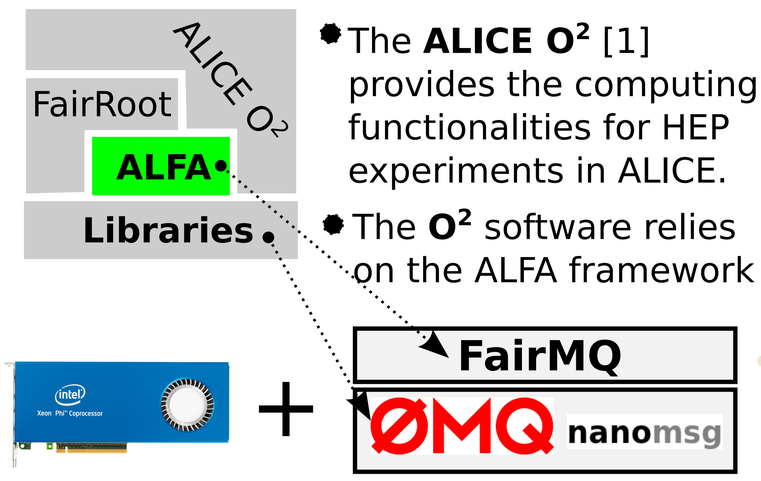
Optimizing the transport layer of the ALFA framework for the Intel® Xeon Phi coprocessor

Aram Santogidis
aram.santogidis@cern.ch

Introduction



- The **ALICE O²** [1] provides the computing functionalities for HEP experiments in ALICE.
- The **O²** software relies on the ALFA framework
- ALFA's transport relies on **ØMQ** and **nanomsg** messaging libraries

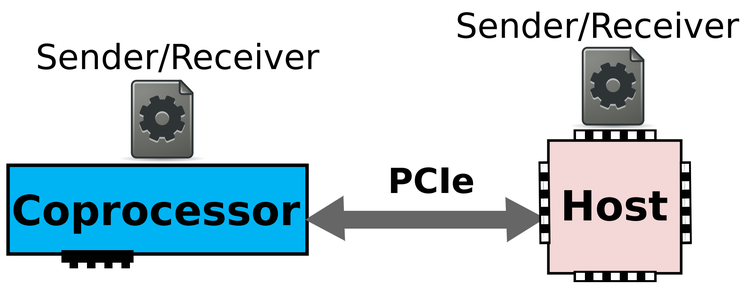
This research effort aims to asses the performance of the transport libraries of ALFA on the Intel Xeon Phi coprocessor and investigate optimization opportunities.

[1] Technical Design Report for the Upgrade of the Online–Offline Computing System The ALICE Collaboration

Methods

Two processes are executed, one on the host and one on the coprocessor.

- One *sender* and one *receiver*
- Transfer 1 GB payload
- Varying message sizes [64KB-128MB]
- We compare the average transfer rates



ZeroMQ and NanoMSG were cross-compiled for the Intel Xeon Phi architecture.

The libraries native benchmarks shipped with the code were used:

perf/**local_thr**	-	Receiver
perf/**remote_thr**	-	Sender

In order to measure the performance of the SCIF native transport protocol of Intel Xeon Phi coprocessor , the *SCIF-perf-bench* was developed:

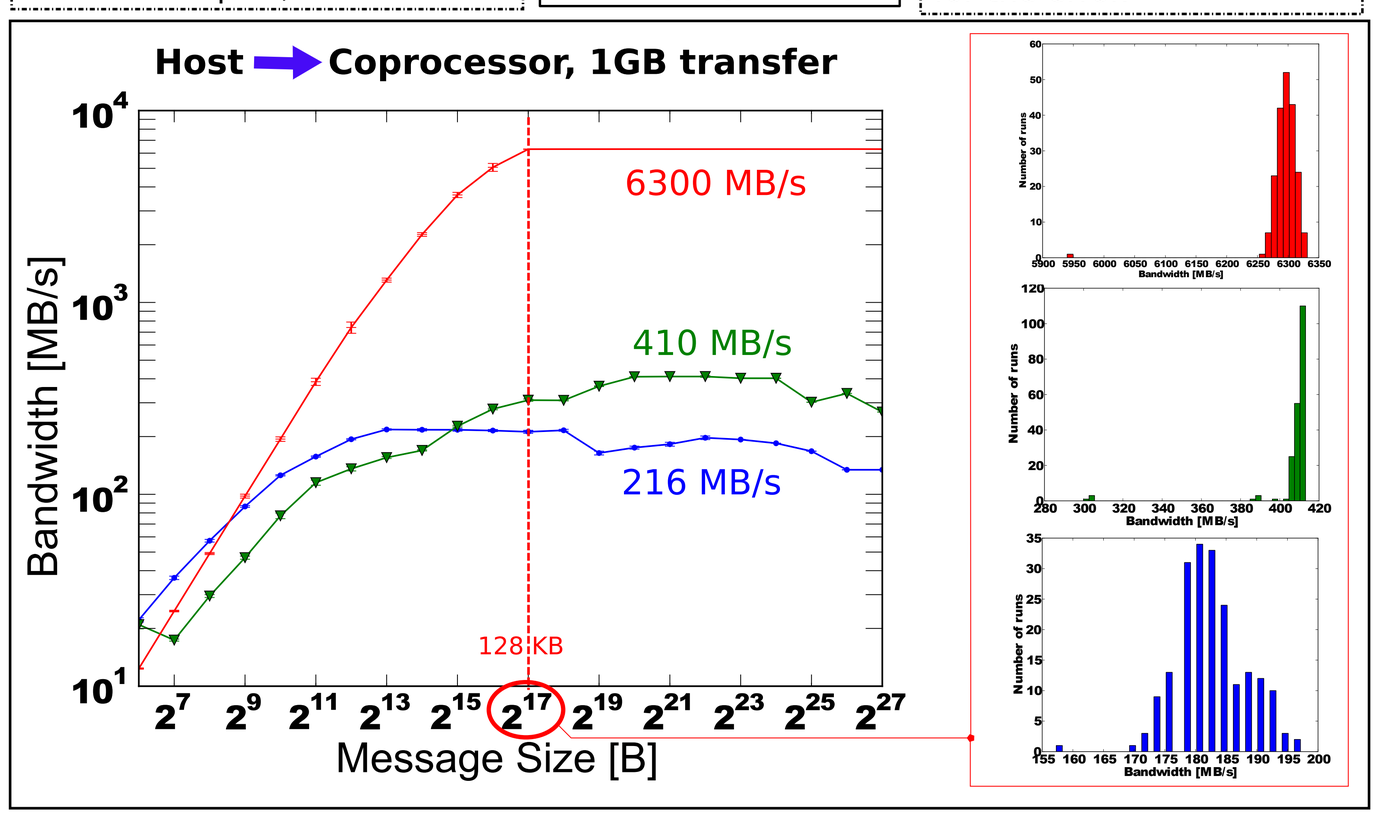
SCIF-perf-bench/**sink**	-	Receiver
SCIF-perf-bench/**source**	-	Sender

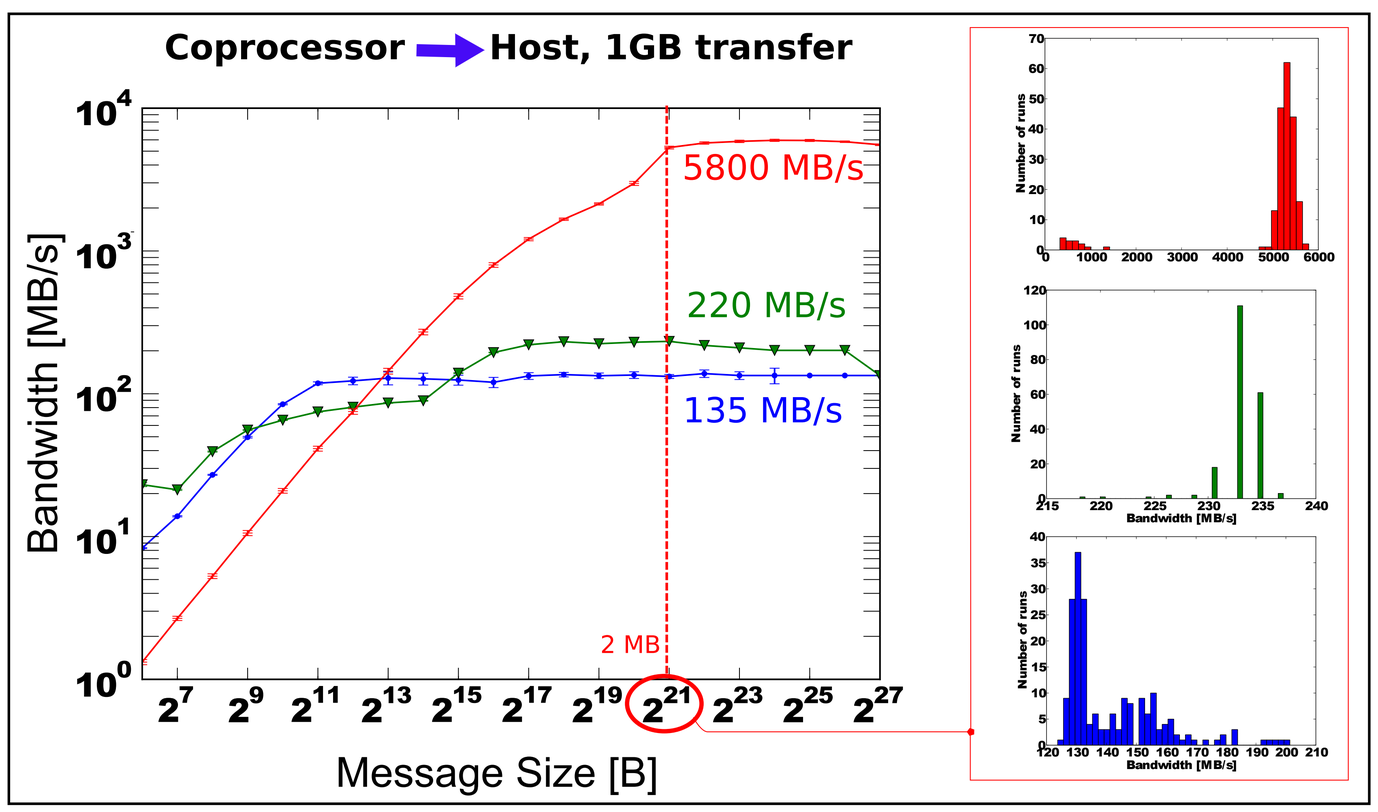
Results

We compared the performance of the libraries against the Intel Xeon Phi coprocessor's native transport, SCIF.

NanoMSG
ZeroMQ
SCIF-perf-bench

Median as Central Tendecy value (200 samples) and Median Absolute Deviation for the error.





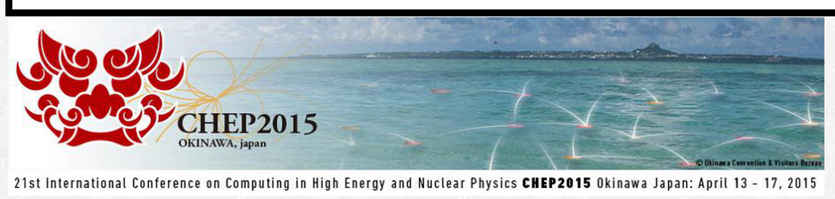
Conclusions and outlook

The performance gap between the libraries and SCIF, for message sizes in the range [1-100 MB], is one order of magnitude in favor of SCIF. This difference can be attributed to the fact that SCIF uses PCIe directly with DMA transfers whereas message libraries use the (single threaded) TCP/IP stack over PCIe with huge software overhead. These facts provide additional motiviation in implementing support for the SCIF protocol in ØMQ.

A high performance transport solution for ALFA on Intel Xeon Phi coprocessor will potentially increase the performance gains of porting complete ALFA devices that encapsulate computation intensive processes. We plan to investigate the possiblity of boosting the performance of such processes on the Intel Xeon Phi coprocessor by taking advantage of the vectorization and parallelization capabilities of the manycore platform.

Acknowledgments

Special thanks to my supervisors Dr. Andreas Hirstius from Intel GmbH and Prof. Spyros Lalis from University of Thessaly for their continuous feedback for this work. Also thanks and regards to Dr. Piotr Umiński from Intel Poland for his help to explain the results of the benchmarks. This research project has been supported by a Marie Curie Early European Industrial Doctorates Fellowship of the European Community's Seventh Framework Programme under contract number (PITN-GA-2012-316596-ICE-DIP)



http://goo.gl/4QeQzE

More information at [cern.ch/manycore]