Technical Report

Efficient Track Reconstruction on Modern Hardware

Thomas Lindemann

01/2018

technische universität dortmund

SFB 876 Verfügbarkeit von Information durch Analyse unter Ressourcenbeschränkung

Part of the work on this technical report has been supported by Deutsche Forschungsgemeinschaft (DFG) within the Collaborative Research Center SFB 876 ”Providing Information by Resource-Constrained Analysis”, project C5.

Speaker: Prof. Dr. Katharina Morik
Address: TU Dortmund University
Joseph-von-Fraunhofer-Str. 23
D-44227 Dortmund
Web: http://sfb876.tu-dortmund.de

1 Introduction

Particle physics has become a massively data-intensive discipline. Huge particle accelerators—such as the Large Hadron Collider (LHC) [1] at CERN—produce vast amounts of experimental data—4 TB/s in the case of the LHCb experiment [2] at CERN—which often must be processed in real time. Named after the b-quark, LHCb is one of the four big experiments at CERN. The general scope is to explain the matter/anti-matter asymmetry. The main focus is the study of particle decays involving beauty and charm quarks. In the LHCb Project, a continuous stream of hits is produced by the several stages of the LHCb detector. Given the low probability of observing an "interesting" collision, physicists produce a vast number of collision experiments in the hope of finding a few interesting ones. Thus, the event data have to be processed in real time, since there are no capabilities to store all collision event permanently with the current storage technology. Analyzing these data volumes has become the key limitation of the domain: any improvement in analysis performance translates into better insights on the physics side.

In this report, we present the results of our experiments of our current work with the HybridSeeding track reconstruction algorithm.

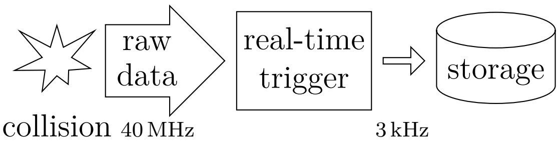

Figure 1: LHCb trigger system.

The LHCb experiment operates at a frequency of 40 MHz; that is, 40 million collisions are performed every second (year-round). For every collision, detectors record about 100 kB worth of data, resulting in a raw data stream of about 4 TB/s. [3]

Our approach is to use modern low-power hardware devices, because we expect this hardware to have more powerful compute capabilities while spending the same amount of energy. We reimplemented some algorithms of the LHCb Trigger successfully for execution on energy efficient low-power hardware and for a distributed execution on a low-power cluster. The experiments have shown that using modern low-power hardware improves the energy efficiency drastically while the event processing time is not increased dramatically.

With (linear) scalability in place, resource efficiency becomes a primary concern for large-scale data processing systems (in order to keep cost low; better throughput can always be achieved by adding more cores). With our approach, therefore, we aim for (i) low-cost hardware and (ii) high energy efficiency—both aspects that match the characteristics of low-power processors with ARM and Intel architectures. We tailored our experiments to run on large-scale, low-cost, low-power ARM and Intel installations.

2 The LHCb Experiment at CERN

The Large Hadron Collider (LHC) at CERN is the world's largest particle accelerator, located near Geneva (Switzerland). About 100 m under the earth surface, protons are accelerated to near-light speed and then made to collide with one another. As a consequence of the collision, new, unstable particles may form up, but quickly decay into smaller decay products.

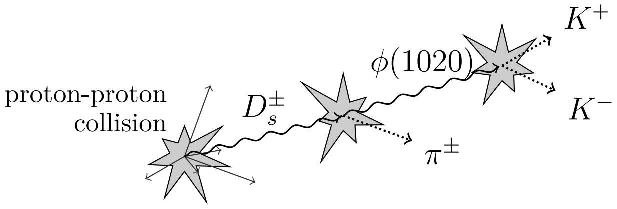

Figure 2: $D_s^\pm \rightarrow \phi(1020)\pi^\pm \rightarrow K^+K^-\pi^\pm$ decay channel. A $D_s^\pm$-meson decays into a pion ($\pi^\pm$) and two kaons (K^+ and K^-), which will be seen by the LHCb detector system.

Figure 2 illustrates this for the decay channel $D_s^\pm \rightarrow \phi(1020)\pi^\pm \rightarrow K^+K^-\pi^\pm$. A $D_s^\pm$-meson decays into a $\phi(1020)$ and a pion ($\pi^\pm$); the former further decays into two kaons of opposite charge (K^+ and K^-). In practice, the $D_s^\pm$ and $\phi(1020)$ will travel a few centimeters before they decay.

The decay products (here the $\pi^\pm$, K^+, and K^-) can be detected through a series of detectors, which are placed several meters away from the primary collision vertex, as illustrated in Figure 3. [2]

Many possibilities (decay channels) exist according to which the colliding protons might form new particles and decay afterward. Only few of them, however, are of interest to the physicists (such as the above $D_s^\pm \rightarrow \cdots \rightarrow K^+K^-\pi^\pm$ channel). A key part of the analysis, therefore, is to test whether the particles observed by the detectors match a decay channel of interest and filter out others. To this end, recorded energies and particle momentums are added up for each step in the decay channel and according to the rules of physics (preservation of energy and momentum).

This part of the analysis, therefore, acts as a filter to the input data stream. But a highly selective one: only 10^{-12} to 10^{-15} of all collisions are "interesting" to the physicists in this sense.

3 Software Trigger Model

3.1 Current Trigger Model

The current Software Trigger at the LHCb Project is composed of state of the art Intel Xeon machines which are arranged in a farm of about 20000 cores. The total number of cores depends on the number of machines which is limited by constraints in energy consumption, thermal discharge, space and expenses.

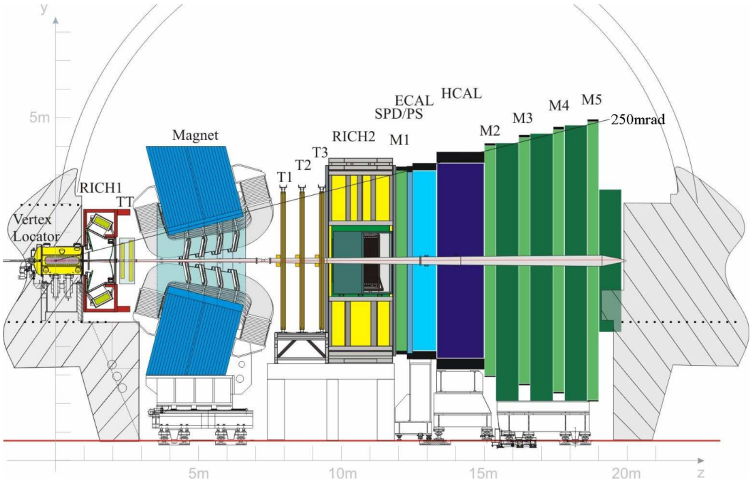

Figure 3: LHCb detectors. The proton beam is located horizontally in the center of the picture. Protons collide near the "vertex locator" on the left; decay products pass a magnetic field before they are detected in several layers of detectors.

In the last Trigger Upgrade has been decided not to make use of the FPGA Hardware Triggers anymore which had pre-filtered the events to an event rate the Software Trigger can handle. Instead, the Software Trigger is supposed to process all events up to the mentioned event rate of 40 MHz. Figure 4 outlines the current arrangement at the LHCb.

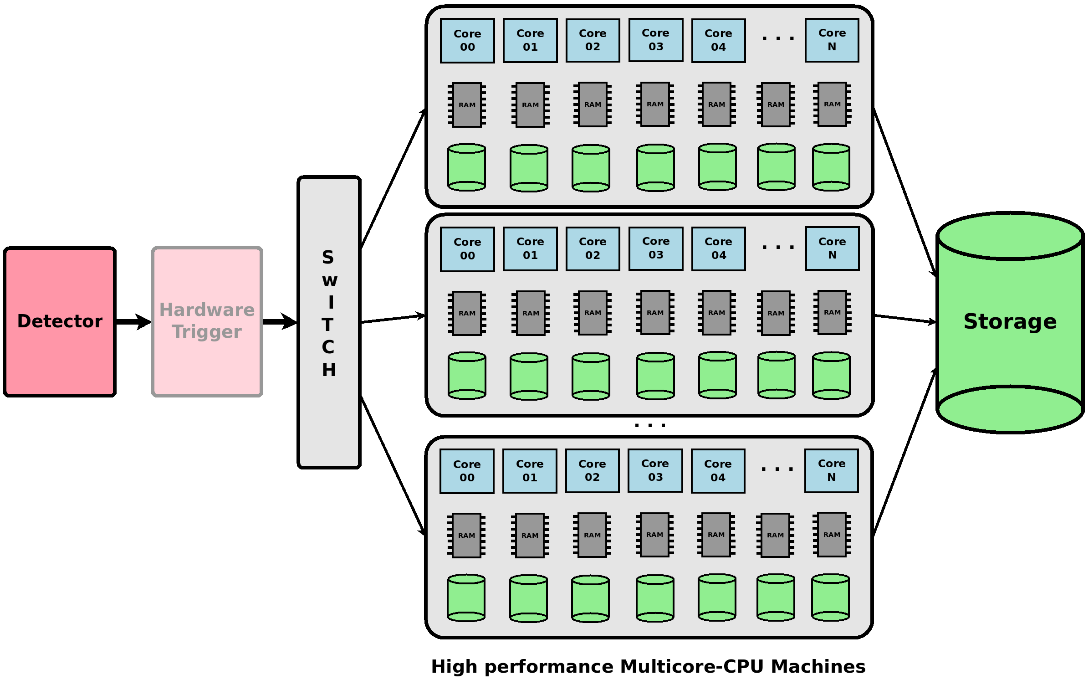

Figure 4: Current Processing Model

3.2 New proposed Trigger Model

Our proposed new concept is to deploy more power efficient processor architectures combined with heterogeneous components which make it possible to make use of the best component architecture for a given problem. In the first step, we took the HybridSeeding Tracking Algorithm from the Trigger, which is an important step in the reconstruction of the an event, and we adapted it to low-power processor architectures by ARM and Intel. In the further step, we also place parts of the algorithm to the embedded GPU of the low-power SOCs.

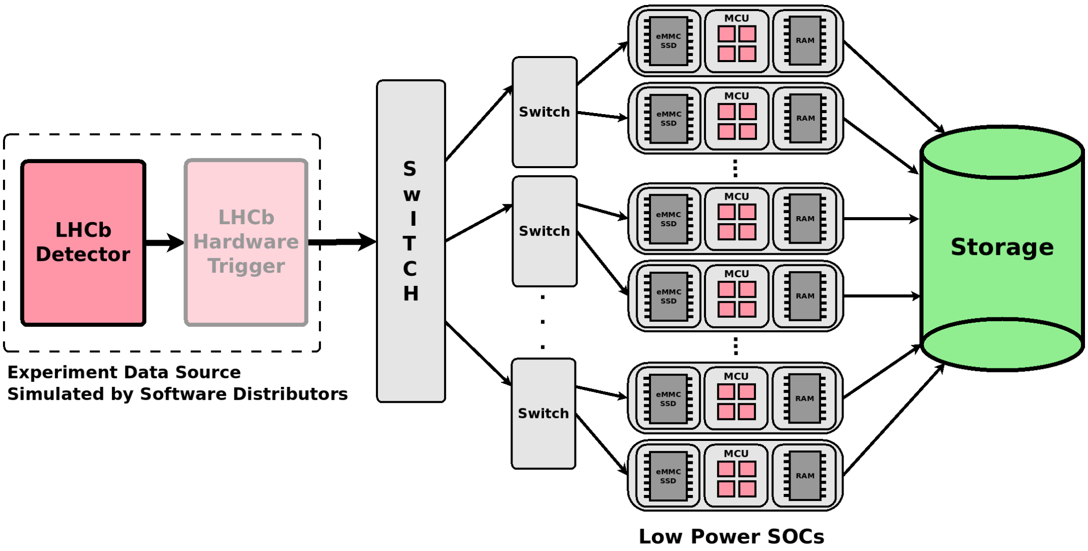

Figure 5: New Tested Event Processing Model

4 Experiments

4.1 Test Systems

For the evaluation of our approach, we have run the HybridSeeding Tracking Algorithm in different configurations on several hardware configurations.

As baseline for the comparison of the efficient modern hardware architectures, we measured the execution time and power consumption of the algorithm on a state of the art reference system, as it is used in the Computing Farm at CERN nowadays. The reference system consists of a dual socket E5-2695 processor configuration and provides 12 physical cores each with hyper-threading technology which is a total number of 48 parallel threads. As candidates for power efficient architectures, we tested ARM's A7, A15, A53 and A73 processors and Intel's Airmont Core in the Celeron N3160, a very similar architecture to the older Silvermont Core. The Celeron N3160 is quad core processor design.

The tested ARM architectures come in several packages, which are assembled to the tested development boards. Thus, the Odroid C2 board provides a quad core Cortex-A53, while the Odroid XU4 has got a heterogeneous octa core architecture with four Cortex-A7 and four Cortex-A15 cores, as the HiKey960 board is a newer generation heterogeneous octa core with four Cortex-A53 and four Cortex-A73 cores. The detailed characteristics of the tested hardware and the reference system is shown in Table 1.

Table 1: Hardware characteristics of our test systems.

Ressource	Xeon Server 2x E5-2695	Odroid C2 ARM A53	Odroid XU4 ARM A7+A15	HiKey 960 ARM A53+A73	UDOO x86 Intel N3160
CPU Amount	2	1	1	1	1
CPU Freq.	2.80 GHz	1.50 GHz	1.4/2.0 GHz	1.85/2.36 GHz	2.24 GHz
Cores	24	4	8	8	4
Threads	48	4	8	8	4
RAM	256 GB	2 GB	2 GB	3 GB	4 GB

The packaging of the ARM based SOCs allows precise execution time measurement of either the smaller core or the bigger one or even both kind of cores working side by side. Having both kinds of processor cores in the same chip package makes it indeed more difficult to measure the power consumption of only one kind of core, because the idle power of the other core is always present. In addition, the SOCs of the low-power board include a low-power GPU, which also has an idle power consumption.

Measuring the power on the reference high-performance Intel Xeon dual socket server has been done by integrated sensors in the management interface of the Fujitsu Primergy Series, because it is the less invasive method. All unnecessary expansion cards have been removed. Nevertheless, it has to be mentioned that measuring at the side of the mains might not perfectly accurate, as there might some losses depending on the power supplies.

4.2 Test Data Set Characteristics

In our experiments (Section 4) we use a data stream subset that contains 56 thousand collision events (i.e., this data set, close to 50 particle tracks were recorded for each collision).

The data set used for the experiments consist of simulated detector data for HybridSeeding Algorithm tests. We used an identical test data set for all tested architectures. Because of the heterogeneity of the events, we have chosen a test large dataset of 56000 events for our experiment, while we measured the time and power consumption. The heterogeneity of the simulated events is caused by a large variation of possible decay channels and resulting number of particle tracks, which have to reconstructed by the HybridSeeding Algorithm. Figure 6 shows a histogram of the event sizes of the test data set.

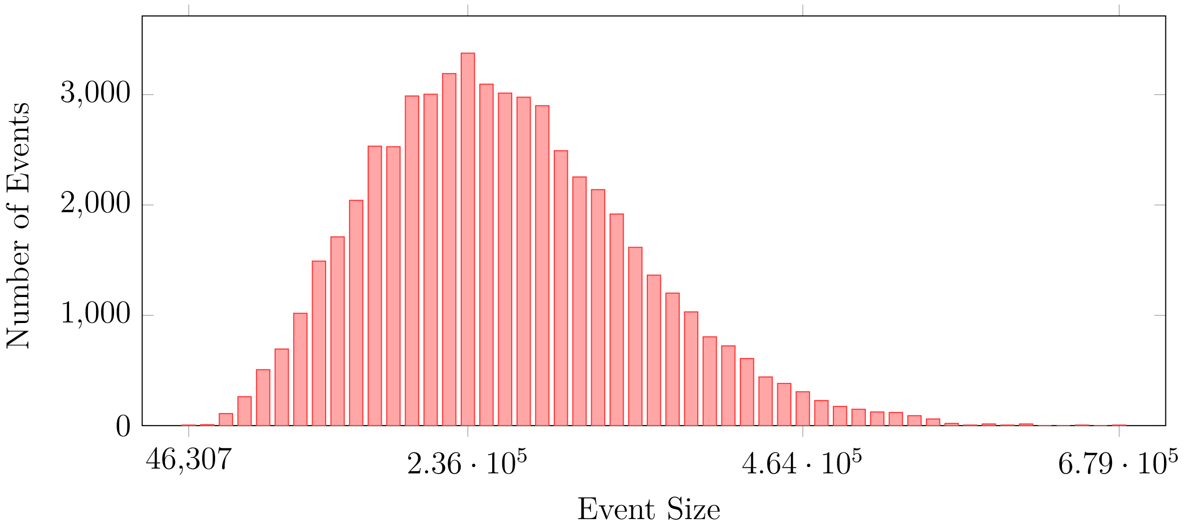

Figure 6: HybridSeeding Algorithm Stream Processing Speed Comparison on Low-Power MCUs and Reference Xeon E5-2695 CPU

4.3 Event Stream Processing

The experiments we made consist of a simulated detector, which sends the events through a TCP connection to several nodes. We have build the distribution system for handling continuous event streams. The overhead of the event distribution has to be as small as possible, so we decided to send the events as packets of Google's Protobuf format.

Due to the large number of processing cores and the resulting processing speed of our high performance reference system, we seeded the events through several distributors and network ports, to make sure that the event stream seeding is not the bottleneck. Because the network transport of the event streams still has a small overhead, we also measured only the path time of the HybridSeeding Algorithm in another experiment and compare the average results in Section 4.4.

Table 2 shows the results of processing 56000 events test data set and the measured time and power. For a better comparison between architectures, we compare two quality metrics, the seeding in events per second and the efficiency in events per power and time, which is events per energy in joule. Figure 7 and Figure 8 visualize the quality metrics in a bar graph.

In the experiment data, we see that the high performance reference system has of course the highest single core performance as expected and a big number of cores, which are additionally split via hyper-threading. A single core of the reference system, which is the same kind of hardware used in the cern computing farm, can do an average amount of 40 events per second, which is about 25 ms per single event including algorithm path time and network reception.

In contrast, the low-power processor architectures reach event rates between 22 and 84 events per second using four cores. Thus, the low-power architecture cores are approximately between 86% and 48% slower than the high performance cores.

Table 2: HybridSeeding Stream Processing Comparison.

Configuration	Processing Cores	Processing Time	Average Power	Events per Second	Events per Joule
Odroid C2 A53	4	1568.83 s	4.39 W	35.70	8.13
Odroid XU4 A7	4	2546.7 s	4.66 W	21.99	4.72
Odroid XU4 A15	4	1517.91 s	7.59 W	36.89	4.86
Odroid XU4 A7+A15	8	1019.81 s	7.88 W	54.91	6.97
HiKey960 A53	4	1360.75 s	6.47 W	41.15	6.36
HiKey960 A73	4	665.37 s	9.22 W	84.16	9.13
HiKey960 A53+A73	8	499.15 s	9.33 W	112.19	12.02
UDOO x86 N3160	4	1071.89 s	5.33 W	52.24	9.80
Xeon E5-2695 - 1 CPU	1	1391.0 s	173 W	40.23	0.23
Xeon E5-2695 - 1 CPU	12	136.46 s	243 W	410.38	1.68
Xeon E5-2695 - 2 CPU	24	72.70 s	360 W	769.76	2.14
Xeon E5-2695 - 2 CPU	48*	56.79 s	394 W	986.09	2.50

*with Hyper-Threading

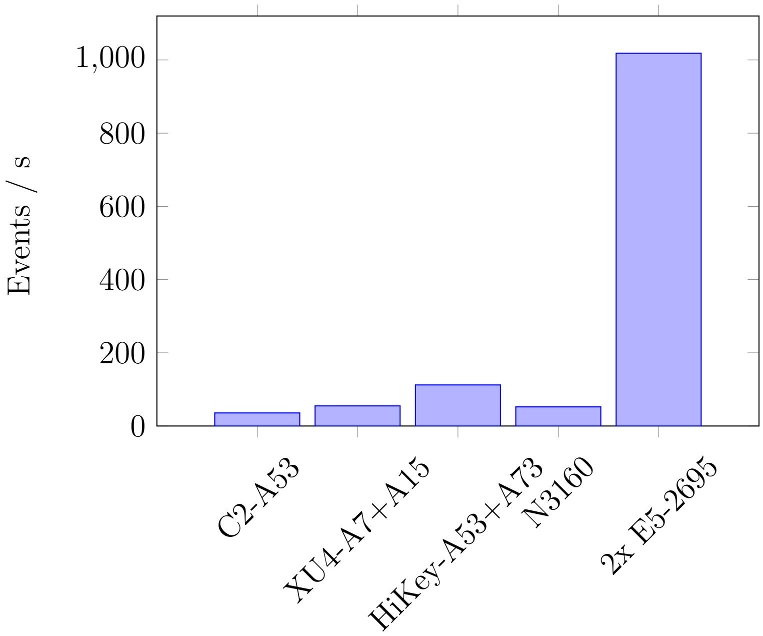

Figure 7: HybridSeeding Algorithm Stream Processing Speed Comparison on Low-Power MCUs and Reference Xeon E5-2695 CPU

The best result has been achieved by the most recent released ARM Cortex-A73 architecture, which is the most efficient in the given package of the HiKey960 board (see Section 4.1)

Furthermore, we observed on the reference system that the event rate is dropping a little bit if many cores are used in parallel, although we provided a sufficient event rate from the distributors. Especially, the hyper-threaded cores of the reference system can just achieve a small speedup, because these virtual cores have to share a physical core.

The main observation of this experiment is proving our approach, that a system concept of low-power processors highly improves the efficiency.

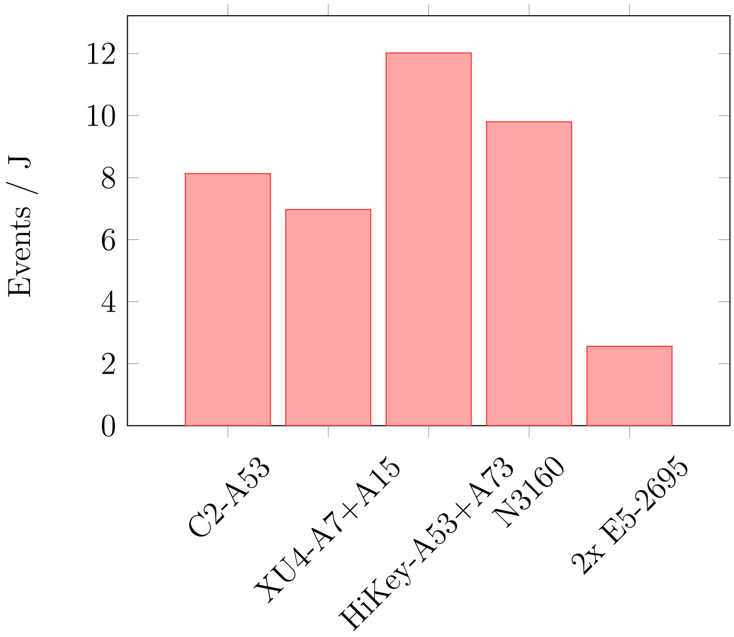

Figure 8: HybridSeeding Algorithm Stream Processing Efficiency Comparison on Low-Power MCUs and Reference Xeon E5-2695 CPU

4.4 Algorithm Path Time Analysis

We also measures the algorithm path time of the HybridSeeding Algorithm, which is only the time for processing events without any effects of the event distribution and reception. With this experiment shall be proven that the performance of the different architectures is not falsified by other factors.

Table 3: HybridSeeding Algorithm Path Time Table.

Processor	Average Algorithm Path Time
Odroid C2-A53	101.64 ms
Odroid XU4-A7	165.56 ms
Odroid XU4-A15	78.58 ms
HiKey960-A53	86.69 ms
HiKey960-A73	37.66 ms
UDOOx86 N3160	74.07 ms
Xeon E5-2695	24.85 ms

In a direct comparison of only one core, one can see that the performance of a single core low-power core is not far away from the high-performance system's. Especially for the newer generation ARM cores.

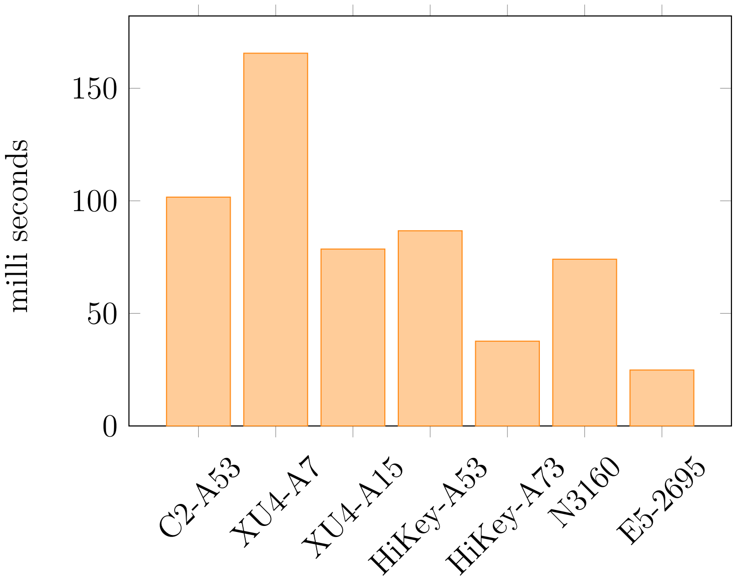

Figure 9: HybridSeeding Algorithm Path Time Comparison

The high-performance system Xeon E5-2965 processor achieves an average single thread performance of 24.85 ms per event while the ARM Cortex A73 reaches an average of 37.66 ms per event for the same test data set.

We can see that the single core performance for the HybridSeeding Algorithm is just about 31% slower and shows that in this specific use case, the overall performance of the Xeon is basically achieved by the bigger count of parallel cores and is the reason why the energy efficiency, which includes time and power consumption, is so much better.

The main reason why the single core performance of the high-performance processor is that in these kind of stream processing workloads is no use of special characteristics of the high-performance system, like very large caches, shared l3 caches, etc.

4.5 Scalability

A key design goal of our approach is to achieve (linear) performance scalability with the amount of resources used for processing. To evaluate whether we achieved this goal, we scaled our experiment to run on a varying number of nodes within our ARM Cortex-A53 cluster.

Figure 10 shows the resulting throughput rates. As can be seen in the figure, the system indeed meets our goal of linear scalability. This means that, by adding more compute nodes, our approach could easily be configured to meet the throughput demands of the experiments at CERN.

Our approach of the low-power device trigger has been created with the idea of a scalable number of small MCUs, which can be adapted to the event frequency. Thus, we assumed that our approach scales out linear over the number of processing nodes in the system as well, even in a cluster of low-power ARM Cortex-A53 SOCs.

For verifying our approach on scalability, we also did some experiments with a partial load of the low-power cluster, while measuring the events per second rate and the events per energy. Table 4 lists the tested configurations and the resulting experiment data in events per second and per joule.

Table 4: Scalability Test

Experiment				Events per Second	Events per Joule
Node Count	Processing Cores	Time Overall	Power Consumption		
1	4	1568.83	4.4	35.70	8.13
2	8	802.7	7.8	69.76	8.99
3	12	536.26	11.34	104.43	9.29
5	20	325.14	18.31	172.23	9.53
10	40	173.27	36.49	323.20	9.53
15	60	114.48	56.13	489.17	9.25
20	80	93.94	74.46	596.13	9.26
25	100	78.28	90.05	715.38	9.50
30	120	68.55	106.75	816.92	9.48
35	140	64.22	120.39	872.00	9.30
40	160	58.28	140.43	960.88	9.36

As illustrated in Figure 10 the events per second plot shows are linear increasing of the system performance of the experimental system over an increasing count of nodes in the cluster. Figure 11 illustrates the plot of the events per joule over an increasing node count. The efficiency is almost constant as expected with some small amount of variation.

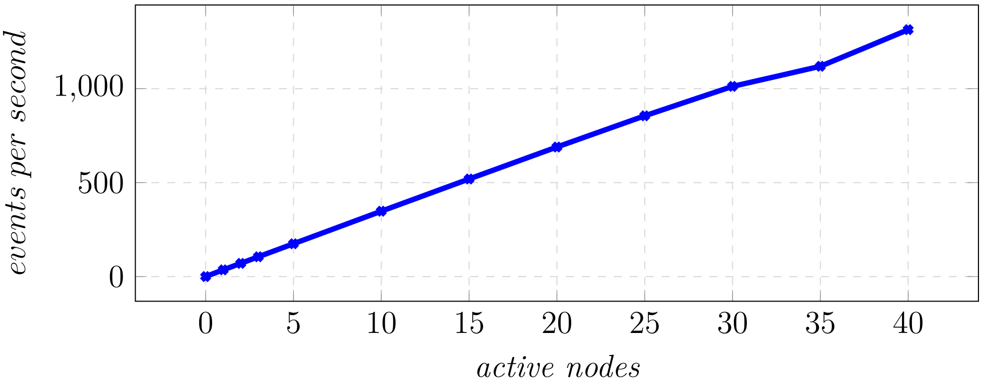

Figure 10: Run time of HybridSeeding over an increasing number of C2 Nodes.

The system, thereby, does not lose its favorable resource efficiency characteristics. Figure 11 shows how the events per joule metric changes as the system is scaled to larger node counts. As can be seen, already for small configurations the system reaches its full research efficiency (and does not degrade afterward). We indeed expect that the results of this experiment are applicable to other low-power hardware and other algorithms.

The reason for the good scalability is that the events can be processed independently, so that the problem class of the Software Trigger at CERN is trivial parallelizable.

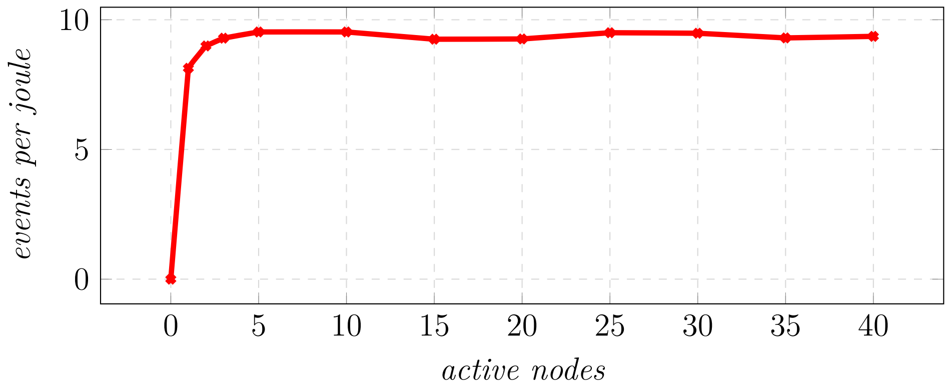

Figure 11: Efficiency of HybridSeeding over an increasing number of C2 Nodes

5 Conclusion

Using the example of the HybridSeeding Algorithm, we could show that the use of hardware optimized for power efficiency is a suitable alternative for the event processing in the Software Trigger of the LHCb Project at CERN. The power consumption optimized processors only for the tested algorithm is just about 31% slower, but it is a factor of approximately 4.7 times more efficient in processed events per joule. We also noticed that there is a small overhead for the event distribution, which is slightly increased for a bigger number of small compute units compared to the multi core high performance reference system, but the experiments also show that the relative per-core-performance drops on the high-performance machine as well, when the number of parallel processing cores is increased.

6 Future Work

Our current work is to migrate the HybridSeeding Algorithm to GPU hardware, because is part of the Tracking and which is as described a major workload for the event reconstruction in the Software Trigger. We expect the GPU to be able of computing the basic operations done by the HybridSeeding Algorithm very fast in parallel. A challenge is the heterogeneity of the events, as described in Chapter 4.2. The number of parallel processing units in high performance GPUs is very big, so that it would be necessary to process a stack of events in parallel, which is hindered in efficiency by the event heterogeneity. Our aimed solution is to use low-power ARM Mali GPUs, this approach is not just obvious, as we already propose the use of low-power CPUs, it is a also better fitting approach to compute every event individual with a bigger number of GPUs in parallel.

The goal we aim for is a large scalable efficient system of heterogeneous units, which is able to place all operations to the processor that can perform the operation best, what means most efficient.

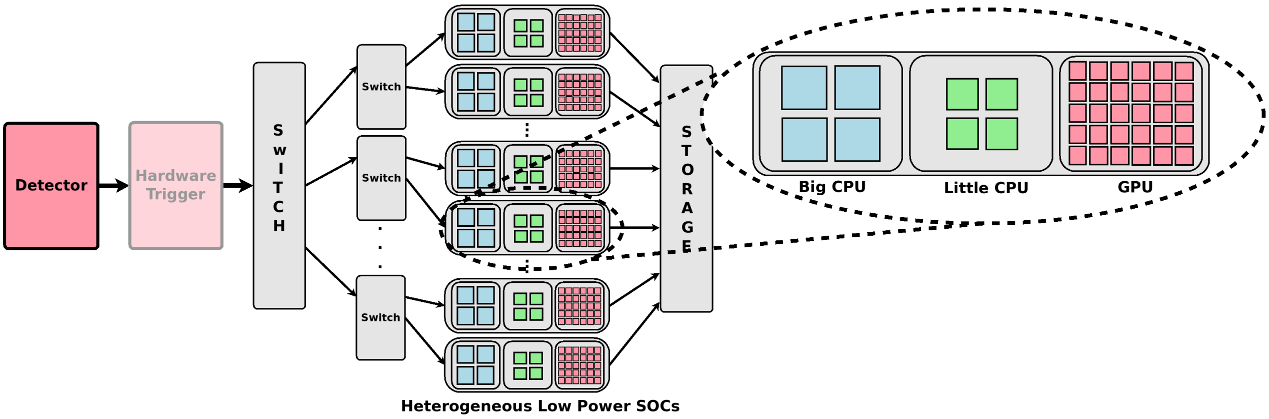

Figure 12: Future Processing Model

Acknowledgment

The authors would like to thank the collaboration partners from the Experimental Physics Department E5A of the TU Dortmund University in the SFB876-C5 collaboration, which is provided by the Research Project SFB876.

References

[1] L. Evans and P. Bryant, "Lhc machine," Journal of Instrumentation, vol. 3, no. 08, p. S08001, 2008.

[2] The LHCb Collaboration, "The lhcb detector at the lhc," Journal of Instrumentation, vol. 3, no. 08, p. S08005, 2008.

[3] "LHCb Trigger and Online Upgrade Technical Design Report," Tech. Rep. CERN-LHCC-2014-016. LHCB-TDR-016, May 2014. [Online]. Available: https://cds.cern.ch/record/1701361