Performance measurement of the DAQ and its optimization

Martin Zemko

COMPASS Front-end, Trigger and DAQ Workshop

3rd March 2020

CTU

CZECH TECHNICAL UNIVERSITY IN PRAGUE

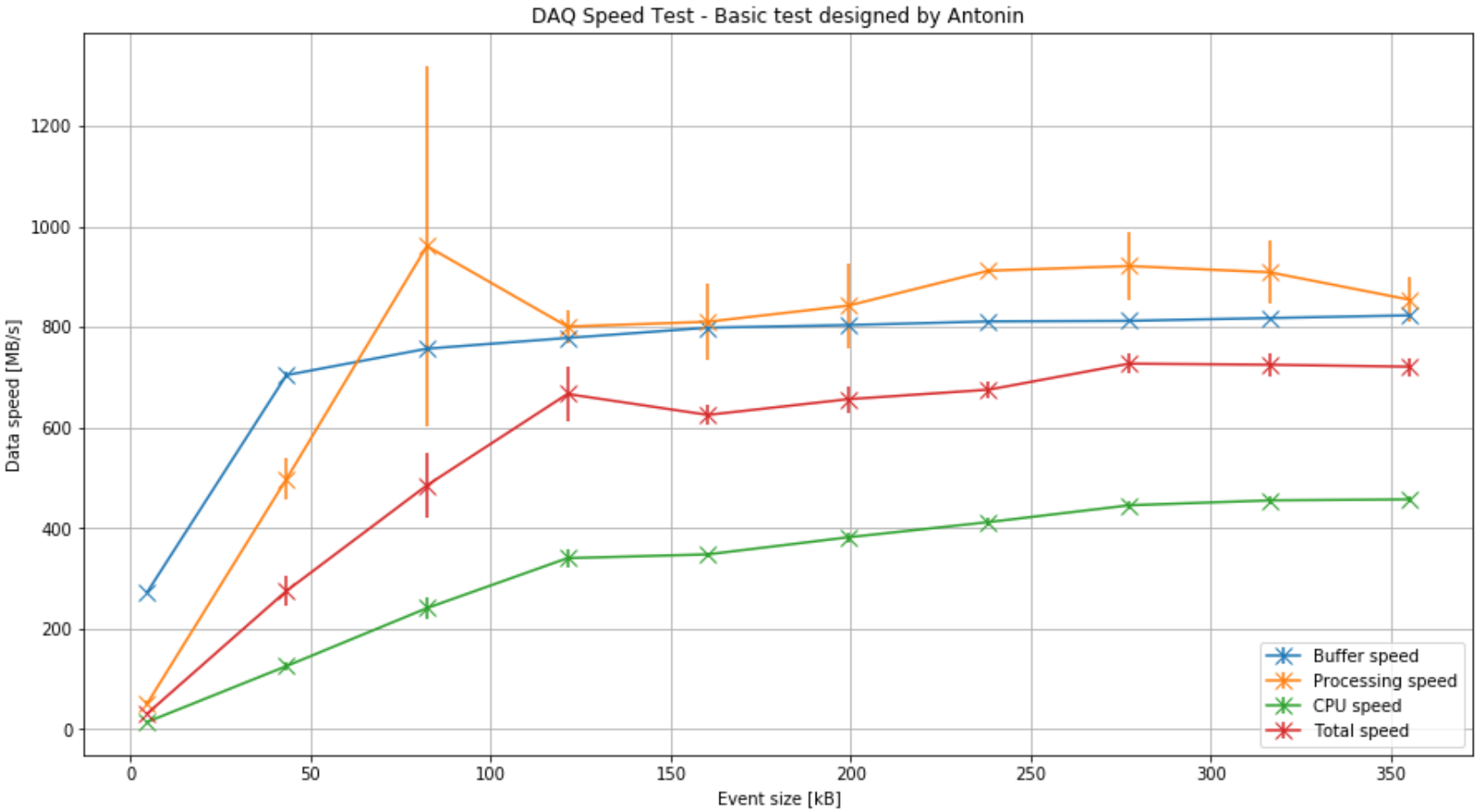
Overview

- Initial DAQ performance
- Anomalies in the DAQ
- Measurements taken during the dry run
- Implementation of the CRC32 checksum
- NUMA optimizations
- Benchmarking

Initial status of the DAQ

- During the normal run the average processing rate was at the level of **100 MB/s**
- The goal is to achieve **at least 1 GB/s** sustainable rate
- A proper measurement was required in order to identify the bottleneck
- Artificial event generator has been developed (by Josef and Antonín) for DAQ speed tests
- Universal measuring script have been created and used for measurements

Initial measurement of the DAQ software

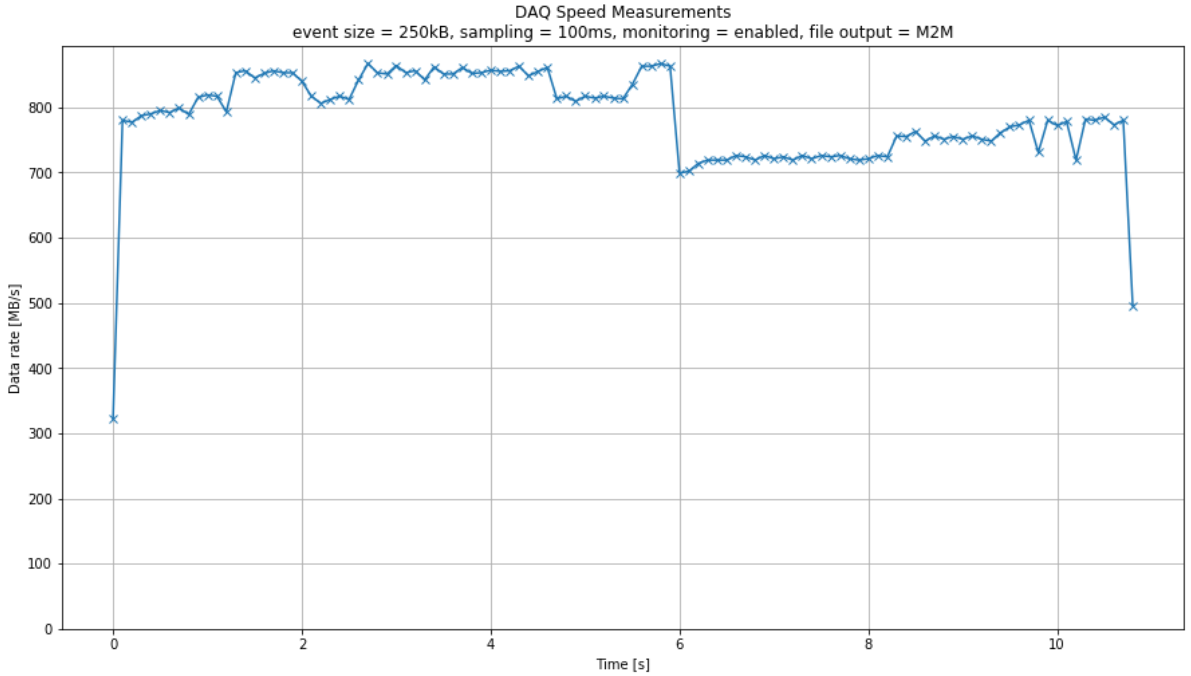


Identifying the bottleneck

- Processing speed of the DAQ software has been around 700 MB/s in average
- Therefore, the bottleneck must have been before the DAQ software, which could be the Spillbuffer card
- However, a proper evidence was required (later confirmed during the dry run)
- Another task was to explain anomalies in the DAQ data rate

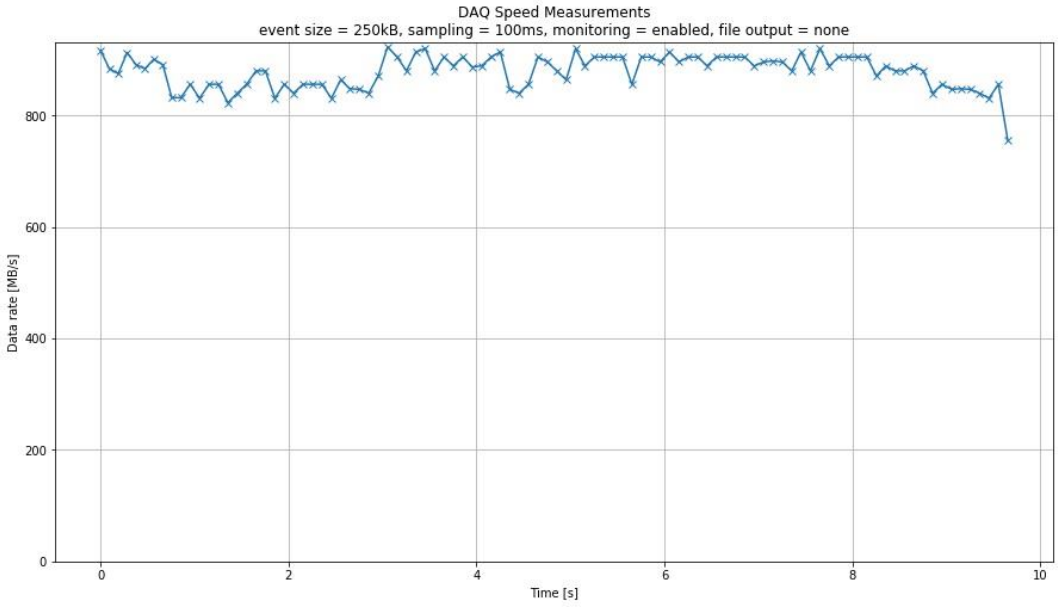
Anomalies in the DAQ software

- Significant fluctuations were observed in the DAQ performance
- Processing data rate was rather unstable
- This strange behavior had to be investigated



Identifying the source of fluctuations

- After another set of tests, the fluctuation has been finally explained
- They were caused mainly by:
  - Incompatible sizes of memory blocks and processing (logic) blocks
  - Timers with different timeout values



Measurements taken during the dry run

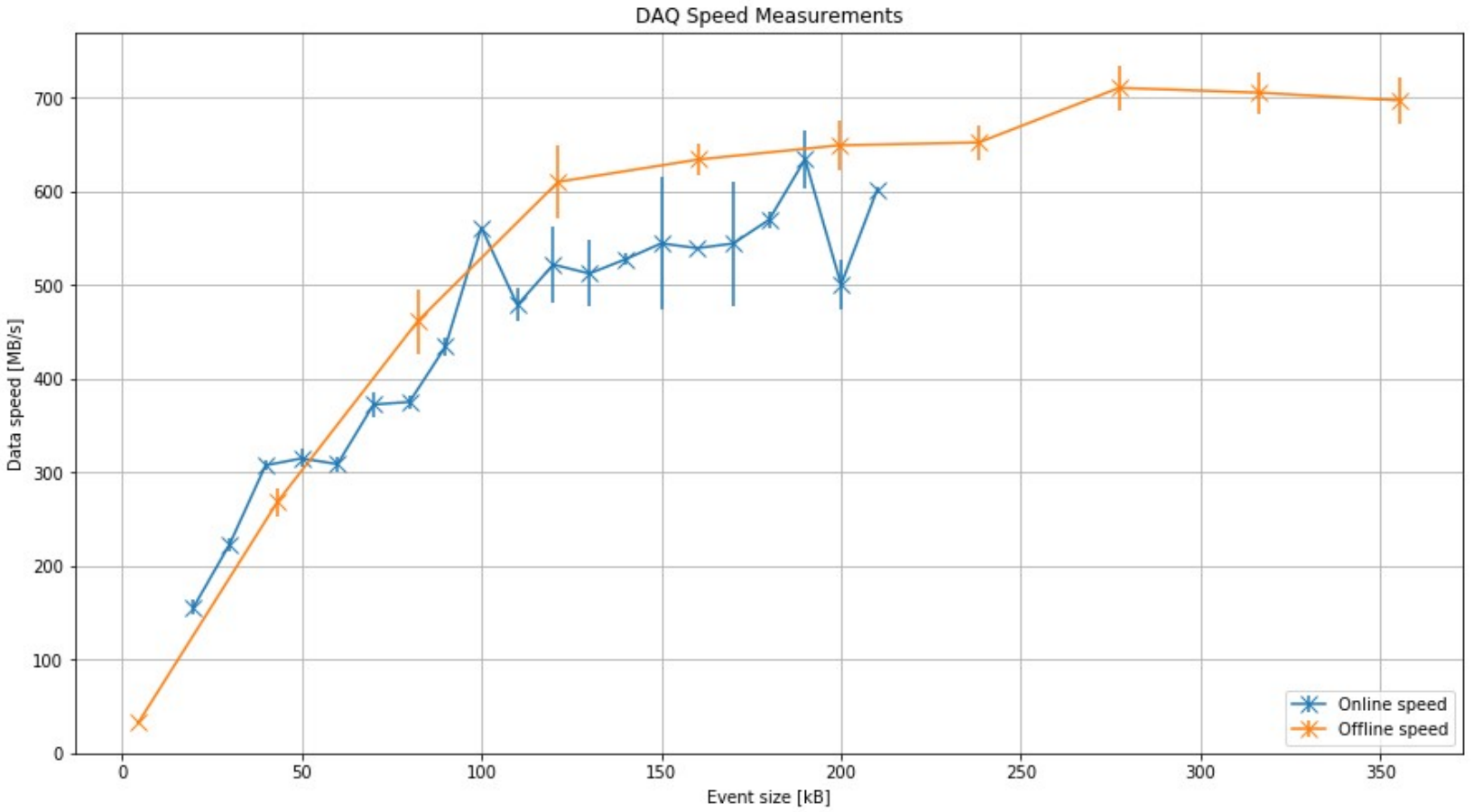
- Speed tests performed during the dry run proved that the overall performance of the DAQ was limited to **150 MB/s**
- The Spillbuffer was suspected as a limiting factor

- **In-memory buffering** has been implemented to find out software readout speed
- Data were accumulated in the RAM memory, and then, readout at once
- DAQ software was able to process data almost 5 times faster
- This result confirmed our concerns about the Spillbuffer
- Besides that, physics data with various event sizes have been recorded, so, they could be used in later tests

Measurements taken during the dry run

- Another bottleneck was represented by the pre-processing thread
- This thread is responsible for:
  - communication with the Spillbuffer driver,
  - preparation of data for processing,
  - memory allocations, DMA requests, etc.
- A new Spillbuffer driver and readout thread are required in order to further increase the data speed
- They are tightly coupled and must be developed together
- Currently, the processing speed of the DAQ software is limited **to 700 MB/s**

DAQ performance during the dry run



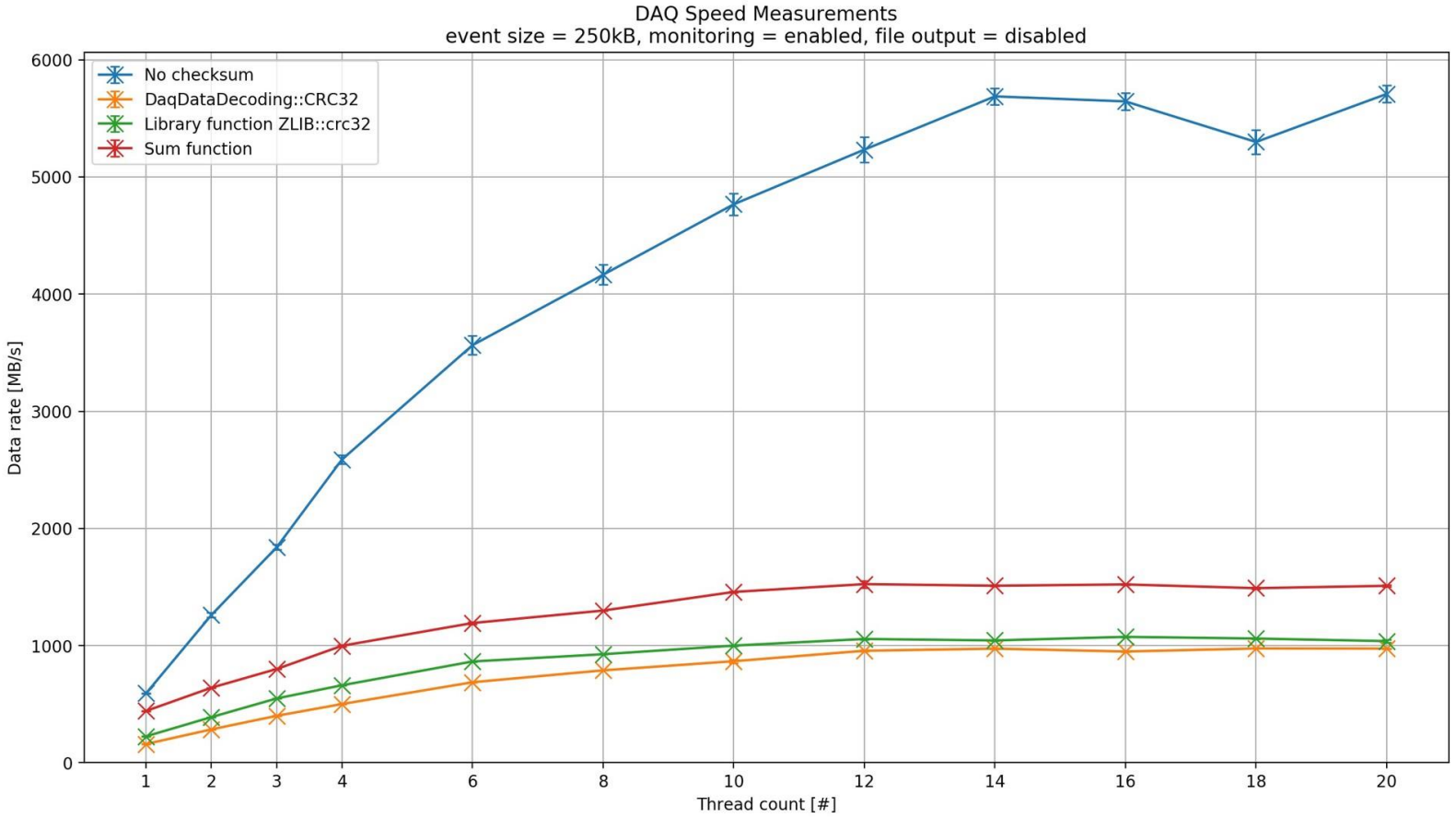
CRC32 checksum implementation

- CRC32 checksum will be implemented in the new DAQ
- Its impact on the overall performance of the DAQ had to be evaluated at first
- **DAQ transformation module** has been chosen to accommodate the CRC32 due to its parallel nature
- Performance of the transformation process was measured with resulting data rate **at the level 5.5 GB/s**
- With the CRC32 enabled, its performance dropped down **to 600 MB/s**
- Some optimization was required

CRC32 checksum implementation

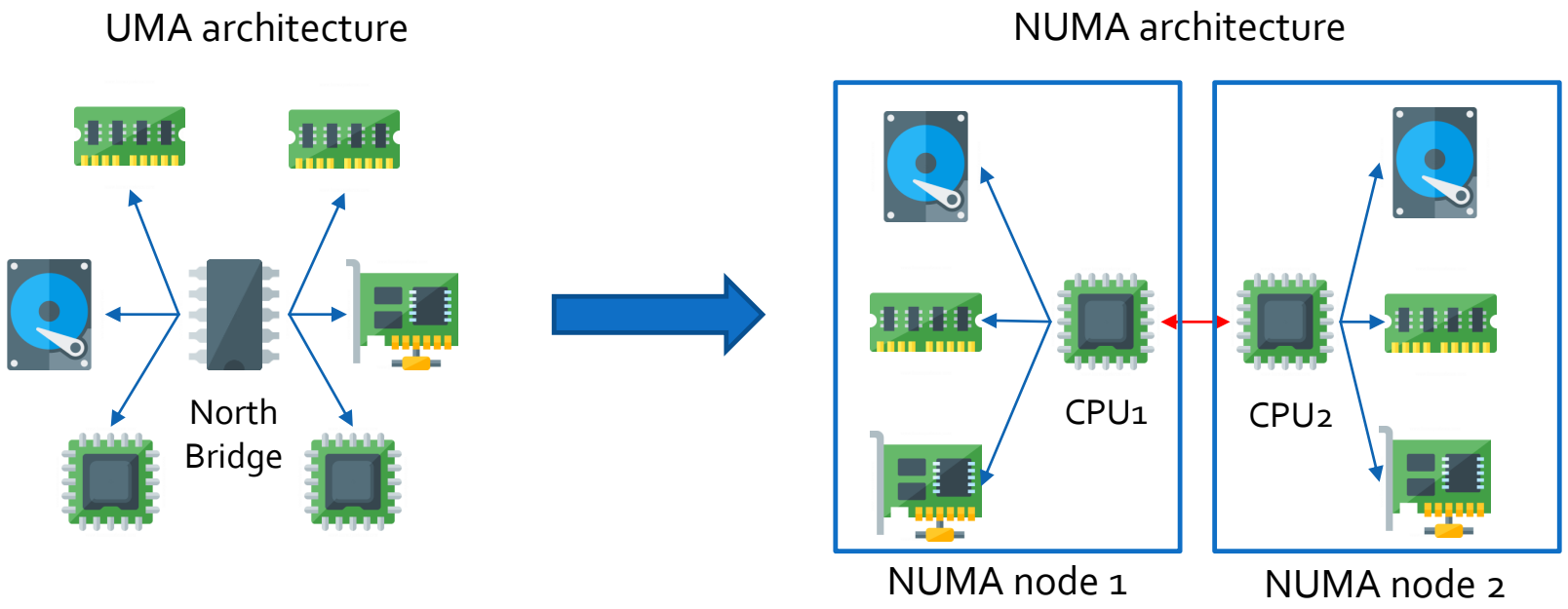
- Several optimization methods have been tested and validate:
  - Using optimization flags in the compiler
  - Precomputed table for CRC32 calculation
  - Manual prefetching of memory
  - Use of inline functions
  - Different kernel profiles for the CentOS 7 – the **„network-performance"** profile has showed up as the optimal one
- Eventually, the combination of methods mentioned above increased the performance **to approx. 1 GB/s**
- Although, the optimization continued with the NUMA balancing

Results of the CRC32 optimization



NUMA-aware optimizations

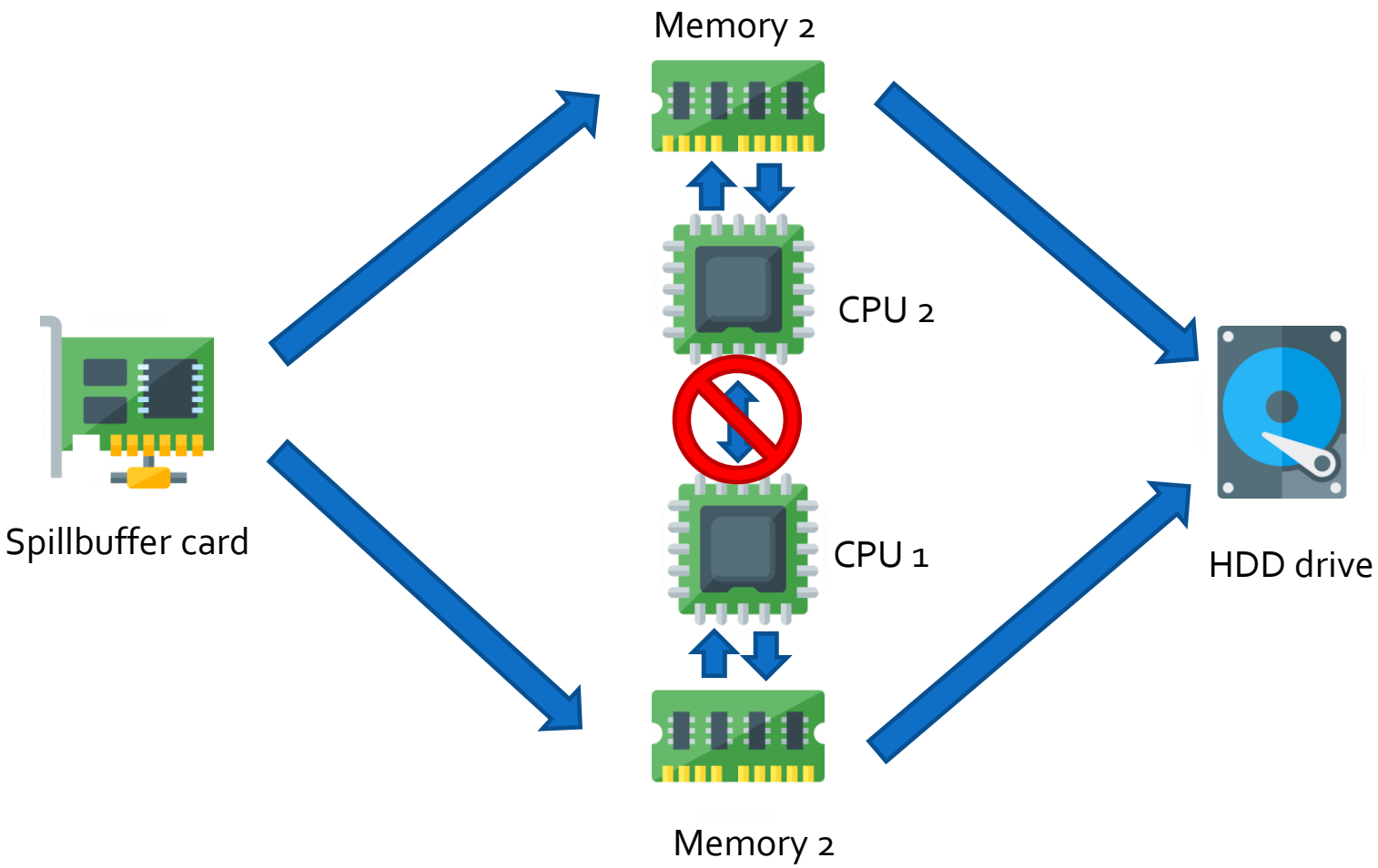
- NUMA stands for Non Uniform Memory Architecture
- That means some memory modules can be closer to some CPUs
- Accessing the distant RAM causes delays and the CPU stalls



NUMA in COMPASS DAQ

- Each readout server contains **two CPUs** Intel Xeon CPU E5-2620
- They behave **as one CPU** towards to software and user
- NUMA scheduler automatically balances the load across the CPUs
- This approach usually works fine in normal use, but fails in high-performance computing
- Tests showed that **only one CPU was active** during the readout process
- Increasing the number of threads did not help

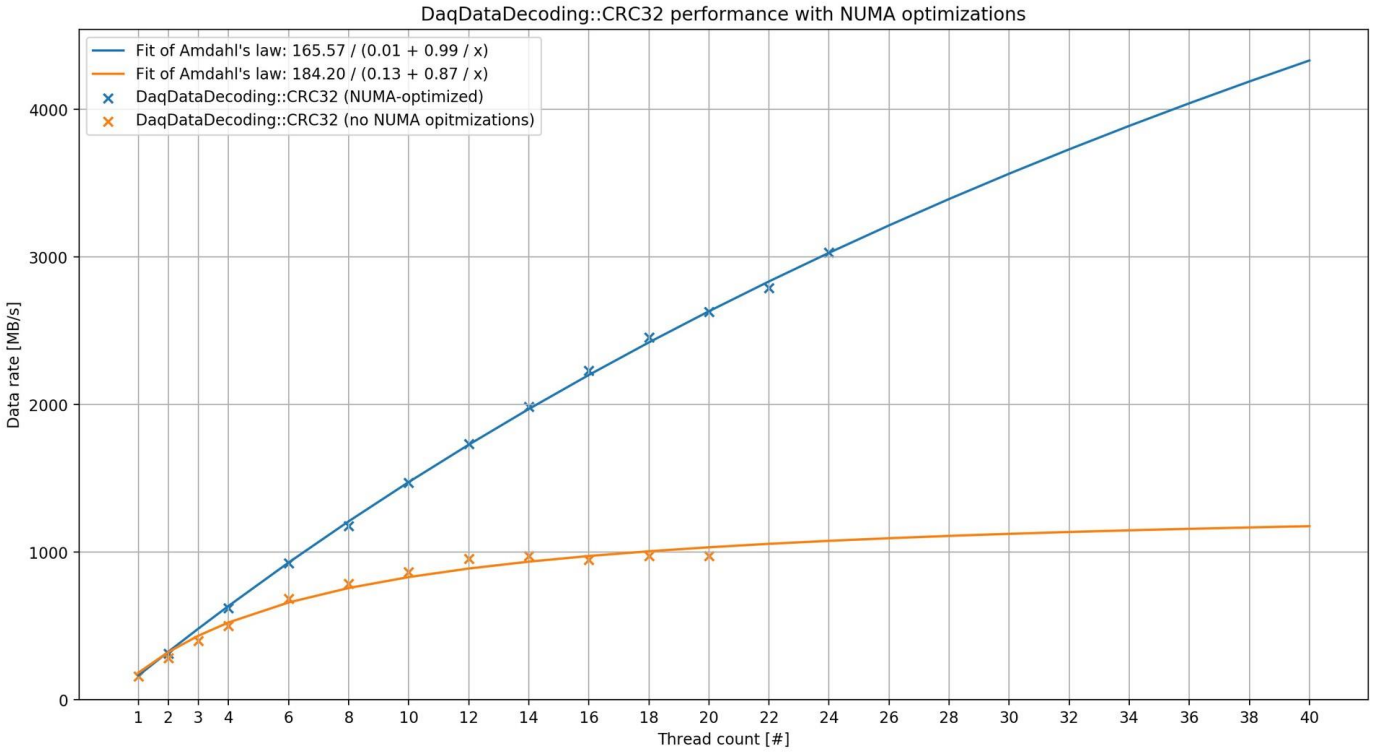
NUMA-aware optimizations



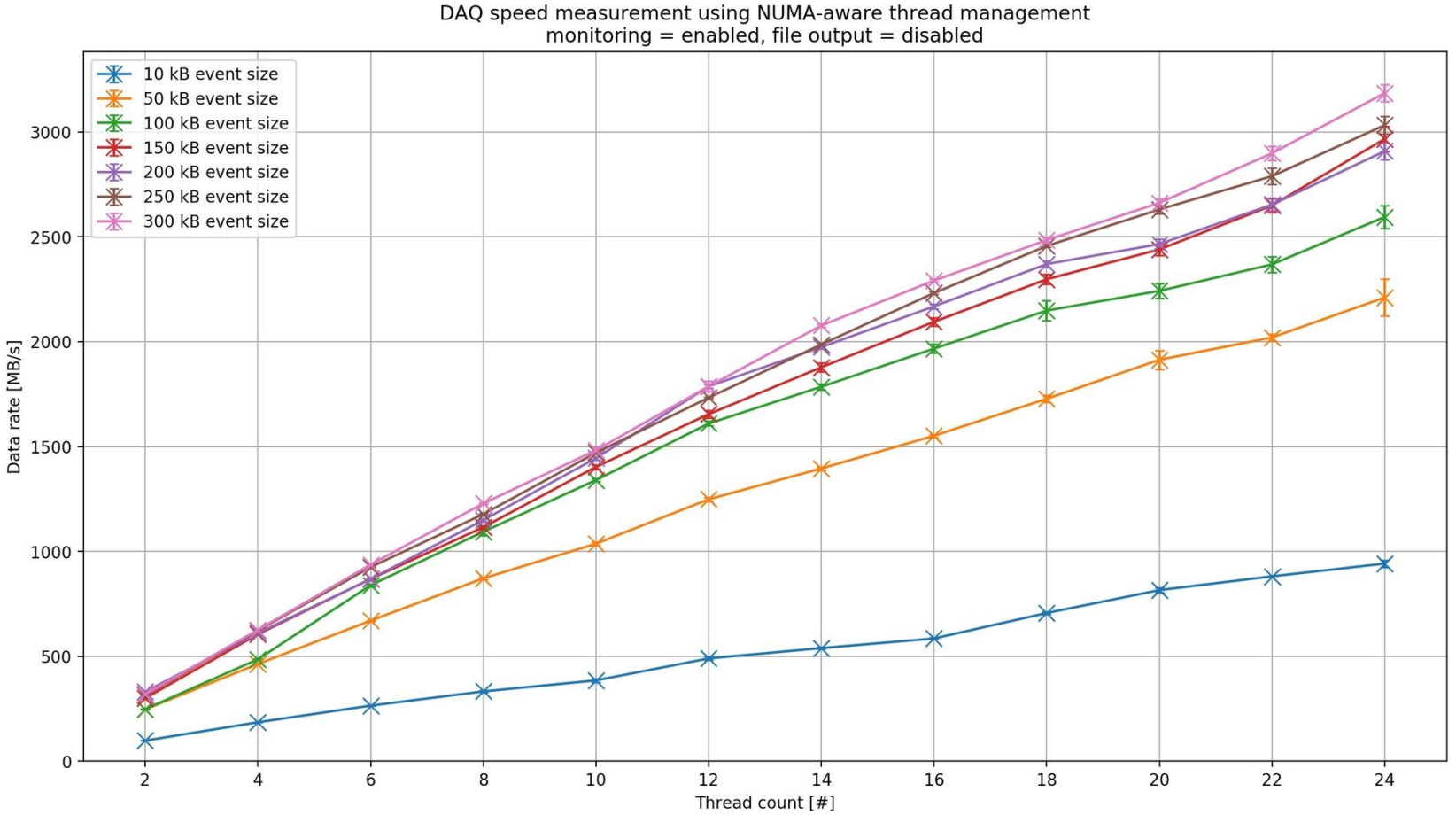
NUMA-aware optimizations

- More intelligent balancing has been implemented
- It uses the NUMA-aware approach, i.e. events are always processed on the closer CPUs
- This prevents CPUs from accessing the distant memory
- Currently, the load is distributed evenly across the CPUs
- Processing speed is significantly increased, **almost 3 times higher**

NUMA-aware optimizations



NUMA optimization



Hardware benchmarking

- Hardware benchmarking is an important part of the optimization process
- We have to understand what hardware we have and what we can we really achieve with it
- Each piece of hardware is assessed separately (CPU, RAM, HDD)
- Evaluation has been done by the **SYSBENCH** tool

CPU benchmark

Property	Value
Model	Intel Xeon CPU E5-2620 v2
Family	Ivy Bridge
Generation	3rd generation, 22nm
Instruction set	64-bit
TDP	80 W
# of CPUs	2
# of physical cores	12 (6 per CPU)
# of threads	24 (12 per CPU)
Base frequency	2.10 GHz
Turbo frequency	2.60 GHz
Cache	15 MB SmartCache
Bus speed	**7.2 GT/s QPI (~ 14.4 GB/s)**
# of QPI links	2
Supported memory	DDR3 800/1066/1333/1600
Max # of memory chanhels	4
Max memory bandwidth	51.2 GB/s

Memory benchmark

Access mode	# of threads	Block size	Read speed	Write Speed
sequential	1	1 KB	4,31 GB/s	3,56 GB/s
sequential	1	1 MB	12,98 GB/s	10,32 GB/s
sequential	**10**	**1 KB**	**28,78 GB/s**	**4,97 GB/s**
sequential	**10**	**1 MB**	**98,61 GB/s**	**45,06 GB/s**
random	1	1 KB	1,15 GB/S	1,20 GB/s
random	1	1 MB	1,08 GB/s	0,51 GB/s
random	10	1 MB	9,17 GB/S	0,97 GB/s
random	10	1 KB	9,77 GB/s	0,64 GB/s

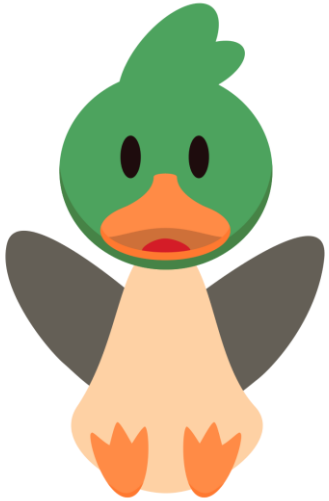
HDD benchmark

- # of drives: 4 per readout computer
- Type: Hitachi 7K4000 (7K3000) 2 TB
- Drive array:
  - RAID-1+0 (pccore11,pccore17)
  - RAID-5 (pccore15)
- More information will be introduced in the next presentation

HDD benchmark

Summary

- Performance of the DAQ software was measured
- Based on the results, several optimization have been introduced and applied
- Main bottlenecks have been identified and will be removed soon
- Performance of the CRC checksum was tested and assessed
- NUMA-aware optimizations have been introduced and their effect assessed
- Hardware of the readout servers was identified and evaluated



Thank you for your attention