

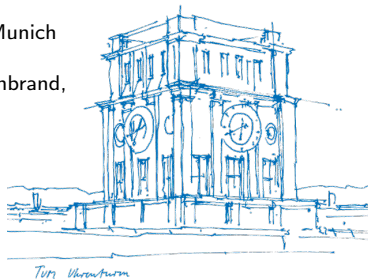
Seminar Efficient Programming of HPC Systems – Frameworks and Algorithms – Kick-off

Erwin Laure

Max Planck Computing and Data Facility & Chair of Computer Architecture and
Parallel Systems
Faculty of Informatics Technical University of Munich

with Material from Alexis Engelke, David Hildenbrand,
Michael Petter, and Josef Weidendorfer

18.04.2023



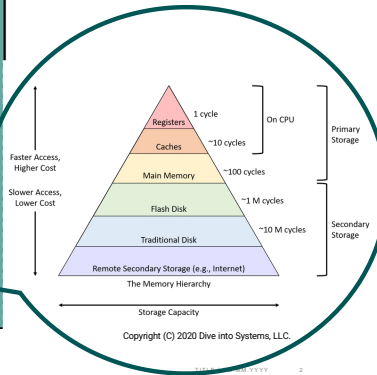
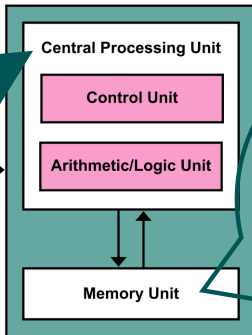
- ▶ Kick-off meeting (today)
- ▶ Literature research + derive article structure
- ▶ Discuss structure with advisor
- ▶ Write draft paper – deadline 2023-06-01

- ▶ Kick-off meeting (today)
- ▶ Literature research + derive article structure
- ▶ Discuss structure with advisor
- ▶ Write draft paper – deadline 2023-06-01
- ▶ Peer-review two other papers – deadline 2023-06-16

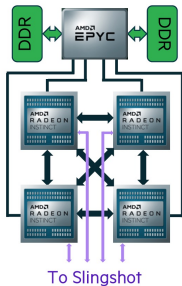
- ▶ Kick-off meeting (today)
- ▶ Literature research + derive article structure
- ▶ Discuss structure with advisor
- ▶ Write draft paper – deadline 2023-06-01
- ▶ Peer-review two other papers – deadline 2023-06-16
- ▶ Incorporate feedback from peers and advisor
- ▶ Final submission + presentation – July 5, 2023

FROM SIMPLE VON NEUMANN ARCHITECTURES TO MODERN HPC SYSTEMS

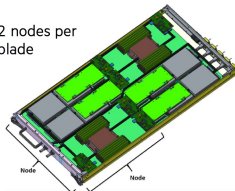
- **Multi-Core**
 - E.g. 36-core Intel IceLake
- **Lots of Optimizations**
 - Pre-fetch
 - Branch prediction
 - FMA
 - Vector
 - Etc.
- **Other features**
 - Encryption
 - Viz
 - Etc.



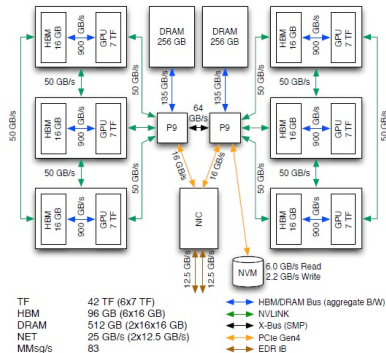
AND THEN WE ALSO ADD ACCELERATORS (GPUS)



2 nodes per blade



COPYRIGHT 2020 HPE

AMD GPU
(ORNL)

HBM & DRAM speeds are aggregate (Read+Write).
All other speeds (X-Bus, NVLink, PCIe, IB) are bi-directional.

AND USE MANY, REALLY MANY OF THESE NODES

- **Frontier Supercomputer @ ORNL:**
 - 9.472 nodes
 - 1,1 EF performance
 - 21 MW power consumption
 - in total over 9 M cores (mostly GPU)



(SOME) CHALLENGES IN PROGRAMMING THESE SYSTEMS

- **Level of parallelism**
 - $O(10^9)$ FPU's
- **Hardware heterogeneity**
 - CPUs, GPUs, other
 - HBM, SSD, object store
- **Programming/Performance Portability**
- **Novel numerical/methodological approaches**

THE GOOD OLD TIMES

- **Programms written in Fortran (or C/C++)**
- **MPI (Message Passing Interface) for moving data across distributed memory**
- **OpenMP for expressing parallelism on shared memory**

Programming Landscape Today

PyTorch

And AI
Frameworks

tensorflow

CUDA OpenMP ROCm/HIP
HSA Programmer APIs
MPI SYCL/One API

memkind OpenACC

Programming Frameworks

Kokkos Alpaka
PETSc

PCI NVMe
Memory APIs

tmpfs

Low-level
interfaces

OS

persistence

Accelerators

Bus/Interconnect

POSIX shared memory

XPMEM

NUMA

- ▶ **Investigate techniques, frameworks, algorithms to efficiently program such systems**
 - ▶ Focus on heterogeneous architectures (GPUs, shared/distributed memory)
- ▶ **Topics**
 - ▶ High-level frameworks (Kokkos, Alpaka, Cabana, PETSc, etc.)
 - ▶ Numerical libraries (SLATE, Ginkgo, heFFTe, etc.)
 - ▶ Data Formats (Mixed-precision, non-IEEE data formats, data compression)
 - ▶ Data Structures and Layouts (AoS-SoA-AoSoA)
 - ▶ Adaptive Mesh Refinement (AMReX, p4est, etc.)
 - ▶ Adaptive (task) Parallelism (HPX, StarPU, Charm++, OpenMP, etc.)
 - ▶ In-Situ Approaches (ADIOS, etc.)
 - ▶ Frameworks for AI (pytorch, tensorflow, etc.)

- ▶ **Investigate techniques, frameworks, algorithms to efficiently program such systems**
 - ▶ Focus on heterogeneous architectures (GPUs, shared/distributed memory)
- ▶ **Topics**
 - ▶ High-level frameworks (Kokkos, Alpaka, Cabana, PETSc, etc.)
 - ▶ Numerical libraries (SLATE, Ginkgo, heFFTe, etc.)
 - ▶ Data Formats (Mixed-precision, non-IEEE data formats, data compression)
 - ▶ Data Structures and Layouts (AoS-SoA-AoSoA)
 - ▶ Adaptive Mesh Refinement (AMReX, p4est, etc.)
 - ▶ Adaptive (task) Parallelism (HPX, StarPU, Charm++, OpenMP, etc.)
 - ▶ In-Situ Approaches (ADIOS, etc.)
 - ▶ Frameworks for AI (pytorch, tensorflow, etc.)
 - ▶ Bring your own topic

- ▶ Find 3-4 suitable publications & suitable websites
- ▶ Describe briefly how the mentioned approach works and helps with achieving efficiency
- ▶ Does the mentioned approach enable a code to be portable across architectures?
- ▶ Discuss advantages and drawbacks

- ▶ Literature and sources
 - ▶ Finding literature and citable sources/references
- ▶ Writing a seminar paper
 - ▶ Structure, style, citing
- ▶ Presentation techniques
 - ▶ Structure, slide design, presentation style

Good to use

- ▶ Books, book chapters
- ▶ Papers (conf./journal)
- ▶ Published articles
- ▶ Manuals

Good to use

- ▶ Books, book chapters
- ▶ Papers (conf./journal)
- ▶ Published articles
- ▶ Manuals

- ▶ Websites with identifiable author
(cite with URL+access date)

Good to use

- ▶ Books, book chapters
- ▶ Papers (conf./journal)
- ▶ Published articles
- ▶ Manuals
- ▶ Websites with identifiable author
(cite with URL+access date)

Try to avoid

- ▶ Wikipedia
- ▶ Facebook, etc.
- ▶ Advertisements
- ▶ Lecture slides
- ▶ Source code
- ▶ **ChatGPT**



Finding literature



- ▶ Starting points: IEEExplore, ACM DL, Google Scholar, arxiv.org, ChatGPT, ...
 - ▶ Select appropriate keywords
 - ▶ Many papers/books accessible freely via the library

- ▶ Starting points: IEEExplore, ACM DL, Google Scholar, arxiv.org, ChatGPT, ...
 - ▶ Select appropriate keywords
 - ▶ Many papers/books accessible freely via the library
- ▶ Other starting point: your advisor

- ▶ Starting points: IEEExplore, ACM DL, Google Scholar, arxiv.org, ChatGPT, ...
 - ▶ Select appropriate keywords
 - ▶ Many papers/books accessible freely via the library
- ▶ Other starting point: your advisor
- ▶ Graph algorithms
 - ▶ Publications of the same author(s)
 - ▶ Publications at the same venue
 - ▶ Cites ... (listed references)
 - ▶ Cited by ...

- ▶ Abstract: Brief summary of area, problem, approach, result
- ▶ Introduction: introduce area, problem, key results, contributions, outline
- ▶ Background: if needed, describe prerequisites

- ▶ Abstract: Brief summary of area, problem, approach, result
- ▶ Introduction: introduce area, problem, key results, contributions, outline
- ▶ Background: if needed, describe prerequisites
- ▶ Main part (approach, evaluation, discussion, etc.)

- ▶ Abstract: Brief summary of area, problem, approach, result
- ▶ Introduction: introduce area, problem, key results, contributions, outline
- ▶ Background: if needed, describe prerequisites
- ▶ Main part (approach, evaluation, discussion, etc.)
- ▶ (*In a paper*: Related Work)
- ▶ Summary & outlook

- ▶ Factual, precise, focused
 - ▶ Stay on topic, no story telling, ...
 - ▶ Limit to important and necessary topics
 - ▶ Don't omit necessary prerequisites

- ▶ Factual, precise, focused
 - ▶ Stay on topic, no story telling, ...
 - ▶ Limit to important and necessary topics
 - ▶ Don't omit necessary prerequisites

- ▶ Avoid forward references
- ▶ Avoid *I*, prefer *we* (or passive voice)
- ▶ *We* only describes the authors, not the reader

- ▶ All work that is not yours **must** be cited
 - ▶ Clearly describe source
 - ▶ But: no wrong/inaccurate attributions
- ▶ Citing styles:
 - ▶ Literal (direct) quote
 - ▶ indirect quote (rephrase) ←strongly preferred
- ▶ Exception: foundations can be assumed (generally first few Bachelor semesters)

The x86 architecture defines
the register CR2 [1].

The x86 architecture defines the
register CR2~\cite{intel2019man}.

The x86 architecture defines the register CR2 [1].

The x86 architecture defines the register CR2~\cite{intel2019man}.

The x86 architecture defines the register CR2. It can be used with the instruction MOV. [1]

The x86 architecture defines the register CR2. It can be used with the instruction MOV.~\cite{intel2019man} (Absatz)

The x86 architecture defines the register CR2 [1].

The x86 architecture defines the register CR2~\cite{intel2019man}.

The x86 architecture defines the register CR2. It can be used with the instruction MOV. [1]

The x86 architecture defines the register CR2. It can be used with the instruction MOV.~\cite{intel2019man} (Absatz)

Valgrind [1] is a tool for run-time instrumentation.

Valgrind~\cite{nethercote2007} is a tool for run-time instrumentation.

The x86 architecture defines the register CR2 [1].

The x86 architecture defines the register CR2~\cite{intel2019man}.

The x86 architecture defines the register CR2. It can be used with the instruction MOV. [1]

The x86 architecture defines the register CR2. It can be used with the instruction MOV.~\cite{intel2019man}
(Absatz)

Valgrind [1] is a tool for run-time instrumentation.

Valgrind~\cite{nethercote2007} is a tool for run-time instrumentation.

Other approaches [1,2,3] ...

Other approaches~\cite{foo,bar,baz}
\dots



Presentation: Content Selection



Presentation for the **audience!**

Presentation for the **audience!**

- ▶ What do you want the audience to take away?
(Not: what can I talk about!)
- ▶ What are the key points?
- ▶ How much content fits into the time slot?

- ▶ Motivation
 - ▶ Why is the topic relevant?
- ▶ Background
 - ▶ Consider referencing information from previous talks
- ▶ Concept
- ▶ Evaluation
 - ▶ How good is the described concept?
- ▶ Conclusions and outlook

- ▶ Important: topics build upon each other!
(avoid forward references)
- ▶ Only give important details
- ▶ Use good/helpful examples
- ▶ Critical discussion of the topic

- ▶ Slides (Beamer)
 - ▶ For use during the talk
 - ▶ Good to prepare
 - ▶ *Backup-Folien* as preparation for questions
- ▶ Whiteboard, blackboard
 - ▶ Permanently needed information
 - ▶ Answering questions
- ▶ Hardware, demonstrators, etc.

- ▶ Check possibilities in advance

- ▶ Prepare slides, etc.
- ▶ Do a dry-run
 - ▶ Always recommended
 - ▶ Helps with uncertainty and time estimation
- ▶ Prepare on-site
 - ▶ Laptop, Beamer, laser pointer, clock, etc.

- ▶ Speak freely
- ▶ Don't go too fast/slow
- ▶ Stay in contact with the audience
 - ▶ Eye contact, position, etc.
- ▶ Usually at least 1 minute per slide
- ▶ Stay in time limit
 - ▶ Optional slides can fill time
 - ▶ Regularly consult a watch

- ▶ **Stay calm**

- ▶ One topic per slide
- ▶ Avoid text
 - ▶ ≤ 8 lines
- ▶ Prefer graphics/illustrations

- ▶ One topic per slide
- ▶ Avoid text
 - ▶ ≤ 8 lines
- ▶ Prefer graphics/illustrations
- ▶ No unused points
 - ▶ Cover everything on the slides in your talk

- ▶ Title page
 - ▶ Title, name, institution, date, location

- ▶ Title page
 - ▶ Title, name, institution, date, location

- ▶ On every other slide: number and title

- ▶ Title page
 - ▶ Title, name, institution, date, location

- ▶ On every other slide: number and title

- ▶ Conclusion
 - ▶ Alle important points on one slide

► Black on white

- ▶ **Black on white**
- ▶ **Black on white**

- ▶ **Black on white**
- ▶ **Black on white**
- ▶ Sufficient contrast
- ▶ Use colors sparingly, but systematically
- ▶ Be careful with gradients
- ▶ No annoying backgrounds (wave textures, etc.)
- ▶ **Animations only with sufficiently added value**

- ▶ Double-check text for typos, etc.
- ▶ Use a readable, sans-serif font
- ▶ Prefer vector graphics (or images with a high resolution)
- ▶ Avoid screenshots/scans
- ▶ Important: citations
- ▶ Listings only with a sufficiently large value

Alignments in brief

>1bl8_A mol:protein length:97 Potassium Channel Protein
 ALIHWRAAGAATVLLAVLLAGSYLAERGAQGLFT
 YPRALWWSVETATTVGYGDLVPTLWGRCAVVMVA
 GITSFGLVTAALATWFWVGREQ

>1orq_C mol:protein length:223 Potassium Channel
 IGDVMEHPLAVELGVSYAALLSVVVVVECTMQLSGEYLV
 RLVLVLDLVLVHWDVAYRAYKSGDPAGYVKITLYEL
 PALVPAGLLALIEGHLAGLGLFRLVRLRLRLILLISRG
 SKFLSALADAADKIRFYHLFGAVMLTVLYGAFAIYIVEY
 PDPNSSIKSVFDALWWAVVTATTVGYGDVVPATPIGKV
 IGLAVMLTGISALTLLIGTVSNMFQKILV

Query= 1bl8_A mol:protein length:97 Potassium Channel Protein
 (97 letters)

>1orq_C mol:protein length:223 Potassium Channel
 Length = 223

Score = 58.5 bits (140), Expect = 4e-14
 Identities = 26/72 (36%), Positives = 43/72 (59%)

**Scoring matrix
 Algorithm to optimize score**

```

Query: 21  GSYIAVLAERGAQGLITYPRALWWSVETATTVGYGDLVPTLWGRCAVVMVAGITS 80
          G++  + E  P + + + ALWW+V TATTVGYGD+ P T G+ + + VM+ GI++
Sbjct: 147 GAFALIVLYPDPNSSIKSVFDALWWAVVTATTVGYGDVVPATPIGKVIGIAVMLTGISA 206

Query: 81  FGLVTAALATWF 92
          L+  ++ F
Sbjct: 207 LTLIGTVSNMF 218
  
```

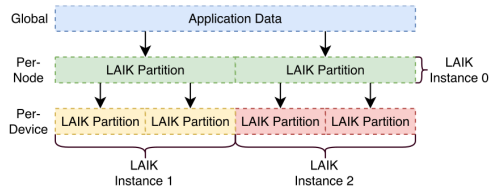


```
\begin{frame}
\frametitle{Die Anti-Folie}
\begin{figure} [ht]
  \centering
  \includegraphics[width=0.95\textwidth]{pictures/antifolie.jpg}
  \caption{Werbe-Folie. Foto von Flickr-Benutzer niallkennedy
    (https://www.flickr.com/photos/niallkennedy/58697220/sizes/l/)}
  \label{fig:gliederung}
\end{figure}
\end{frame}
```

Abbildung: Screenshot of code with insufficient resolution

LAIK (5) – Hierarchische Partitionierung

- multiple Partitionierung auf verschiedenen Ebenen
- Beispiel: inter/intra-node
- sinnvoll für Exascale, heterogene Systeme
- Veränderung des Indexraums muss möglich sein!



- ▶ Summary slide with main take-away points

- ▶ Summary slide with main take-away points
- ▶ NO *Questions* slide!

- ▶ Bring your point to the audience – written or spoken
- ▶ Good literature as starting point
- ▶ Logical structure for paper and presentation
- ▶ Presentation: good preparation is important

- ▶ Bring your point to the audience – written or spoken
- ▶ Good literature as starting point
- ▶ Logical structure for paper and presentation
- ▶ Presentation: good preparation is important

- ▶ Chance to learn 😊

- ▶ Paper 6-8 pages
- ▶ Presentation about 15 Minutes plus questions (5 minutes)

- ▶ Paper 6-8 pages
- ▶ Presentation about 15 Minutes plus questions (5 minutes)

- ▶ Grading
 - ▶ 40% paper
 - ▶ 40% presentation
 - ▶ 20% review
 - ▶ all needs to be positive
- ▶ start literature search now and get in contact with your tutor

ANY
QUESTIONS
?



- ▶ Paper 6-8 pages
- ▶ Presentation about 15 Minutes plus questions (5 minutes)

- ▶ Grading
 - ▶ 40% paper
 - ▶ 40% presentation
 - ▶ 20% review
 - ▶ all needs to be positive

- ▶ communicate **three (3)** topics until April 23
- ▶ final topics and tutor will be assigned next week
- ▶ start literature search now and get in contact with your tutor