EUCALL

The European Cluster of Advanced Laser Light Sources

Grant Agreement number: 654220

WP5 – UFDAC

MS 5.3

Performance evaluation of data transfer and injection algorithms

Lead Beneficiary: HZDR

Authors: Michael Bussmann (HZDR), Fabian Jung (HZDR), Kwinten Nelissen (ELI-ALPS), Balasz Bago (ELI-ALPS), Wassim Mansour (ESRF), Carlos Lopez Cuenza (PSI)

Due date: 30.09.2017
Date of delivery: 30.09.2017

Project webpage: www.eucall.eu

Deliverable Type	
R = Report DEM = Demonstrator, pilot, prototype, plan designs DEC = Websites, patents filing, press & media actions, videos, etc. OTHER = Software, technical diagram, etc.	R
Dissemination Level	
PU = Public, fully open, e.g. web CO = Confidential, restricted under conditions set out in Model Grant Agreement CI = Classified, information as referred to in Commission Decision 2001/844/EC	PU

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 654220

A. Abstract / Executive Summary

This document discusses the current status of data transfer and injection solutions within UFDAC.

It includes results on the main topics in data transfer and injection as identified in the UFDAC work package, namely latency, throughput, scalability and resilience provided by ELI-ALPS, ESRF, HZDR and PSI.

B. Deliverable Report

1 Introduction

Data transfer and injection are important for ultrafast data acquisition in four distinct ways:

1. Latency of the solution. For real time feedback applications latencies must be as short as possible to avoid pile up or loss of important data. For high data rate application latency might exist, but can be hidden by buffering incoming data while performing calculations on previously acquired data in parallel.
2. Throughput of the solution. Throughput is foremost determined by the capabilities of the hardware that is used to transfer the data but also by the software stack used for data transfer.
3. Scalability of the solution. Scalability means that with growing demands the data transfer solution provided can be adapted, so that there is at best a linear relation between increase in throughput and increase in resources needed for data transfer. Scalability also means that solutions can be scaled to different hardware platforms if growing demands in throughput require this.
4. Resilience of the solution. If data production is expensive as is the case at many light sources, data loss during transfer or injection is inacceptable. As such, resilience of solutions is important to minimize the possibility of data loss.

In the following the status of data transfer and injection within UFDAC is summarized and discussed with reference to latency, throughput, scalability and resilience if applicable. Contributions from HZDR, ELI-ALPS, PSI and ESRF are included.

2 Objectives

The objectives are

- Defining the requirements on latency, throughput, scalability and resilience according to the applications at the various RIs
- Measuring prototype implementations to compare with requirements
- Evaluate measurements according to requirements

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 654220

3 Work performed / results / description

ELI-ALPS

Modern High Performance Computing (HPC) centers provide high-speed/low-latency network interconnects for the compute nodes. However, in order to harness the performance of this interconnects in an efficient and effective way a better understanding of data-injection handling is required, which depends on the available hardware and installed software stack.
The main aim of the report is to investigate the bandwidth and latency of these high-speed interconnects for different usage modes as function of the data packet size. For this purpose, two different benchmark programs are used. One them is the OSU benchmarks [1] based on the popular MPI-library and the other is the perftest benchmark [2] which perform data-injection at the lowest level. The dependence of the hardware stack is investigated by performing the benchmark programs on two different cluster, i.e. HZDR – hypnos [3] and NIIF - debrecen2 [4].
At the end of the report software design guidelines are formulated allowing future software developers to select the most optimal software architecture for their use case.

Test platforms
For measuring the bandwidth and latency of the network interconnects two HPC centers, NIIF – debrecen2 and HZDR – hypnos were chosen. The benchmarks were performed with two different software performances tests, i.e. OSU and perftest. The software and hardware configuration are given below.
HPC center – NIIF
Software components:
CUDA 8.0.61
NVIDIA Driver 375.26
Mellanox Driver 2.33.5000
OpenMPI 1.8.5
Red Hat Enterprise Linux Server release 6.8 (Santiago)
Hardware components:
GPUs: 3 x Tesla K20Xm, 6GB
Network Interface card: Mellanox Technologies MT27600
CPU: 2 x Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60GHz
System Memory: 12 GB RAM

HPC center – HZDR
Software components:
CUDA 8.0
NVIDIA Driver 367.48
Mellanox Driver 2.33.8000
OpenMPI 1.8.6
Ubuntu 14.04.5 LTS
Hardware components:
GPUs: 4 x Tesla K20Xm, 4GB
Network Interface card: Mellanox Technologies MT27500 Family [ConnectX-3]
CPU: 2 x Intel(R) Xeon(R) CPU E5-2609 0 @ 2.40GHz
System Memory: 64 GB RAM

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 654220

Data transfer methods

For testing the data transfer performance of HPC centers two different kind of data sources and targets are tested. One of them is the 'Host to Host' transfer. In this case data transfer is performed between the system memories of two compute nodes. As shown in the Figure 1 the data transfer (write or read) goes from the memory of Node A to the memory of Node B through nodes NIC (red arrows).

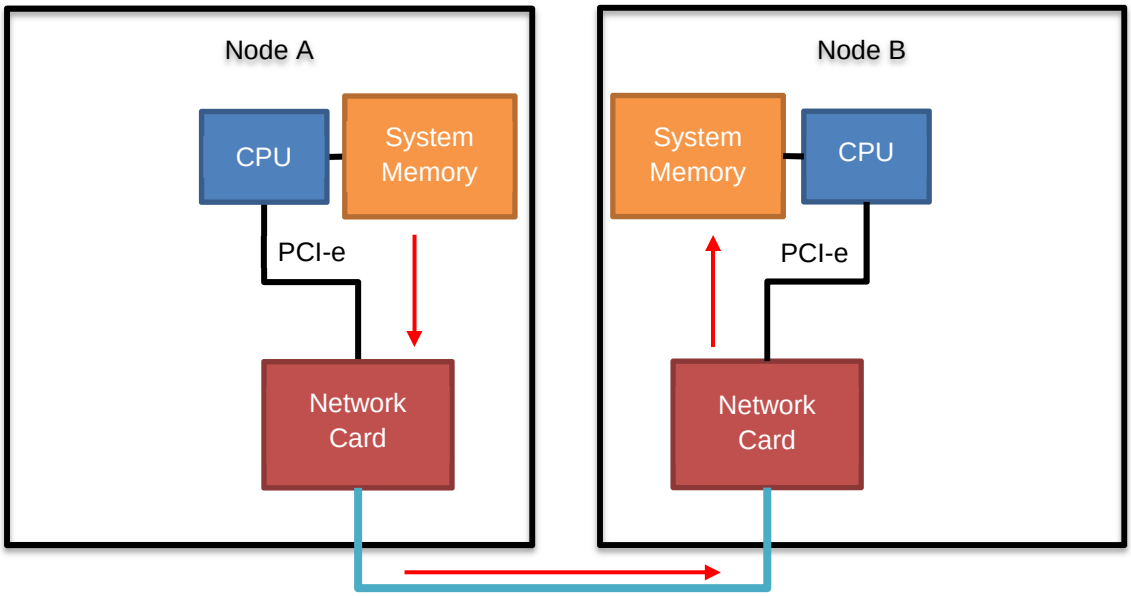


Figure 1 Data flow of Host to Host data transfer

The other tests are the Device to Device tests, where the data goes from the memory of a PCI-e device of one node to the memory of a PCI-e device of another node. In our test environments the considered devices are NVIDIA GPUs, see Figure 2. The main advantage here is that during the data transfer process the System memory is not accessed and data passes the PCI-e buss only for one time.

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 654220

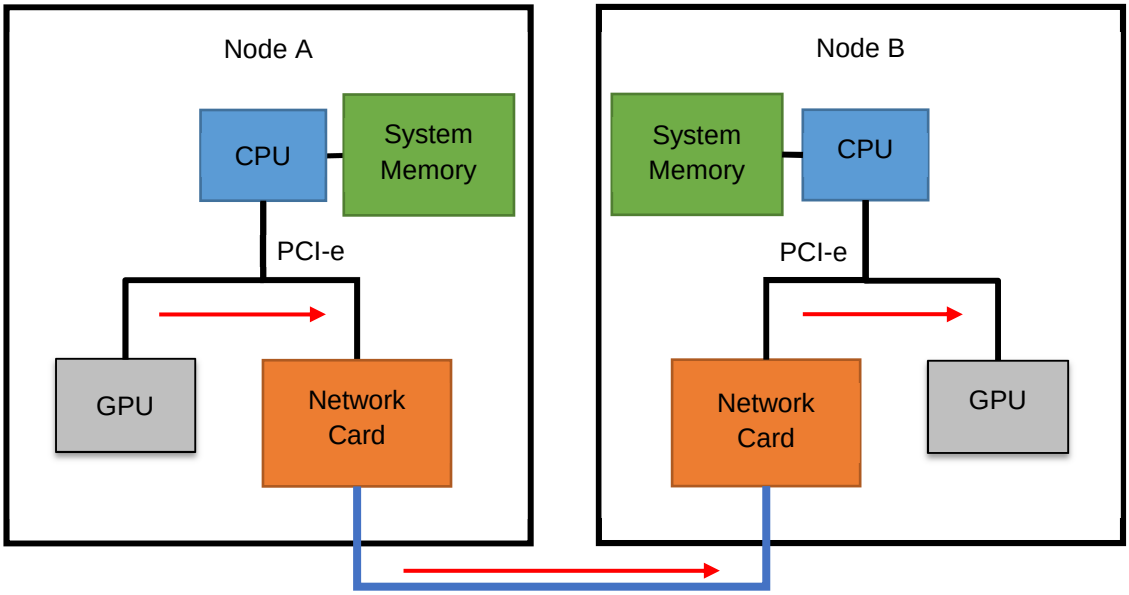


Figure 2 Data flow for Device to Device data transfer

OSU Micro-Benchmarks

The OSU Micro-Benchmark (OMB) software package is tool for testing bandwidth and latency for network connection (http://mvapich.cse.ohio-state.edu/benchmarks/).

Host to host benchmarks

Two types of tests are performed on the different HPC centers. One of them was the bandwidth test (osu_bw) and the other was the latency test (osu_latency).

In bandwidth tests the sender sends back-to-back messages to the receiver and then waits for a reply from the receiver. The receiver sends the reply only after receiving all these messages. This process is repeated for several times and the bandwidth is calculated based on the elapsed time and the number of bytes sent by the sender [1].

The bandwidth for lower message size (size smaller than 8 KB) is 1.5 times higher for the HZDR (Figure 3), however the maximum bandwidth is nearly identical. A possible explanation for this behavior may be caused by driver differences.

In latency tests the sender sends a message to the receiver and waits for the response from receiver. The receiver sends back a message after receiving. After many iteration of this ping-pong tests the average latency values are calculated ([1]).

We discovered an analogous behavior at NIIF for small message sizes (smaller than 32 bytes). The exact reason at the time of writing this report is unknown but is currently under investigation.

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 654220

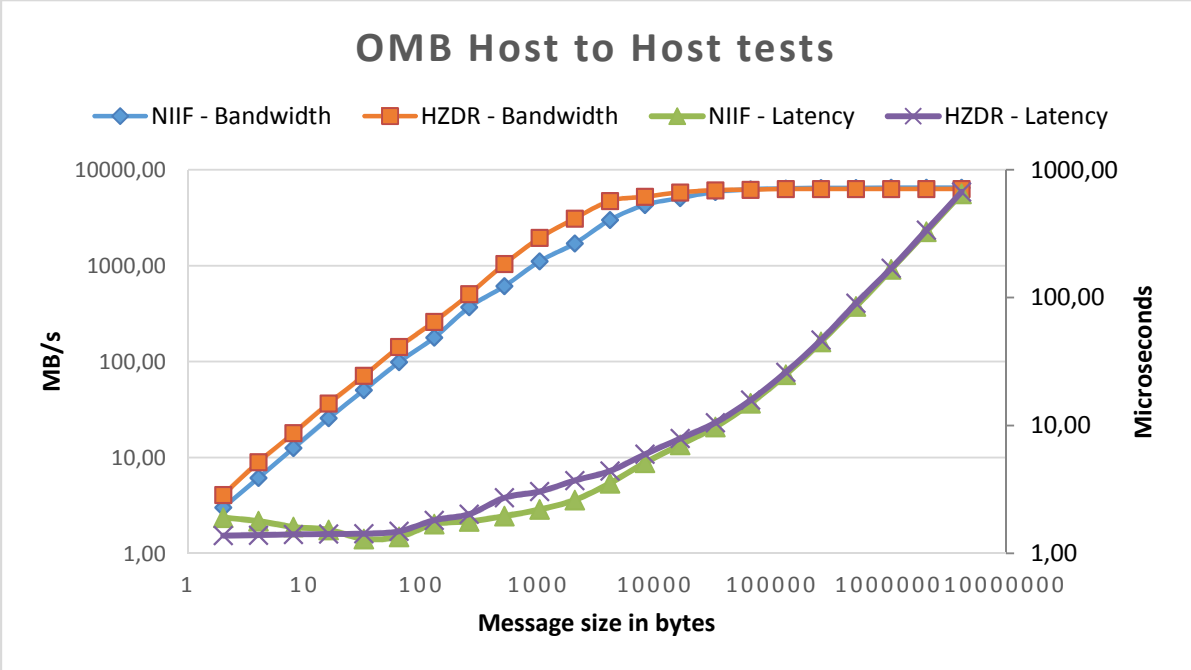


Figure 3 OSU Micro-Benchmarks Host to Host tests at NIIF and HZDR

Device to device benchmarks

The GPU to GPU message passing with CUDA-RDMA could be performed at the HZDR. At NIFF currently the Mellanox OFED GPUDirect RDMA package is not (yet) available. Similar to the host to host benchmark, bandwidth and latency tests were carried out.

In the bandwidth tests between device to device data transfer another interesting behaviour was observed. For message sizes between 1 KiB and 16 KiB the increment of bandwidth speed slows down and at 32 KiB the bandwidth rises significantly (3 times larger than at previous message size).

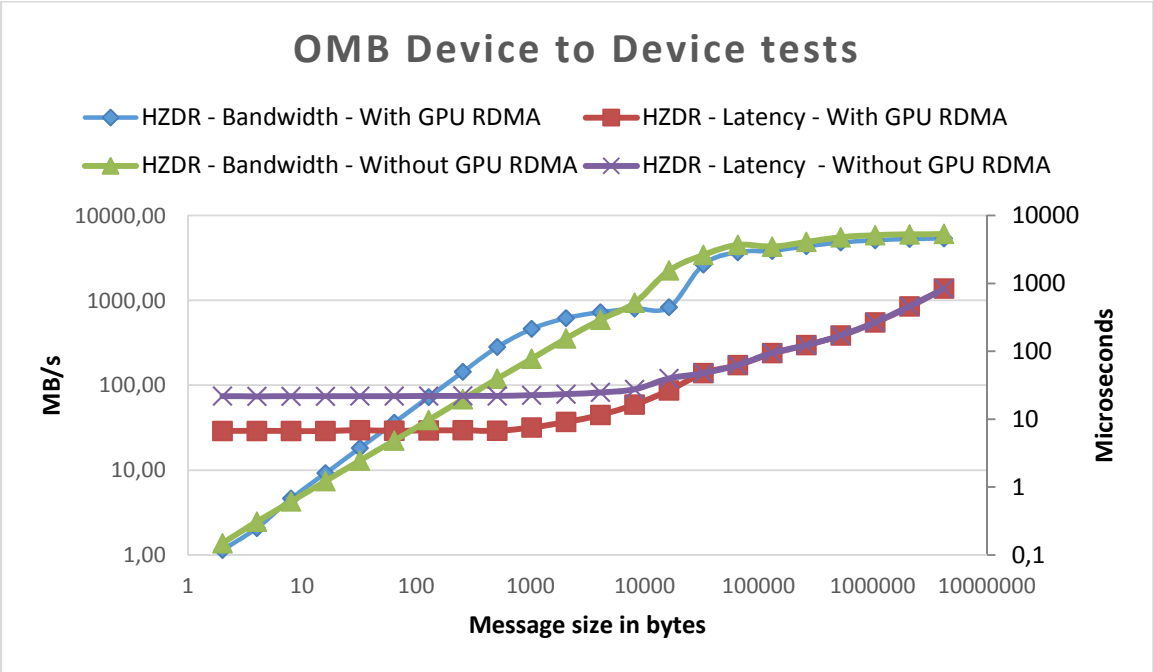


Figure 4 OSU Micro-Benchmarks Device to Device tests at HZDR

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 654220

Conclusion

The OSU Micro-Benchmarks tests have shown the power of high bandwidth interconnections and by using GPU Direct RDMA technology of NVIDIA the GPU based computation can benefit from this interconnection as well. By using GPU RDMA latency is reduced for smaller message size (smaller than 32 KiB)

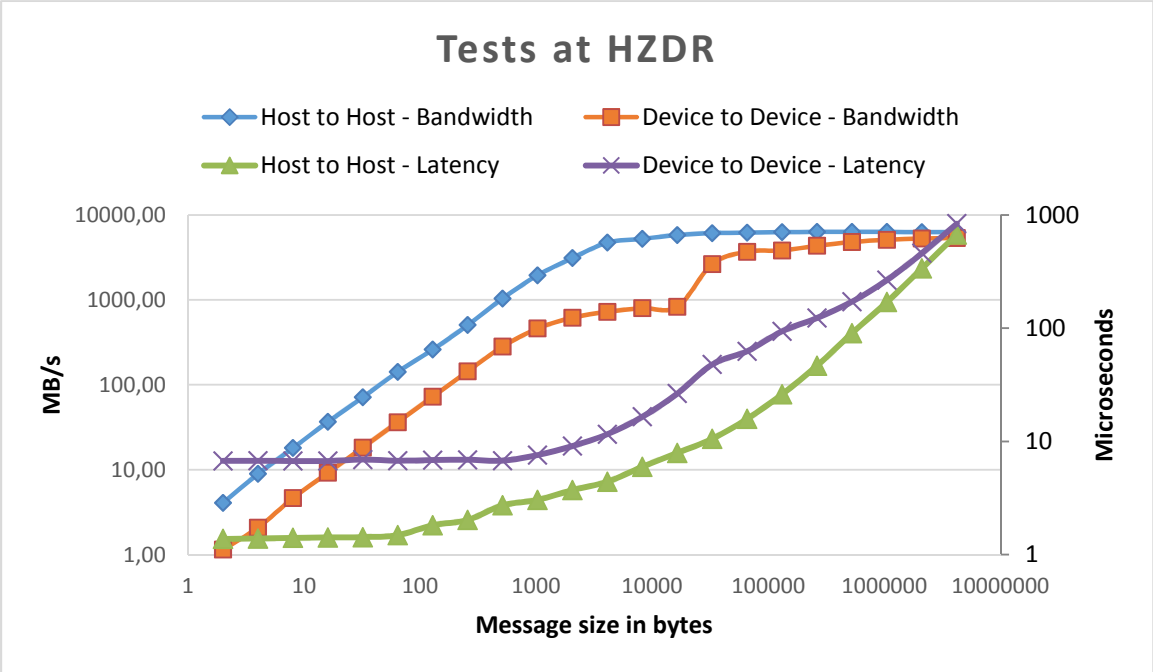


Figure 5 OSU Micro-Benchmark tests for HZDR

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 654220

'Perftest'

A second benchmark software was used to confirm the results of the OMB tests. The 'perftest' is a software tool to test network performance. This tool was selected because of the performance test of NVIDIA for modern HPC platforms [5]. The perftest benchmark uses the directly ibverbs library [6] to send and receive messages with RDMA [7], unlike at OMB tests, where RDMA is supported through Open MPI [8].

Host to host

Two types of tests were used similarly to the OMB tests, the bandwidth (ib_send_bw) and the latency (ib_send_lat).

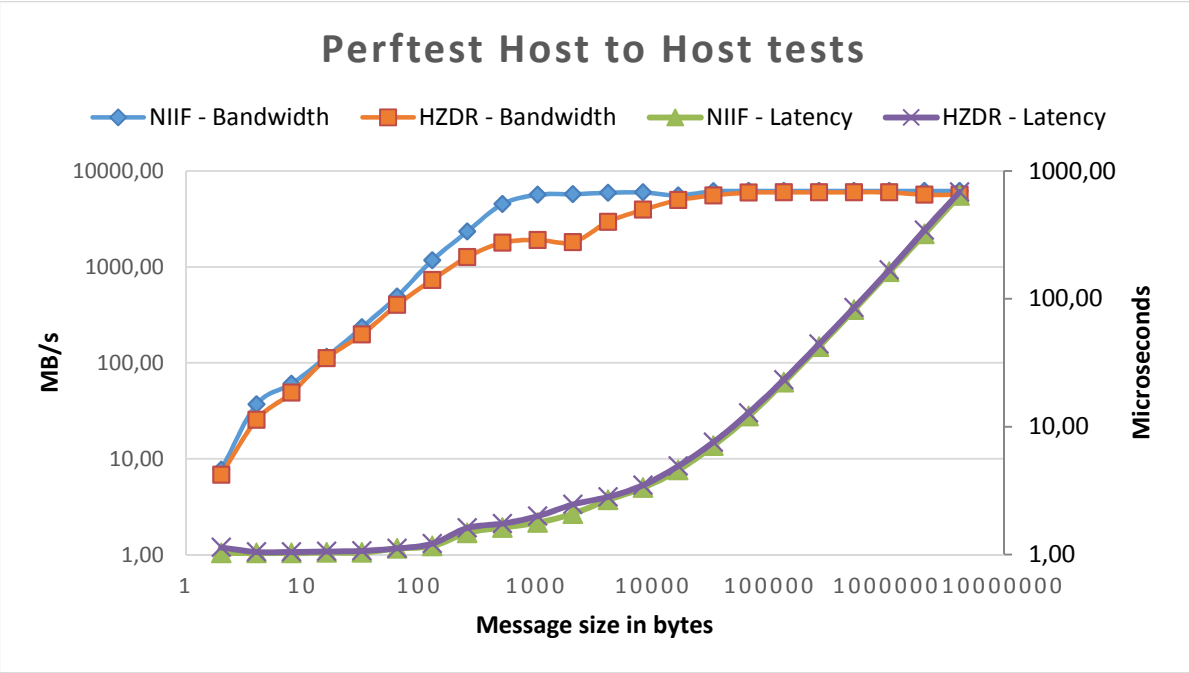


Figure 6 'Perftest' tests Host to Host at NIIF and HZDR

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 654220

Device to device

Only at the compute nodes at HZDR we could perform the GPU to GPU tests using 'perftest' benchmark since the package supports only bandwidth tests with CUDA GPU Direct RDMA. This is unfortunately not yet implemented at NIIF.

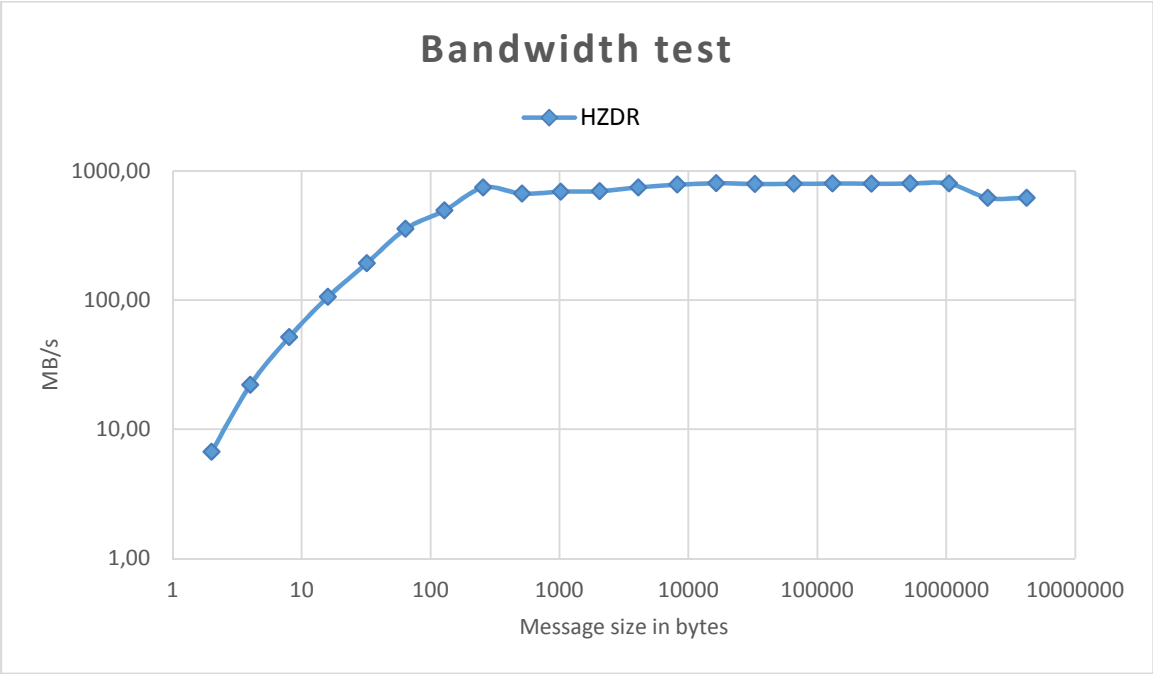


Figure 7 'Perftest' bandwidth test Device to Device at HZDR

Conclusion

The results confirm the outcome of the OMB between host to host, but at the device to device tests an unexpected behavior was discovered. Currently the poor performance of GPU to GPU tests is under investigation.

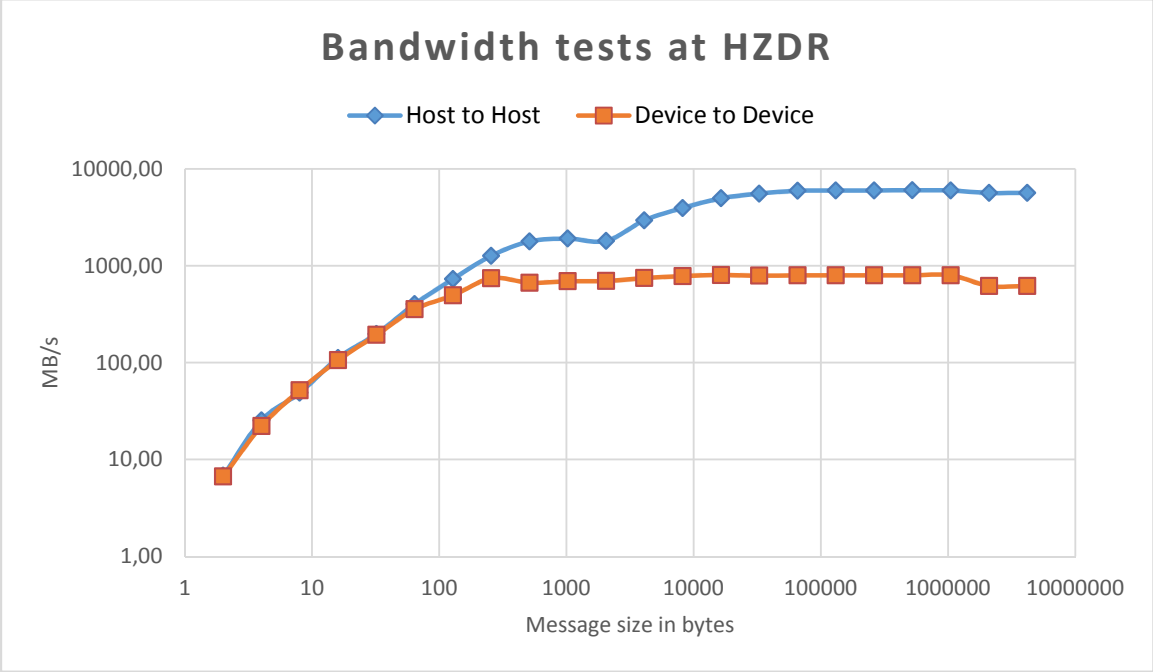


Figure 8 'Perftest' bandwidth tests at HZDR

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 654220

Design guidelines based on conclusion

The results of Host to Host tests show that the Open MPI based data transfer has lower bandwidth than the low-level ibverbs based data transfer for smaller package sizes (size smaller than 16 KiB), however the maximum bandwidth is very similar for both transfer types. Concerning latency, the tests show that the Open MPI based solution has a higher latency than the ibverbs based method. The optimal bandwidth of Host to Host data transfer for Open MPI based solution is found for 16 KiB and 2KiB for ibverbs.

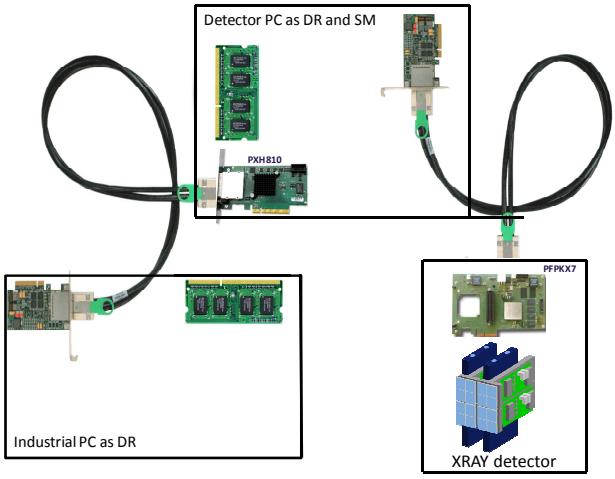
The Device to Device tests show that the maximum bandwidth is smaller 6317 MB/s vs. 5388 MB/s and latency is higher with the Open MPI based solution. The optimal message size is quite large, i.e. 1 MiB. The OMB tests for Device to Device with and without enabling GPU RDMA showed (Figure 4) that the use of a CUDA based solution is worth, especially for smaller message size (smaller than 32 KiB) where latency can be three times smaller and bandwidth can be ten times larger. For the ibverbs based solution an anomaly was found and therefore the peek bandwidth is significantly smaller 6032 MB/s vs. 835 MB/s.

One may note that the development process for Open MPI based solutions is less complicated for Host to Host and for Device to Device (CUDA based solution) and the Open MPI is well known for developers. Therefore, we would recommend, based on the current results, the use of the Open MPI based solution if the message size reach the optimum level of the Open MPI based transfer (16 KiB) and latency is not critical. For GPU based computational tasks it is recommended to use the Open MPI based solution. The low-level ibverbs based solution is preferable, when the message size is small and latency is critical.

Currently we are investigating the reason behind the poor bandwidth with ibverbs Device to Device data transfer and the strange break of the growth of bandwidth with Device to Device tests.

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 654220

RASHPA is an integrated hardware and software solution for scalable data transport and injection via PCIe. A first prototype has been developed within EUCALL WP5 UFDAC:



In the hardware prototype depicted above, RASHPA has been integrated to the smartpix, a medipix3 based detector currently under development at the ESRF. A commercial board, PFPKX7 from Techway was used to implement RASHPA. The PFPKX7 is a Kinext-7 PCIe FPGA board.
The prototype supports multiple destinations feature thanks to the use of PCIe switch. Copper cables were used to build and test routable network although fibre optics cabling is also available. In this implementation, two types of PC are used: a so-called detector PC having 64GB of internal DDR, and Gen3x16 PCIe endpoint, and an industrial PC with only 4GB on internal DDR and Gen1x1 PCIe endpoint. The PFPKX7 supports Gen2x4 PCIe, thus the link connecting the detector to the detector PC is negotiated to Gen2x4 whereas the virtual link connecting the detector to the industrial PC is limited to Gen1x1.

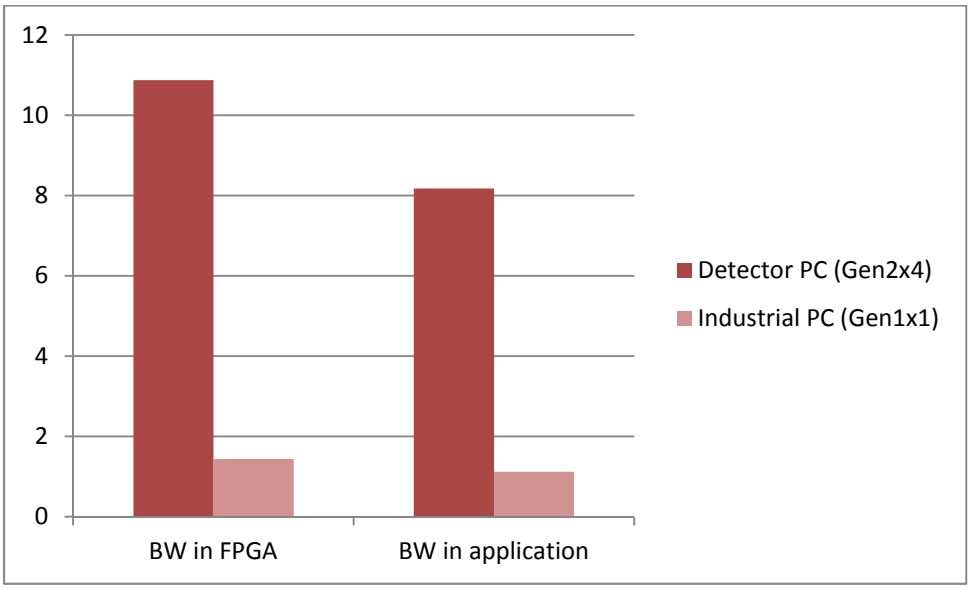


Figure 9 Data throughput in Gbps internally within the PFPKX7 FPGA (left) and inside the application (right)

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 654220

Figure 9 shows the data throughput of the data acquisition process in Gbps. Two ways were investigated to measure the data throughput. The first one measures the throughput internally within the FPGA, whereas the second one detects whenever data are available to the application.

Measured throughput when Gen2x4 PCIe endpoint is the target is 10.88Gbps (68% of the Gen2x4 maximum bandwidth) when measured within the FPGA and 8.18Gbps (51.15%) when measured at the application level. Similarly measured bandwidth at the Gen1x1 is 1.44Gbps (71.99%) and 1.12Gbps (56.28%) when measured at the FPGA and application level respectively.

These losses in the data throughput are due to the con configuration of the DMA which is imposed by RASHPA as well as the restriction of the PCIe packet size limited to 4Kbytes. Some other latency can be added when throughput is measured at the application level due to the processor execution time.

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 654220

The CRACEN solution aims at providing a resilient and scalable solution for data transfer and injection that allows for optimizing communication based on topologies, supports load balancing and a variety of communication software stacks. Currently, UDP, TCP and MPI are implemented as back ends for communication, which can be used and mixed in one application with a single communication interface.

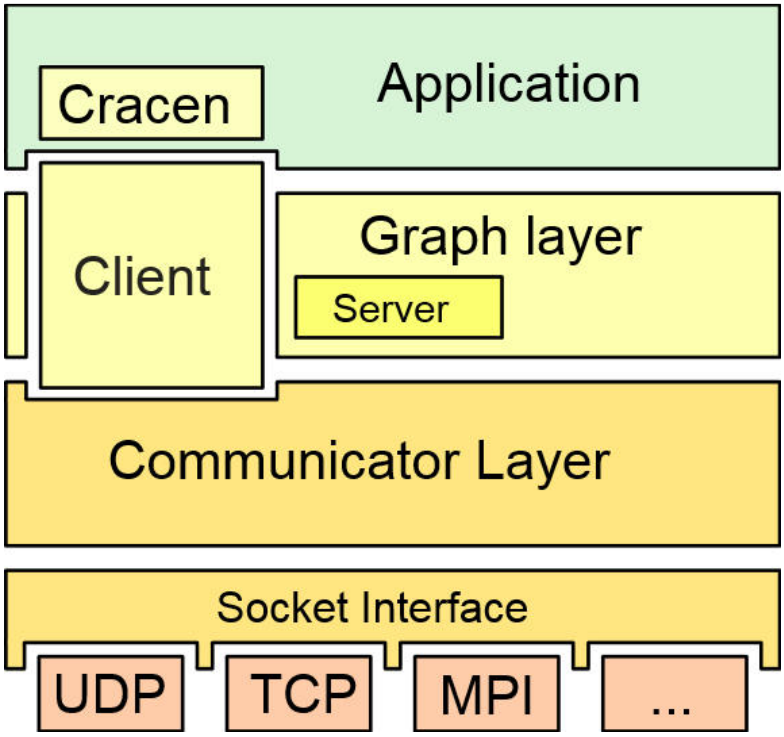


Figure 10 The CRACEN design includes a communciation layer on top of a socket interface and a graph layer for describinf communication topologies and their mappings to network topologies

CRACEN consists at its core of a single communication layer placed on top of a socket interface realized in a variety of back ends, see Figure 10.

CRACEN allows for inherent serialization and deserialization of data. Communication within applications is described via graph-based topologies that can be mapped to existing interconnect topologies to optimize throughput. As such, the communication logic inherent to a certain software solution can be optimized for a chosen underlying hardware interconnect solution and is scalable between hardware solutions.

The basis for both load balancing and resilience lies within parallel buffering streaming data between data analysis steps, allowing to describe any application by a graph-interconnected succession of tasks, see Figure 11. Oversubscription and buffering allows for scalability, load balancing and resilience at the same time.

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 654220

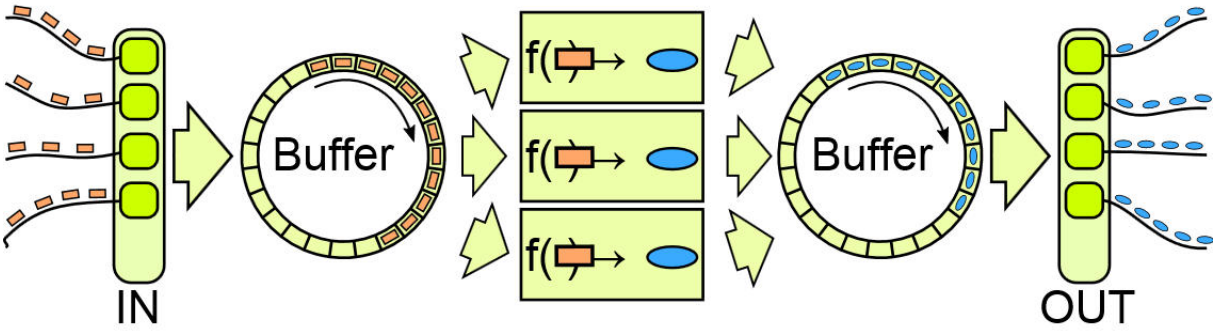


Figure 11 Applications using CRACEN can describe data transfer and analysis by a graph-interconnected sequence of tasks with inherent buffering

Scalability in terms of throughput is linear as seen in *Figure 12*. This of course depends on the underlying hardware configuration and selected communication topology, but the data shows that the CRACEN software stack does not prevent linear strong scaling.

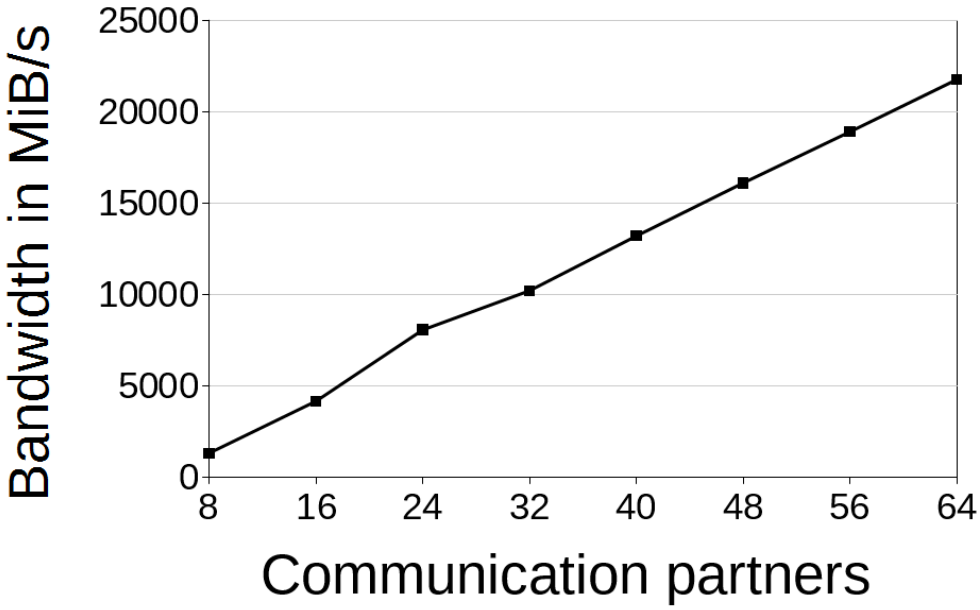


Figure 12 Strong scaling of throughput shows that throughput scales linearly with the number of communication partners

Resilience and load balancing are achieved by asynchronous communication and penalty-based scheduling of communication operations depending on local buffer size and response time for each communication partner within the topology. With close to empty buffer state and quick response time communication partners receive more data than partners with close to full buffer states and long response times. This means that malfunctioning partners will not be provided with data.
Resilience was tested with respect to throughput as seen in Figure 13.

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 654220

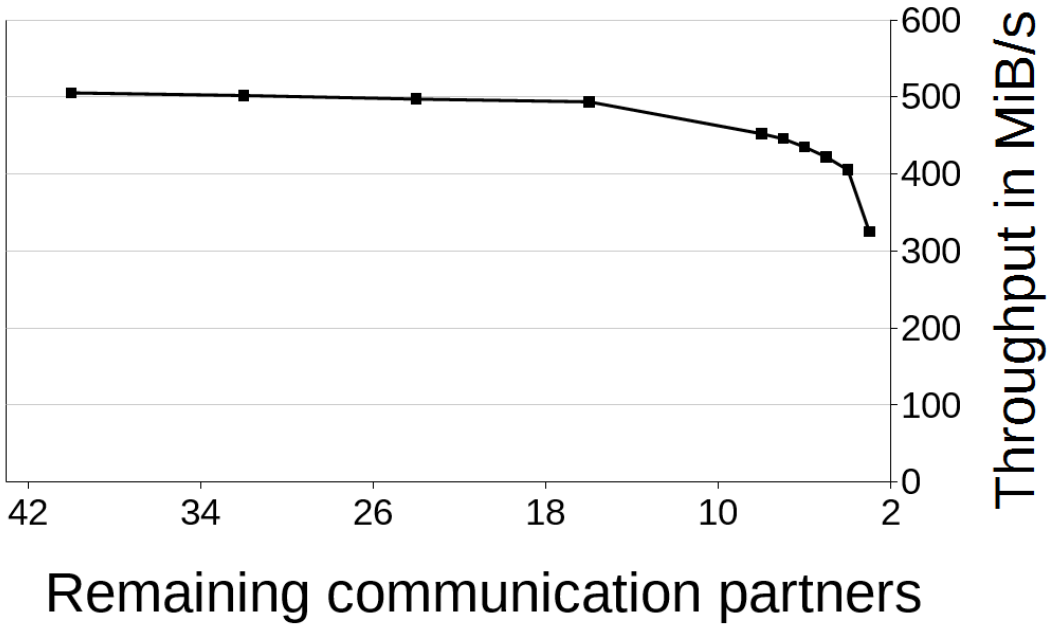


Figure 13 Resilience test. By intentionally decreasing the number of working communication partners throughput gradually decreases, as more partners experience failures.

CRACEN is available as Open Source and will in a first test be used to scale the single-node based solution for the PSI Jungfrau calibration algorithm described in the following section to scale to more than one Jungfrau module.

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 654220

The work done at PSI is a crossover with the imaging milestone MS 5.2, Evaluation of test case implementations for processing of digitizer data and image analysis. It evaluates the role of data transfer in optimizing the overall throughput of a GPU-based solution to the online Jungfrau detector calibration from signal to photon count rate developed together with HZDR.

A GPU-based algorithm for online conversion of data from dynamic gain switching charge integrating detectors such as AGIPD, Jungfrau, or Gotthard, into number of photons has been developed.

Hardware setup

The performance evaluation of this solution has been carried out on a server equipped with an Intel Xeon E5-2690 and two NVIDIA Tesla P100 GPUs. The link between the GPUs and the CPU is a dedicated PCIe 3.0 x16 per GPU. Figure 14 shows a description of the setup.
In order to evaluate the performance of the GPU-application in terms of throughput and latency, a set of test case data from experiments carried out with the JUNGFRAU detector has been used. This dataset is stored in the server disk.

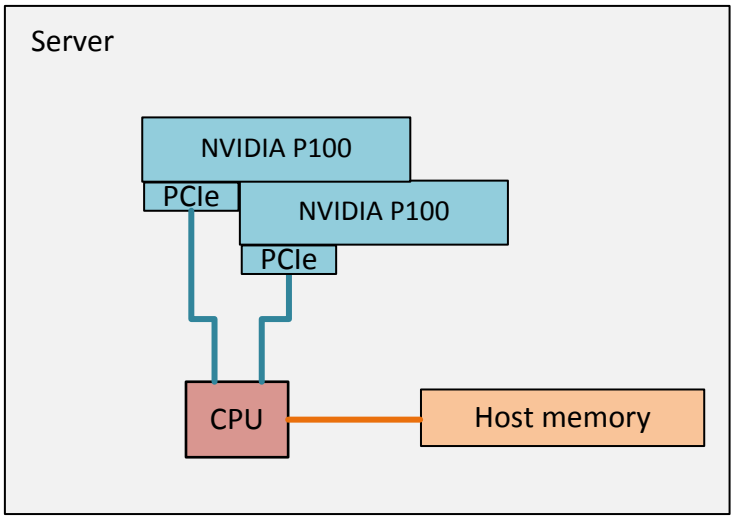


Figure 14 Server setup for the measuremnt

Software application

The GPU-based application converts the dynamic gain switching integrated charge from each pixel of the detector into the number of photons that hit the pixel. To do so, the application processes blocks of a determined number of full images (frames) at the time, that is, the received charge integrated data received from the detector is buffered until the specified number of full images are received and is processed at once. Once the processing has been performed, the converted data is sent back to the CPU for visualization of the converted data and/or storage.
Additionally, the application is able to update the values of pedestal correction for each pixel based on the measured values when the pixel has not been hit by any photon.

Performance evaluation

The GPU application consists of a main CUDA kernel that performs the conversion algorithm for every single pixel in addition to the update of the pedestal correction. As we have mentioned before,

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 654220

this processing can be carried out in processing blocks (sets of full images) of different sizes. Figure 15 shows how the CUDA kernel processing time increases linearly with the size of the processing block. Given the size of 1MB for a JUNGFRAU frame (1024x512 pixel), we can see that the current CUDA kernel is able to process 1GB of data in less than 20ms. The execution time of the CUDA kernel is not influenced significantly by the number of CUDA streams. The same applies to the number of GPUs, provided that the same block size is processed by each GPU.

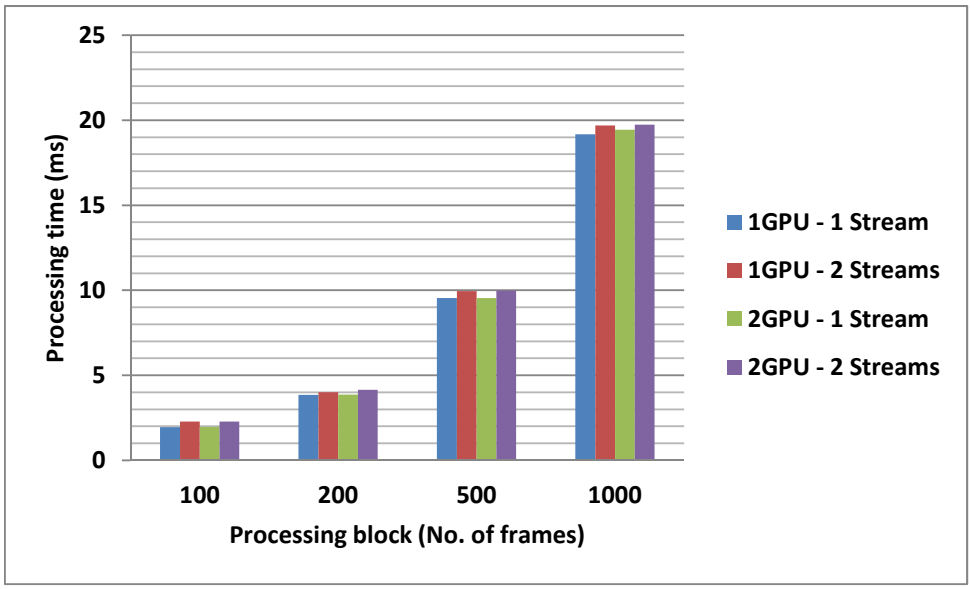


Figure 15 CUDA kernel processing time for different processing block sizes and setup configurations

In terms of data throughput, **Error! Reference source not found.**, we can see how it scales up almost linearly with the number of GPUs used, and this relationship becomes more linear as the size of the processing block increases.

From Figure 16 we can also see that the maximum throughput is bandwidth-limited by the PCIe 3.0x16 link, which is close to the limit for the data transfers from the CPU to the GPUs (around 10GBps). This means that there is still room to perform additional computations on the received data apart from the conversion and pedestal correction the CUDA kernel currently carries out.

From the results we do not see any improvement when using more than one processing stream, which should boost the overall data throughput. We believe that the reason is that the GPU application has not been adapted to the new PASCAL architecture (and the new CUDA unified memory) to benefit of the dual DMA engine that should allow us to overlap the kernel execution with memory transfers from CPU-to-GPU and GPU-to-CPU.

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 654220

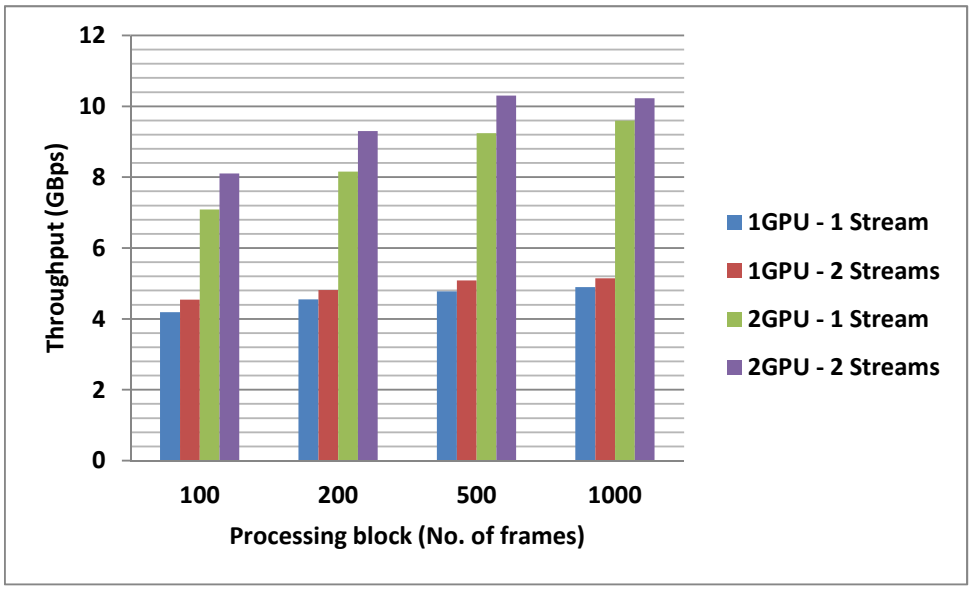


Figure 16 Throughput for different processing block sizes and setup configurations

On the other side, the higher throughput of greater processing blocks translates into greater processing latencies as Figure 17 shows. From the experimental results the maximum processing latency increases linearly with the volume of data processed at once and it is not impacted by the use of one or more GPUs. However, as it happens with the maximum throughput, we do not benefit from using multiple processing streams that should allow us to reduce the latency significantly.

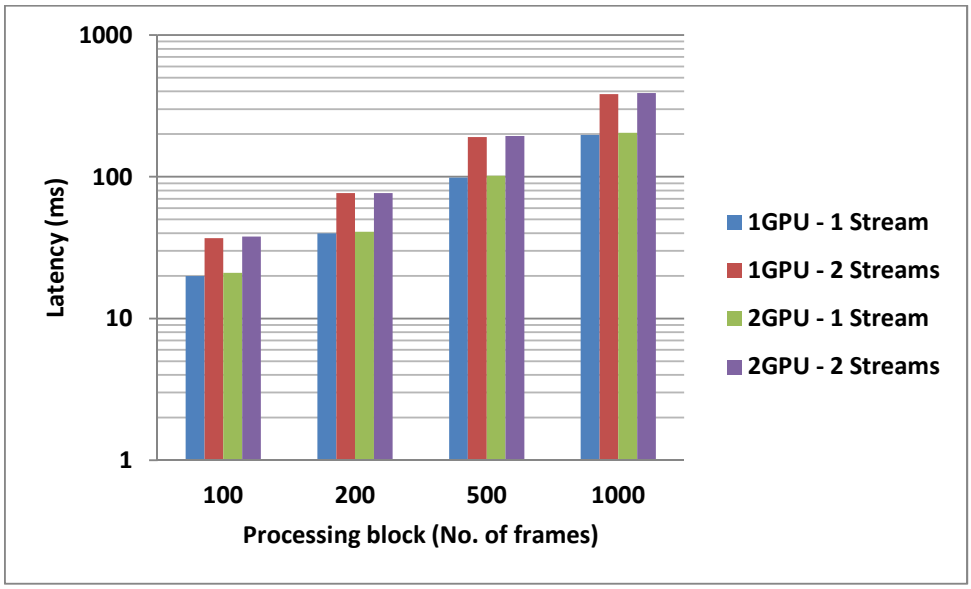


Figure 17 Maximum processing latency for different processing block sizes and setup configurations

It is worth mentioning that tests carried out in KEPLER architectures showed the benefit from using 2 different processing streams to increase the overall throughput and reduce the maximum latency. For this reason, it is necessary to adapt the GPU application to the new PASCAL architecture.

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 654220

4 Synergy Aspects

The work done at ELI-ALPS in collaboration with HZDR is a comprehensive write up of measurements of data throughput and latency for CUDA-based NVIDIA GPUs and their respective host systems as well as of the Infiniband interconnect. Such configurations are a prototypical and very general solution to high data rate imaging solutions. As such the results given here are of extreme importance also for high data rate applications at light sources.

RASHPA is a versatile PCIe based solution that can be adapted by other RIs. As it is already prototyped using a smartpix detector used also at other RIs, portability between RIs is ensured. It is scalable and its hardware stack is based on the widely available PCIe interface.

The PSI application of Jungfrau detector calibration developed together with HZDR is a good example of the interplay between data transfer and analysis that lies at the heart of UFDAC, viewing ultrafast data acquisition as an integrated task of transfer and analysis of data. The solution developed here is currently being evaluated by ELI-ALPS for possible reuse.

The HZDR CRACEN solution aims at a scalable, resilient data transfer solution that is flexible in both communication topology and communication stack. It is GPL Open Source and can be adapted to work for different hardware solutions, with emphasis on DAQ chains connecting a high data rate experiment to a compute centre for online analysis. It is foreseen to use CRACEN for scaling the PSI detector calibration presented here to 32 Jungfrau modules using CRACEN and to test it at ELI-ALPS.

The various applications presented here also shows the different aspects of data transfer and injection with GPU and FPGA based solutions, giving a comprehensive overview of the various setups envisioned for such solutions at the various RIs.

5 Conclusions

We present four prototype setups for determining efficient data transfer and injection, one using FPGAs (ESRF), two on GPUs (ELI-ALPS, HZDR) and one that uses a frontend FPGA with subsequent GPU-based analysis (PSI).

The data throughput typically seen are close to what can be expected from the underlying hardware solution and are in the range of few Gbps with freely available hardware.

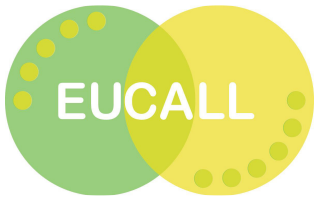
Latency can become an issue and has to be mitigated by more complex approaches that also increase scalability.

First studies of scalability suggest that scaling in terms of throughput produces new challenges and subsequent analysis is needed to better optimize scalability. Scaling in terms of portability is possible, but depends on the final throughput achievable with the software stack used and the underlying hardware.

Finally, resilience can be achieved for some communication software stacks, but especially those like MPI over Infiniband optimized for throughput and latency pose inherent problems for resilient solutions.

With INTEL Omnipath and IBM/NVIDIA Nvlink under development it is mandatory to stay open for future solutions, which is a great challenge for long-term development at current RIs.

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 654220



6 References

[1] “OSU Micro-Benchmarks,” 6 9 2017. [Online]. Available: http://mvapich.cse.ohio-state.edu/static/media/mvapich/README-OMB.txt.

[2] “perftest,” 7 9 2017. [Online]. Available: https://github.com/lsgunth/perftest/blob/master/README.

[3] “NIIF,” 7 9 2017. [Online]. Available: https://niif.hu/en/supercomputing.

[4] “HZDR,” 7 9 2017. [Online]. Available: https://www.hzdr.de/db/Cms?pOid=12231&pNid=852.

[5] “NVIDIA Accelerated Computing,” 7 9 2017. [Online]. Available: (https://devblogs.nvidia.com/parallelforall/benchmarking-gpudirect-rdma-on-modern-server-platforms/).

[6] “libibverbs,” [Online]. Available: http://www.rdmamojo.com/2012/05/18/libibverbs/. [Accessed 7 9 2017].

[7] “RDMA,” [Online]. Available: https://en.wikipedia.org/wiki/Remote_direct_memory_access. [Accessed 7 9 2017].

[8] “Open MPI,” [Online]. Available: https://www.open-mpi.org/. [Accessed 7 9 2017].

7 Publications

https://github.com/ComputationalRadiationPhysics/cracen2

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 654220