Agile, eXtensible,
fast I/O Module
for the cyber-physical era

Brief introduction to the AXIOM SW stack

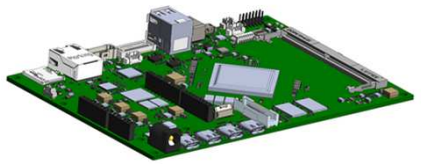
Paolo Gai, Bruno Morelli, Evidence Srl, Italy

{pj,bruno}@evidence.eu.com

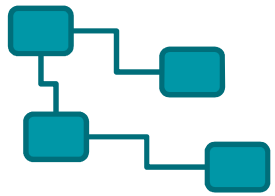
Applicazioni industriali e real-time su dispositivi eterogenei multi/many-core
T3Lab, Bologna, 18/10/2017

The AXIOM Project in one slide

- We are designing a small embedded board
  That bridges High Performance Computing (HPC)
  … and Cyber Physical Systems (CPS)



- We connect a set of boards together
  using high-speed transceivers of Xilinx Zynq Ultrascale+
  … RDMA for fast transfers!



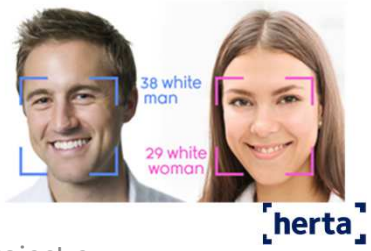
Cluster setup!

- We develop a common programming paradigm
  OmpSs@Cluster → OpenMP on the cluster on top of GASNet
  OmpSs@FPGA → Transparent FPGA acceleration

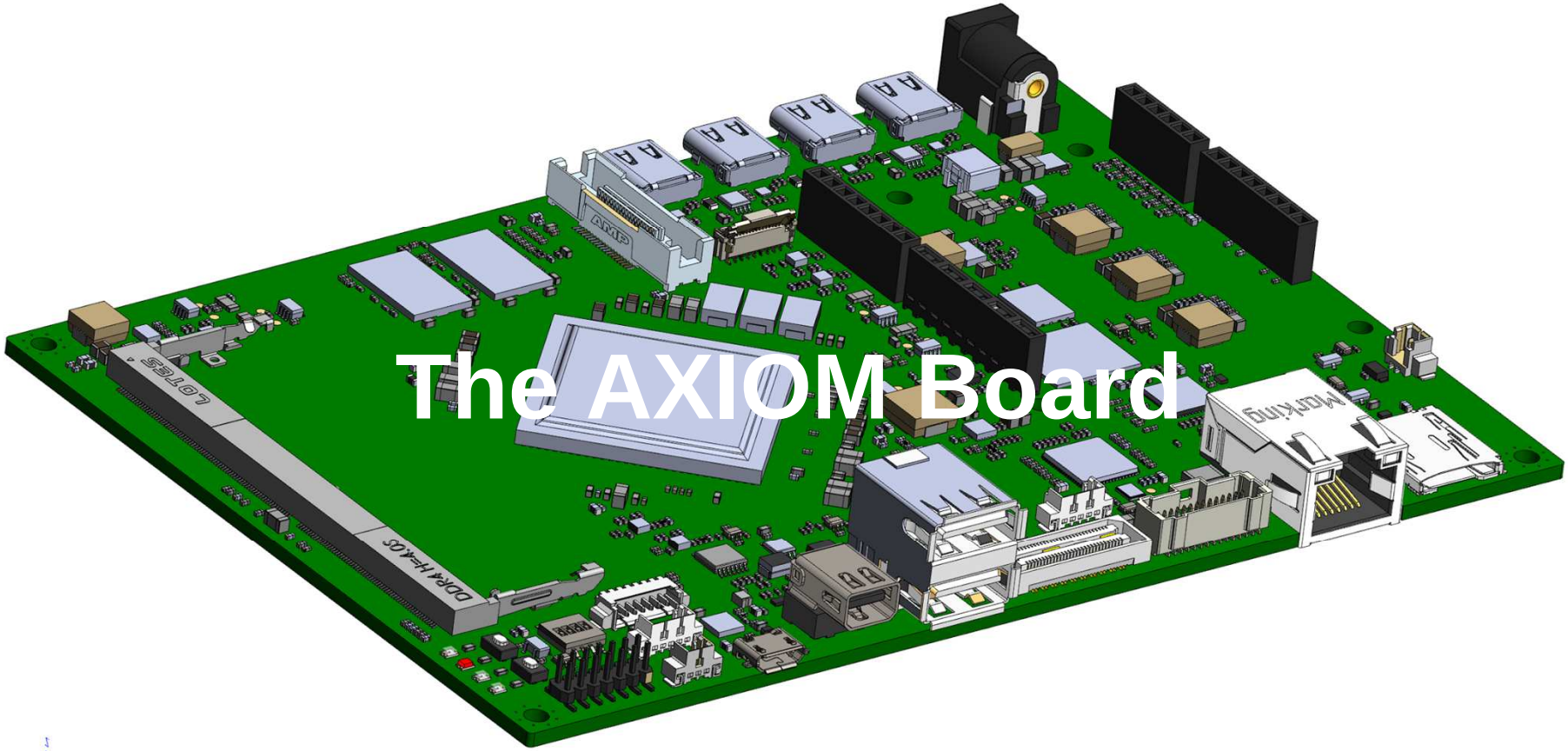
OpenMP

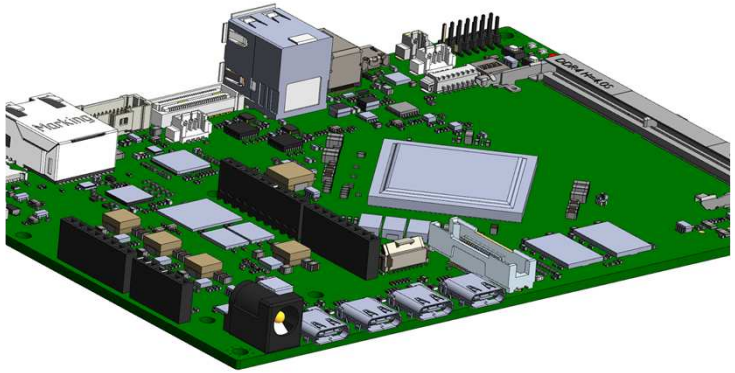
Contributing to the standard

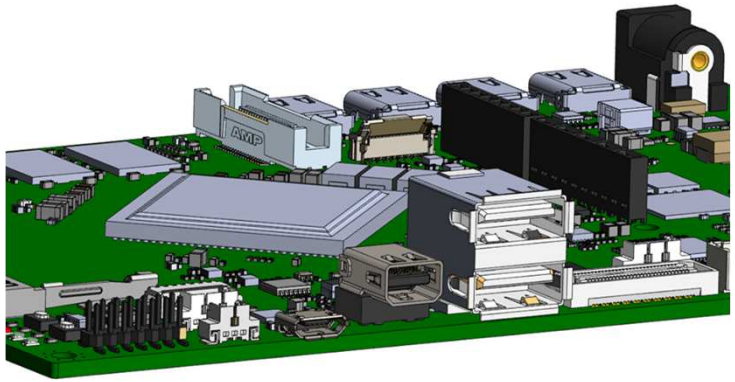
- We use it for Video and audio processing
  Smart surveillance, speech recognition



The AXIOM Board







AXIOM Board: characteristics

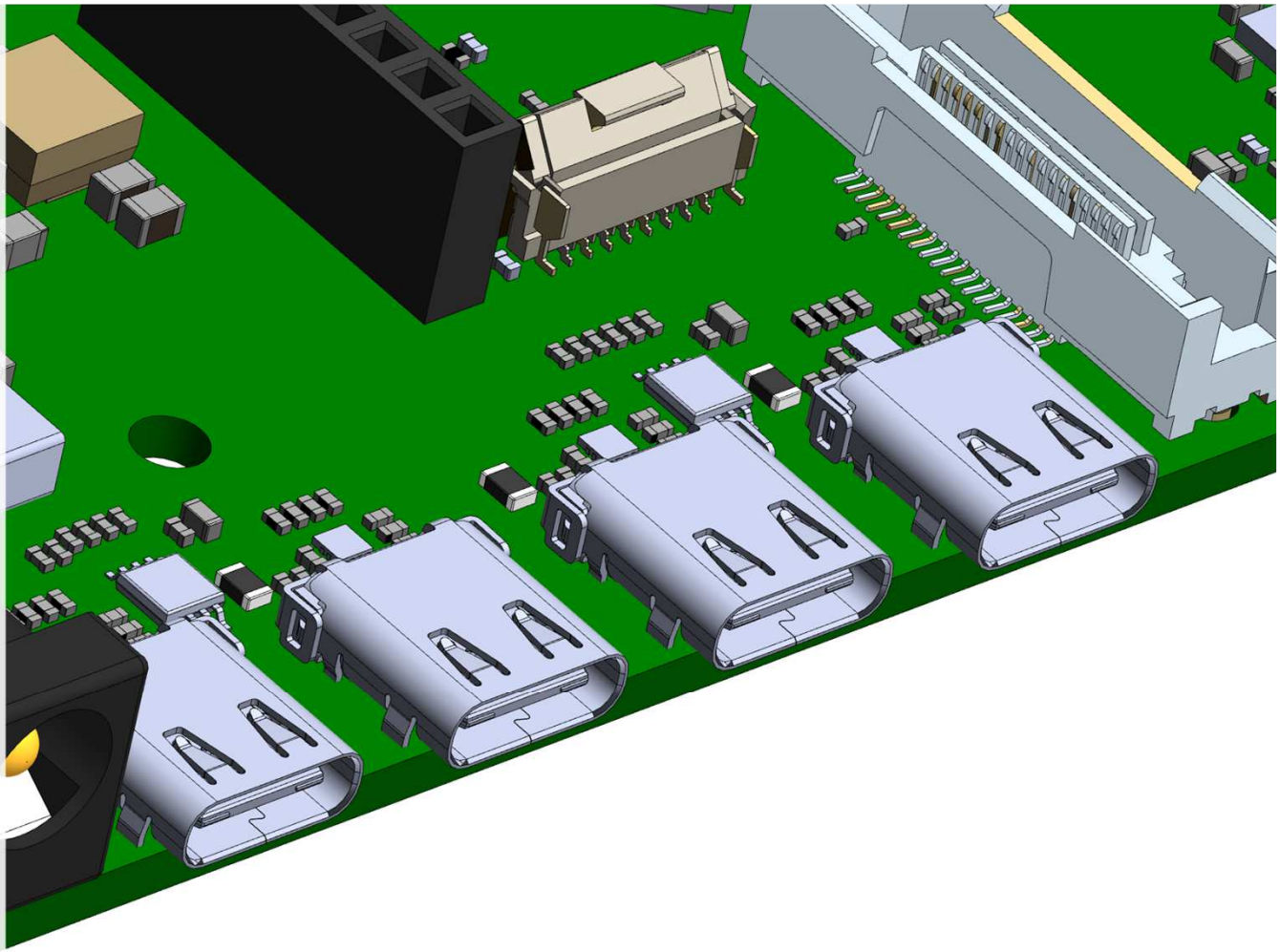
- Small form factor (160cm x 109cm)
- Xilinx Zynq Ultrascale+ ZU9EG
- socket SO-DIMM DDR4 for the PS RAM
- 1Gb DDR4 for the PL RAM
- 8 to 32 GB di eMMC
- Boot from QSPI, eMMC, uSD card, JTAG
- Standard connections (USB, Ethernet, Video output)
- Camera input
- Trace port for software tracing
- Power management measurement possible

AXIOM Board: AXIOM Link

USB Type C connector

Used to get a high-speed connection between boards

Standard connector with special care for signal integrity

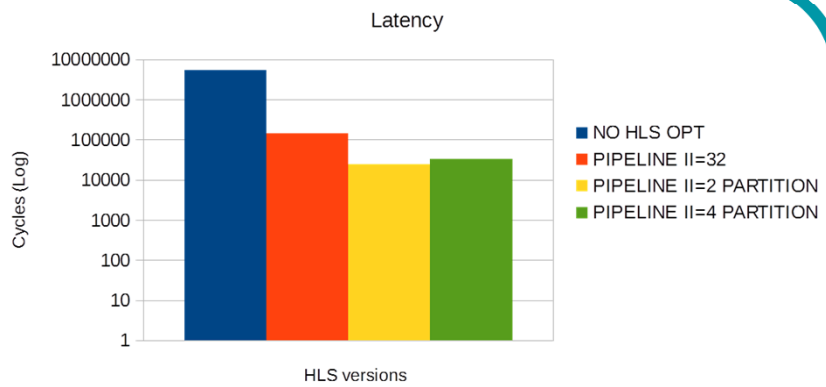


Easy programmability via OmpSs

Only 3 lines of code to
- accelerate code on FPGAs
- distributed code across several AXIOM boards

Application	*Seq - DMA version*	*pthread version*	*OmpSs version*
Cholesky	71	26	3
Covariance	94	29	3
64x64	95	39	3
32x32	95	39	3

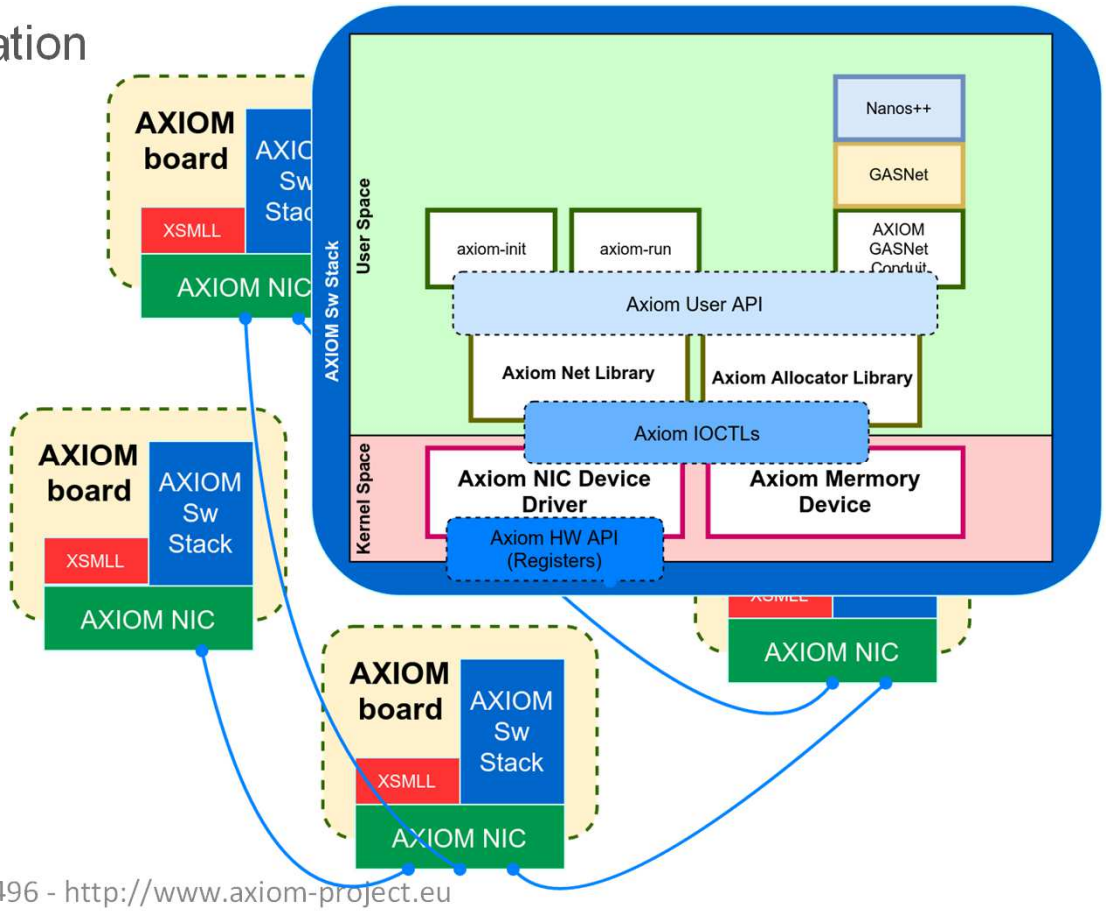
```
1 #pragma omp target device(fpga, smp) copy_deps
2 #pragma omp task in(a[0:64*64-1], b[0:64*64-1]) \
4 void matrix_multiply(float a[64][64],
5                      float b[64][64],
6                      float out[64][64]) {
7     for (int ia = 0; ia < 64; ++ia)
8         for (int ib = 0; ib < 64; ++ib) {
9             float sum = 0;
10            for (int id = 0; id < 64; ++id)
11                sum += a[ia][id] * b[id][ib];
12            out[ia][ib] = sum;
13        }
14 }
15 ...
16 int main( void ){
17 ...
18  matrix_multiply(A,B,C1);
19  matrix_multiply(A,B,C2);
20  matrix_multiply(C1,B,D);
21 ...
22  #pragma omp taskwait
```



OmpSs matrix multiply performance
- 1.7 Gflops using 64x64 blocking size
- 4.0 Gflops using 128x128 blocks

First complete software stack now available!

- Real board and QEMU Emulation
- AXIOM-Link software specs available
- Device drivers
- Memory allocator
- Utility apps
- GASNet Spawner
- OmpSs@Cluster

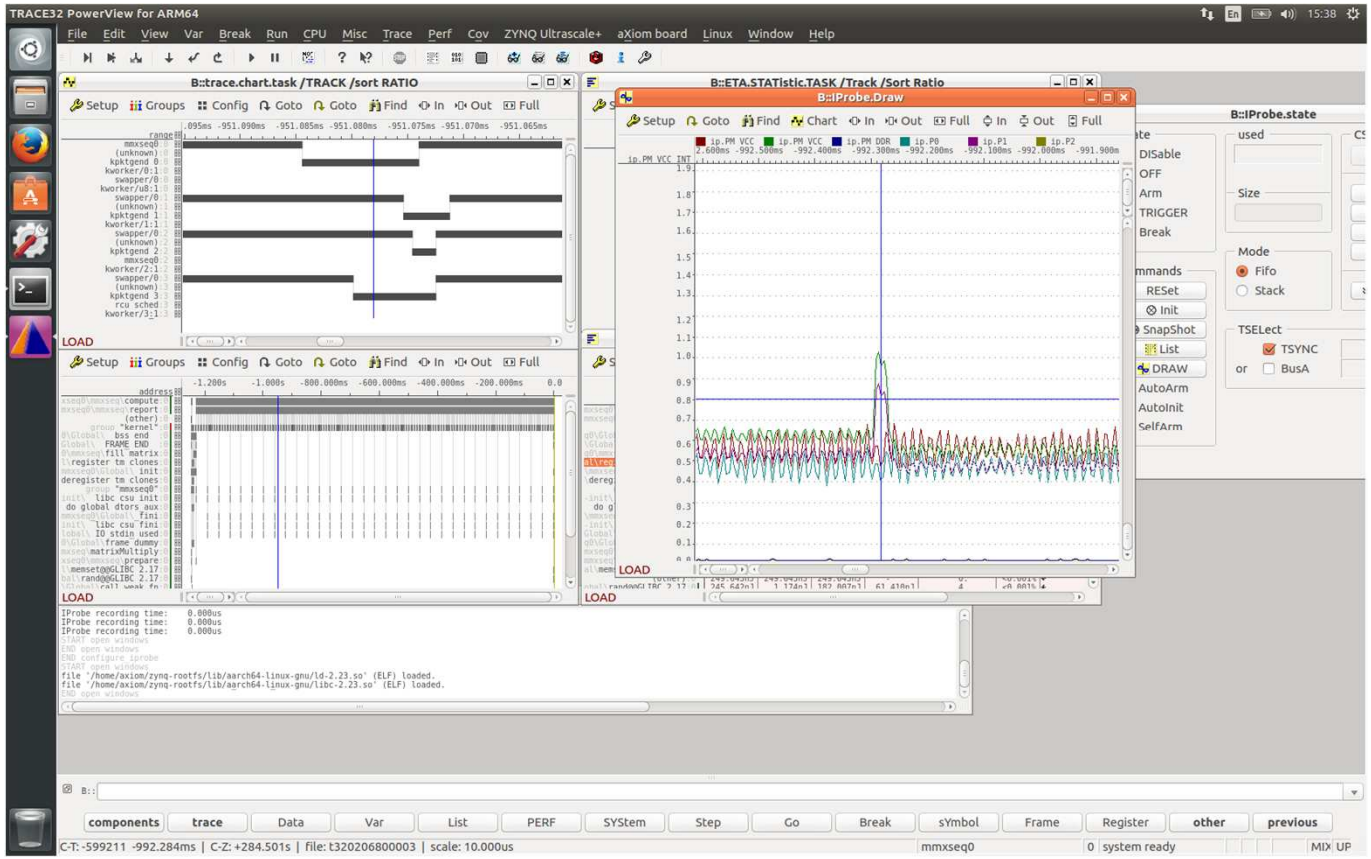


Some performance results…

- RDMA async
  - 9.6 Gbps
- RDMA sync
  - 7.7 Gbps
- LONG
  - 1.1 Gbps
- LONG multi-thread (8 threads)
  - 3.7 Gbps
- RAW
  - 0.285 Gbps

- MatrixMultiply with OmpSS@cluster over AXIOM conduit
- Block Size = 4 - Matrix Size = 800
  - 2 boards (3 working threads per board)
    - Execution time: 5296 msec (*6=31776)
  - 1 board (4 working threads)
    - Execution time: 7852 msec (*4=31408)
- Block Size = 4 - Matrix Size = 1000
  - 2 boards (3 working threads per board)
    - Execution time: 11020 msec (*6=66120)
  - 1 board (4 working threads)
    - Execution time: 16412 msec (*4=65648)

2 demos

- Cluster setup
- Lauterbach TRACE32 Trace and Power measurement

Lauterbach TRACE32 Trace and Power