

BENCHMARKING AND CO-DESIGN examples from the DEEP and EPI projects

26.09.2023 I ADAC Symposium I Estela Suarez (FZJ/JSC & UniBonn)



Mitglied der Helmholtz-Gemeinschaft

- Co-design and Benchmarking
- Experiences
 - DEEP Projects → System Level
 - EPI \rightarrow Processor Level
- Lessons Learned
- Summary



Co-design – my personal definition

- Study interaction between
 - application code,
 - system **software**,
 - hardware components,
 - and system architecture
- to find the modifications at each of those four levels
- that bring overall best
 - performance and
 - energy efficiency



CENTRE

Role of Benchmarking in Co-design

- Characterize applications through representative
 - Synthetic benchmarks
 - Mini-applications
 - Large scale use cases
- Evaluate different software versions/options
- Compare different hardware devices
 - Run benchmarks on hardware prototypes and systems
 - Model/simulate different architectural features
- Determine best combination of resources for given workload mix
 - Diverse application portfolio (not only an individual use-case)





- Co-design and Benchmarking
- Experiences
 - DEEP Projects → System Level
 - EPI \rightarrow Processor Level
- Lessons Learned
- Summary





- Focus: system-level architecture
 - Modular Supercomputing Architecture

Project Activities

- Hardware prototyping
- System Software development
- Application porting
- Application Selection (part of proposal)
 - 6-7 codes and partners
 - Large scale codes plus benchmarks
 - Variety of scientific fields



- Focus: chip-level microarchitecture
 - Arm CPU and RISC-V accelerator

Project Activities

- Chip design, emulation & tape-out
- Low-level Software (e.g. compilers)
- Benchmarking
- Application Selection (during project)
 - 16 partners, >40 codes
 - Benchmarks, mini-apps, kernels
 - Variety of scientific fields



	Projects	European Processor Initiative
Project Focus	- System-level architecture (Modular Supercomputing Architecture)	- Processor development (CPU and accelerator)
Project Activities	 Hardware prototyping System software development Application porting and benchmarking 	 Processor design Processor modelling Processor prototyping Low-level software (e.g. compiler) Application porting and benchmarking
Application & Benchmark Selection	 Conscious selection, part of proposal 6-7 codes and partners Variety of fields Additionally synthetic benchmarks 	 During first months of project >40 codes (large codes, mini-apps, synthetic benchmarks) 16 partners
Co-design focus	- Selection and balance of system components (CPU skew, accelerator choice, number of nodes, etc.)	 Finding impact of microarchitecture features onto application performance Simulation based (gem5, VPsim, MUSA)
Benchmarking strategy	 Use-cases of large-scale applications Some synthetic benchmarks 	 Synthetic benchmarks and mini-apps Some large scale codes



- Co-design and Benchmarking
- Experiences
 - DEEP Projects → System Level
 - EPI \rightarrow Processor Level
- Lessons Learned
- Summary





Co-Design of a Hardware Prototype



Fixed parameters

- System architecture: MSA
- Design targets:
 - **Cluster**: highest Byte/Flop ratio
 - Booster: highest energy efficiency
 - DAM(*): highest flexibility & memory
- Installation time: 2020
- Budget: ~3.5 MEuro
- Providers:
 - Integration: Megware
 - Interconnect: EXTOLL
- (*)DAM = Data Analytics Module



Design choices

- Node number in each module
 - Relative size of modules
- Node design
 - **Cluster**: CPU type and SKU (#cores, DDR size, etc.)
 - Booster: CPU type and accelerator (type and #)
 - Data Analytics Module:

CPU and accelerator type(s)



Application-driven HW+SW developments





Computation vs. Communication balance ratio #cores/memory bandwidth

-

-

-

• 5) Re-run step (3 and 4) on final system and compare with baseline

Communication and $I/O \rightarrow$ memory and network bandwidth

- <u>Note</u>: code itself has also changed / improved in between

12

Benchmarking Steps

to give Co-design input

- 1) Define use cases representative for each application
 - Including input data sets
- 2) Integrate codes in benchmarking environment
 - JUBE: <u>https://www.fz-juelich.de/en/ias/jsc/services/user-support/jsc-software-tools/jube</u>

4) **Performance analysis** and measurement **→** extract quantitative co-design input

Performance and scaling behaviour for each application part \rightarrow # nodes/module

Compute intensive kernels \rightarrow ratio between CPU and acceleration parts

• 3) Run use-cases on representative hardware







DEEP-EST prototype





DEEP-EST Prototype 55 Cluster + 75 Booster + 16 Data Analytics 100 Gbit Extoll + InfiniBand + Eth 800 TFlop/s



https://www.fz-juelich.de/en/ias/jsc/systems/prototype-systems/deep_system



DEEP-EST prototype





DEEP-EST Prototype 55 Cluster + 75 Booster + 16 Data Analytics 100 Gbit Extoll + InfiniBand + Eth 800 TFlop/s



https://www.fz-juelich.de/en/ias/jsc/systems/prototype-systems/deep_system



14

Example: xPic (Space Weather Simulation)

- Field solver: 6× faster on Cluster
- **Particle solver**: 1.35 × faster on **Booster**
- **Overall performance gain:**

1× **28% × gain** compared to Cluster alone **node** 21% × gain compared to Booster alone

8× **38% × gain** compared to Cluster only nodes **34% × gain** compared to Booster only

 3%-4% overhead per solver for C+B communication (point to point)





KATHOLIEKE UNIVERSITEI



4096

2048

#cells per node

#particles per cell

Compilation flags

-xMIC-AVX512 (Booster)		
ps	JÜLICH	

-openmp, -mavx (Cluster)



- Co-design and Benchmarking
- Experiences
 - DEEP Projects \rightarrow System Level
 - EPI \rightarrow Processor Level
- Lessons Learned
- Summary



EPI: Co-design and Validation (Benchmarking based)





European Processor

epi

Initiative

EPI Co-design Scope

- Focus on giving quality feedback to HW/SW designers
 - co-design between application developers and chip designers
- Multi-level suite of benchmarks
 - from very low- synthetic benchmarks to high-level applications
- Multi-level models & simulators
 - Analytical models, high level
 - Simulators (e.g. gem5, VPSim, MUSA)
 - Reference platforms (e.g. A64FX, Graviton-3)



EPI Benchmark Suite



>40 codes, in the fields:

- Biophysics
- Biology/Medicine
- Earth Sciences/Climate
- HEP & Fusion
- Material Sciences
- CFD
- Hydrodynamics
- PDE
- Image / Media

- Automotive
- Cryptography
- HPDA
- Machine Learning
- Deep learning
- Cloud
- Data Base
- Reference benchmarks: (HPL, HPCG, Stream, DGEMM...)





European Processor

Initiative

EPI Chip Simulation

- Goal
 - Understand impact of architectural parameters on application performance

• Simulations of chip microarchitecture

- Detailed representation of chip elements (CPU, caches, network-on-chip, memory hierarchy)
- Capability to change features

JSC contributions

- Develop models (gem5) that accurately represent the EPI Rhea platform (Arm-based CPU)
- Analyse design trade-offs with benchmarks
- Give feedback to chip developers







Example: benchmarking on gem5 simulator

• Prefetcher evaluation with (by N.Ho, JSC)





- L.Zaourar et al., SC ws. proceedings *"Multilevel simulation-based co-design of next generation HPC microprocessors"*, http://hdl.handle.net/2128/29249
- Mitglied der Helmholtz-Gemeinschaft

• Roofline model comparisons (by A.Portero, JSC)





gem5

A64FX

Graviton3

Mitglied der Helmholtz-Gemeinschaft

Suarez – 2023

22

Example: Application Evaluation

Goal: evaluate the MiniGhost benchmark:

- gem5 (ARM) model
- AWS EC2 Graviton2 (Neoverse N1, ARMv8.2)
- AMD Epyc x86
- Intel Xeon x86

Conclusions

- MiniGhost: good scalability with # of cores
- gem5 model:
 - Similar performance than off-the-shelf Intel/AMD CPUs
 - Underperforms (2×) against similar microarchitecture (Graviton-2 / Neoverse N1)







- Co-design and Benchmarking
- Experiences
 - DEEP Projects → System Level
 - EPI \rightarrow Processor Level
- Lessons Learned
- Summary



Lessons Learned

Challenges:

- Technical / Practical
 - Hard to extract <u>quantitative</u> co-design input
 - Even harder for full workload mixes
 - Lack of clear baseline reference
 - $_{\circ}$ $\,$ codes, system-SW and -HW evolve simultaneously
 - Hard to pin-point & quantify co-design effect
 - Design decisions strongly cost-driven
 - Limited time-frame to apply co-design input

Strategic / Logistic / Organisational

- Application developers are rewarded for scientific runs (not for benchmarking or co-design)
- Code-selection not always by pure scientific/technical criteria
- Some details protected by commercial IP

Opportunities:

- Technical / Practical
 - Potential for optimisations in performance, energy efficiency, and scientific throughput
 - Tailor system to application portfolio
 - Enable own approaches to system architecture
 - Learn and understand each others language (from application to hardware design)
- Strategic / Logistic / Organisational
 - Real impact on product development roadmap
 - Real impact on application porting and performance improvements
 - Target open source simulation framework, with open benchmark suite incl. workload mixes



- Co-design and Benchmarking
- Experiences
 - DEEP Projects → System Level
 - EPI \rightarrow Processor Level
- Lessons Learned
- Summary







- Benchmarking is a critical tool for Co-Design (both at system and component level)
- Challenges on extracting quantitative requirements from applications and pin-point the impact of individual inputs on the final design
- **Opportunities** for performance and energy efficiency **improvements**, if we invest on and **apply** a **systematic**, **data-driven**, **community effort**



THANK YOU!



www.deep-projects.eu

@DEEPprojects

@deep-projects

The **DEEP Projects** have received funding from the European Commission's FP7, H2020, and EuroHPC Programmes, under Grant Agreements n° 287530, 610476, 754304, and 955606.

The EuroHPC Joint Undertaking (JU) receives support from the European Union's Horizon 2020 research and innovation programme and Germany, France, Spain, Greece, Belgium, Sweden, Switzerland





www.european-processorinitiative.eu

@EuProcessor

European Processor Initiative

The **EPI project** has received funding from the European High Performance Computing Joint Undertaking (JU) under Framework Partnership Agreement No 800928 and Specific Grant Agreement No 101036168 EPI-SGA2.

The JU receives support from the European Union's Horizon 2020 research and innovation programme and from Croatia, France, Germany, Greece, Italy, Netherlands, Portugal, Spain, Sweden, and Switzerland.



