Limago: an FPGA-based Open-Source 100 GbE TCP/IP Stack

Mario Ruiz[1], David Sidler[2], Gustavo Sutter[1], Gustavo Alonso[2] and Sergio López-Buedo[1,3]

[1]High-Performance Computing and Networking Research Group,
Autonomous University of Madrid, Spain
[2] Systems Group, Department of Computer Science, ETH Zürich, Switzerland
[3]NAUDIT High-Performance Computing and Networking, Spain
mario.ruiz@uam.es

HPCN High Performance Computing and Networking

Systems@ETH Zürich

naudit High Performance Computing and Networking

Motivation

- Network is becoming a bottleneck in current datacenter applications.
- New approaches are being explored to maximize the network efficiency and to tailor its functionality to the actual needs.
- In-network data processing.
- Network-attached paradigm.
- Provide a platform for further research in programmable networks.
- Starting point 10 Gbit/s stack by Sidler *et al.* [1]

[1] Sidler, David, et al. "Scalable 10Gbps TCP/IP stack architecture for reconfigurable hardware." *2015 IEEE 23rd Annual International Symposium on Field-Programmable Custom Computing Machines*. IEEE, 2015.

Challenges

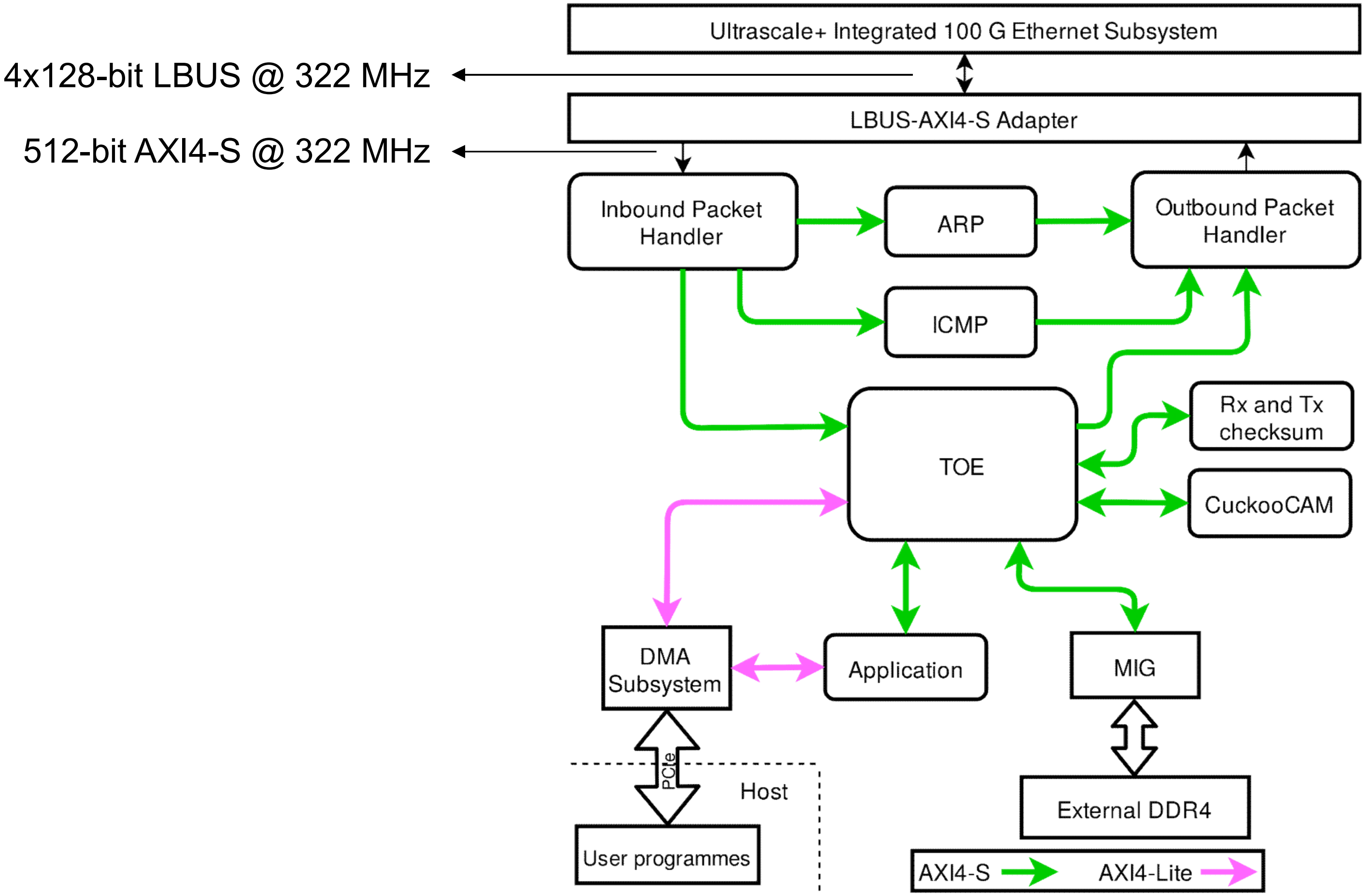
- Datapath 8x, clock frequency 2x
- Scalability with increasing network bandwidth.
- Flexible and high-productivity methodology. Vivado-HLS
- Widen applicability.
- Long Fat Pipe Issue.

$$RTT(s) \ \times LinkCapacity(b/s) > BufferSize(b)$$

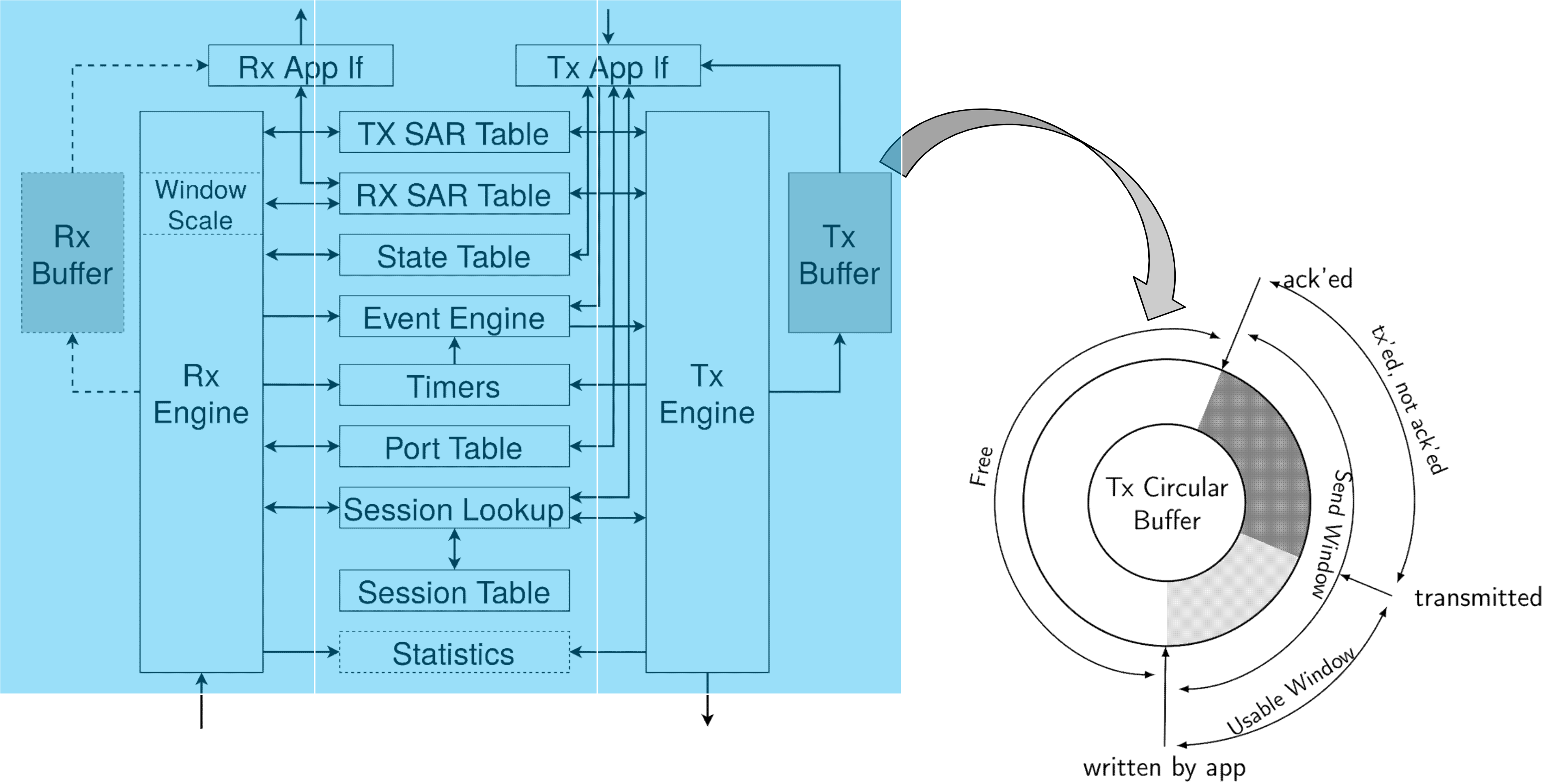
- One’s complement checksum [2].
- CAM. New design based on cuckoo hashing (HLS).
- DRAM bandwidth.

[2] Sutter, Gustavo, et al. "FPGA-based TCP/IP Checksum Offloading Engine for 100 Gbps Networks." *2018 International Conference on ReConFigurable Computing and FPGAs (ReConFig)*. IEEE, 2018.

Limago at a Glance

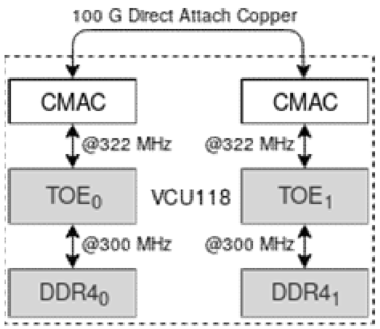


TOE

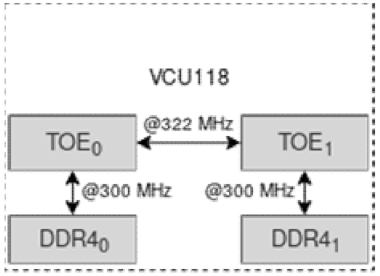

Rx App If
Tx App If
TX SAR Table
RX SAR Table
Window Scale
State Table
Event Engine
Timers
Port Table
Session Lookup
Session Table
Statistics
Rx Buffer
Rx Engine
Tx Engine
Tx Buffer
ack'ed
tx'ed, not ack'ed
Free
Tx Circular Buffer
Send Window
transmitted
Usable Window
written by app

Experiments

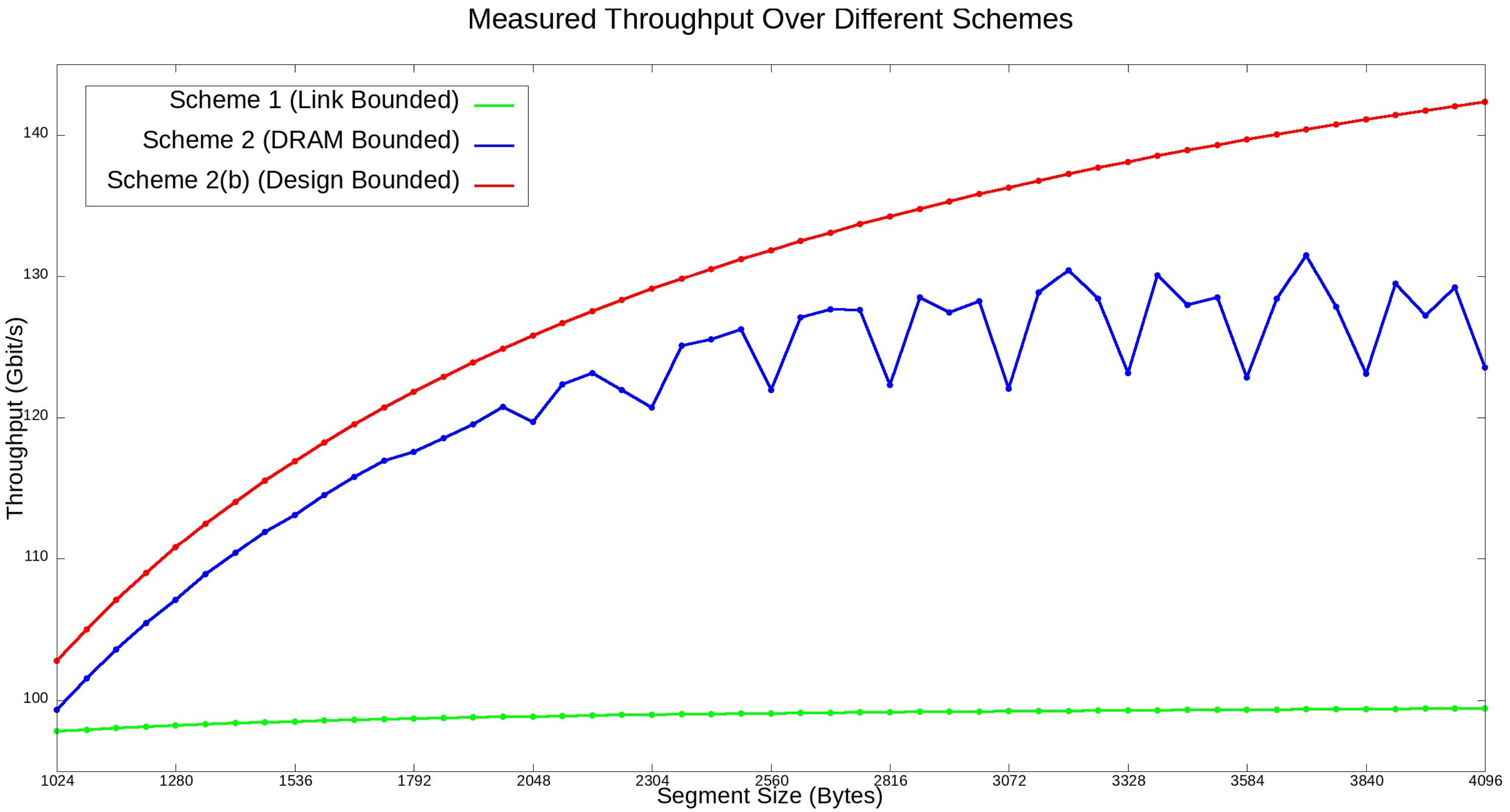
Limago to Limago (running iperf2 one connection)



a) Scheme 1

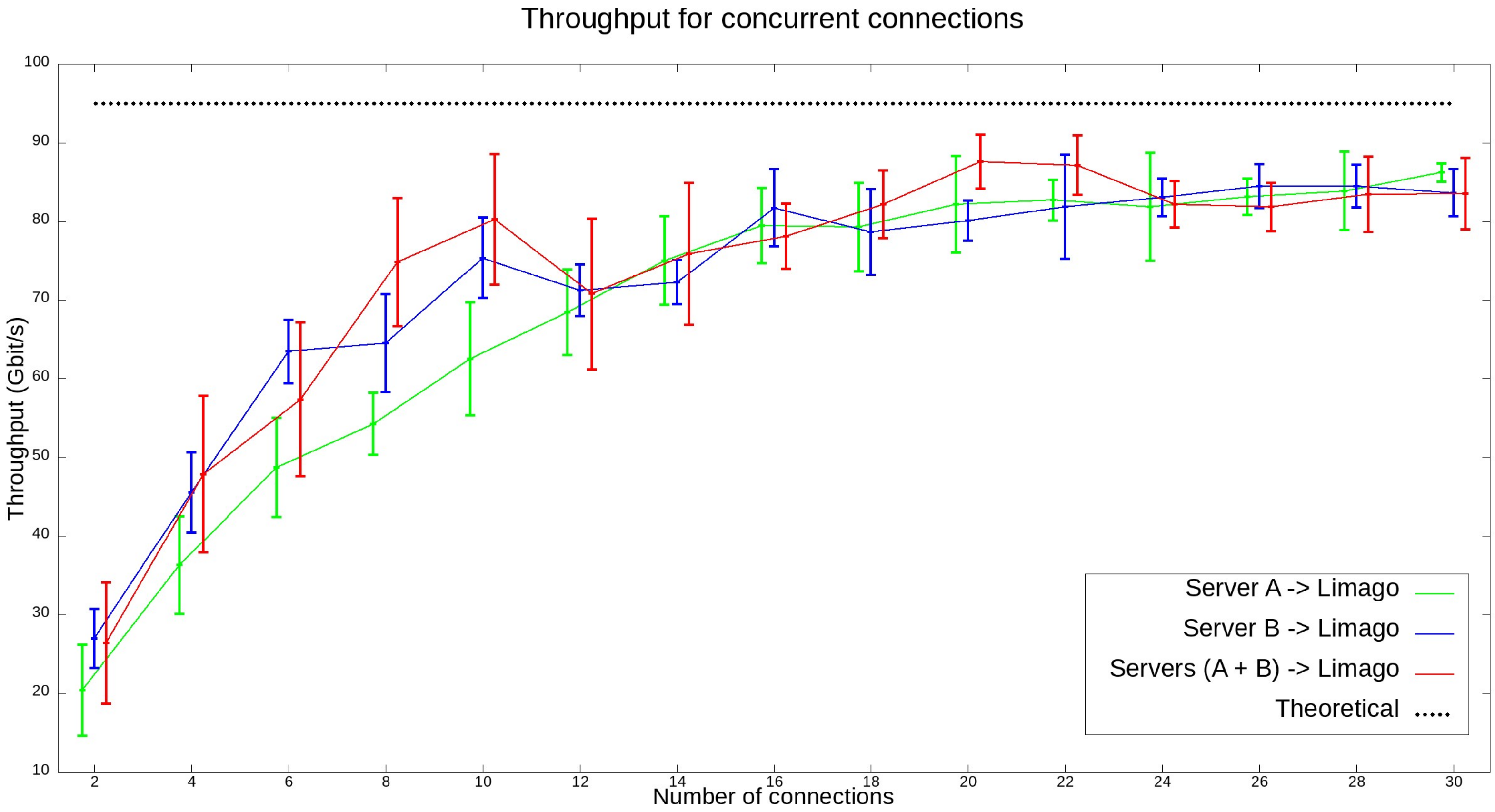


b) Scheme 2

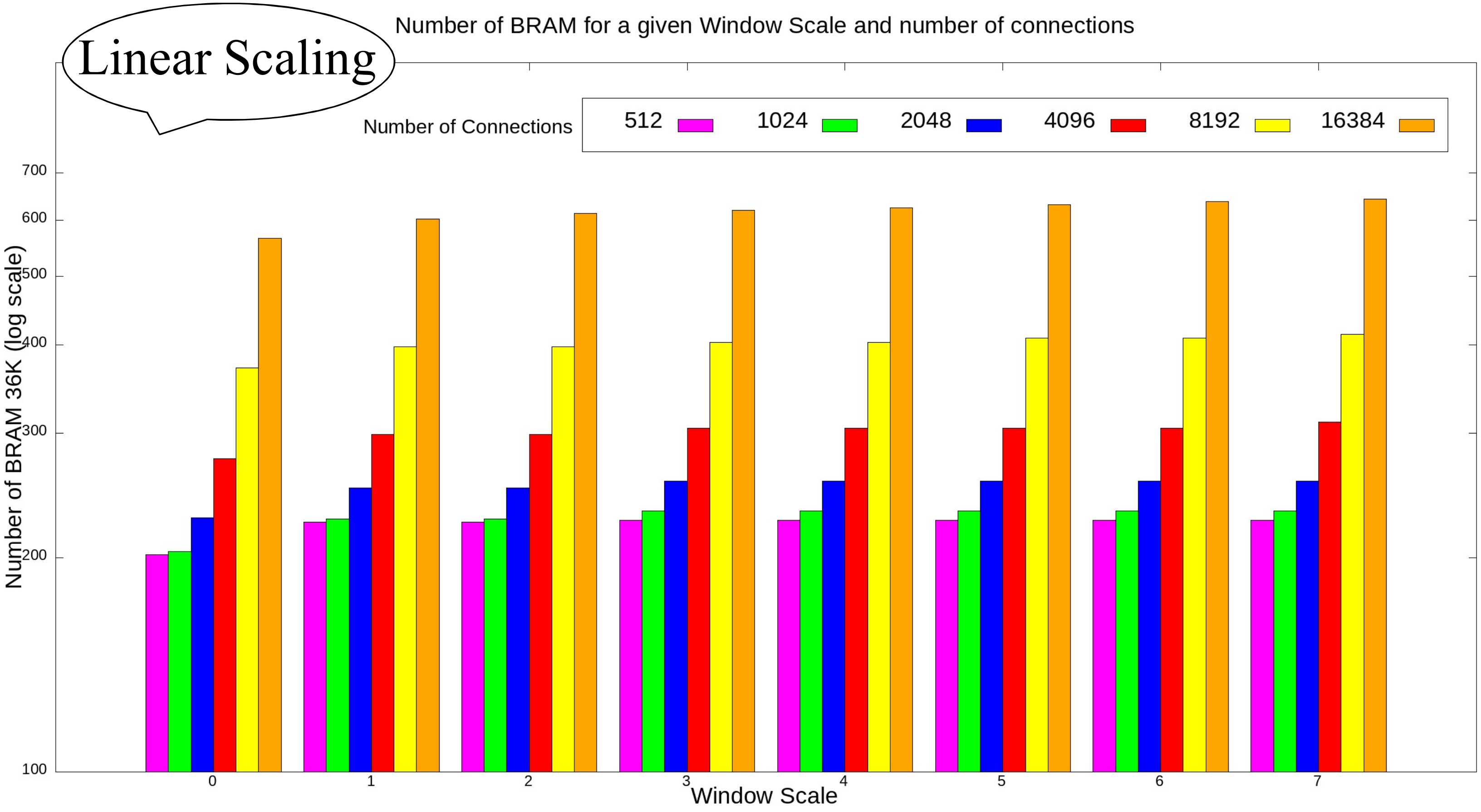


Experiments

Server(s) to Limago (running iperf2)



Resource consumption (TOE)



Conclusions

- ✓ Open-Source implementation.
- ✓ Support for multiple connections and Window Scale.
- ✓ Mostly written in C/C++ using Vivado-HLS.
- ✓ 7,456 lines of C/C++ and 1,482 lines of HDL.
- ✓ Future work includes support for packet reordering and selective acknowledgement (using HBM).

VCU118	LUT	FF	BRAM
10 G	6.6 %	3.6 %	17.1 %
100 G	10.1 %	7.5 %	20.4 %
Difference	1.55x	2.1x	1.2x

Just
20 % more BRAM
for 10x throughput

Visit our poster for further details

Check out our github!