

BEAST Lab Preliminary Meeting

LRZ

Dr. Josef Weidendorfer, josef.weidendorfer@lrz.de

LMU

Minh Thanh Chung, minh.thanh.chung@ifi.lmu.de

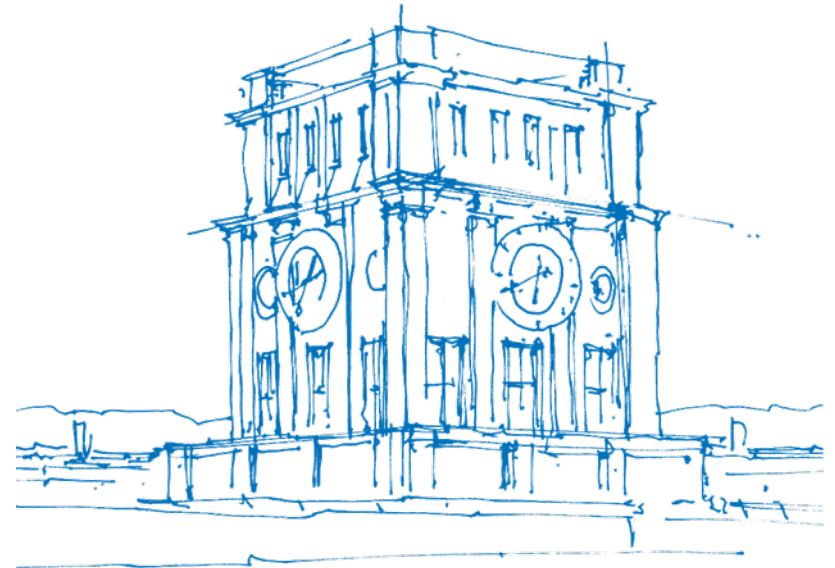
Dr. Karl Fűrlinger, karl.fuerlinger@ifi.lmu.de

TUM

Vincent Bode, vincent.bode@tum.de,

Dennis-Florian Herr herrd@in.tum.de

Bengisu Elis, bengisu.elis@tum.de



TUM Uhrenturm

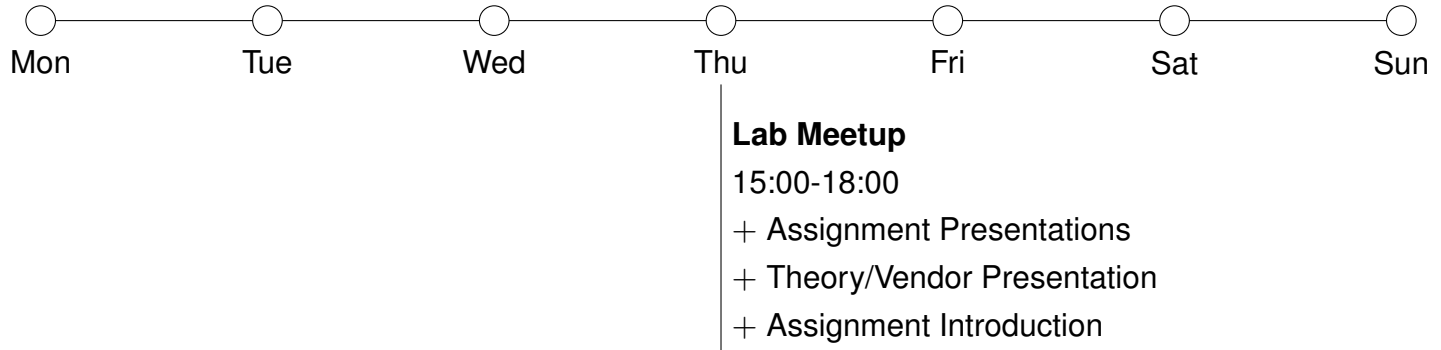
Table of Contents



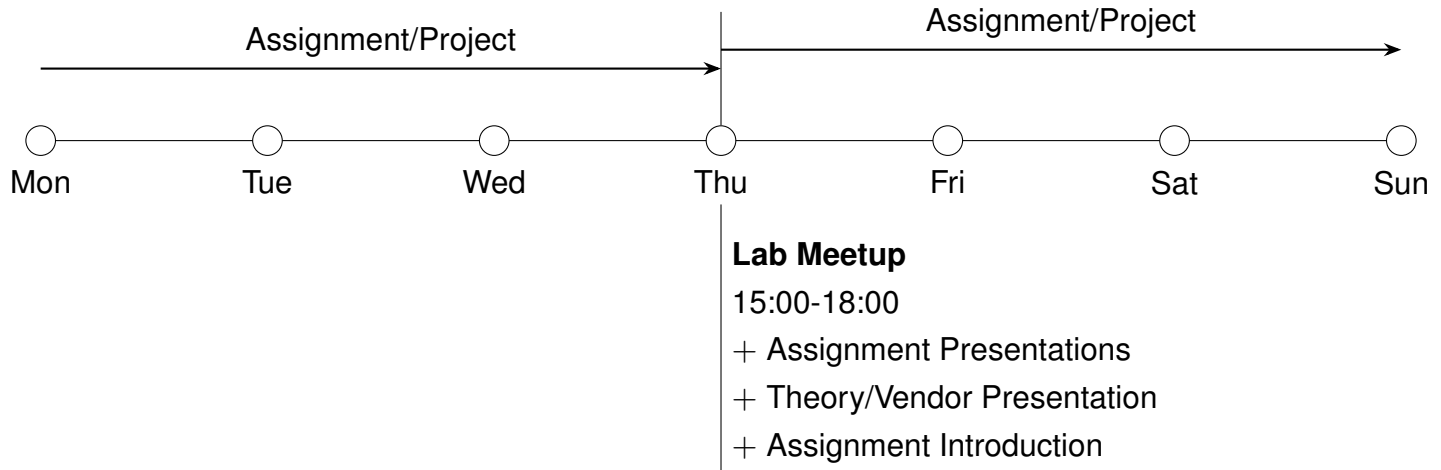
Course Organization

Introduction to BEAST

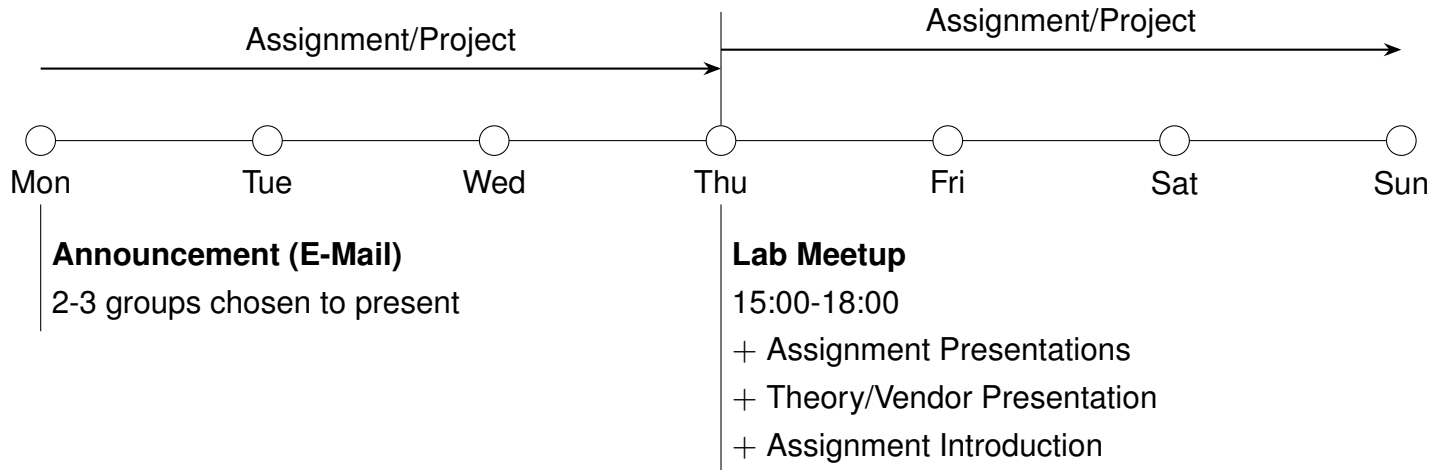
Weekly Schedule



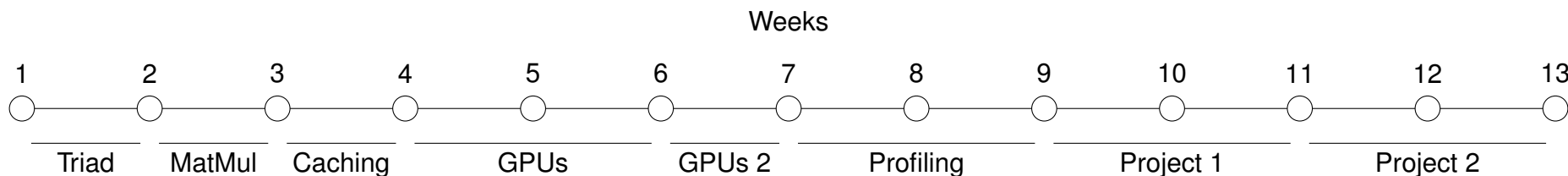
Weekly Schedule



Weekly Schedule



Tentative Semester Overview



Organization

- Note: This is preliminary based off of last semester and is subject to improvements
- 6 Assignments
 - 1 week each (except on holidays)
- 2 bigger Projects
 - 2 weeks each
- Student groups of 3 (Bachelor) or 2 (Master)

Previous Vendor Talks



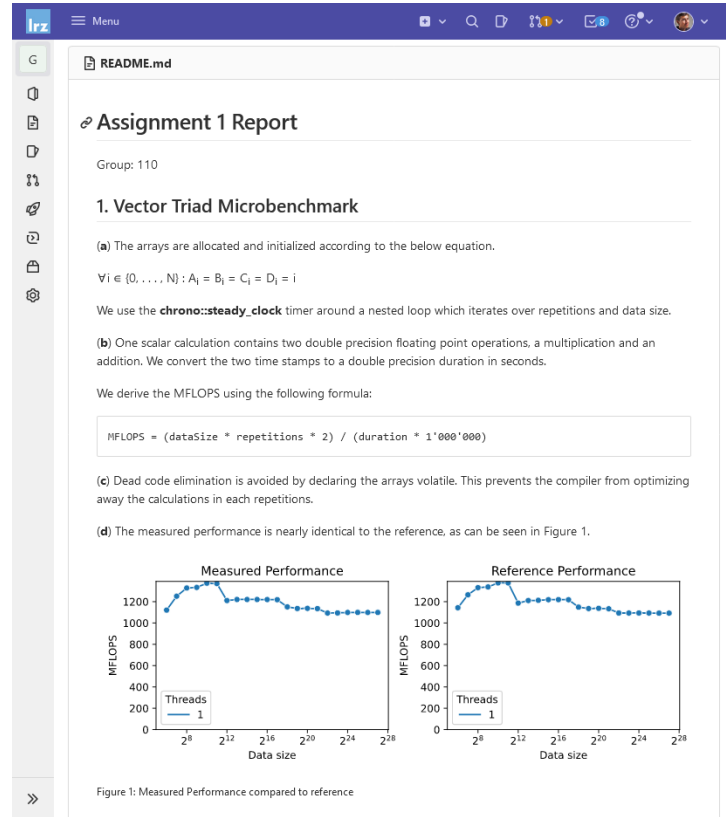
Deliverables/Grading

Git Repository

- **Assignment/Project Report** in Markdown
- Your Code
- CI Jobs (not graded)

Presentation

- No slides. Go through the report
- Talk about what you learned
- Get feedback from advisors



irz Menu

README.md

Assignment 1 Report

Group: 110

1. Vector Triad Microbenchmark

(a) The arrays are allocated and initialized according to the below equation.

$$\forall i \in \{0, \dots, N\} : A_i = B_i = C_i = D_i = i$$

We use the `chrono::steady_clock` timer around a nested loop which iterates over repetitions and data size.

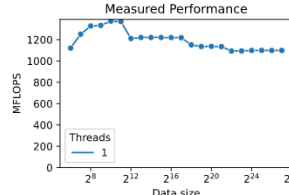
(b) One scalar calculation contains two double precision floating point operations, a multiplication and an addition. We convert the two time stamps to a double precision duration in seconds.

We derive the MFLOPS using the following formula:

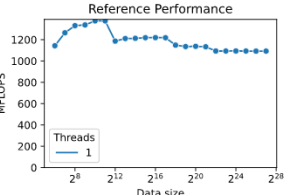
$$\text{MFLOPS} = (\text{dataSize} * \text{repetitions} * 2) / (\text{duration} * 1'000'000)$$

(c) Dead code elimination is avoided by declaring the arrays volatile. This prevents the compiler from optimizing away the calculations in each repetitions.

(d) The measured performance is nearly identical to the reference, as can be seen in Figure 1.



Measured Performance



Reference Performance

Figure 1: Measured Performance compared to reference

Next Steps

Register on Matching System

- We will prioritize you if you attended today
- Open until 27.07.2022
- Wait for announcement of matching results (04.08.2022)

Group Preferences

- Only after matching has ended
- Send us by e-mail (bengisu.elis@tum.de)
- No preferences submitted → we will match you

Attend Course Kickoff

- At university if everything goes according to plan
- We hope to see you there :)

The screenshot shows the 'matching.in.tum.de' website. At the top, there is a navigation bar with 'matching.in.tum.de', 'FAQs', 'Vincent Bode', and the TUM logo. A blue banner below the navigation bar states 'Matching SS22 starts on the 10.02.2022'. The main content area is titled 'Matching platform' and features a search bar and a 'Show 10 entries' dropdown. Below this is a table with columns for 'Title', 'Students Begin', 'Students End', 'Courses Begin', and 'Courses End'. The table lists three entries: 'Seminars SS2022', 'Practical Courses WS2022', and 'Practical Courses WS2022'. Below the table, it indicates '3 matching instances' and shows 'Showing 1 to 3 of 3 entries' with 'Previous' and 'Next' buttons. The bottom section of the screenshot shows a similar table but with the message 'No matching instances.' and 'Showing 1 to 1 of 1 entries'.

Up Next: Introduction to BEAST

Preliminary Meeting: BEAST Lab WS 22/23

July 21, 2022

Collaboration among 3 institutions

LMU
TUM
LRZ

LMU – MNM/Prof. Kranzlmüller
(Karl Furlinger, Minh Chung, Sergej Breiter)

TUM – CAPS/Prof. Schulz
(Bengisu Elis, Dennis-Florian Herr, Vincent Bode)

LRZ - Future Computing Group
(Josef Weidendorfer, Amir Raofy)

We want you to learn about **performance properties of current architectures**

- Be able to understand and explain performance effects seen from measurements
- Get a deeper understanding of current system designs (CPU / GPU)

Part 1: get started with small codes across systems

- We show key hardware design concepts + a parallel programming model (OpenMP)
- We give you typical small HPC code examples
- You run measurements of different scenarios across systems, compare / discuss results
- We all discuss results in weekly meetings, starting with presentations of groups

Structure:

Memory on CPU (Triad / Traversal) → Compute on CPU (MM) → ... on GPU → Tools

Focus: Experimental Evaluation



We want you to learn about **performance properties of current architectures**

- Be able to understand and explain performance effects seen from measurements
- Get a deeper understanding of current system designs (CPU / GPU)

Part 2: make use of gained knowledge

- We assign randomly one system to each group
- We give you some larger typical HPC code examples
- You tune the code to get best single-node performance (3 week time)
- We discuss intermediate/final experiences/results in weekly meetings

Evaluation of Single-Node Performance



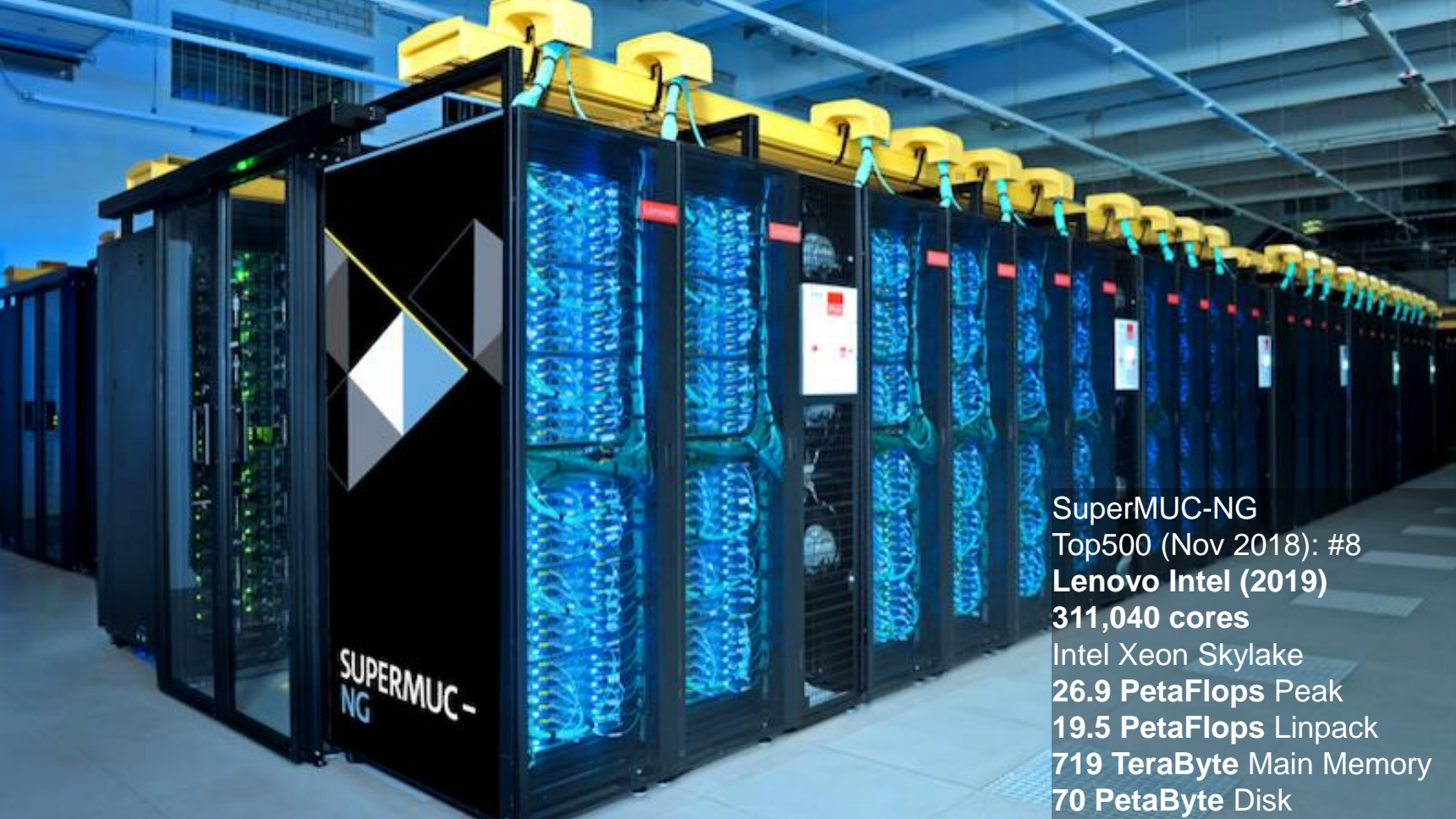
Target Architectures for the Lab

CPUs

- Intel Icelake (ISA: x86-64 + AVX512)
- AMD Rome (ISA: x86-64 + AVX2)
- Marvell ThunderX2 (ISA: ARM AArch64 + Neon)
- Fujitsu A64FX (ISA: ARM AArch64 + SVE)

GPUs

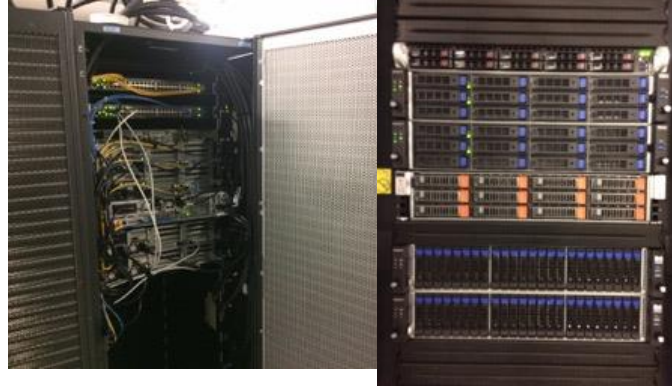
- NVidia V100
- AMD MI-100



SuperMUC-NG
Top500 (Nov 2018): #8
Lenovo Intel (2019)
311,040 cores
Intel Xeon Skylake
26.9 PetaFlops Peak
19.5 PetaFlops Linpack
719 TeraByte Main Memory
70 PetaByte Disk

BEAST – Bavarian Energy Architecture and Software Testbed

The LRZ Future Computing Testbed



- Help decide about next large system
 - Get experience on benefits of various future architectures for LRZ codes
 - Find best configuration: how much money to spend on compute / memory / network?
 - Enable migration planning: educate own staff / port LRZ tools / prepare courses
 - Support vendor collaboration
- Enable research studies on new technologies
 - Forward looking: LRZ services around future platforms, novel usage models
 - more experimental: FPGAs, AI accelerators, integration of heterogeneity (QC)
 - In partnership with selected researchers from Munich universities

Lot of work to do! Engage students for student work (BA, MA): This Lab!

The Testbed – Available Hardware



2 racks, each with 6 PDUs (for power measurements)

- Max power consumption per rack: 35 kW

Top to bottom (picture from last year)

- 3 switches (Infiniband 200Gb/s HDR), 2x 48port 1Gb/s Ethernet
- Login 1U “testbed.cos.lrz.de”
- **2x AMD Rome** GPU server 2U: “rome1” / “**rome2**”
- Storage 2U with homes
- **2x Marvell ThunderX2** GPU server 2U: “thx1” / “**thx2**”

Not shown:

- HPC CS500 Management 2U + **8 nodes A64FX** “cs1” – “cs8”
- **2x Intel IceLake** GPU server 2U: “ice1” / “**ice2**”



Intel Skylake (available as fallback)

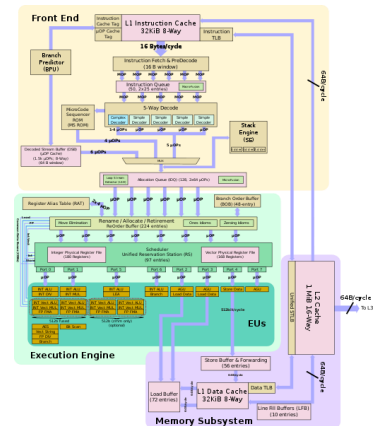
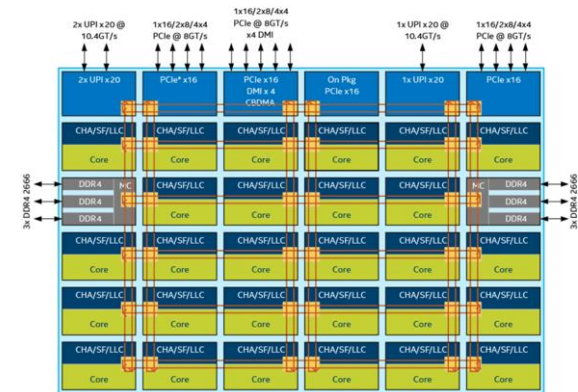


SuperMUC-NG

- Here: only Single-Node experiments
- Node
 - 2 sockets with Intel Skylake Xeon Platinum 8174
 - 2x 24 = 48 cores
 - 2x 512bit vector units per core (8 x DP FMA)
 - 2 threads per core ("Hyper-Threading")
 - 2.3 GHz base (currently: 2.5 GHz), 14nm
- 96 GB main memory

Links

- <https://doku.lrz.de/display/PUBLIC/Hardware+of+SuperMUC-NG>
- [https://en.wikichip.org/wiki/intel/microarchitectures/skylake_\(server\)](https://en.wikichip.org/wiki/intel/microarchitectures/skylake_(server))



Intel Icelake

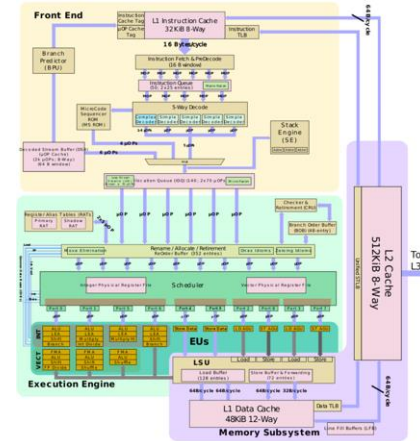


Two systems in BEAST

- 2 sockets Intel Xeon (Icelake) Platinum 8360Y
 - 2x 36 = 72 cores
 - 2x 512bit vector units per core (8 x DP FMA)
 - 2 threads per core ("Hyper-Threading")
 - 2.4 GHz base, Intel 10nm
- 512 GB main memory, 1.5 TB Optane NVRam

Links

- [https://en.wikichip.org/wiki/intel/microarchitectures/ice_lake_\(server\)](https://en.wikichip.org/wiki/intel/microarchitectures/ice_lake_(server))
- https://en.wikichip.org/wiki/intel/microarchitectures/sunny_cove



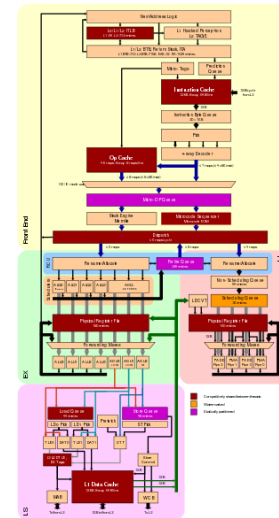
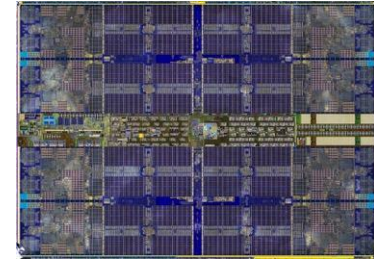
AMD Rome

Two systems in BEAST

- 2 sockets with EPYC 7742
- 2x 64 = 128 cores (“Zen2”)
 - Chiplet design: IO-Die + 8x CCX-Dies (2x 4-core)
 - 2x 256-bit vector units per core (4 x DP FMA)
 - 2 threads per core
 - 2.25 GHz base, TSMC 7nm
- 512 GB main memory
- 2x AMD Radeon MI-100 GPUs
 - 7nm, 32GB HBM, PCIe4

Link

- https://en.wikichip.org/wiki/amd/microarchitectures/zen_2



Marvell ThunderX2

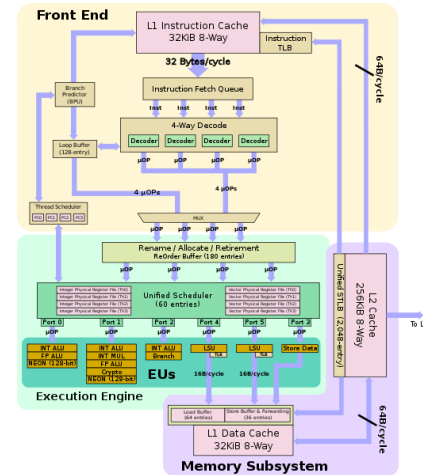
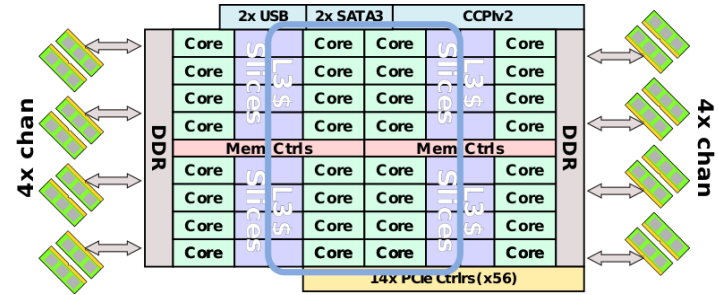


Two systems in BEAST

- 2 sockets with ThunderX2 CN9980
 - 2x 32 = 64 cores (“Vulcan”)
 - 128-bit vector units (2 x DP FMA)
 - 4 threads per core
 - 2.2 GHz base, 16nm
 - 512 GB main memory
 - 2x Nvidia V-100
 - Volta, 32GB HBM, PCIe3

Link

- <https://en.wikichip.org/wiki/cavium/microarchitectures/vulcan>



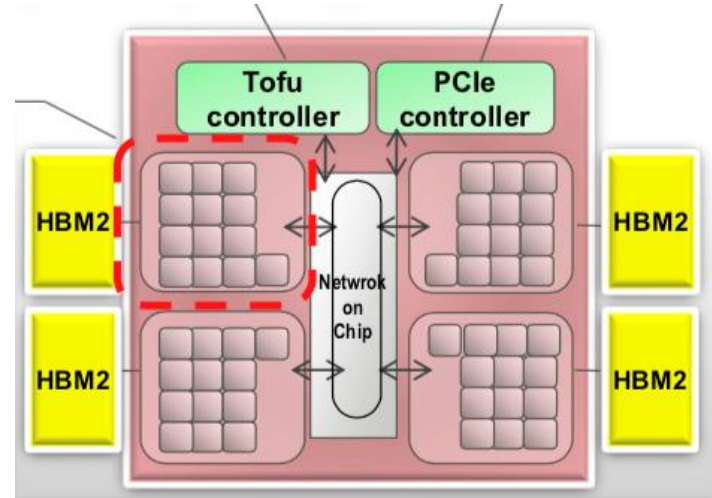
Fujitsu A64FX

HPE CS500 in BEAST

- 8 nodes with one A64FX CPU (“NSP1”)
- 48 cores per CPU
 - 2x 512bit vector units per core
 - 1.8 GHz, TSMC 7nm
 - 4 NUMA domains
- 32 GB HBM2

Link

- https://en.wikipedia.org/wiki/Fujitsu_A64FX



[Fujitsu: The 1st SVE Enabled Arm Processor: A64FX and Building up ARM HPC Ecosystem, 2019]



Leibniz Supercomputing Centre
of the Bavarian Academy of Sciences and Humanities