PREPARING FOR EXTREME HETEROGENEITY IN HPC

VETTER AND COLTON GRAINGER (SCRIBE)

1. Introduction to ORNL

Funded by the Office of Science (`https://en.wikipedia.org/wiki/Office_of_Science`). - Research labs - Oak Ridge National Laboratory (Oak Ridge, TN) - Argonne National Laboratory (Lemont, IL) - Fermi National Accelerator Laboratory (Batavia, IL) - Lawrence Berkeley National Laboratory (Berkeley, CA) - National Center for Atmospheric Research (Boulder, CO)

- Security labs
  - Los Alamos National Laboratory (Los Alamos, NM)
  - Nevada National Security Site (Las Vegas, NV)

2. Outline

- ambiguous futures
- applications and software systems in a state of crisis
- portable programming models and performance prediction
  - openACC into FGPAs

3. Exascale computing (ECP)

> Exascale computing refers to computing systems capable of at least one exaFLOPS, or a billion billion (i.e. a quintillion) calculations per second. Such capacity represents a thousandfold increase over the first petascale computer that came into operation in 2008. `https://en.wikipedia.org/wiki/Exascale_computing`

3.1. applications.

From `https://www.exascaleproject.org/project/`:

Chemistry and Materials Applications - EXAALT: Molecular Dynamics at Exascale for Materials Science - ExaAM: Transforming Additive Manufacturing through Exascale Simulation - GAMESS: Enabling GAMESS for Exascale Computing in Chemistry and Materials - LatticeQCD: Lattice Quantum Chromodynamics for Exascale - NWChemEx: Tackling Chemical, Materials, and Biomolecular Challenges in Exascale - QMCPACK: Quantum Mechanics at Exascale

Co-Design - AMReX: Block-Structured Adaptive Mesh Refinement (AMR) Co-Design Center - Application Assessment - CEED: Center for Efficient Exascale Discretizations - CODAR: Co-Design Center for Online Data Analysis and Reduction at Exascale - COPA: Co-Design Center for Particle Applications - ExaGraph: Co-Design Center for GraphEx - ExaLearn: Co-Design Center for Exascale Machine Learning Technologies - Proxy Applications

Data Analytics and Optimization - CANDLE: Exascale Deep Learning Enabled Precision Medicine for Cancer - ExaBiome: Exascale Solutions for Microbiome Analysis - ExaFEL: Data Analytics at Exascale for Free Electron Lasers - ExaSGD: Optimizing Stochastic Grid Dynamics at Exascale - Urban: Multiscale Coupled Urban Systems

Earth and Space Science Applications - E3SM-MMF: Cloud-Resolving Climate Modeling of the Earth's Water Cycle - EQSIM: High-Performance, Multidisciplinary Simulations for Regional-Scale Earthquake Hazard/Risk Assessments - ExaSky: Computing the Sky at Extreme Scales - ExaStar: Exascale Models of

Date: 2019-05-22.
url: `true`.

Stellar Explosions - Subsurface: Exascale Subsurface Simulator of Coupled Flow, Transport, Reactions, and Mechanics

Energy Applications - Combustion-Pele: Transforming Combustion Science and Technology with Exascale Simulations - EXAWIND: Exascale Predictive Wind Plant Flow Physics Modeling - ExaSMR: Coupled Monte Carlo Neutronics and Fluid Flow Simulation of Small Modular Reactors - MFIX-Exa: Performance Prediction of Multiphase Energy at Exascale - WDMApp: High-Fidelity Whole Device Modeling of Magnetically Confined Fusion Plasmas - WarpX: Exascale Modeling of Advanced Particle Accelerators

National Security Applications - ASC ATDM LANL Application - ASC ATDM LLNL Application - ASC ATDM SNL Application

3.2. **E4S.**

> E4S exists to accelerate the development, deployment and use of HPC software, lowering the barriers for HPC users. E4S provides containers and turn-key, from-source builds of more than 80 popular HPC products in programming models, such as MPI; development tools such as HPCToolkit, TAU and PAPI; math libraries such as PETSc and Trilinos; and Data and Viz tools such as HDF5 and Paraview. `https://e4s-project.github.io/`

4. Motivating trends

- "GlobalFoundries forfeit 7nm Manufacturing"
- "HPE to acquire Cray supercomputing"

Which (Kurzweilian) wave of computing are we entering?

1. electromechanical
2. mainframes
3. minicomputers
4. personal computers
5. networking and internet
6. integrated circuits
7. transition technologies
   - chip design: quantum, neuromorphic, advanced digital, emerging memory devices

> Clearly, we will be able to build such component-crammed equipment. Next, we ask under what circumstances we should do it. The total cost of making a particular system function must be minimized. To do so, we could amortize the engineering over several identical items, or evolve flexible techniques for the engineering of large functions so that no disproportionate expense need be borne by a particular array. Perhaps newly devised design automation procedures could translate from logic diagram to technological realization without any special engineering. It may prove to be more economical to build large systems out of smaller functions, which are separately packaged and interconnected. The availability of large functions, combined with functional design and construction, should allow the manufacturer of large systems to design and construct a considerable variety of equipment both rapidly and economically. newly devised design automation procedures could translate from logic diagram to technological realization evolve flexible techniques for the engineering of large functions more economical to build large systems out of smaller functions, which are separately packaged. [*Cramming More Components onto Integrated Circuits*, Gordon Moore, 1965]

- maintaining and improving programmer productivity
-

5. Programming for heterogeneous computing

Here's the current "out of control" approach.

system	notes	cores
MPI, Legion, HPX, Charm++, etc.	OpenMP, Pthreads, U-threads, etc.	OpenACC, CUDA, OpenCL, OpenMP4, SYCL, Kokkos

5.1. **Challenges in FPGA computing.**

Reconfigurable computing is a computer architecture combining some of the flexibility of software with the high performance of hardware by processing with very flexible high speed computing fabrics like field-programmable gate arrays (FPGAs). https://en.wikipedia.org/wiki/Reconfigurable_computing

Kernel-pipelining transformation optimization

- kernel execution model in OpenACC
  - device kernels can communicate with each other only through device global memory
  - synchronizations between kernels are at the granularity of a kernel execution

TODO

```
#pragma acc data copyin (a) create (b) copyout (c)
{
    #pragma acc kernels loop gang worker present (a,b)
    TODO loop here
    #pragma acc kernels loop gang worker present (b,c)
    TODO loop here
}
```

“CUDA vs FPGA?”[1]. Stack Overflow. Retrieved May 22, 2019. “Learning OpenCL is now sufficient to program FPGAs - altera.com/products/design-software/... and given the primary importance of power combined with the lower power requirements of FPGAs for similar performance, FPGAs now offer an attractive alternative to GPU for many problem types. – ProfNimrod Jan 28 ’16 at 15:01”

5.2. **Standards for FPGCs at ORNL.**

- OpenCL offices an open standard portable across diverse heterogeneous platforms; much higher than HDL, but complex for typical programmers.
- OpenARC is an open source C compiler developed at Oak Ridge National Laboratory to support all features in the OpenACC 1.0 specification. https://en.wikipedia.org/wiki/OpenACC

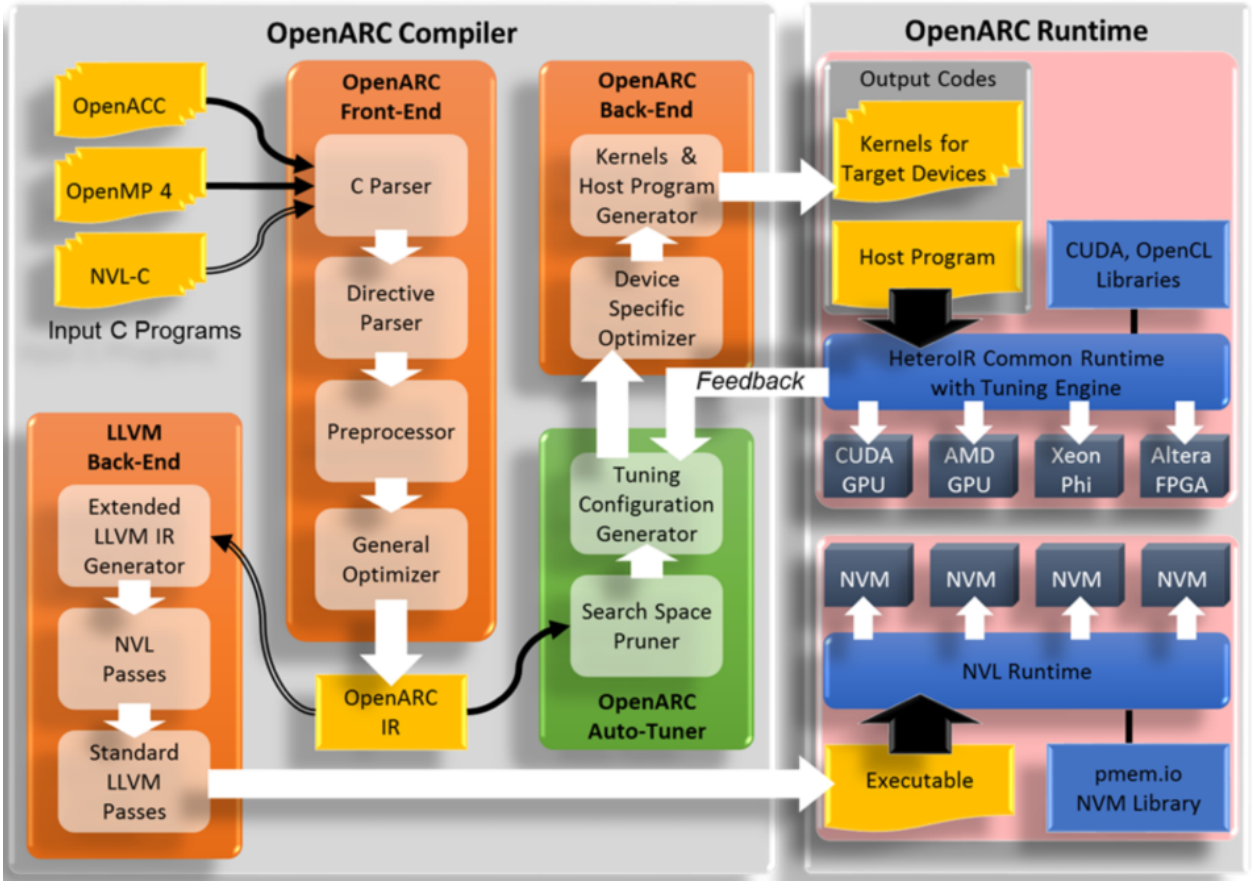
  “OpenACC to FPGA: A Directive-Based High-Level Programming Framework for High-Performance Reconfigurable Computing”[2]. Retrieved May 22, 2019. “Accelerator-based heterogeneous computing has become popular solutions for power-efficient high performance computing (HPC). Along these lines, Field Programmable Gate Arrays (FPGAs) have offered more advantages in terms of performance and energy efficiency for specific workloads than other accelerators. Nevertheless, FPGAs have traditionally suffered several disadvantages limiting their deployment in HPC systems, mostly due to the challenges of programmability and portability. We present a directive-based, high-level programming framework for high-performance reconfigurable computing. It takes a standard, portable OpenACC C program as input and generates a hardware configuration file for execution on FPGAs. We implemented this prototype system in our open-source OpenARC compiler, which uses the Altera OpenCL compiler as its backend. Preliminary evaluation of the proposed framework on an

[1] https://stackoverflow.com/questions/317731/cuda-vs-fpga

[2] https://sc18.supercomputing.org/proceedings/tech_poster/tech_poster_pages/post229.html

> Intel Stratix~V with five OpenACC benchmarks demonstrates that our proposed FPGA-specific compiler optimizations and novel OpenACC pragma extensions assist the compiler in generating more efficient FPGA programs."

See also Vetter's poster[3].



5.3. **FGPAs against GPUs, overall performance.**

- FPGAs prefer applications with deep execution pipelines, e.g., FFTs, SpMul.
- GPUs prefer MatMul and Jacobi calculations.

6. Emerging memory systems

TODO: define nonvolatile memories (see *A Survey of Software Techniques for Using Non-Volatile Memories for Storage and Main Memory Systems*)

> DRAGON: Direct Resource Access for GPUs Over NVM `https://github.com/pakmarkthub/dragon`

- Three memory spaces
  - GPU mem (GM)
  - Host mem (HM)
  - NVM as primary storage
- modified GPU driver
  - manages data movement and coherency

[3] `https://sc18.supercomputing.org/proceedings/tech_poster/poster_files/post229s2-file2.pdf`

intel/caffe: Caffe is a deep learning framework made with expression, speed, and modularity in mind. It is developed by the Berkeley Vision and Learning Center (BVLC) and community contributors. (This fork of BVLC/Caffe is dedicated to improving performance of this deep learning framework when running on CPU, in particular Intel® Xeon processors.) `https://github.com/intel/caffe`

7. Summary

1. See the DOE report from the ORNL.

   "Extreme Heterogeneity 2018 - Productive Computational Science in the Era of Extreme Heterogeneity: Report for DOE ASCR Workshop on Extreme Heterogeneity (Technical Report) | OSTI.GOV"[4]. Retrieved May 22, 2019. "The 2018 Basic Research Needs Workshop on Extreme Heterogeneity identified five Priority Research Directions for realizing the capabilities needed to address the challenges posed in this era of rapid technological change. This report captures the outcomes of that workshop. In the context of extreme heterogeneity, it defines basic research needs and opportunities in computer science research to develop smart and trainable operating and runtime systems, programming environments, and predictive tools that will make future systems easier to adapt to scientists' computing needs and easier for facilities to deploy securely."

2. exascale computing has ambiguous future
3. complexity is the next major hurdle
   - heterogeneous computing
   - deep memory with NVM
4. software solutions?
   - programming
     - memory
       - DRAGON
       - NVL-C
       - Papyrus
     - heterogeneity
       - OpenACC to FPGAs
       - Clacc for LLVM
5. "impacts on software application and application design"

[4] `https://www.osti.gov/biblio/1473756`