Exascale-class applications on the reconfigurable EuroEXA prototype: lessons learnt and progress so far

Presenter – Paul Carpenter
Barcelona Supercomputing Center
paul.carpenter@bsc.es

Leader WP2 Applications and Software Technical Manager, EuroEXA
Leader WP3 Enablement of Software Compute Node, ExaNoDe

© 2017–2019 EuroEXA and Consortia Member Rights Holders
This research project is supported by the European Commission under the "Horizon 2020 Framework Programme" with grant number 754337

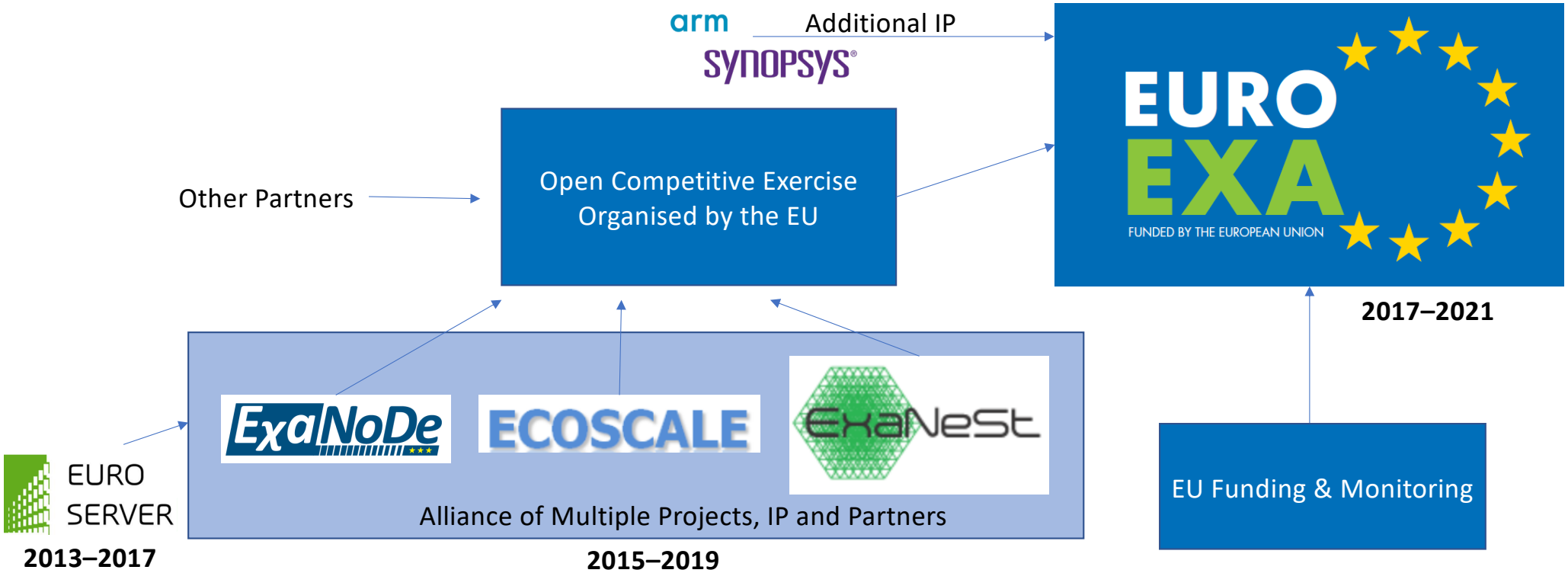
EuroEXA

- Co-design and demonstrate 1 PF+ testbed in operational environment

- Technologies developed in EUROSERVER, ExaNoDe, ExaNeSt, ECOSCALE projects
  - FPGA acceleration
  - UNIMEM (global address space)
  - Hardware memory compression

- Deliver available performance to full-scale production applications
- Achieve balance between compute resources and demands of applications

EuroEXA in context



EuroEXA partnership

- H2020 FETHPC-01-2016
- September 2017 – February 2021 (3.5 years)
- Budget €20 M

EuroEXA path: prototypes

- Three testbeds to be deployed at STFC, Daresbury, UK

Testbed 1

50 nodes of new technology for **software development**

Built and installed

2018

Testbed 2

~500 nodes and new infrastructure technologies to **test scaling**

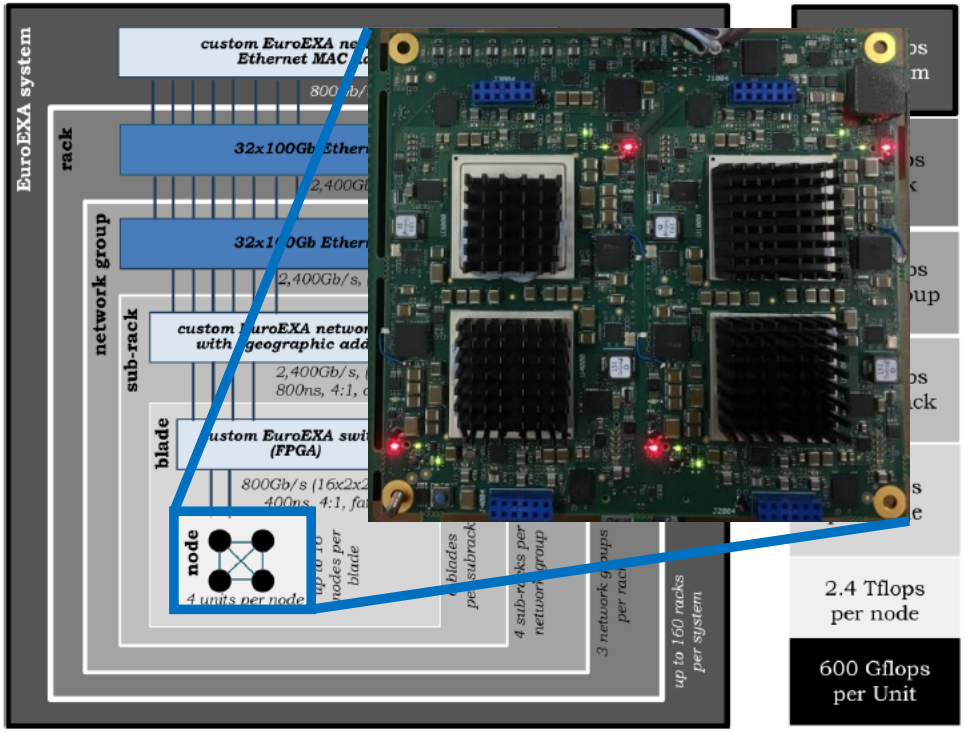
2019

Testbed 3

Test **new node and processor technologies** that will ultimately project to exascale

2020

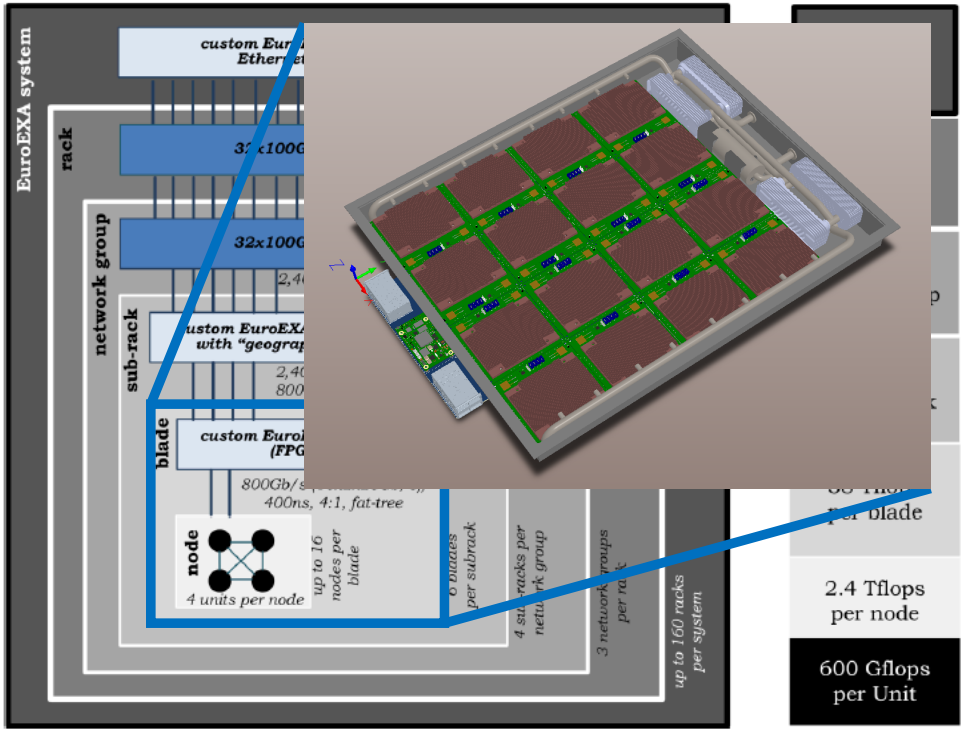
System architecture and technology: Testbed 1 compute node



Initial technology from FORTH (QFDB)

- 12 cm x 13 cm board
- Xilinx Zynq UltraScale+
  - 4 ARM Processors and 4 FPGA Accelerators
- M.2 SSD
- 4 x SODIMMs + Onboard RAM
- Daughterboard style
- 160 Gb/s of I/O

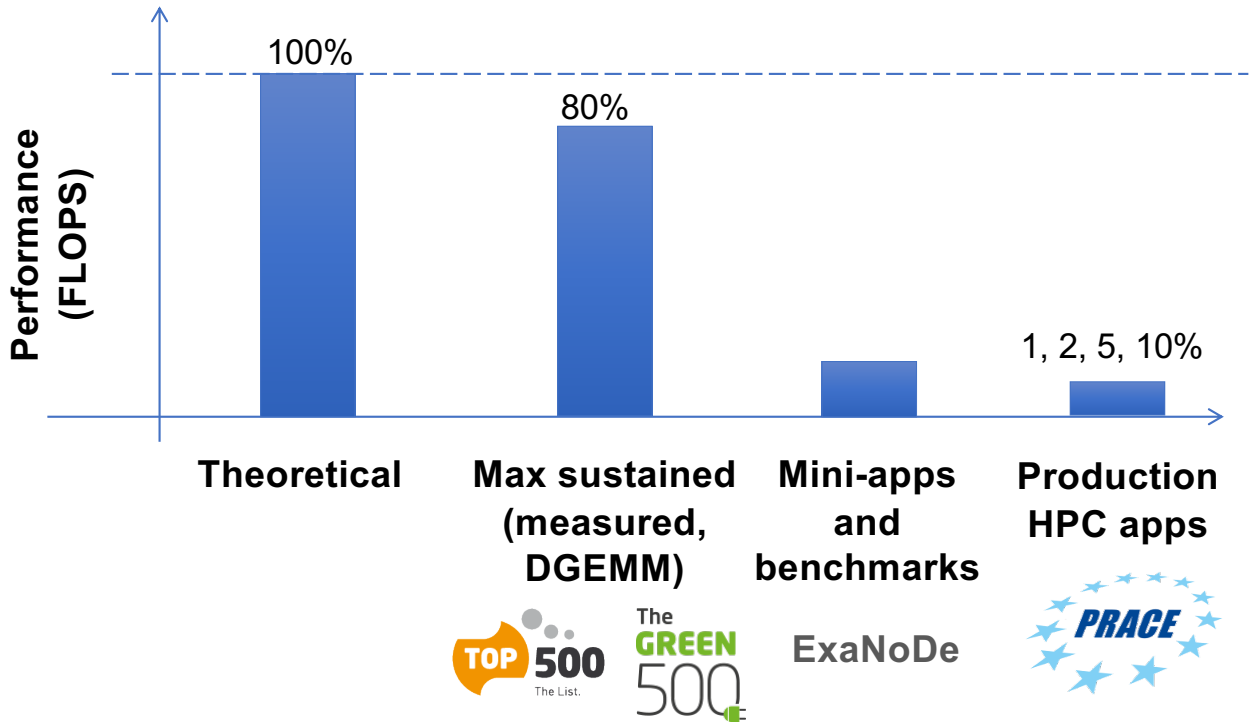
System architecture and technology: Blades



Technology from Iceotope:

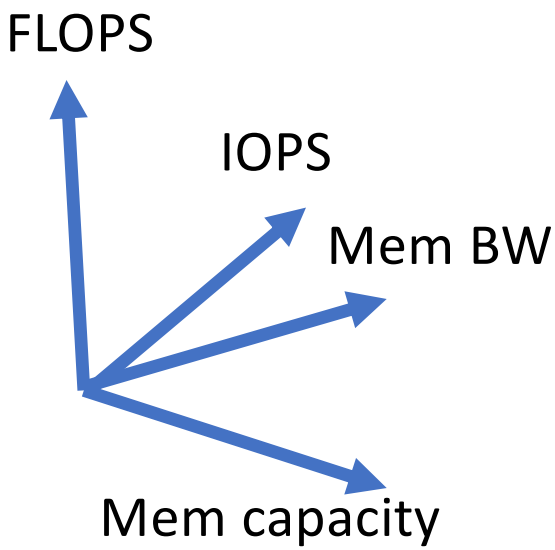
- 16 Node half depth 1u chassis
- 2 x 3.2 kW per u (back2back)
- Total Liquid Cooling technology
- 48 V DC distribution
- Hot water out, chiller-less operation

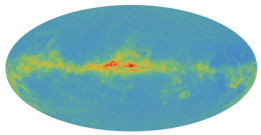
Why focus on applications? Theory vs. Practice



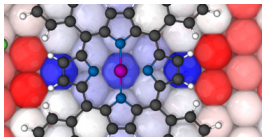
- Real apps not just LINPACK
- e.g. TaihuLight
  - 93 PF and 6 GF/W for HPL
  - But 0.3% of peak performance for more realistic HPCG

EuroEXA: co-design, demonstration and evaluation using exascale-class apps

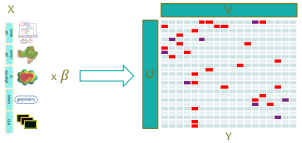




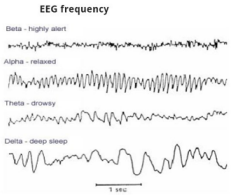
AVU-GSR



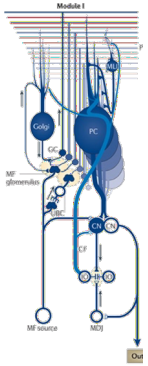
Quantum Espresso



SMURFF

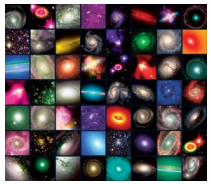


Neuromarketing

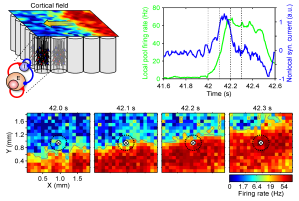


InfOli

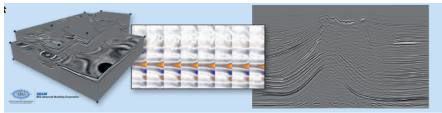
NEMO



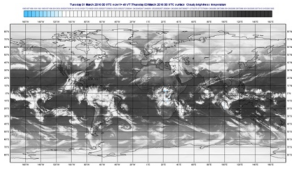
Astronomy image classification



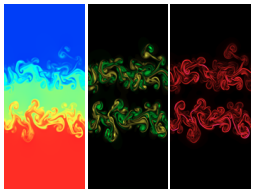
NEST/DPSNN



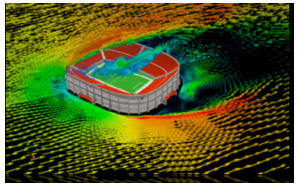
FRTM



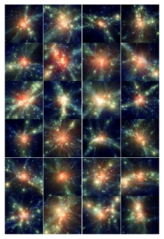
IFS



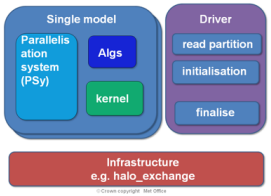
LBM



Alya



GADGET

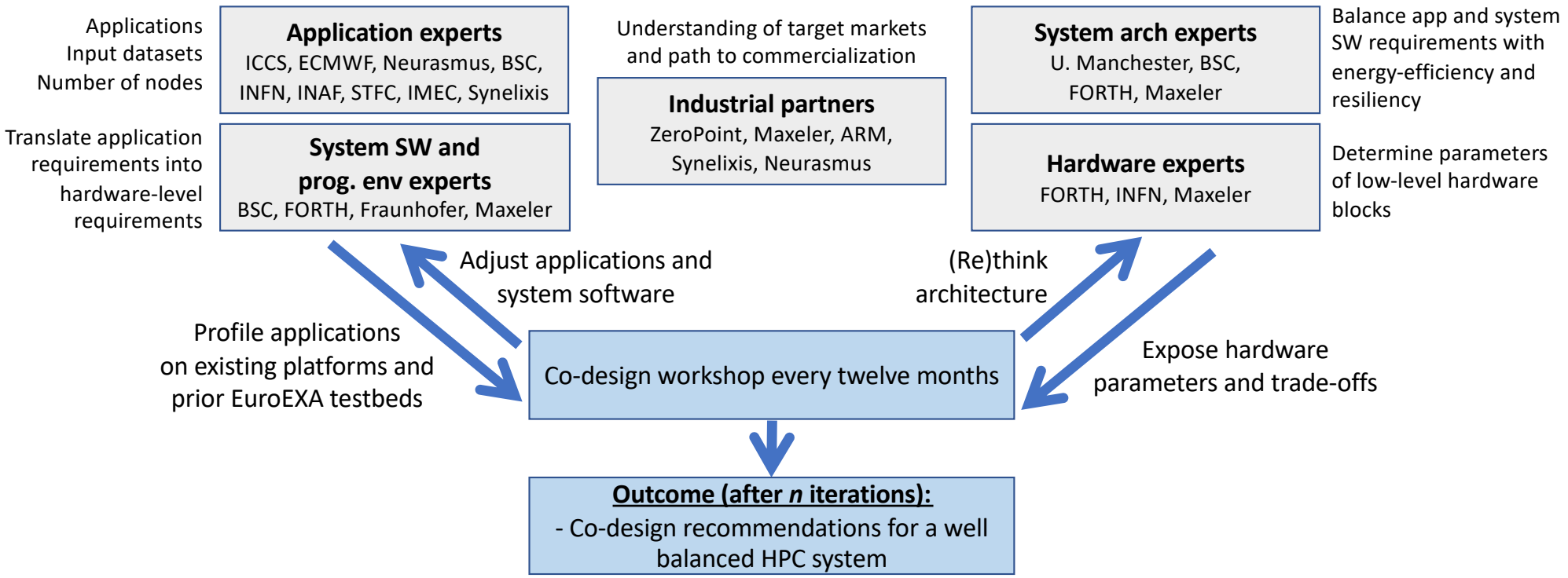


LFRic

Why these 14 applications?

- 10 strong European application partners
- 14 applications covering three important application domains
  - Climate and weather (LFRic, NEMO, IFS)
  - Physics and energy (LBM, Alya, GADGET, AVU-GSR, FRTM, Astronomy image classification, Quantum Espresso)
  - Life science and bioinformatics (NEST, Neuromarketing, InfOli, SMURFF)
- These domains will require exascale computing in near future
- ALYA, GADGET, NEMO and Quantum ESPRESSO part of PRACE UEABS
- Applications for porting, demonstration, evaluation, extrapolation, co-design

Co-design for EuroEXA architecture



Co-design organization

- Understand scope
  - Is it real co-design? Or just pretending to be co-design?
  - What parameters are really open for co-design? hardware input
- Form Co-design Working Group with all needed expertise
- Conference calls and face-to-face plenary co-design workshop
- Recommendations document and deliverable to commission
- **Decision** made by project as a whole

How to quantify trade-offs

Quantify performance impact of the design parameter: ***Perf =f(L2 cache size)?***

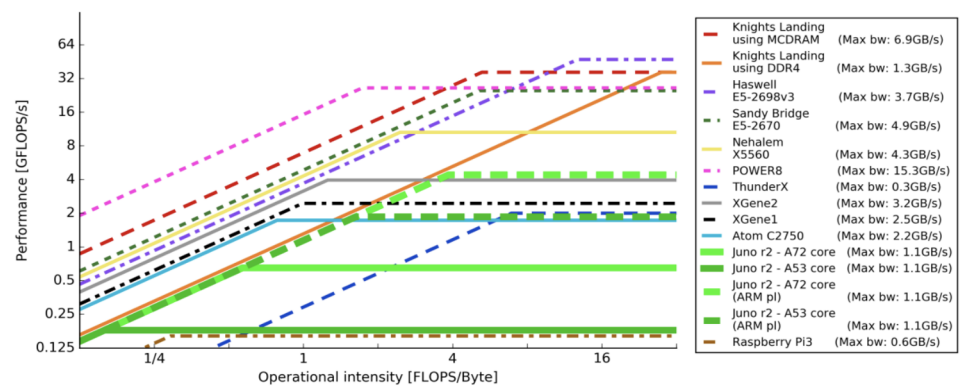
How can we do this?

1. Simulation
   a. Detailed, cycle-accurate hardware simulators *vs.* simplified models
   b. Full application *vs.* Kernels
2. Analytical models
   a. Level of detail?
   b. Full application *vs.* Kernels
3. Profiling
   a. Application execution on real systems (EuroEXA-like?)
   b. Full application *vs.* Kernels
   c. Performance counters, binary instrumentation tools
4. State-of-the-art analysis
   a. What do others (HPC domain) say about this design parameter?

State of the art analysis: Experience from ExaNoDe project

- Based on experience from ExaNoDe project
  - Summarized in public ExaNoDe D2.2 Report on the ExaNoDe Design Guidelines
- http://exanode.eu/wp-content/uploads/2017/04/D2.2.pdf

- Sustained vs theoretical FLOPS
- Sustained vs theoretical memory BW
- Memory access latency
- FLOPS/byte



Co-designed parameters

- Cache sizes for ASIC based on simulation
- ZeroPoint MaxiMem hardware memory compression
  - Design parameters and trade-offs
  - Applications and special simulator
- UNIMEM parameters
- FPGA and SoC interfaces
- Network topology and trade-offs
  - => Second year of project

CONFIDENTIAL

Experience and model of ported applications: Accelerator FPGA: *ZU9 is too small!*

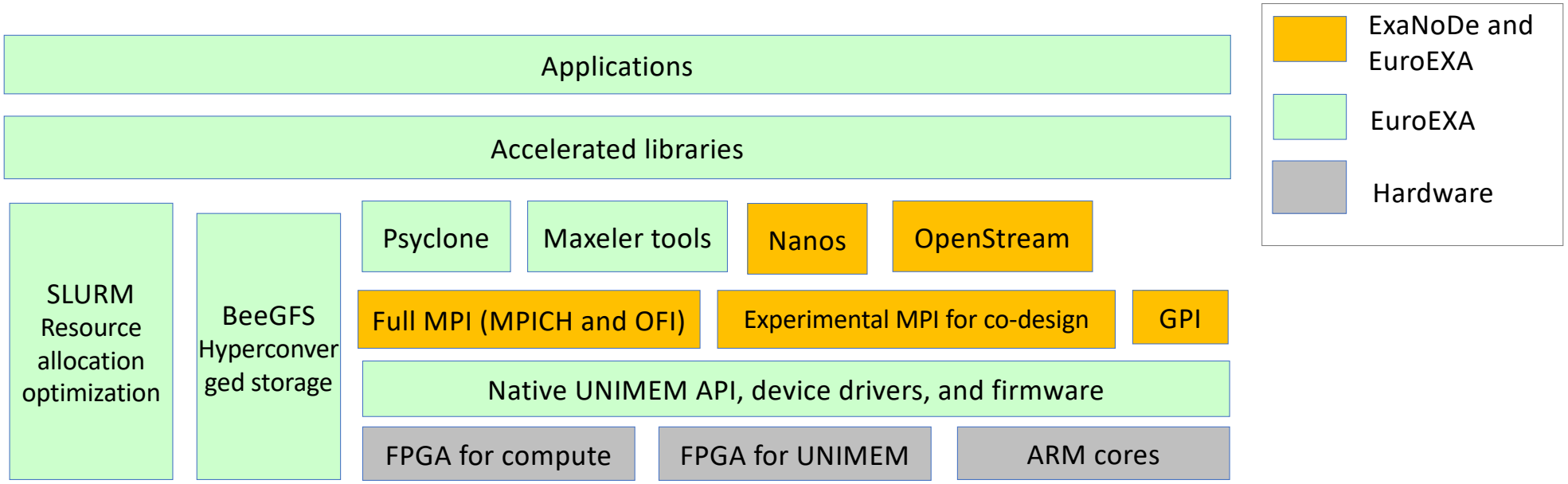
Original plan: QFDB
Four Xilinx Zynq UltraScale+ ZU9P

New proposal: CRDB
Xilinx Zynq UltraScale+ ZU9P for interconnect
Xilinx Zynq UltraScale+ VU9P for acceleration

Advantages of larger FPGA

- More FPGA resources to help compute-heavy applications
- Higher I/O bandwidth and memory bandwidth
- HPC applications already demonstrated on VU9P
  - Maxeler: BQCD, NEMO, QE, SpecFEM3D
- Compatibility with Maxeler and Amazon AWS EC2 F1
- Lower proportion of compute FPGA used for interconnect
- Removes heterogeneity among compute FPGAs

EuroExa software stack



Motivation: Programming environment

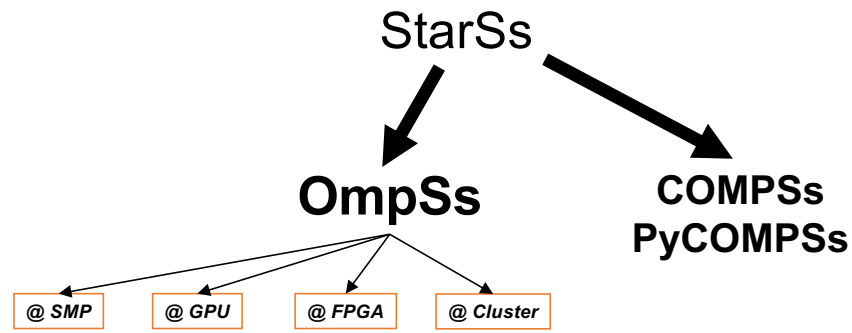
- Complex machines, specialization, heterogeneity
  - Deep memory hierarchies
  - Accelerators: GPUs, FPGAs, vector, TPU, etc.
  - Timing variability: thermals, DVFS, near threshold, interconnect energy saving
  - Manual tuning of data placement, scheduling, explicit communication
    => long development times, uneconomic, correctness and performance issues
  - Users often do not know how many nodes to ask for
- Complex execution environments and workflows
  - Interactive and urgent computing
  - Virtualization => timing variability?
  - Dynamic resource sharing
- Complexity + variability = divergence between mental models and actual behavior
- Need standardized programming model/language

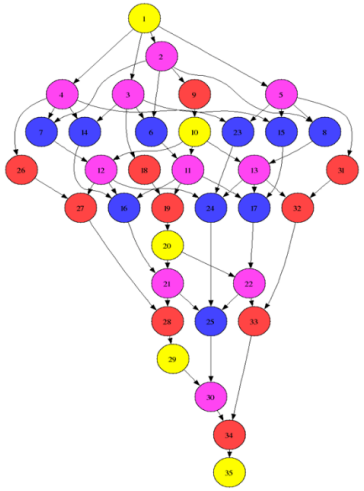
EuroEXA programming environment

- EuroEXA has unique hardware capabilities
  - UNIMEM communications
  - FPGA acceleration
- Applications will target portable programming models / communication libraries
  - MPI, GPI, OpenStream, OmpSs, MaxJava and PSyClone
- Optimization for EuroEXA architecture done by programming environment
  - Building on the work in ExaNoDe project
  - EuroEXA brings greater maturity, especially in FPGA support
  - Tuning and optimization through EuroEXA's full applications

OmpSs parallel programming model

- StarSs family: key concepts
  - Sequential task-based program
  - View of a single address/name space
  - Happens to execute in parallel: automatic runtime computation of dependencies
- Productivity and portability





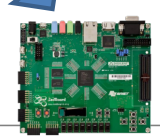
- **Long-term research at BSC, Barcelona**
  - 10+ years' investment
  - Tasking, task dependencies, heterogeneity, multiple address spaces, etc.
- **Pushing ideas to OpenMP standard**
  - Task dependencies in OpenMP 4.0
  - Priorities in OpenMP 4.5
- **Mercurium compiler**
- **Nanos runtime system**

OmpSs supports SMPs, GPU, clusters, even FPGAs

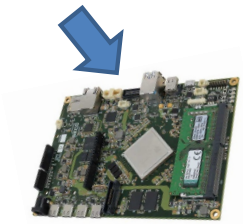
- FPGA support uses Vivado HLS as target

```
#define BS 128

void matrix_multiply(float a[BS][BS], float b[BS][BS],float c[BS][BS])
{
#pragma HLS inline
  int const FACTOR = BS/2;
#pragma HLS array_partition variable=a block factor=FACTOR dim=2
#pragma HLS array_partition variable=b block factor=FACTOR dim=1
  // matrix multiplication of a A*B matrix
  for (int ia = 0; ia < BS; ++ia)
    for (int ib = 0; ib < BS; ++ib) {
#pragma HLS PIPELINE II=1
      float sum = 0;
      for (int id = 0; id < BS; ++id)
        sum += a[ia][id] * b[id][ib];
      c[ia][ib] += sum;
    }
}
```



Zynq-7000 Family
2x Cortex-A9 cores + FPGA (32-bit platforms)



SECO AXIOM Board
Zynq U+ XCZU9EG-ES2
4x Cortex-A53 cores + FPGA (64-bit platforms)

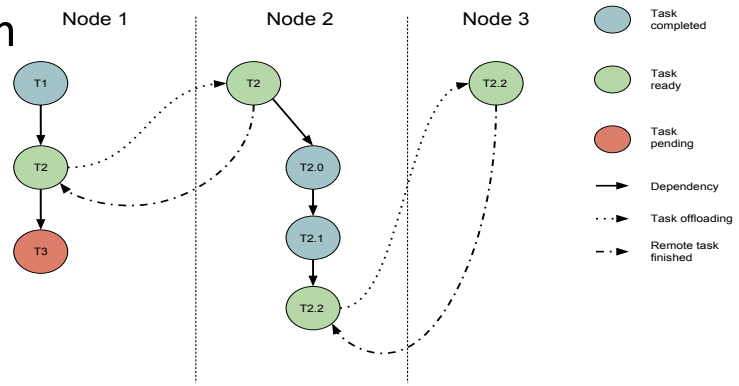
Trenz Electronics Zynq U+
TE0808 XCZU9EG-ES1

Towards
Discrete FPGAs

Xilinx ZCU 102
XCZU9EG-2FFVB1156

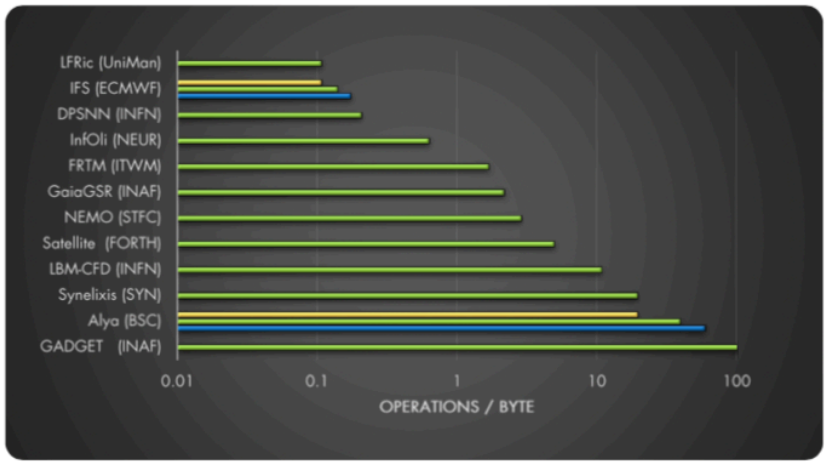
OmpSs Cluster

- Distributed memory task-based programming model
  - Parallelism defined through tasks and dependencies via OmpSs programming model
  - Tasks are transparently offloaded to different nodes
  - Scheduling based on locality and load balancing
  - Task graphs distributed across nodes
- Common virtual memory layout
- Each node has an instance of the Nanos 6 runtime system
  - Start program with mpirun
  - Main function is a task on one core of one node
- Task creation overhead kept low via nested tasks



EuroEXA application requirements

- Extensive application requirements exercise in first six months of project



Arithmetic intensity

Memory footprint per node

Porting so far

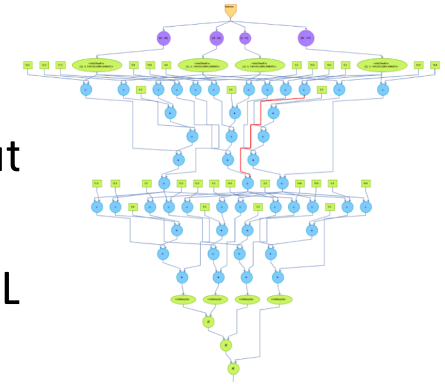
- First step: porting to ARM (straightforward)
- Porting kernels to FPGA
  - Currently Testbed 1 prototype using Zynq UltraScale+
  - With a variety of programming models
- Challenges
  - Learning curve
  - Compilation time
  - Double precision in FPGAs
  - Integration!

Performance: SMURFF inference for chemogenomics

- Fixed point refinement:
  - 16-bit fixed
  - varying integer part

- Compared to CPUs:
  - FPGA (ZU9): 480 GMulAdd/s
  - CPU: 35 GMulAdd/s

- Fixed Point Pipeline
  - Single cycle throughput
  - 80 cycle latency
  - Uses 90% of FPGA MUL units



- Floating Point Pipeline
  - Single cycle throughput
  - 1057 cycle latency
  - 1333% of FPGA MUL units

Astronomy image classification

- CNN inference for on-board satellite
- Comparison against space-qualified Kepler GPU, similar technology to FPGA
  - (More recent GPUs 4x better energy efficiency)

Table 4: Architecture comparison with GPU

	ZCU-102	QFDB	K2200
Clock Frequency(MHz)	250	250	1124
Throughput(Images/s)	1084	4334	2000
Latency(s)	0.003	0.003	0.06
GFLOPS	66.1	265	122.5
Total On-chip Power(Watt)	11.8	47.3	300
Energy Consumption(Joule)	108.8	109.1	1.5K
Images/Joule	91.6	91.6	6.66

Conclusions and vision

- Co-design and demonstrate 1 PF+ testbed in operational environment
- Vision is exascale machine for Europe

- High performance for full-scale production applications
- Co-design led to SoC design choices and change in FPGA
- Complete software stack and optimized portable programming models
- Ongoing porting of applications to the architecture