Advanced Parallel Computing

06.05.2019

Students:
Jona Neef
Nikolas Krätzschmar
Philipp Walz

Exercise 2

2.1 Reading

Christoph Lameter. 2013. An overview of non-uniform memory access.

In the paper "An Overview of Non-Uniform Memory Access" Christoph Lameter describes the Non-Uniform Memory Accesses, which are available in almost every system today, because each processor has its own memory very close to the execution unit due to its performance.

Numa Support optimizes process execution in most cases without the user having to intervene. However, there are also additional Numa Configuration Tools, which are mainly used in high performance applications, where very good knowledge of the hardware and software is required.

I accept the contents of the paper, because the NUMA problem is described very well at the beginning and also a deeper insight is given, how the whole thing is used in Linux.

Fabien Gaud. 2015. Challenges of memory management on modern NUMA systems.

In their paper "Challenges of memory management on modern NUMA systems", the team around Fabien Gaud evaluates characteristics and features of non-uniform memory access systems.

First, the authors give a brief introduction to the topic with an example of a modern NUMA system consisting of four nodes and various interconnect links. It is explained that current x86 NUMA systems are cache coherent which supports compatibility but also aggravates performance. Although remote accesses on modern NUMA systems take only 30% longer than local accesses, this latency can increase extremely if congestions appear on the memory controller or on the interconnect.

Furthermore, they give a detailed explanation about conducted experiments that compare performance differences between single- and multi-threaded applications on NUMA systems.
Because single-threaded applications did not produce memory congestion, the difference between local- and remote accesses stayed in a range of 20%.

On the other hand, with multi-threaded applications, the two NUMA policies first-touch and interleave have a great effect on performance. The paper shows that the first-touch policy (which is used in Linux by default) improves locality, but it can also increase the imbalance of memory allocations among different nodes. This can cause memory congestions, which further reduces overall performance.

Finally, the authors propose different solution approaches such as manual NUMA policy optimizations, AutoNUMA and Carrefour, a memory-placement algorithm with focus on traffic management.

The paper concludes with the prediction that the growing amount of cores per NUMA system cause that these performance effects will continue to be a concern in the future.

We strongly accept the authors work and share their opinion that NUMA effects play a major role in performance evaluations and optimizations. Furthermore we think that not only OS developers but also application software developers should consider these architectural impacts in their code.

2.2 Pointer Chasing Benchmark

1) Latency Analysis

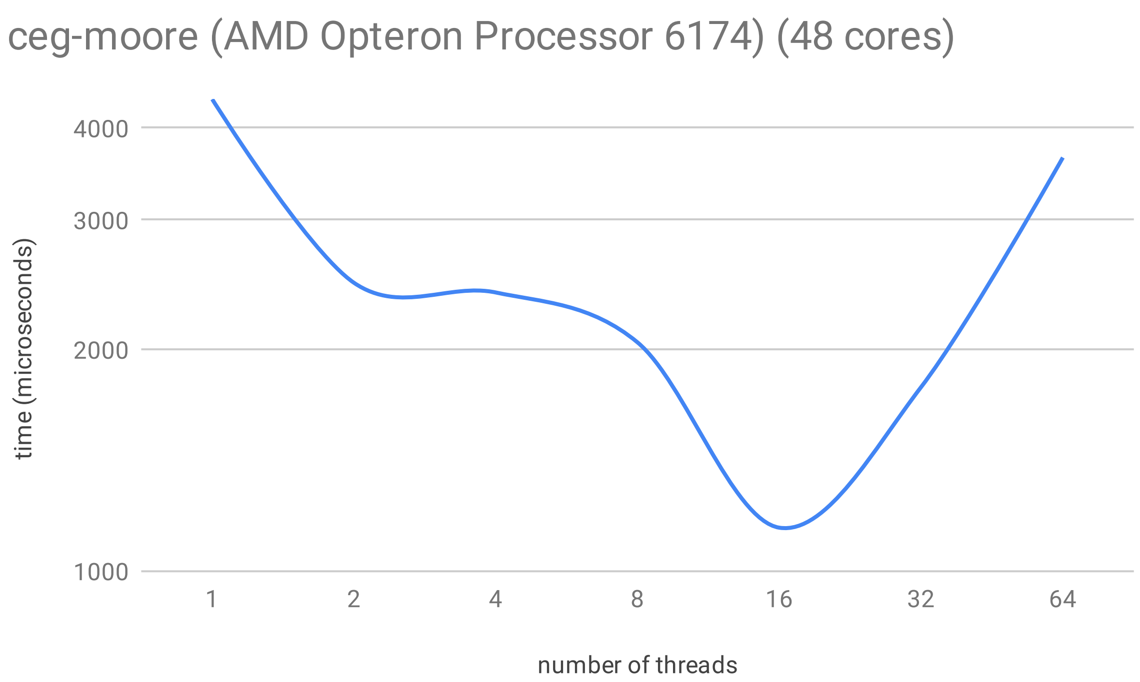
AMD Opteron 6174 Specifications

Description	Value	
Core Count	12	-
Freequency	2.2 GHz	-
Total Cache	19.6 MB	-
L1 Cache	64 KB (Data) + 64KB (Instruction) (per core)	1.2 ns
L2 Cache	512 KB (per core)	3.0 ns
L3 Cache	12 MB (per socket)	15.0 ns
Main Memory		56 ns
TLB miss penalty	-	80 cycles = 36 ns
TLB/Cache associativity	direct mapped	

2.3 Multi-threaded load bandwidth

First Machine: AMD Opteron Processor 6174

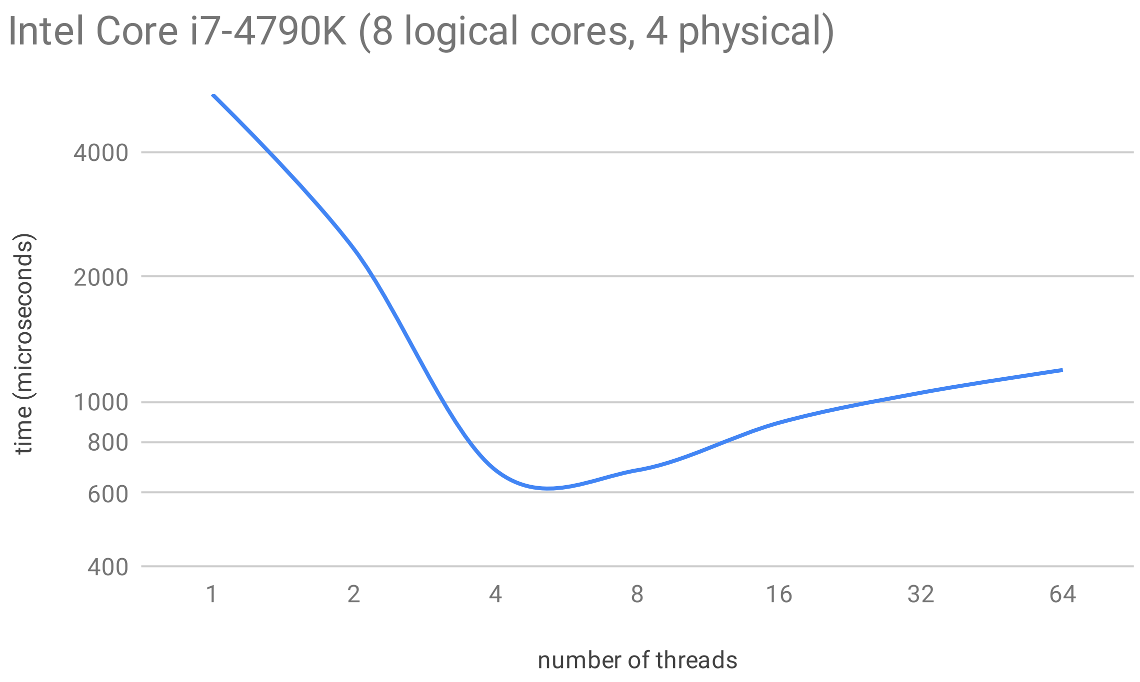
Memory controller	The number of controllers: 1 Memory channels: 4 Supported memory: DDR2-1066, DDR3-1333, LV DDR3-1066 DIMMs per channel: 3 Maximum memory bandwidth (GB/s): 42.7



Second Machine: Intel Core i7-4790K

Memory Specifications

Max Memory Size (dependent on memory type) ?	32 GB
Memory Types ?	DDR3-1333/1600, DDR3L-1333/1600 @ 1.5V
Max # of Memory Channels ?	2
Max Memory Bandwidth ?	25.6 GB/s
ECC Memory Supported ‡ ?	No


Intel Core i7-4790K (8 logical cores, 4 physical)
time (microseconds)
4000
2000
1000
800
600
400
1
2
4
8
16
32
64
number of threads