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

IWES 2017 – 2nd Italian Workshop on Embedded Systems

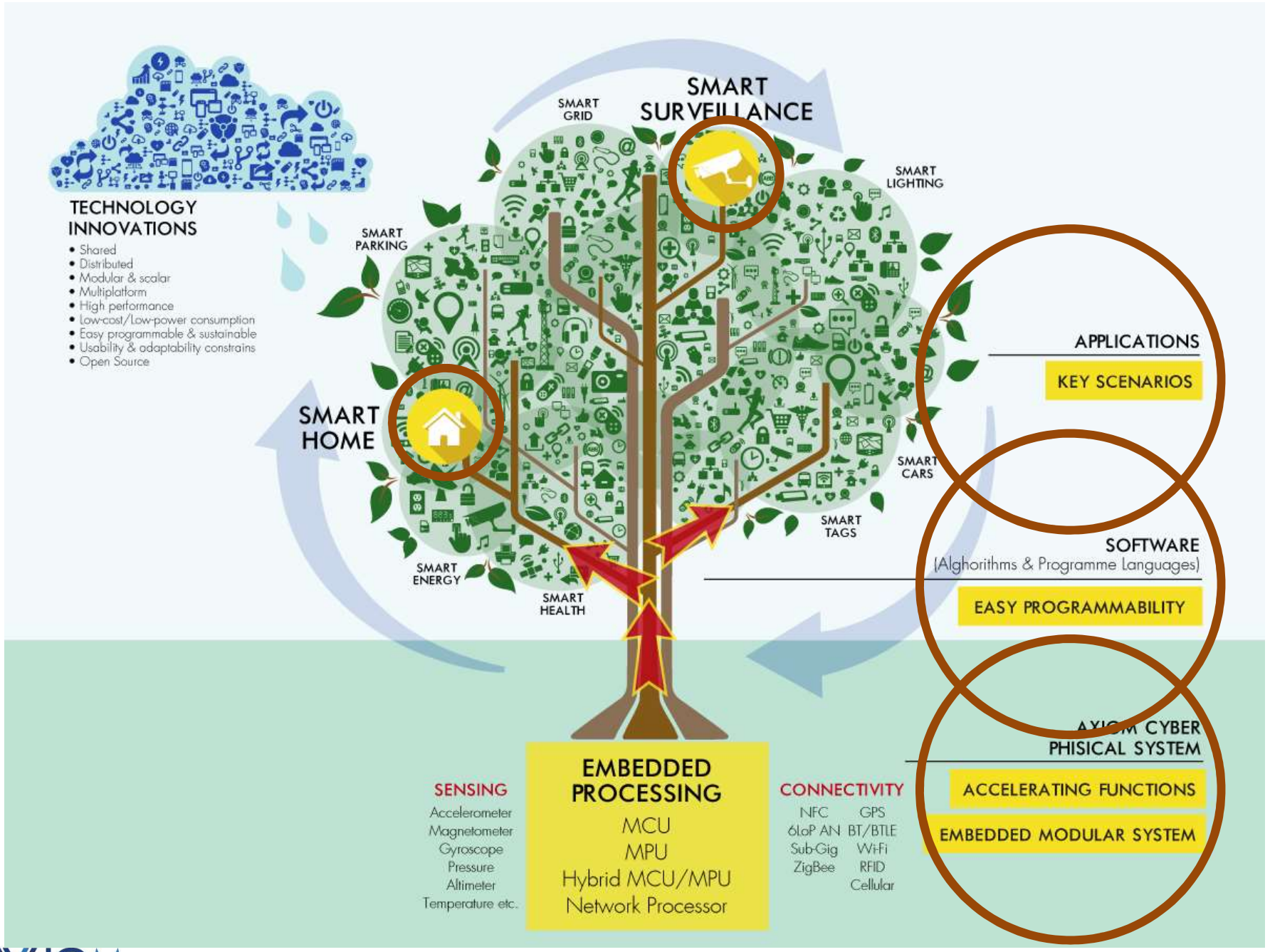
Rome, 7-8 September 2017

The AXIOM-board: bringing programmability, acceleration, scalability into a 64-bit hand-size board

Roberto Giorgi

University of Siena, Italy (Project Coordinator)

University of Siena
(Coordinator Partner)

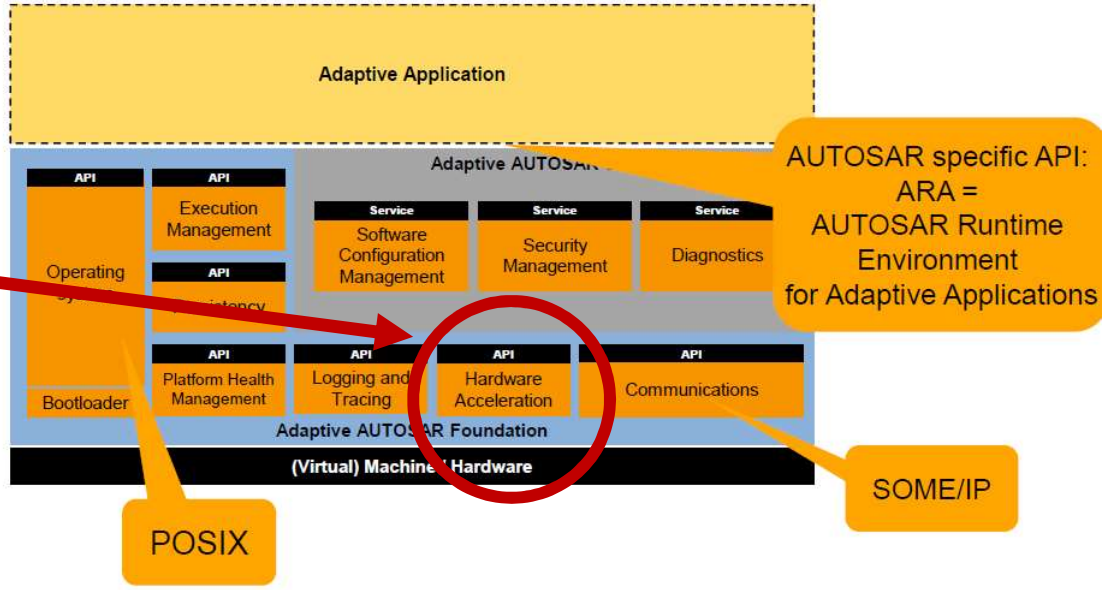

TECHNOLOGY INNOVATIONS
• Shared
• Distributed
• Modular & scalar
• Multiplatform
• High performance
• Low-cost/Low-power consumption
• Easy programmable & sustainable
• Usability & adaptability constrains
• Open Source
SMART GRID
SMART SURVEILLANCE
SMART LIGHTING
SMART PARKING
SMART HOME
SMART CARS
SMART TAGS
SMART ENERGY
SMART HEALTH
APPLICATIONS
KEY SCENARIOS
SOFTWARE
(Alghorithms & Programme Languages)
EASY PROGRAMMABILITY
PHISICAL SYSTEM
ACCELERATING FUNCTIONS
EMBEDDED MODULAR SYSTEM
SENSING
Accelerometer
Magnetometer
Gyroscope
Pressure
Altimeter
Temperature etc.
EMBEDDED PROCESSING
MCU
MPU
Hybrid MCU/MPU
Network Processor
CONNECTIVITY
NFC
6LoP AN
Sub-Gig
ZigBee
GPS
BT/BTLE
Wi-Fi
RFID
Cellular

Highlights of this talk

1) Exploring the concept of ”scalable embedded system”
2) Indicating a way to achieve such scalability by supporting special threads called Data-Flow Threads (DF-Threads)
3) Illustrating how these concepts are integrated in the AXIOM project, **which is focused to build a scalable Single Board Computer**

Vehicle Architecture

- Expected total number of ECUs: 120*
  - 5 - 10 domain controllers will run with adaptive platform
- Classical and adaptive AUTOSAR will cohexist
- Hardware acceleration is needed



[*Stefan Voget, Continental, TAPPS Workshop keynote, May 2017]

Continental

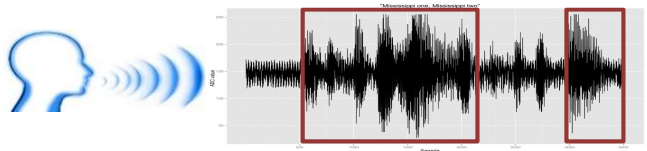
10.05.2017
Dr. Stefan Voget © Continental AG

AXIOM OBJECTIVES

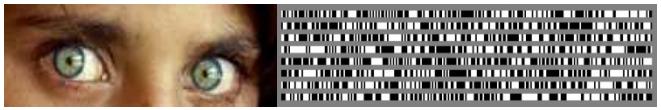
- **OBJ1) Producing a small board that is flexible, energy efficient and modularly scalable**
  - A as AGILITY, i.e. flexibility: FPGA, fast-and-cheap interconnects based on existing connectors like SATA
  - Energy efficiency: low-power ARM, FPGA
  - Modularity: fast-interconnects, distributed shared memory across boards
- **OBJ2) Easy programmability of multi-core, multi-board, FPGA**
  - Programming model: Improved OmpSs → X as EXTENSIBILTY
  - Runtime & OS: improved thread management
- **OBJ3) Leveraging Open-Source software to manage the board**
  - Compiler: BSC Mercurium
  - OS: Linux
  - Drivers: provided as open-source software by partners
- **OBJ4) Easy Interfacing with the Cyber-Physical Worlds**
  - Platform: integrating also Arduino support for a plenty of pluggable board (so-called "shields") → "IO" as I/O
  - Platform: building on the UDOO experience from SECO
- **OBJ5) Enabling real time movement of threads**
  - Runtime: will leverage the EVIDENCE's SCHED_DEADLINE scheduler (i.e. EDF) included Linux 3.14, UNISI low-level thread management techniques
- **OBJ6) Contribution to Standards**
  - Hardware: SECO is founding member of the Standardization Group for Embedded Systems (SGET)
  - Software: BSC is member of the OpenMP consortium

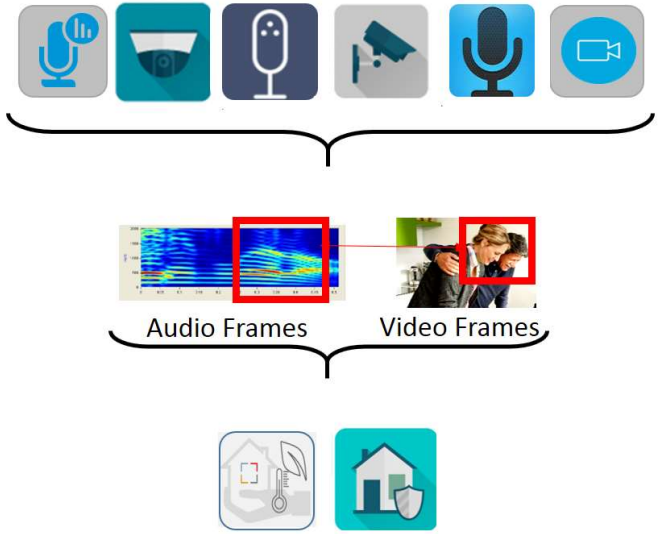
Smart Living/Home scenario

- **Speaker identification** is the identification of a person from characteristics of voices (*voice biometrics*).



- **Iris recognition** is the process of recognizing a person by analyzing the random pattern of the iris.





SLH application demo on the AXIOM Evaluation Platform (AEP)

```
analog@analog:~/AXIOM/AXIOM_Biometric_identification_V2.2$ time ./AXIOM_Bio_Identification_Gcc
Change pipe status OK!
Speech size = 600
Speech size = 1200
Find a new eye ...
Rejected by Segmentation/Normalization procedure
Find a new eye ...
Rejected by Segmentation/Normalization procedure
Find a new eye ...
Rejected by Segmentation/Normalization procedure
Find a new eye ...
Testing...
Find a new eye ...
Eye rejected for sobel!
Find a new eye ...
Testing...
Find a new eye ...
Testing...
Iris recognition done!
Speech size = 1800
Speech size = 2400
Speech size = 3000
Start sfbcep with audio.speechSize = 3020
Start normFeat ...
Writing to: gstsfbcep
Start energyDetector ...
Start normFeat ...
Writing to: gstsfbcep
ComputeTest_GMM MALE...
(ComputeTest) No Channel Compensation
Compute-T-Test MALE...
Compute-Z-Test MALE...
ComputeTest_GMM FEMALE...
(ComputeTest) No Channel Compensation
Compute-T-Test FEMALE...
Compute-Z-Test FEMALE...
Speaker recognition done!
Iris recognition and speaker identification completed!

VIMAR speaker identification proc has identified : AXIOM_User_03
VIMAR iris recognition proc  has identified : AXIOM_User_03

real    0m33.448s
user    0m41.030s
sys     0m0.880s
analog@analog:~/AXIOM/AXIOM_Biometric_identification_V2.2$
```

Audio trigger

Iris Recognition

Speaker identification

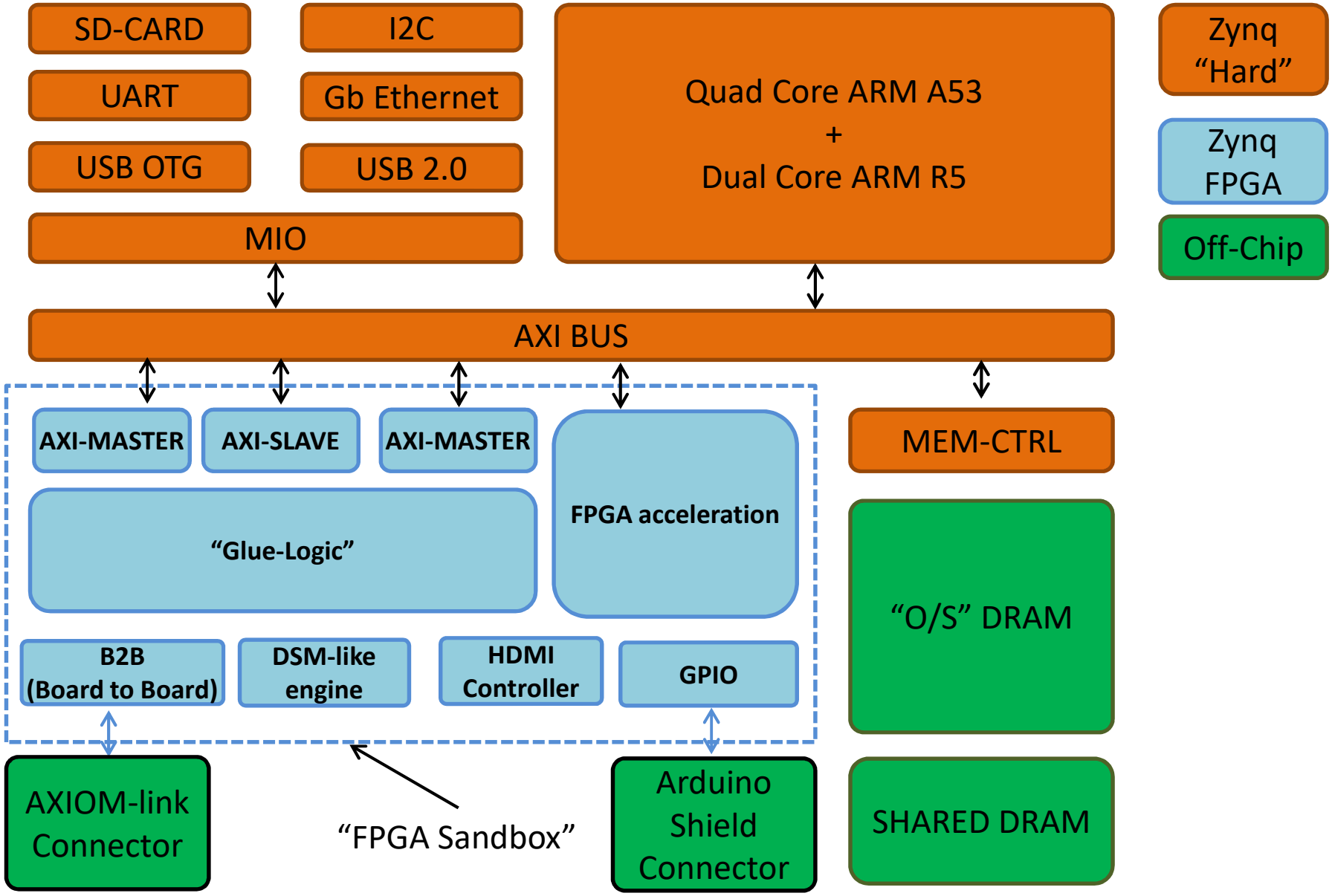
Identification done!

The T-800's POV from Terminator 2
Carolco Pictures / Tristar

AXIOM – THE MODULE-v2

- KEY ELEMENTS
  - K1: ZYNQ FPGA (INCLUDES 6 ARM CORES)
  - K2: ARM GP CORE(S)
  - K3: HIGH-SPEED & INEXPENSIVE INTERCONNECTS
  - K4: SW STACK – OMPSS+LINUX BASED
  - K5: OTHER I/F (ARDUINO, USB, ETH, WIFI, …)

AXIOM-v2 Architectural Template

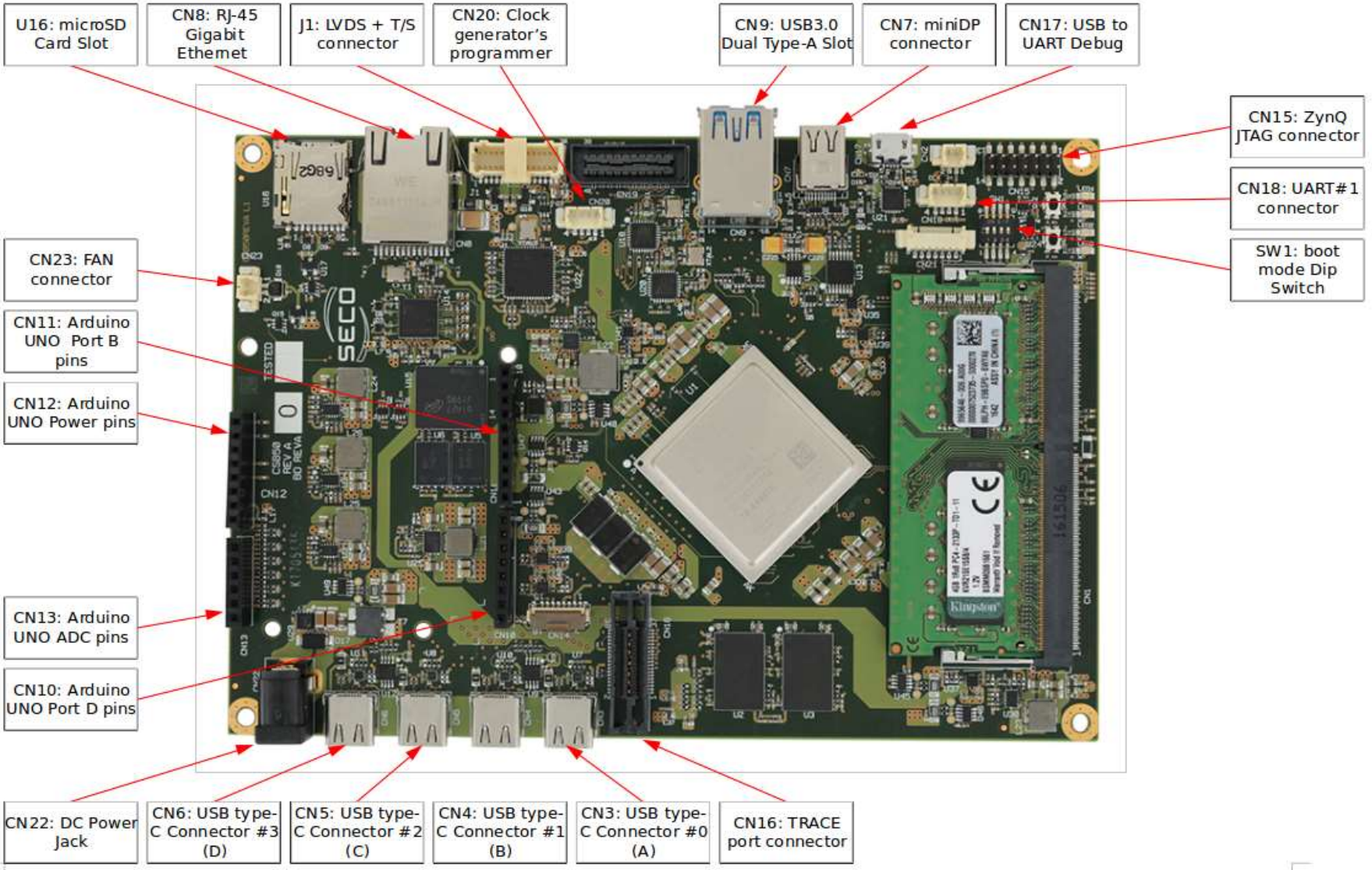


WHY OMPSS

```
1 #pragma omp target device(fpga, smp) copy_deps
2 #pragma omp task in(a[0:64*64-1], b[0:64*64-1]) \
3                  out(c[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
23 }
```

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

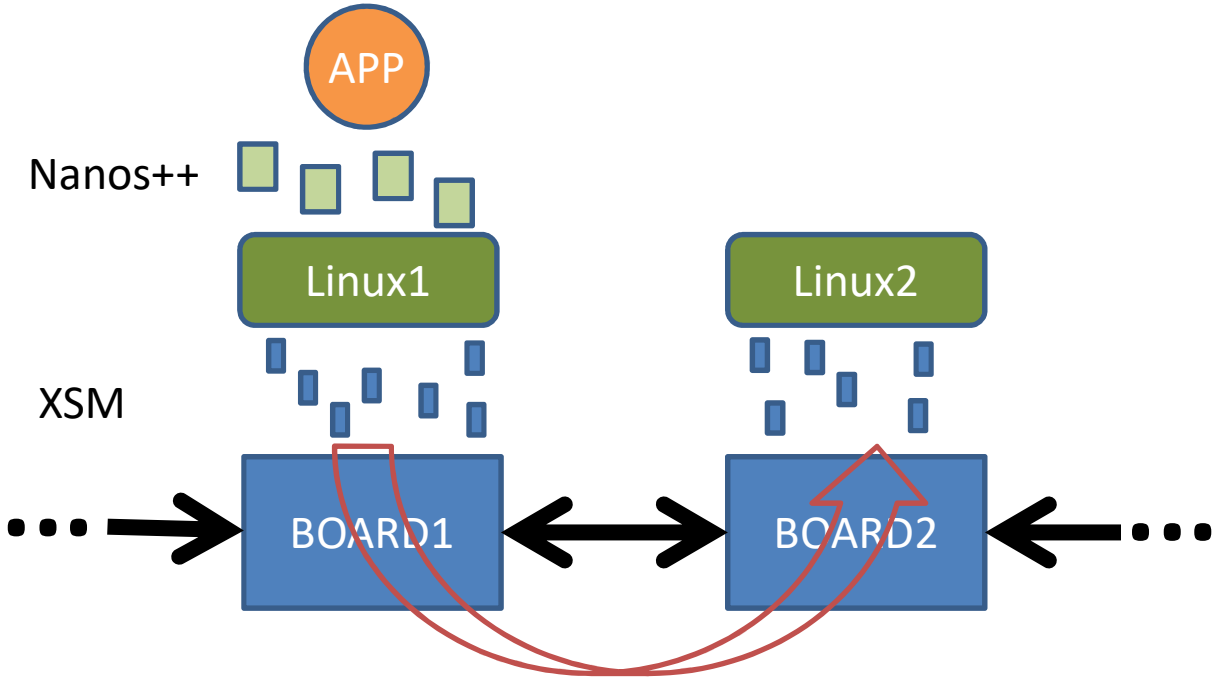
The AXIOM-BOARD (about 10x15 cm)



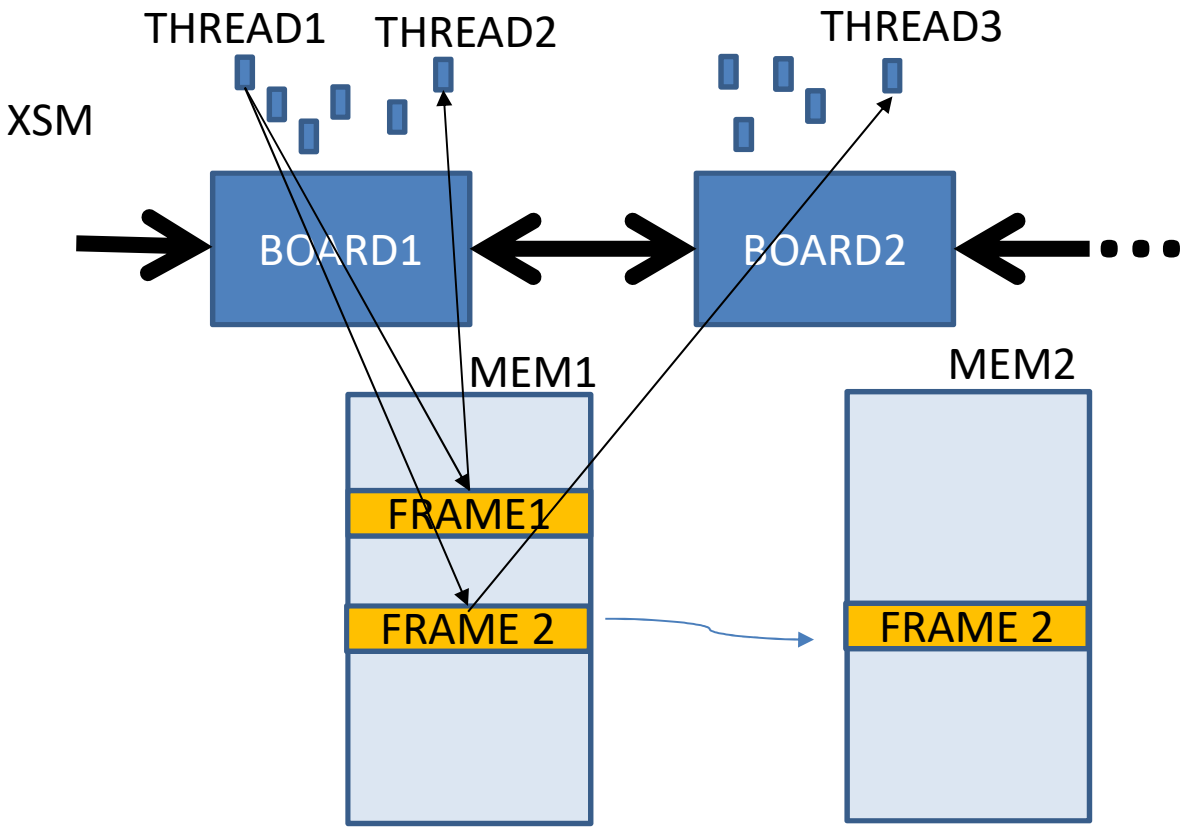
Disclaimer: subject to changes without notice.

Testing Environment

- Problem to analyze



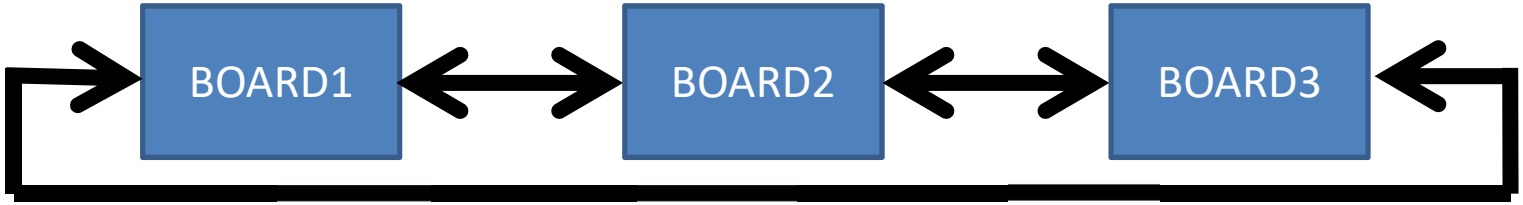
Data movement

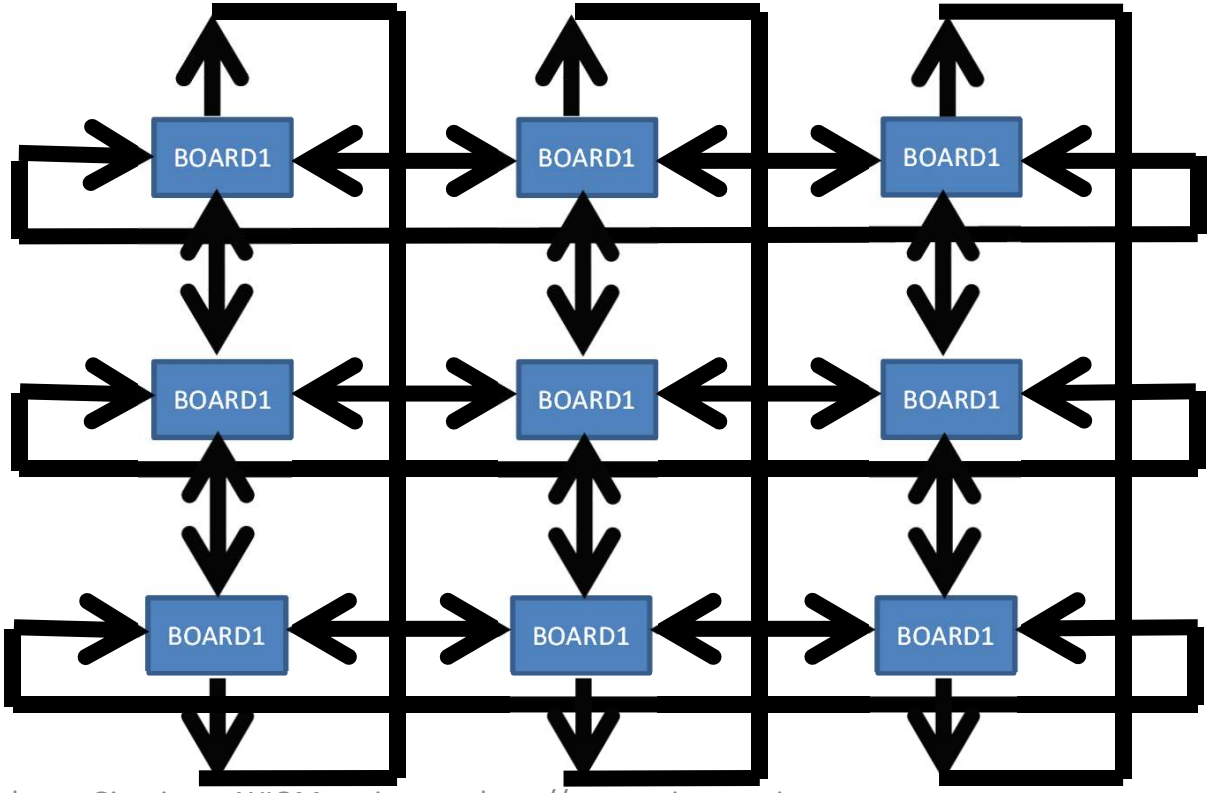


4x 10Gbit/s via USB-C connectors

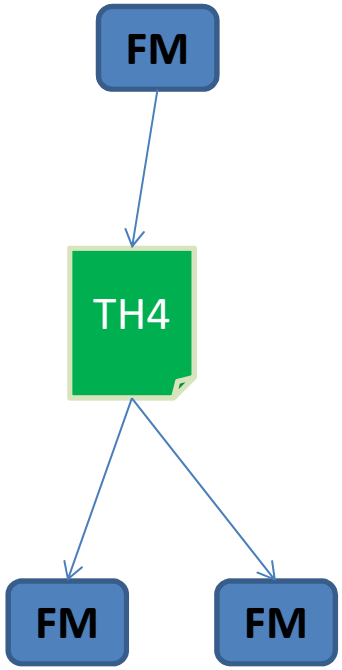
Several Topologies are possible

- E.g. ring or 2D torus





XSMLL -- XSM Low Level

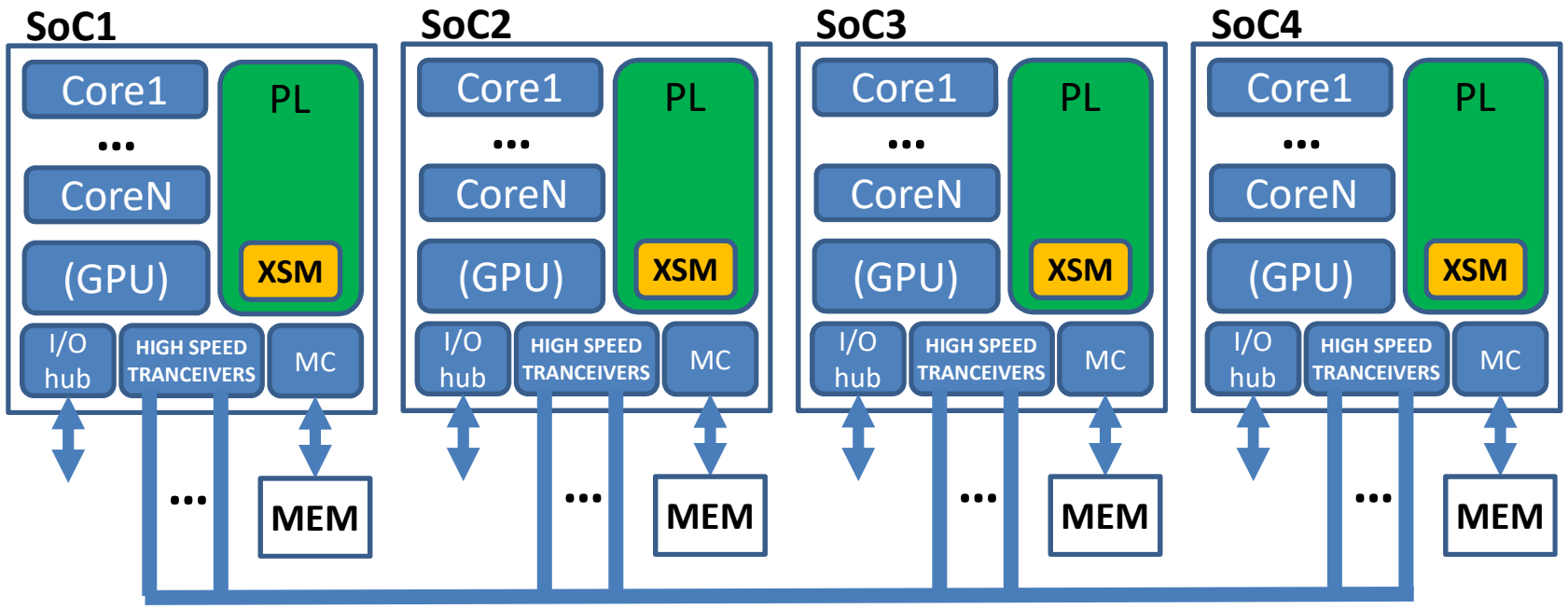


- X-thread (new incarnation of DF-thread)
  - A function that expects no parameters and returns no parameters.
    - The body of this function can refer to any memory location for which it has got the pointer through XSM function calls (e.g., xpreload, xpoststor, xsubscribe, ...). An X-thread is identified by an object of type xtid_t (X-thread identifier). In other words:

```
typedef void (*xthread_t)(void)
```

- INPUT_FRAME, OUTPUT_FRAME
  - INPUT_FRAME: A buffer which is allocated in the local memory and contains the input values for the current X-thread.
  - OUTPUT_FRAME: A buffer which is allocated in the local memory and contains values to be used by other X-threads (consumer X-threads)
- SYNCHRONIZATION_COUNT
  - A number which is initially set to the number of input values (or events) needed by an X-thread. The SYNCHRONIZATION_COUNT has to be decremented each time the expected data is written in an OUTPUT_FRAME.

4-board AXIOM System



Modeled SoC

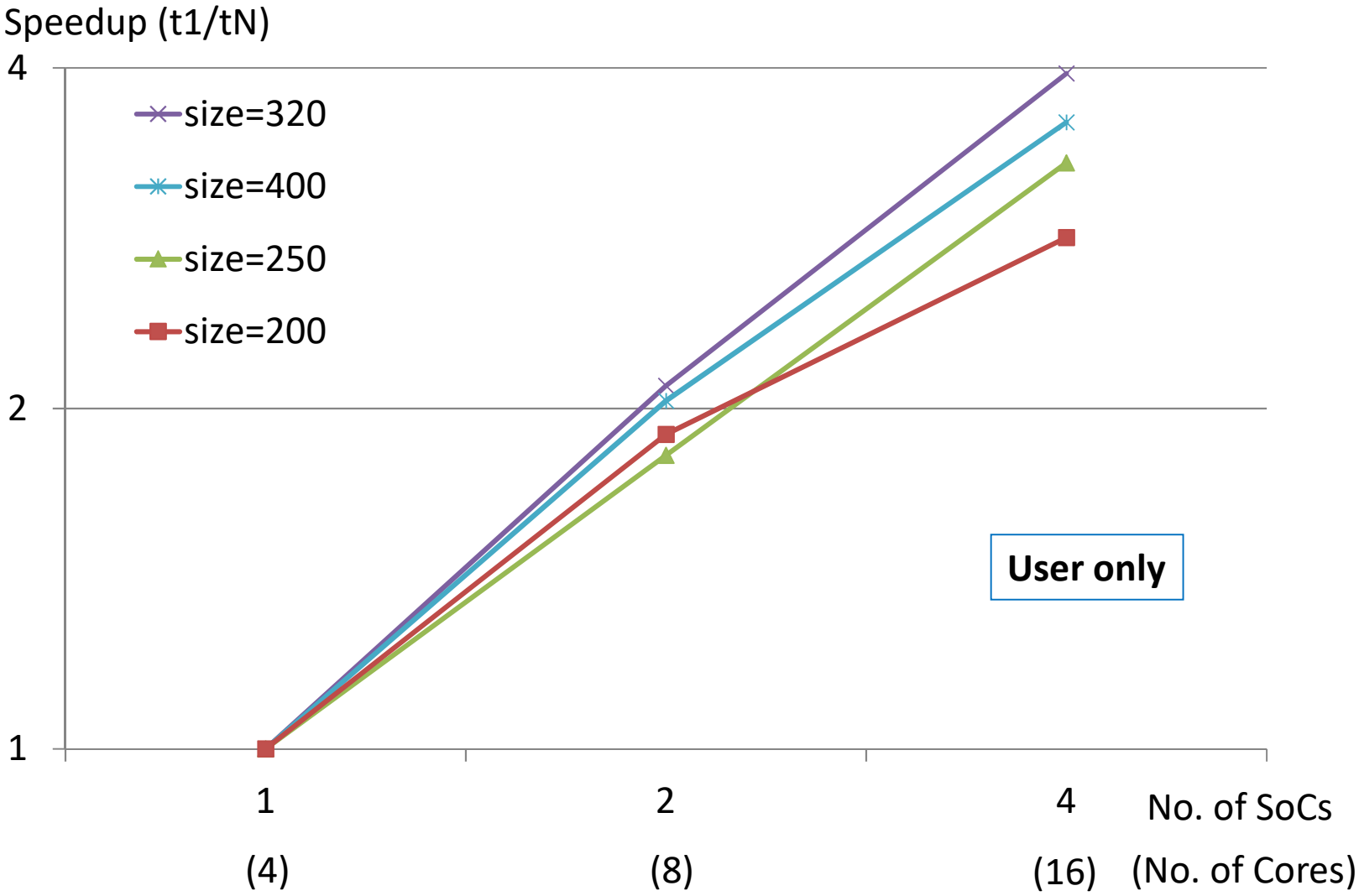
Parameter	Description
SoC	4-cores connected by a shared-bus, IO-hub, MC, high-speed transceivers
Core	1GHz, in-order superscalar
Branch Predictor	two-level (history length=14bits, pattern-history table=16kB, 8-cycle misspredicition penalty)
L1 Cache	Private I-cache 32 KB, private D-cache 32 KB, 2 ways, 3-cycle latency
L2 Cache	Private 512 KB, 4 ways, 5-cycle latency
L3 Cache	Shared 4GB, 4 ways, 20-cycle latency
Coherence protocol	MOESI
Main Memory	1 GB, 100 cycles latency
I-L1-TLB, D-L1-TLB	64 entries, full-associative, 1-cycle latency
L2-TLB	512 entries, direct access, 1-cycle latency
Write/Read queues	200 Bytes each, 1-cycle latency

Matrix-Multiply on COTSon/XSM

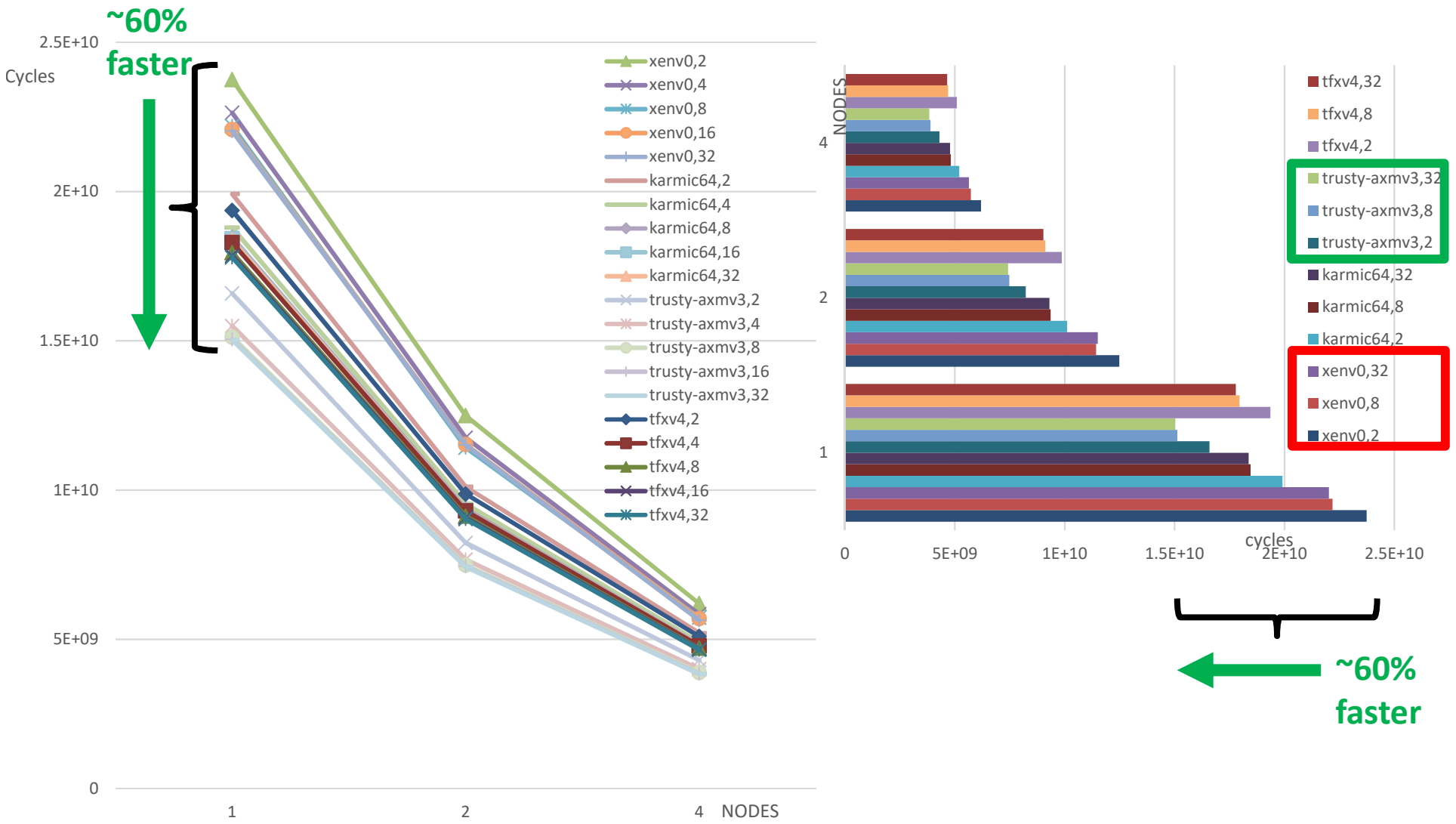
http://cotson.sourceforge.net

- Some experiments have been performed on the COTSon/XSMLL with the following parameters
  - Square Matrix size *n* and block size *b*:
    - n=160,200,250,320,400,500,640,800,1000,1280,1600,2000 b=5,10,25,50
    - n=128,256,512 b=8
  - Different programming models
    - OpenMPI, Cilk
  - Different execution models
    - XSMLL, Standard
  - Different Linux Distributions
    - Ubuntu 9.10 (karmic64), 10.10 (tfxv4), 14.04 (trusty-axmv3), 16.04 (xenv0)

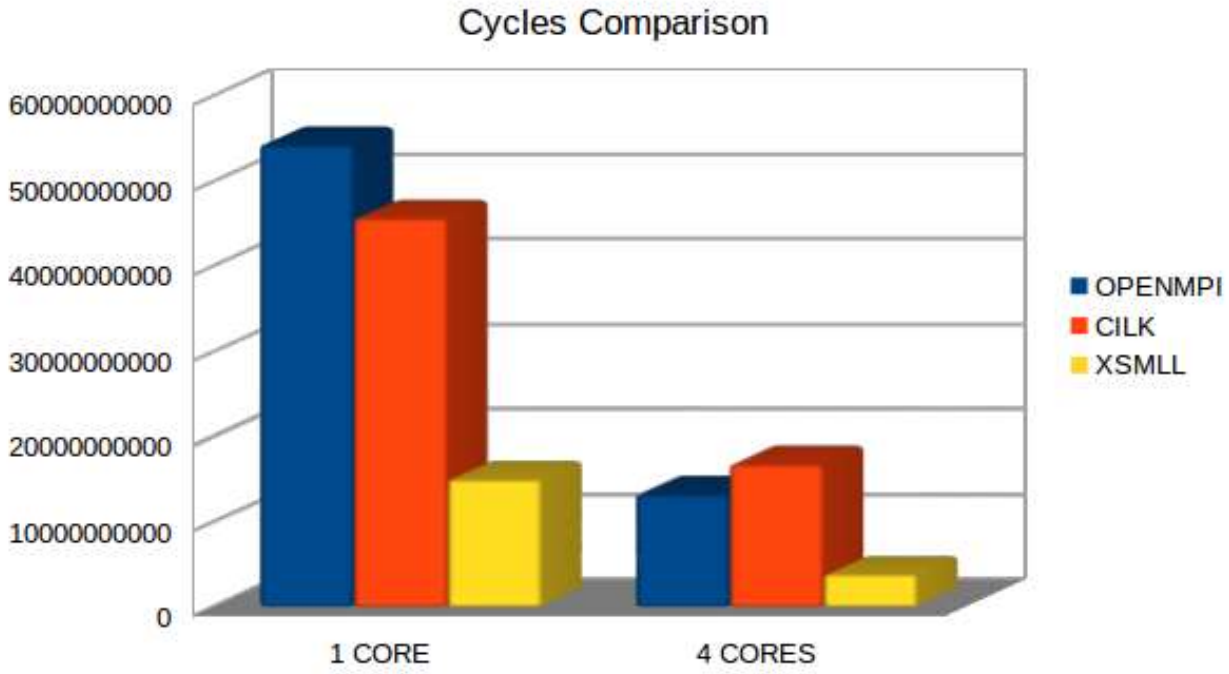
Strong Scaling for benchmark “Dense Matrix Multiplication”



The OS dependency



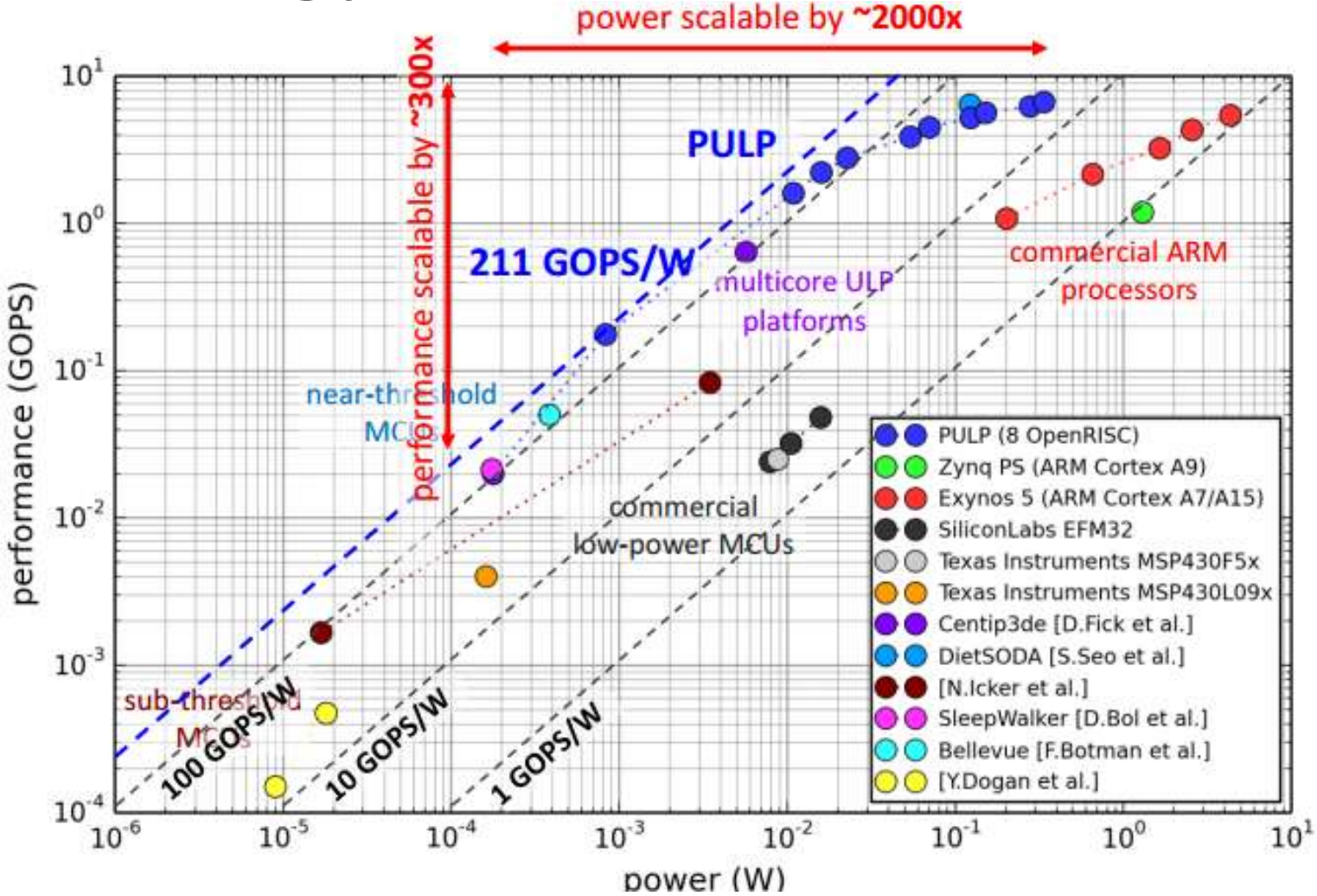
XSMLL vs OpenMPI vs Cilk



	1 NODE (1CORE)	4 NODES/CORES	**XSMLL SPEEDUP**	
OPENMPI	54281301097	13223633943	**3.63**	**3.49**
CILK	45645234077	16738179585	**3.05**	**4.41**
XSMLL	14941500251	3792215176		

* For CILK we are using 4 cores instead of 4 nodes

Energy Efficient Processors



ETH ZURICH – EXPLORING RISC-V FOR THE PULP ARCHITECTURE 30/6/2015

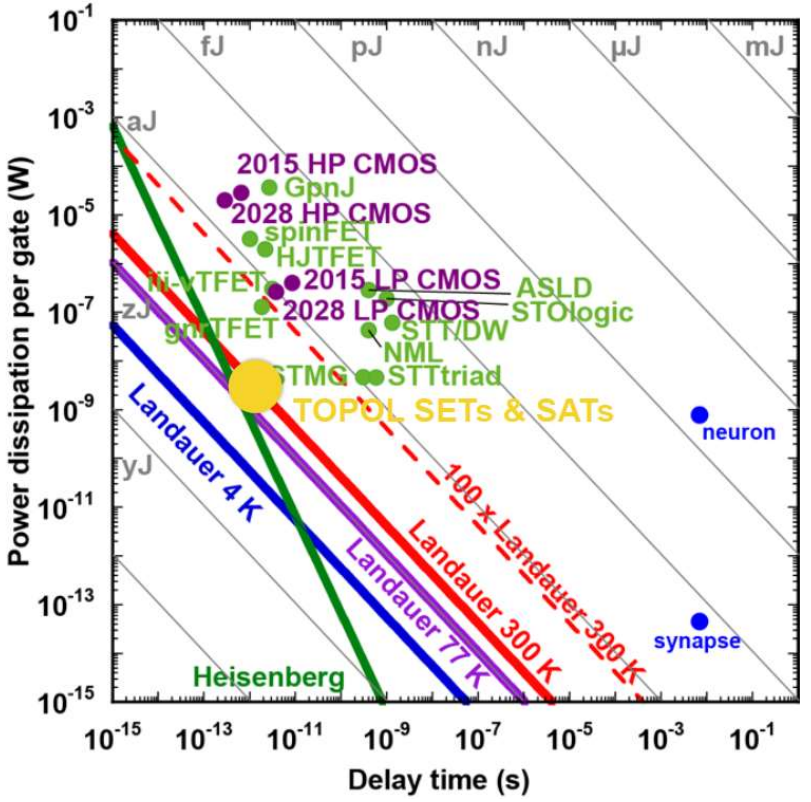
Toward Zero-Power Computing

- **Feynman's principle**

R. Feynman (from "Feynman's Lecture Notes on Computation"):
"Logic or arithmetic could be done with
the power that converges to zero,
even if the number of operations
in a program approaches infinity;
the power for communications
can never approach to zero,
and only to infinity,
if the length of communication lines
approaches infinity
and the number of instructions in a program
approaches infinity."

- **"There is no limit to the minimum of energy required to operate a computer"***

(Luca Gammaitoni, The Future Technology Summit, 24th Sept. 2015)

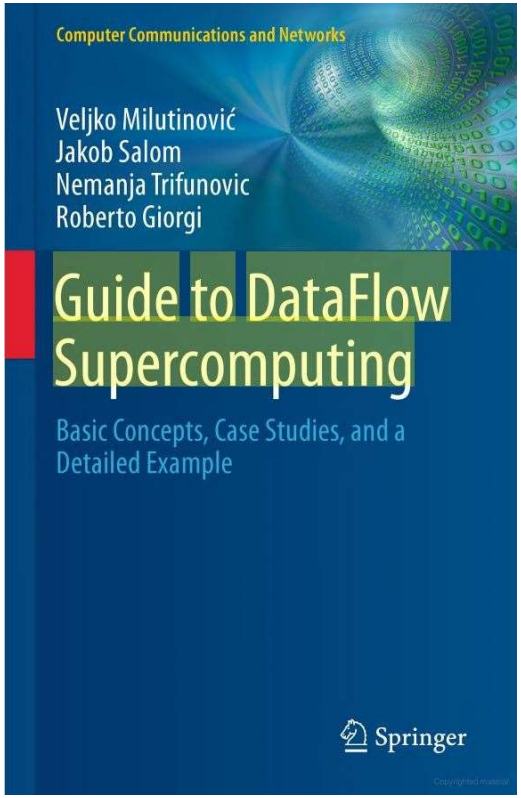


[D. Paul, **ICT-Energy - Strategic Research Agenda]**

Dataflow Approach

V. Milutinovic, N. Trifunovic, R. Giorgi:

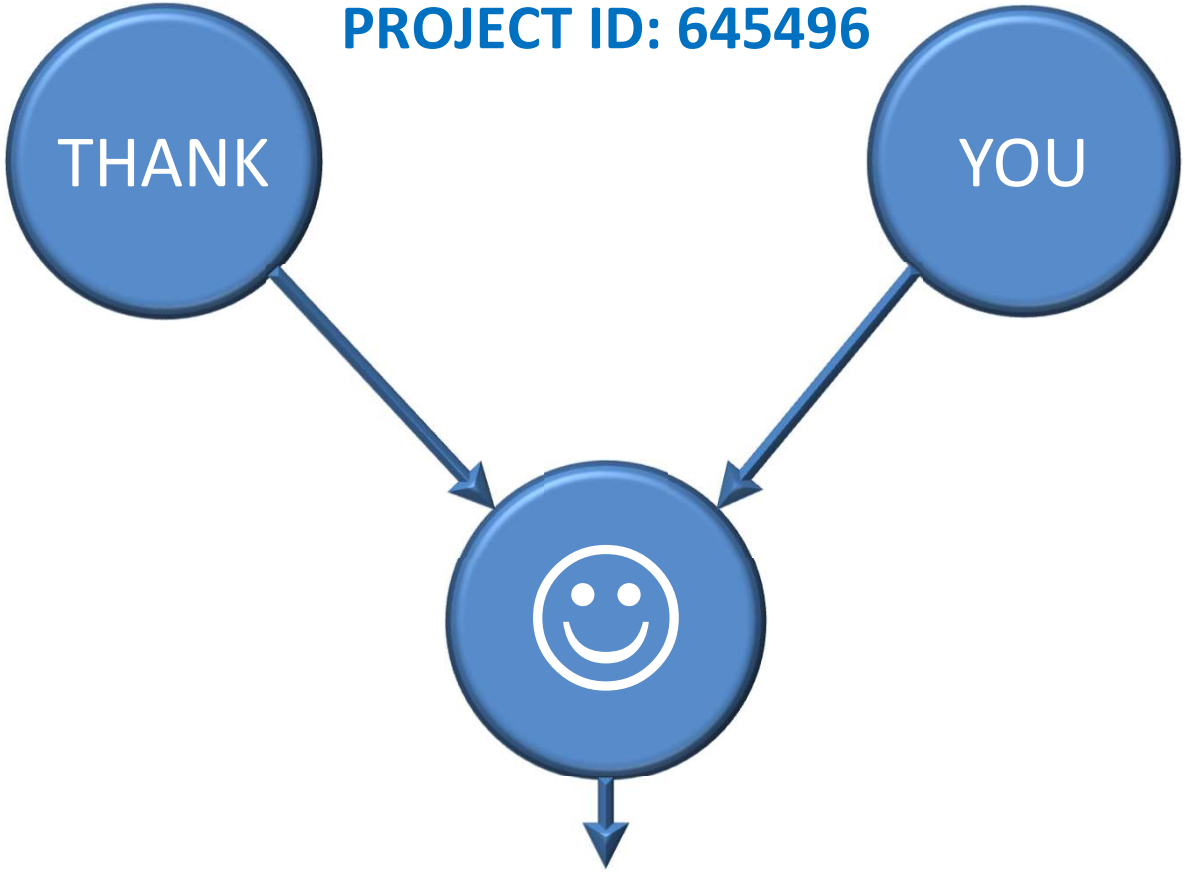
The **control flow** approach is based on unavoidable communications (with logic and arithmetic as primary issues for computing, but secondary contributors to power consumption); the **dataflow approach** is based on logic and arithmetic (with non-zero communications present only if the dataflow compiler is not smart enough, and consequently is not able to generate the execution graph with only zero-length communications lines between neighboring arithmetic and/or logic units). "



M. Milutinovic, J. Salom, N. Trifunovic, R. Giorgi, "Guide to DataFlow Supercomputing", Apr 2015

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

PROJECT ID: 645496



Questions?

AXIOM

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

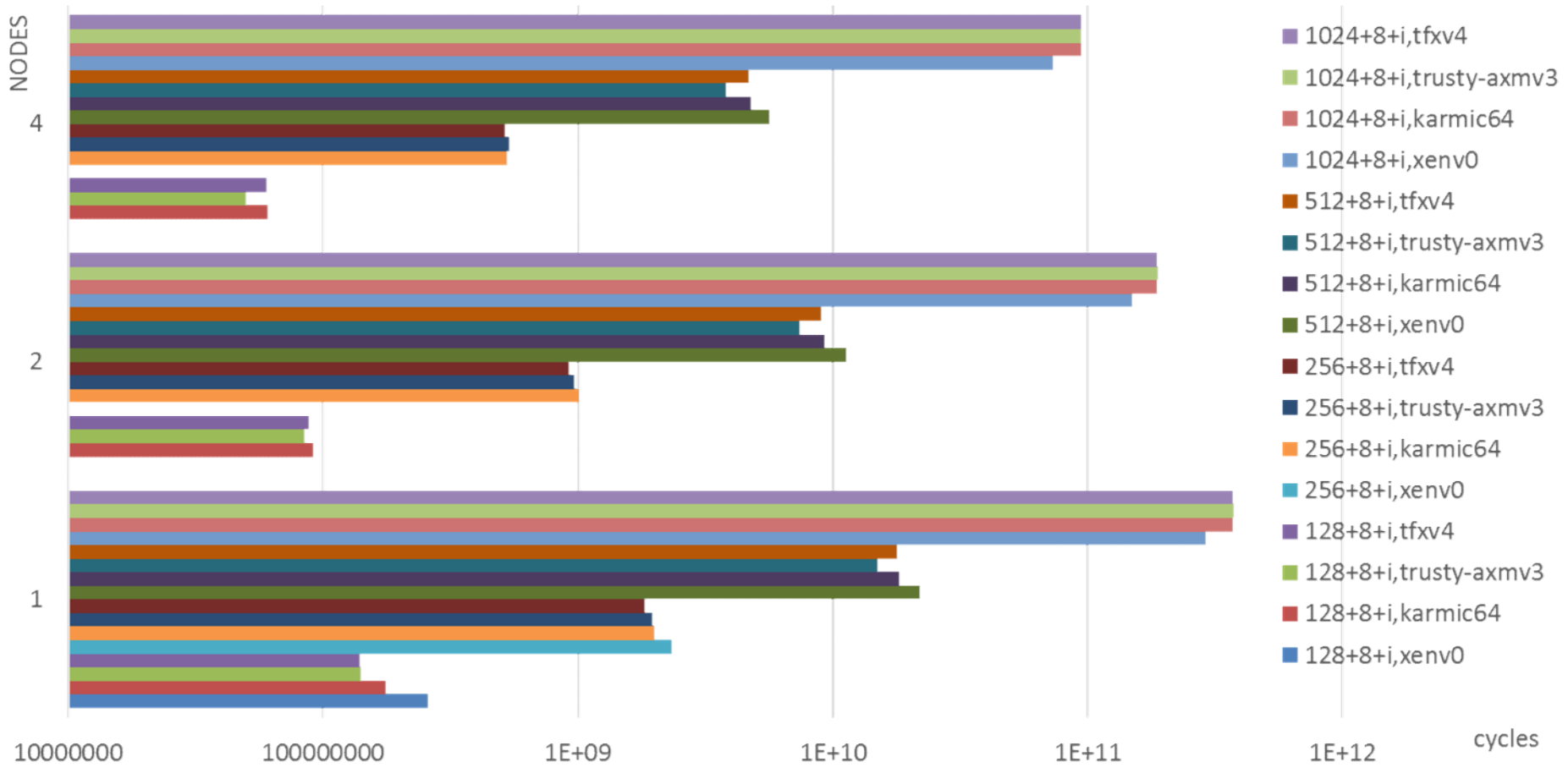
PROJECT ID: 645496

Roberto Giorgi –- AXIOM project --- http://www.axiom-project.eu

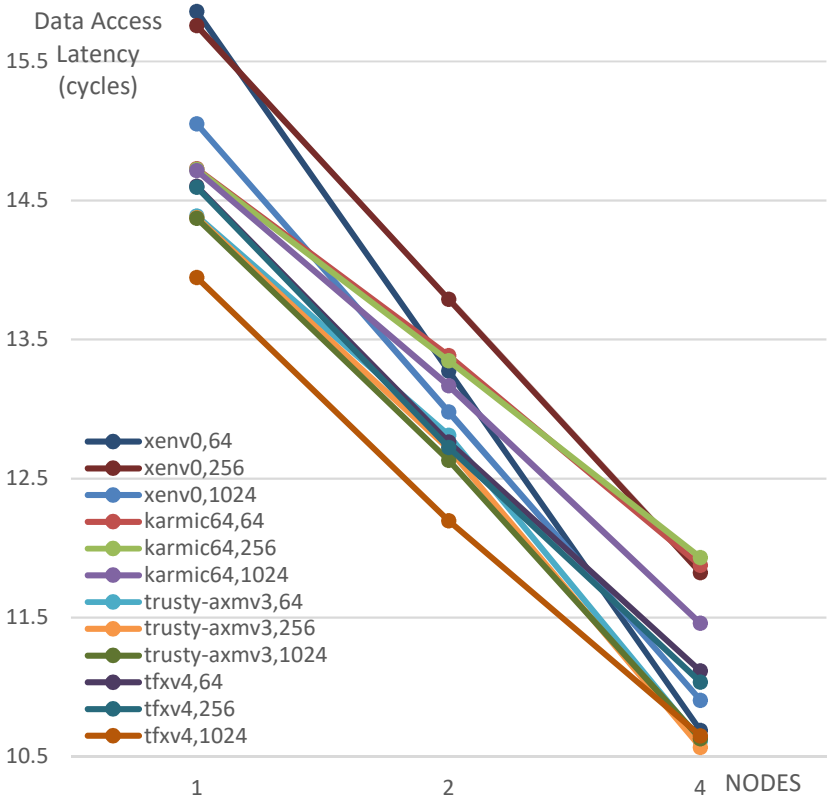
The AXIOM-board: bringing programmability, acceleration, scalability into a 64-bit hand-size board

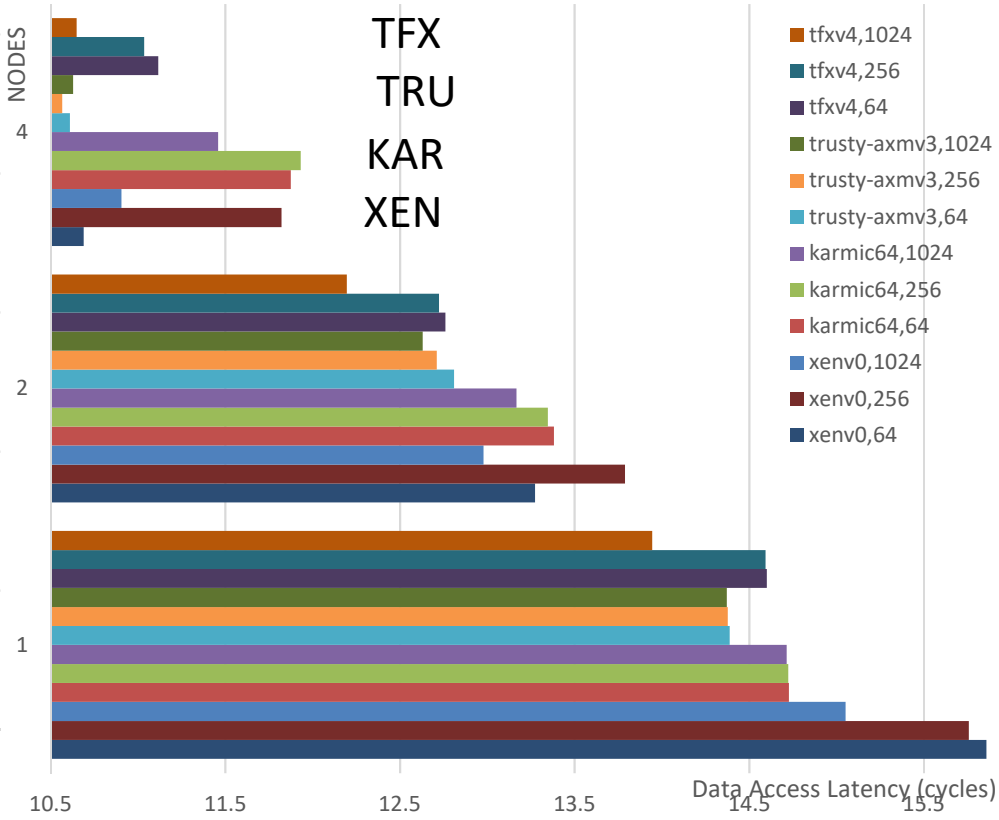
University of Siena
(Coordinator Partner)

Multi-HD Experiment (2)

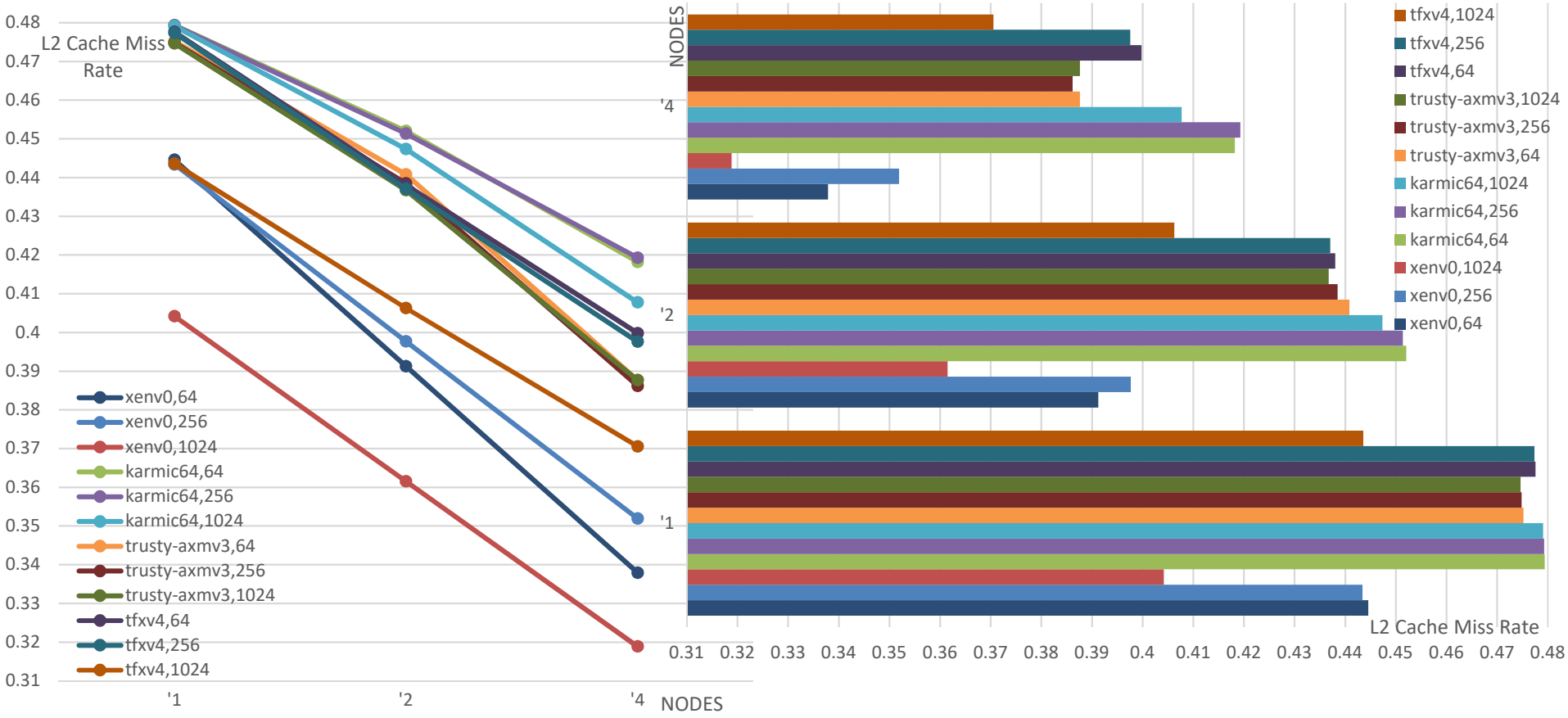


Data Access Latency

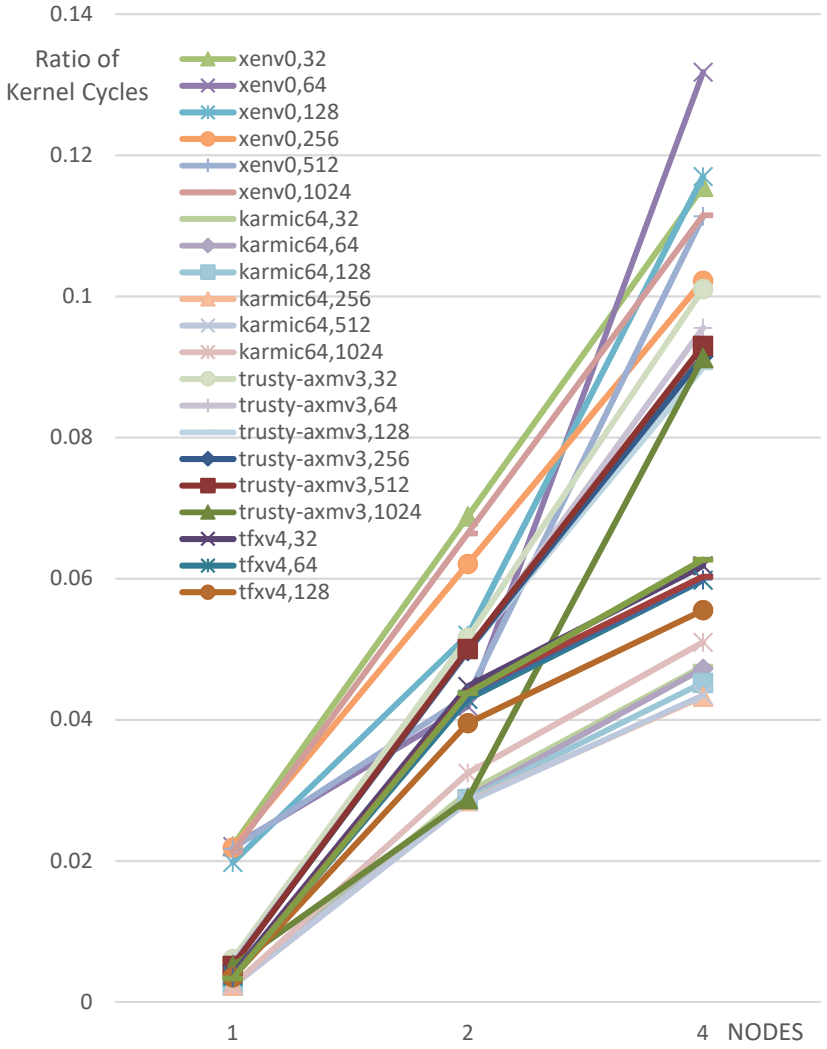


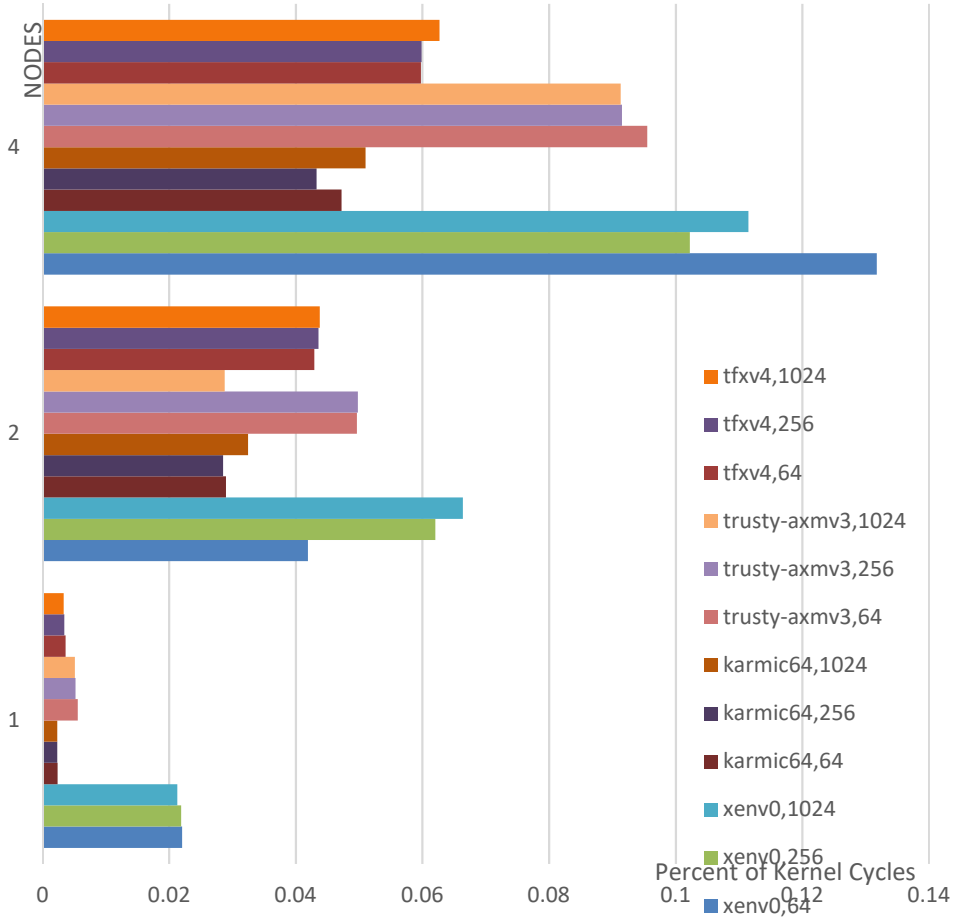


L2 Cache Miss Rate



Ratio of Kernel Cycles vs Total Cycles





T312

AXIOM

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

PROJECT ID: 645496

Roberto Giorgi –- AXIOM project --- http://www.axiom-project.eu

Scalable Embedded Systems: Towards the Convergence of High-Performance and Embedded Computing

University of Siena
(Coordinator Partner)

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EVIDENCE
EMBEDDING TECHNOLOGY

herta security

FORTH
Institute of Computer Science

SECO

VIMAR
energia positiva