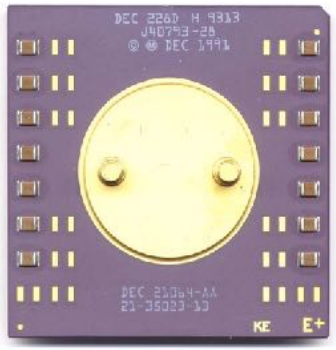
Challenges of the Future Computing Technology Landscape

Salman Habib
Argonne National Laboratory
habib@anl.gov

Living in the past — 1980/2000

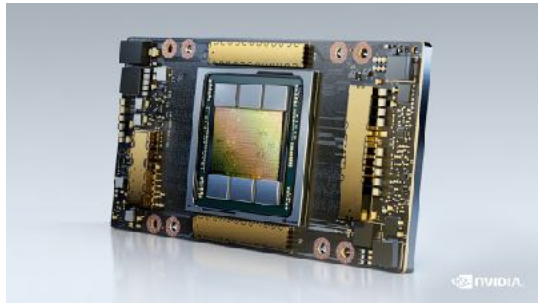
- Moore's law reigns
- Life is good
- 2x rule of thumb for effort on performance
- Cost/performance ratios very favorable
- Scientific computing rides on general computing advances
- Global parallelism not difficult to address



DEC Alpha CPU

Facing the present — 2000/2025

- (Conventional) Moore's law ends
- Life starts becoming nontrivial
- Cost/performance stalls
- Local concurrency must be faced (GPUs)
- Computing advances fragment
- Scientific computing roadmap begins to look unclear



NVIDIA A100 GPU

Anton 3 ASIC
DE Shaw Research

Confronting the future — 2025+

- Multiple tech roadmaps, but no major changes (too much inertia)
- Life is tough
- Significant specialization
- Scientific computing roadmap may require actual planning

Argonne NATIONAL LABORATORY

Experiments as computing technology drivers

- **Requirements set primarily by throughput**
  - Data ‘velocity’ in burst and quasi-continuous modes controls computational requirements
  - Needs vary across experiments (size, one-shot vs. multi-pass, technology history, community preferences, workflow complexity, other technical requirements)
  - Related computing (e.g., detector simulation) tasks may be run on the same type of compute platform
  - Experiments stress the total computing environment — computing as well as IO and data management, thus SmartNICs and processing-in-memory can play useful roles (essentially distributed computing tasks for data in flight and in repose)
  - Most experimental workflows have limited arithmetic intensity (low flops/byte ratio) and limited data reuse
  - HEP is not alone — similar issues are faced in other fields (e.g., light sources)

General observations

- **Advantages of general purpose computing (e.g., CPUs)**
  - Advances in hardware directly translate to improvements in application performance
  - Higher-level software stack largely independent of (local) hardware implementation
  - Responsibility for performance optimization lies largely outside the realm of (high-level) applications (reliance on compilers)
  - Relatively fixed set of algorithms — relatively narrow focus on improved implementations; algorithmic development not directly connected to underlying hardware
  - Although technology advances may be rapid, the effect on software development cycles (traditionally long) is relatively small

- **Disadvantages**
  - Stagnation in application performance as the underlying technology stalls
  - Performance engineering over a finite set of algorithms can only produce limited gains
  - Different approaches to computing naturally arise to fight performance stalls
  - Possible danger of being left in the “slow lane” as hardware/software technology evolve in different directions; software development cycles need to be sped up!

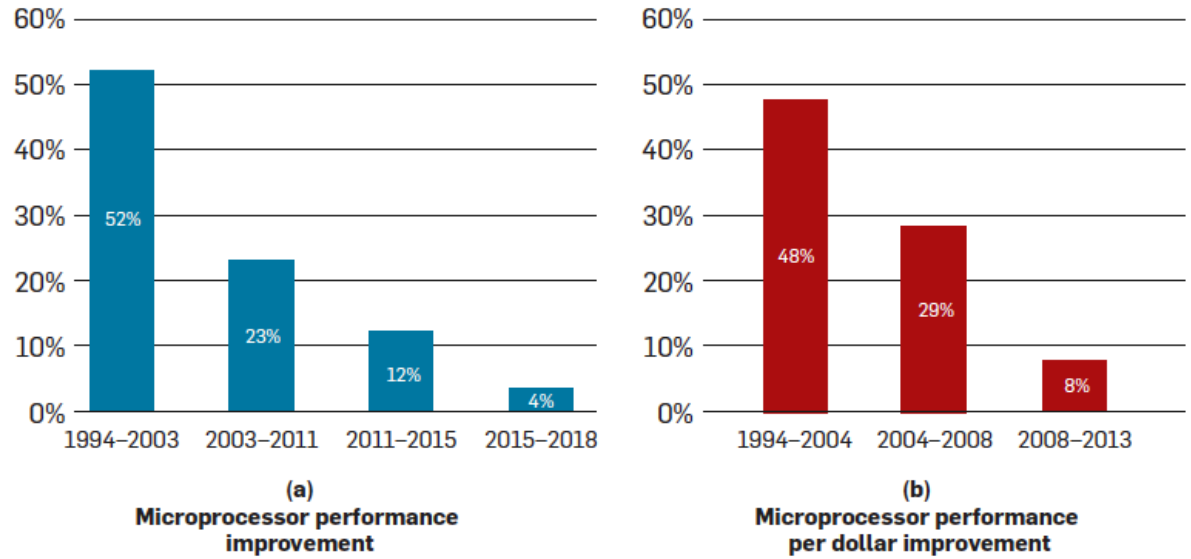
Trend towards heterogeneity

- **Transistor limitation issues**
  - Dennard scaling — reduce transistor sizes but keep electric fields constant (increased transistor count, increased performance, reduced supply voltage keeps power use constant)
  - Transistor density increase led to complex architectures capable of multiple optimizations (out-of order execution, speculation, pipelining, cache hierarchy —), adding more general capability
  - Power consumption limits (leak currents, frequency and supply-voltage limits) + finite power budget drives trend to multiple (simpler) cores and customization (algorithmic or restricted parallelism)

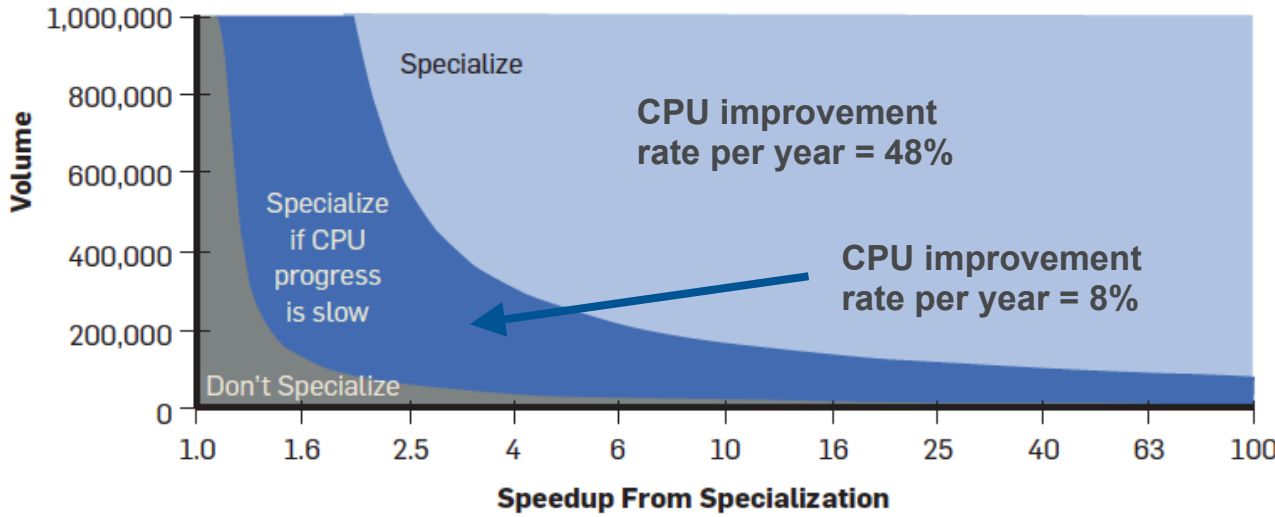
- **Software/application ramifications**
  - Higher-level software stack can no longer ignore lower-level hardware realities, no more “riding the wave” of Moore’s law
  - Algorithm choices controlled/restricted by low-level architecture — worst-case scenarios can involve poor trade-offs for many scientific applications (small winner pool, large loser pool — less diversity in scientific applications)
  - Management of hardware specialization is the major challenge going forward — involves all aspects of problem specification, solution strategies, algorithmic implementations, overall software environment, includes management of heterogeneity at local and system-scale levels

Trend towards specialization

- **Winners**
  - Applications that can exploit significant parallelism
  - Computational tasks can be arranged in stable configurations with a regular cadence
  - Limited memory accesses for a fixed computational effort
  - Allow use of fewer degrees of precision (e.g., AI/ML applications)
  - Low-power applications
- **Specialization trends**
  - Slowdown in CPU performance gain makes specialization more attractive
  - As the threshold for specialization is lowered, more applications can benefit from it (good)
  - This can be a driver for fragmentation (bad)



Credit: Thompson & Spanuth CACM 2021

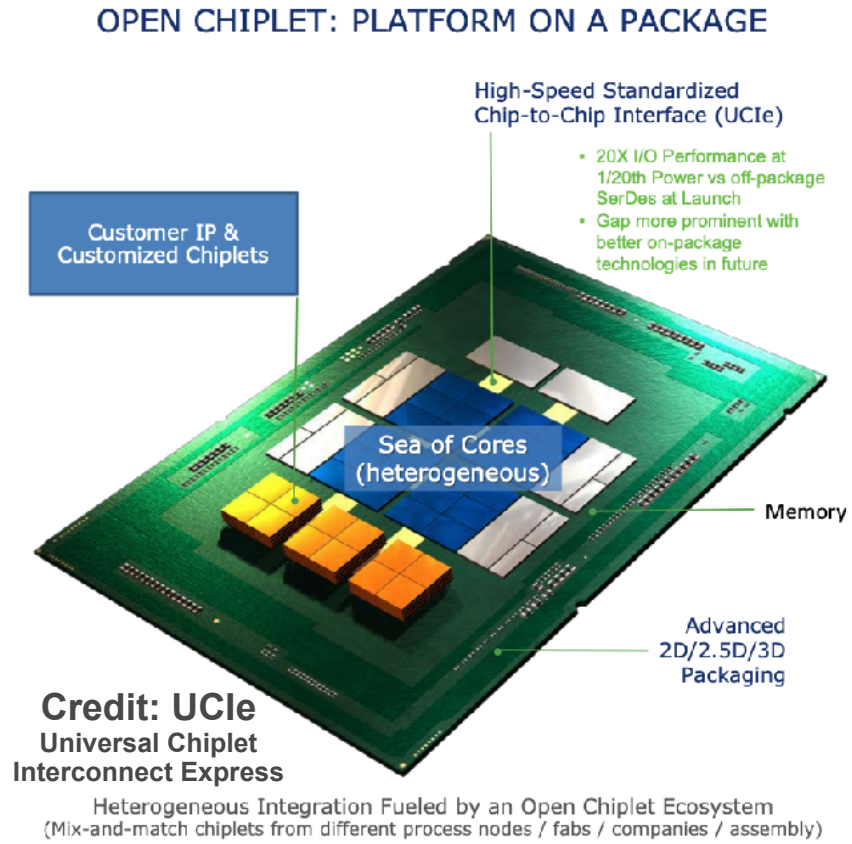


Economic model for the rationale behind specialization

Argonne NATIONAL LABORATORY

Ramification for experiments: Co-design vs. specialization

- **Future technologies**
  - Diversity of approaches appears to characterize the 2025+ technology roadmap
  - Low-power CPU, CPU/GPU, FPGA, AI/ML architectures (e.g., TPUs), all compete in this space
  - Specialized architectures (ASICs) can be part of the solution, even for off-line computing — negatives are cost and potential obsolescence as hardware technology evolves
  - Co-designed approaches that focus on algorithmic flexibility and ability to leverage vendor and open source methodologies for portability are likely preferred
  - Hardware co-design possibilities may be better than in the past via chiplet integration (supported by major players — AMD, Google, Intel, Meta, TSMC, —)
  - Software stack will require more low-level expertise than in the past, but physicists can still be targeted as the primary code writers via use of performant higher-level frameworks (e.g., as in present-day AI/ML)



A Speculative Summary

- The trend towards heterogeneity and specialization is irreversible, as CPU performance gains will remain modest
- Over the next decade, no radically new technology is likely to get us back to the status quo (photonics, quantum, neuromorphic)
- There will be winners and losers, the losers are those who:
  - Cannot get a worthwhile performance boost from specialization/co-design
  - Do not have large enough requirements (or enough funding) to pay for specialization
- For experiments to avoid being losers, they should:
  - Actively consider new algorithmic approaches to solving their problems, if possible (e.g., AI/ML approaches, data restructuring)
  - Actively consider coordinating mechanisms to form a big enough market (this will require a combination of co-design and perhaps a more limited version of specialization) — commercial or public clouds could also aid in providing this function
  - Avoid the trap of just asking for more CPU funding ;-)

Acknowledgments: Argonne HACC team, HEP-CCE project;
Ray Bair, Andrew Chien, Rob Ross, Neil Thompson