UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Bringing FPGAs to HPC Production Systems and Codes

Christian Plessl

Paderborn Center for Parallel Computing &
Department of Computer Science

Paderborn University, Germany

Paderborn Center for Parallel Computing

R2HC Workshop – Nov 11, 2018 – Supercomputing 2018

State of the FPGA Union

Highly Capable FPGA Devices

Example: Intel Stratix 10 GX2800 FPGA

- > 900,000 configurable logic blocks
  - up to 4 Boolean functions of 8 inputs
- 5760 hardened arithmetic units (DSP)
  - fixed point and IEEE 754 SP floating-point
- > 11,000 independent SRAM blocks
  - width/depth/ports highly configurable
- integrated DDR4-2666 memory controllers
- 96 serial transceivers, up to 28.3 Gbps
- typically about 300-600MHz
- power consumption 50-225W

100 TERRA-OPS

10 single-precision TFLOPS

20 TB/s internal SRAM bandwidth (full duplex)

300 TB/s communication bandwidth (full duplex)

up to 80 GFLOPS/W

Increasingly Productive FPGA Tools

- Traditional EDA Software
  - hardware synthesis from VHDL, Verilog
  - simulators, place and route

mature but much too cumbersome for general HPC

- General high-level synthesis tools
  - generation of complete accelerators or components from OpenCL or C/C++ with annotations (Intel OpenCL SDK for FPGAs, Xilinx SDAccel)

higher productivity and increasingly good results

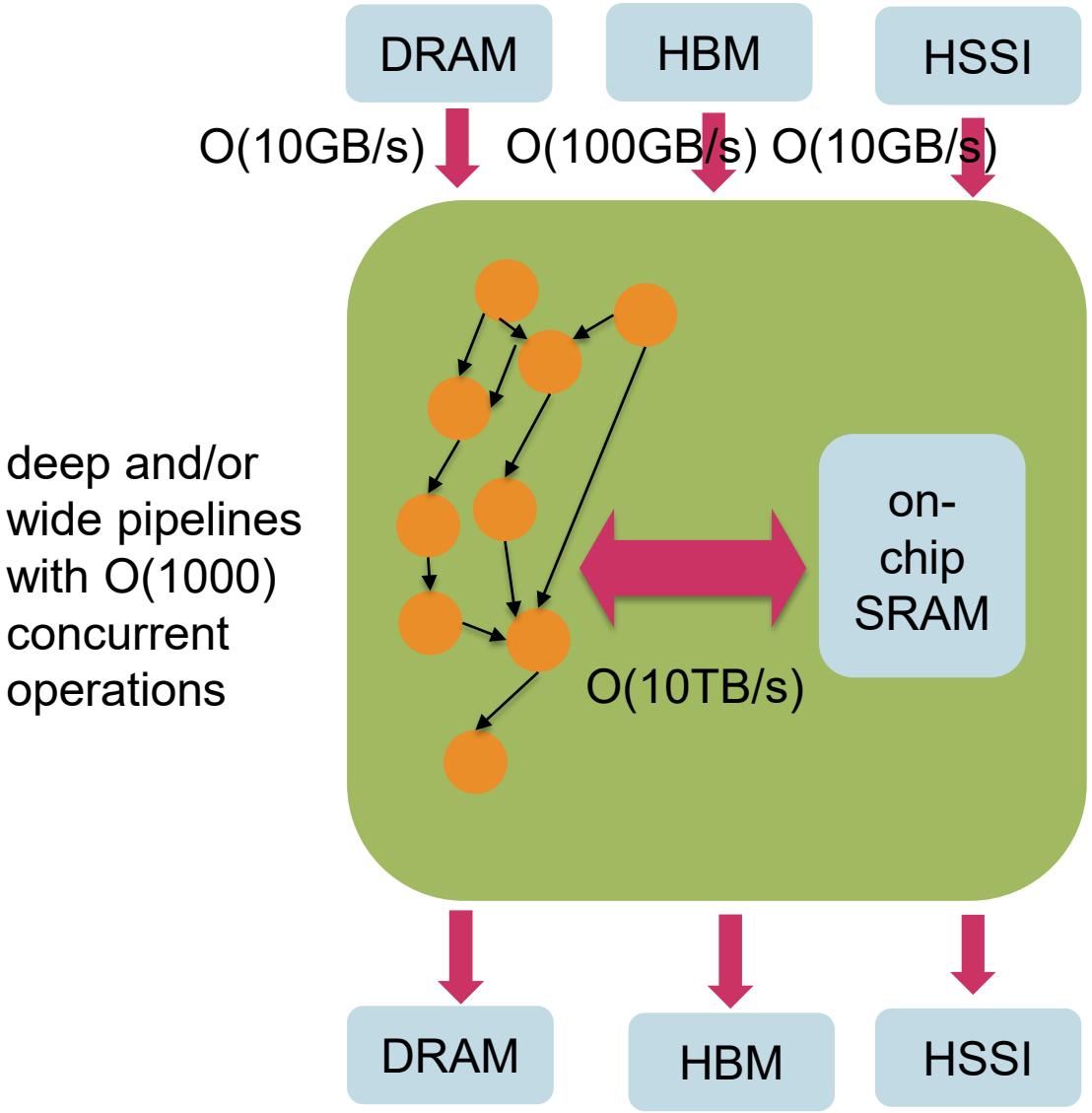
- Domain-specific tools for important niches
  - networking (P4)
  - digital signal processing (MATLAB/Simulink toolboxes)
  - deep learning inference (Xilinx xDNN, Intel DLA)

even higher productivity but narrow scope

- Libraries and ready-to-use applications

almost inexistent for HPC

Opportunities for FPGAs



- Compute-bound applications
  - customization of operations and data formats
  - new methods considering FPGA architecture
- Memory-bound applications
  - unrolling and data flow computing with very deep pipelines
  - application-specific, distributed memory architectures
- Latency-bound applications
  - speculative or redundant execution
- I/O-bound applications
  - on-board network interfaces
  - direct FPGA-to-FPGA communication

Demonstrated Benefits for Proof-of-Concept Codes

- Examples from important HPC domains
  - Linear algebra: CG solver for sparse linear equation systems [1]
    - 20-40x faster than CPU
  - Geophysics: 3D convolution [1]
    - 70x faster than CPU, 14x faster than GPU
  - Molecular dynamics [2]
    - 80x faster than NAMD (single core) CPU
  - Bioinformatics (BLAST) [3]
    - 5x faster than optimized, parallel CPU implementation
  - Climate modeling [4]
    - 4 FPGAs 19x faster than two socket CPU, 7x faster than GPU

[1] O. Lindtjorn, R. G. Clapp, O. Pell, O. Mencer, M. J. Flynn, and H. Fu. Beyond traditional microprocessors for geoscience high-performance computing applications. IEEE Micro, Mar.–Apr. 2011.

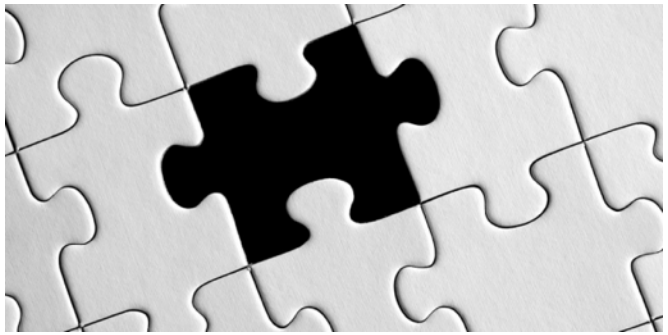
[2] M. Chiu and M. C. Herbordt. Molecular dynamics simulations on high-performance reconfigurable computing systems. ACM TRETS Nov. 2010.

[3] A. Mahram, and M. C. Herbordt. NCBI BLASTP on High-Performance Reconfigurable Computing System. ACM TRETS Jan 2015.

[4] L. Gan, H. Fu, W. Luk et. al. Solving the Global Atmospheric Equations through Heterogeneous Reconfigurable Platforms. ACM TRETS Mar. 2015

What's Missing to Establish FPGAs in HPC?

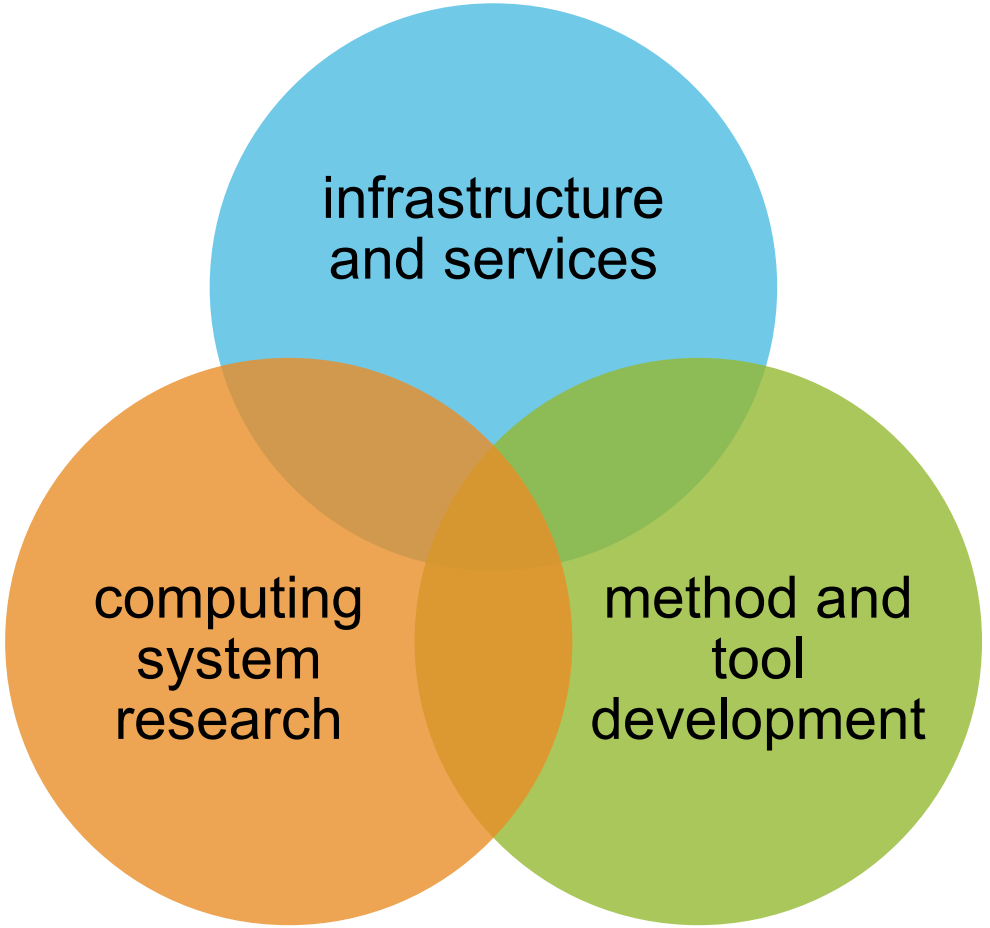
- Hardware
  - move FPGAs from lab to HPC production systems
- Software
  - (open source) HPC applications and libraries using FPGAs
  - HPC-specific development tools & runtime environment
- Community
  - currently: FPGA ∩ HPC ≈ ∅
  - workshops, conferences, journals, user meetings
- Developer training
  - materials addressing HPC developers and CSE
  - best practices and design patterns



FPGAs at the Paderborn Center for Parallel Computing (PC²)

PC² – Competence Center for Innovative HPC

- Scientific institute of Paderborn University
- Service provider and research institution
  - provision HPC infrastructure and services for computational sciences
  - develop methods and tools for simulation and modeling in cooperation with computational scientists
  - perform computing systems research for energy-efficient HPC, emphasis on heterogeneous and accelerated computing with FPGAs
- Long track record in exploring emerging and off the beaten path technologies



Long Experience in FPGA Research and Systems

- Several generations of research groups working with FPGAs in CS and EE
- Multi-year preparation for deployment of FPGAs in production HPC systems

System	Inst	CPU	FPGA	Toolflow	Properties
Maxeler MPC-C	2012	Xeon X5660	4x Xilinx Virtex-6 SX475T	MaxCompiler	MAX3 FPGA card, MaxRing interconnect
Nallatech 385A	2016	Xeon E5-1260v2	Intel Arria 10 GX1150	Intel OpenCL	Nallatech 385A FPGA card
IBM S812L	2016	POWER8	Xilinx Virtex-7 VX690T	Xilinx OpenCL	AlphaData 7V3 FPGA board
Micron Workstation	2016	Intel i7-5930K	Xilinx Kintex-7 UltrascaleKU115	Xilinx OpenCL	Pico AC-510 FPGA card with Hybrid-memory cube
XCL cluster	2017	Xeon E5-1630v4	Xilinx Virtex-7 VX690T + Xilinx Kintex Ultrascale KU115	Xilinx OpenCL	8-nodes, 1 AlphaData 7V3 and 1 8K5 FPGA cards each
HARP cluster	2017	Xeon E5-v4	Intel BDW+FPGA hybrid CPU/FPGA	Intel OpenCL, HDL	10-node cluster with 1 BDW+FPGA processor per node
Noctua Cluster	2018	Xeon SKL 6148	Intel Stratix 10 GX2800	Intel OpenCL	16 nodes, 2 Nallatech 520N FPGA cards each

selected FPGA systems at PC²

lab testbed production

Relevance of FPGAs to PC²

- Noctua project 2018-22
  - research grant for next generation HPC system (10M€) and data center building (15M€)

- FPGAs play a strategic role our roadmap
  - exploration of FPGAs in production HPC machines
  - port libraries and real scientific applications to FPGAs
  - work on parallel FPGA implementations (MPI, PGAS)
  - study performance and energy trade-offs

- Investment complemented by research, development and support efforts
  - infrastructure accessible for free for researchers in Germany
  - international collaborations possible and desired, negotiated on case-by-case basis

Noctua HPC System (Phase 1)

- Cray CS500 Cluster System
- 256 CPU nodes
  - 2 x Intel Xeon Skylake Gold 6148, 2 x 20 Cores, 2.4GHz
  - 192 GB RAM
- 16 FPGA nodes
  - 2 x Intel Stratix 10 GX2800 (Nallatech 520N boards)
    PCIe 3.0 x16, 4 x 8GB DDR4 channels
  - per board 4 QSFP28 ports
  - currently worldwide biggest and most modern FPGA installation in academic HPC system
- 100 Gbps Intel Omni-Path network
- 700 TB Cray ClusterStor L300N storage system

Early access since 9/2018, general availability est. 12/2018

Selected Current Work

Our "Bringing FPGA to HPC" Strategy

Leverage

- target applications contributing high load to our HPC clusters
- focus on parts FPGAs are known to have potential
- widely-usable infrastructure improvements

Application Competence

- in-depth application and method knowledge is key (selection of right methods and benchmark applications)
- co-development with work with code owners

Sustainability

- protect results form abandonment and bitrot
- release as open source, build reusable libraries

Development Flows and System Integration

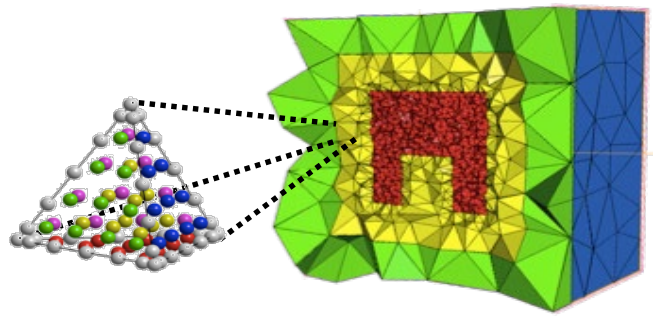
- Focus on Intel/Xilinx OpenCL toolflows for FPGA
  - encapsulates applications in common infrastructure (some forward compatibility, security)
  - familiarity of some HPC developers with OpenCL or CUDA
  - abstraction level allows for code co-development and maintenance by application owners

OpenCL

- Automated provisioning of different SDK and BSP versions through Slurm
  - interactive and batch use of FPGA nodes
  - job submission with required BSP information: scheduling to FPGAs with requested BSP if present, or re-configuration and reboot
  - simple in theory, but fragility of BSP and driver reconfiguration requires careful handling of edge cases and self-tests

Target Application: MIDG2*

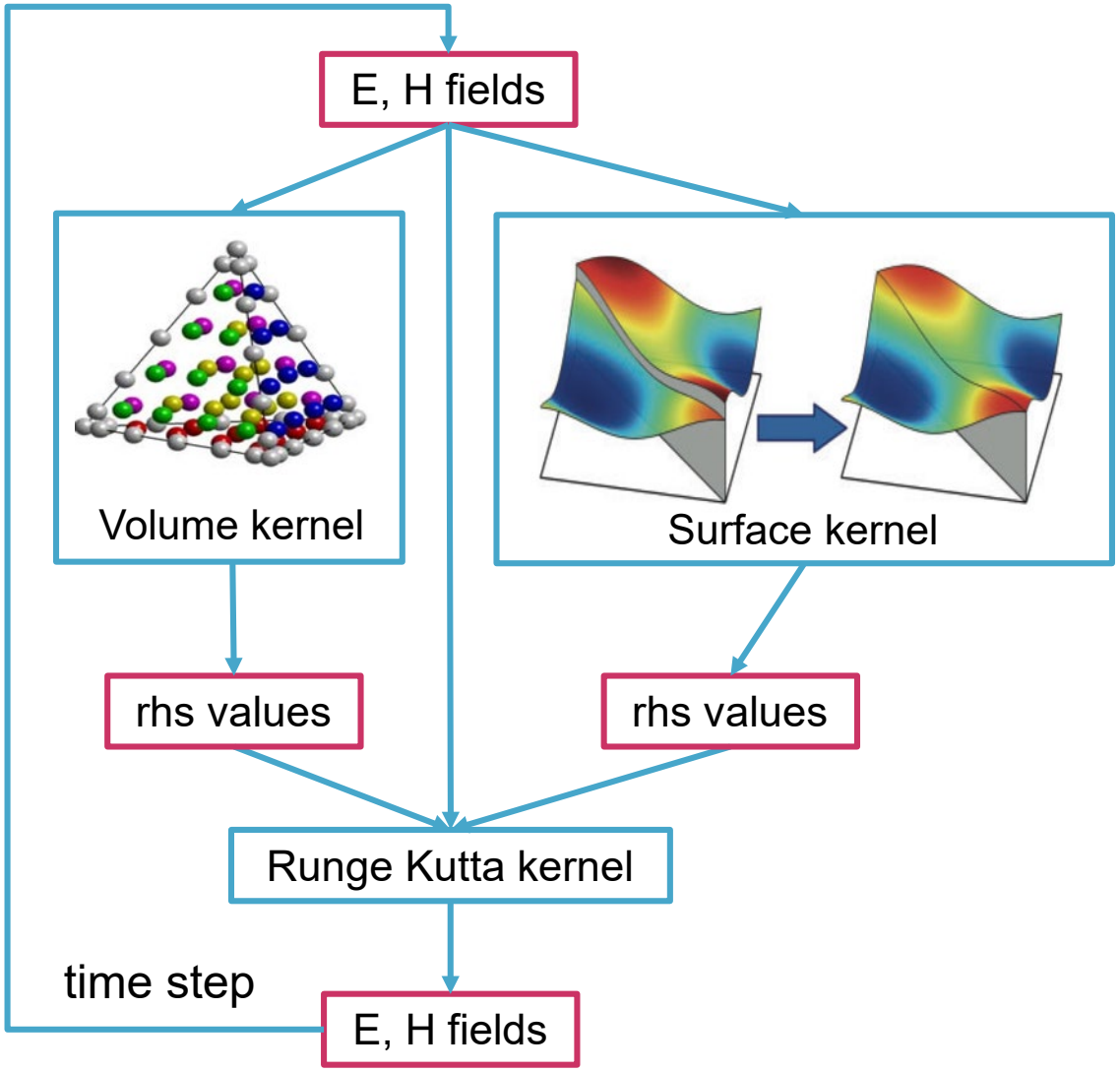
- Time-Domain Nodal DG solver for Maxwell's Equations
  - in-house code by group of Prof. Jens Förstner
  - based on MIDG2 by T. Warburton (https://github.com/tcew/MIDG2)
  - applications: nanophotonics and astrophysics



- State of the art method
  - high numerical quality and stability
  - unstructured 3D mesh
    - adapted to material boundaries and regions of interest
    - non-linear materials and multi-physics
    - no global stiffness matrix required

- Suitability for FPGAs
  - high arithmetic intensity, can be controlled by polynomial order
  - local computations, favorable computation to communication ratio
  - suitable for FP32 computation

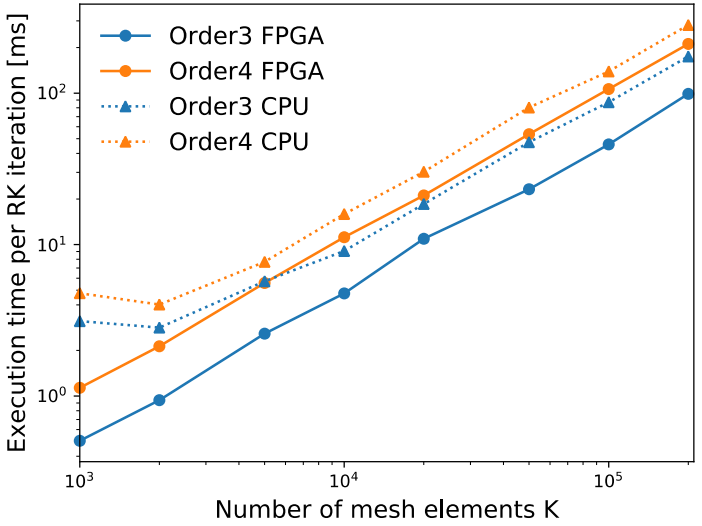
MIDG2* FPGA Implementation



- Method works on tetrahedral meshes
  - E and H field is defined at nodal points in volume and at surface
  - typ. mesh sizes 10^3–10^6 elements
- Algorithm divided in three kernels running on FPGA
  - Volume kernel
  - Surface kernel
  - Runge-Kutta kernel
- Decoupling of memory accesses
  - overlapping indirect memory access for element i+1 with processing element i

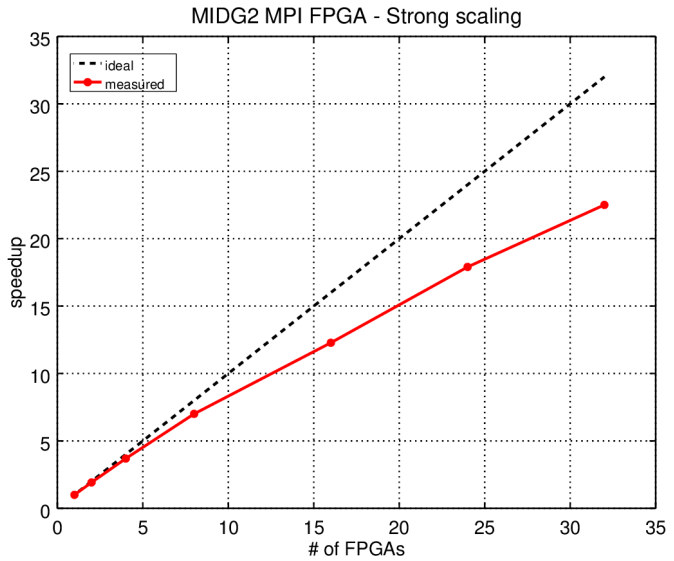
MIDG2* Early Results

- First phase: Arria 10 GX1150 vs. 2x Xeon E5-2670v1
  - two main kernels with >100 GFLOP/s design points
  - using local RAM as buffers and constant memory
  - achieving high off-chip bandwidth through decoupled access
  - FPGA outperforms dual-socket Xeon by ~2x



- Current phase
  - Stratix 10 port functional, with performance headroom
  - scaling to 32 FPGAs with host transfer + MPI
  - working on direct FPGA-to-FPGA communication

[Kenter et al., OpenCL-based FPGA Design to Accelerate the Nodal Discontinuous Galerkin Method for Unstructured Meshes, FCCM'18]



strong scaling multi-FPGA implementation on Noctua

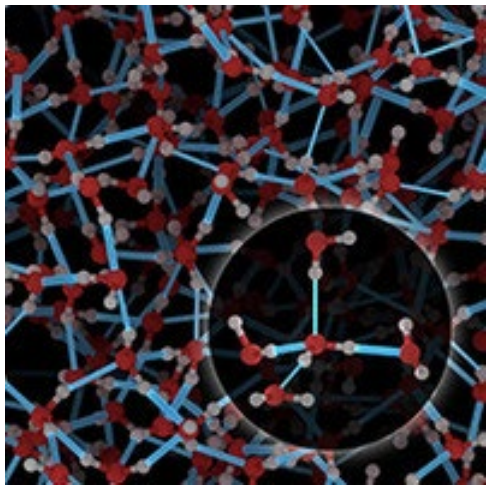
Target Application: CP2K

- CP2K
  - widely used open-source code for molecular dynamics
  - comprises many methods and usage modes
  - cooperation with Prof. Thomas D. Kühne (Theor. Chemistry)

- Promising components for FPGAs

  - approximate linear algebra for linear scaling electronic structure methods (small dense and large sparse matrices)

  - 3D FFT for efficient computation of electrostatic interaction or orbital representations in periodic structures



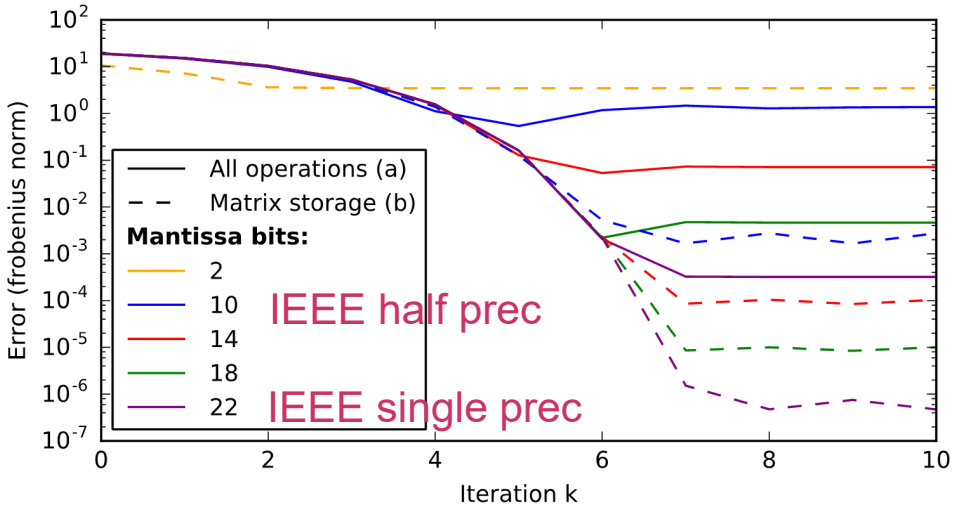
network of hydrogen bonds in water

Approximate Linear Algebra suitable for FPGAs

- Iterative computation of approximate inverse p-th root of symmetric, positive definite matrix A

$$\mathbf{B} = \mathbf{A}^{-1/p} \qquad \boldsymbol{B}_{k+1} = \frac{1}{p}\left((p+1)\boldsymbol{B}_k - \boldsymbol{B}_k^{p+1}\boldsymbol{A}\right)$$

  - well suited for low precision computing
  - initial convergence not not influenced by precision
- FPGA opportunities
  - custom floating-point formats
  - mixed precision implementation
  - reduction of data transfers from/to accelerator
- Status
  - low-precision DGEMM for Xilinx FPGAs (>300 GFLOPS FP16)
  - Stratix 10 still lots >5x headroom (clock speed, resource utilization, ...)



approximation error for custom floating point formats

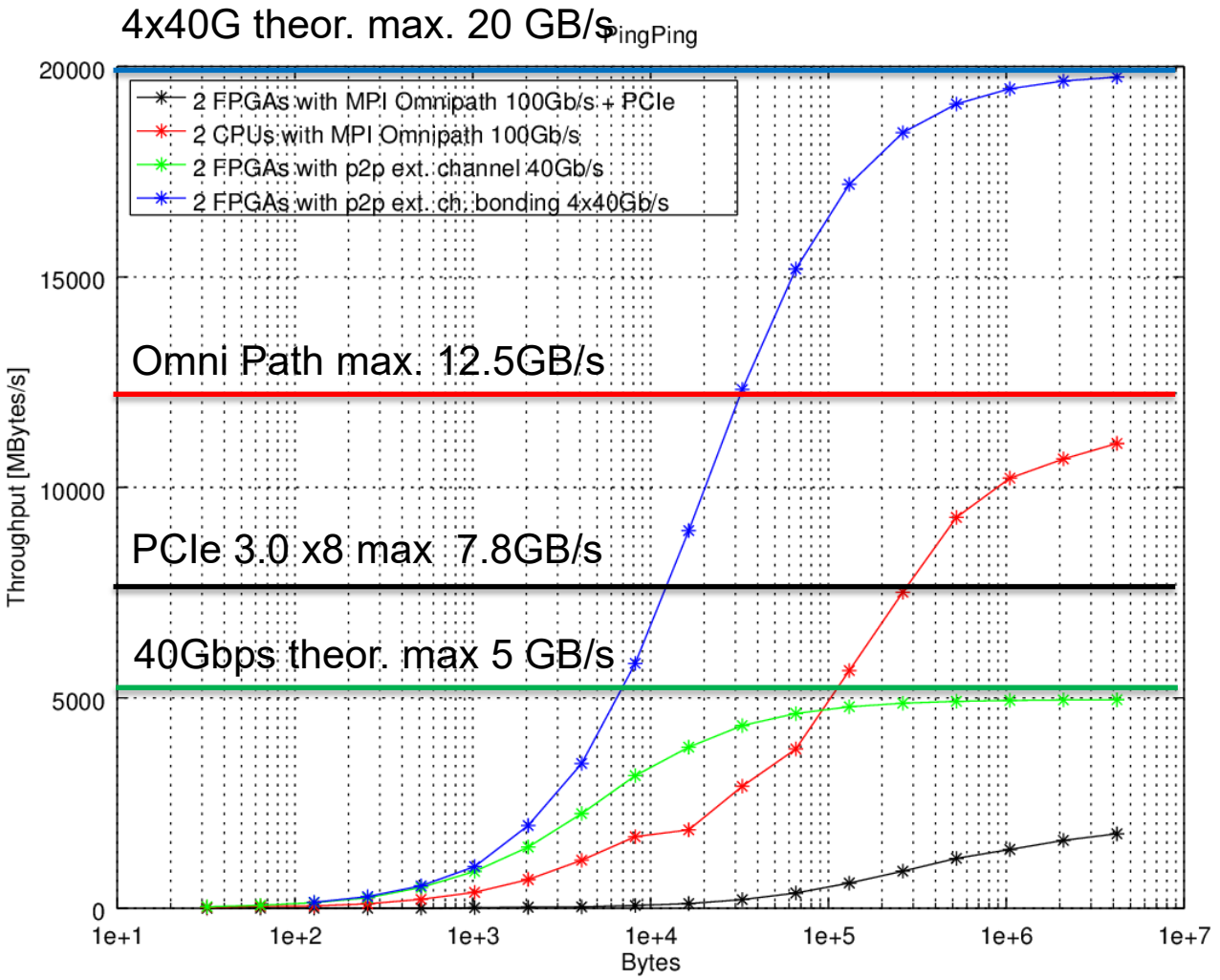
$$\left\| B_k - A^{-1/2} \right\|_F = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} \left| b_{ij} - \alpha_{ij} \right|^2}$$

error metric (Frobenius norm)

- Richters, Lass, Walther, Plessl, Kühne: **A General Algorithm to Calculate the Inverse Principal p-th Root of Symmetric Positive Definite Matrices.** Communications in Computational Physics, 2018.
- Lass, Kühne, Plessl: **Using Approximate Computing for the Calculation of Inverse Matrix p-th Roots**. IEEE Embedded Systems Letters. 2018.

Dedicated Network between FPGAs

- Use cases for dedicated network between FPGAs in HPC
  - OpenCL kernels that communicate via channels between cards
  - faster and lower latency connection than MPI via host (avoid PCIe bottleneck)
  - application-specific communication topologies and patterns
- Nallatech 520N card and OpenCL SDK for 18.0.1
  - 4 QSFP+28 network ports per card with up to 100G (current BSP limitation 40G)
  - PCIe 3.0 x16 (current BSP limitation x8)



Promoting Reuse

- All our developments will be released as open-source

- Dense linear algebra
  - development of library functions for matrix multiply (and later convolutions) in progress
  - contribution to libXMMS planned

- Sparse linear algebra
  - developed new massively parallel algorithm for approximate inverse p-th roots of sparse matrices, can be combined with approximate dense inversions
  - integration in libDBCSR in progress

- Fast Fourier Transforms
  - single-precision 3D FFT (16^3, 32^3, 64^3), optimization and further versions on-going
  - proof-of-concept integration in CP2K completed and validated
  - create own FFT library, evaluate feasibility of MKL/FFTW-compatible wrappers

Lessons Learned and Conclusion

Lessons: Procurement of HPC Systems with FPGAs

- You will live on the ~~leading~~ bleeding edge
- FPGAs are extremely exotic for HPC OEMs
  - bid evaluation, benchmarks, and acceptance criteria may ruin the complete deal
- We fared well with procuring a tested solution rather than components
  - makes OEM accountable for overall solution
  - validation of FPGA card in specific server, drivers, toolflow, BSP, workload manager integration
  - handling multi-user/application/tool version/BSPs is still challenging
- There will be a gap between specification and reality
  - its a long chain: FPGA device, board, BSP, software tools, driver, ...
- Substantial lead time from FPGA device/card announcement to mature SKU

Conclusions

- The FPGA ecosystem has substantially improved
  - FPGA can compete head to head with other architectures
  - high-level toolflows finally provide productivity and efficiency (~30% of FPL'17 papers use HLS)

- The time is right to move from proof of concept to actual, parallel HPC applications
  - about 40% of FPL'17 paper target data center/HPC topics
  - field is still stuck in proof of concept case studies
  - publish your work as open-source applications and libraries

- In addition there is still a need to improve the foundations
  - stability and sustainability of software and hardware stack
  - better support for HPC languages and libraries (Fortran, OpenMP, OpenACC, MPI)

- Join us in the effort or bringing FPGAs to HPC

christian.plessl@uni-paderborn.de
https://pc2.uni-paderborn.de
Twitter: @plessl // @pc2_upb