanced analysis

4.1 Cache Levels

After an initial startup curve with very short arrays the graph seems to fit the expected cache speeds pretty well with L1d cache beeing fastest and L3 Cache being slowest and expectedly a sudden drop when you have to fall back to RAM instead of caches.

4.1 Compiler Flags

Compiler: g++ that is default installed on the machines (no need to change anything) Flags:

- `-std=c++1y` ⇒ using `c++14` to get access to `std::chrono` to calculate speed
- `-o dataAccessSpeed.out` ⇒ output file (so we can reference it in the python code)
- `-O3` ⇒ optimization level 3 (highest) so we hopefully mainly test the array sizes and not how well we optimized the code manually

4.1 Memory Bandwidth Ara Cluster

Our node peaked at around 55 GB/s and the changes in the plot are mainly just the different cache levels (with the smaller caches being significantly faster than the bigger ones) and the sudden drop when we have to fall back to RAM. The only change we couldn't figure out was the spike at 240 KB.