SCIENTIFIC SOFTWARE PRODUCTIVITY – CASE STUDIES, CHALLENGES, OPPORTUNITIES & POTENTIAL SOLUTIONS

Sunita Chandrasekaran
Assistant Professor, University of Delaware
Dept. of Computer & Information Sciences
schandra@udel.edu

Collegeville Workshop July 2020

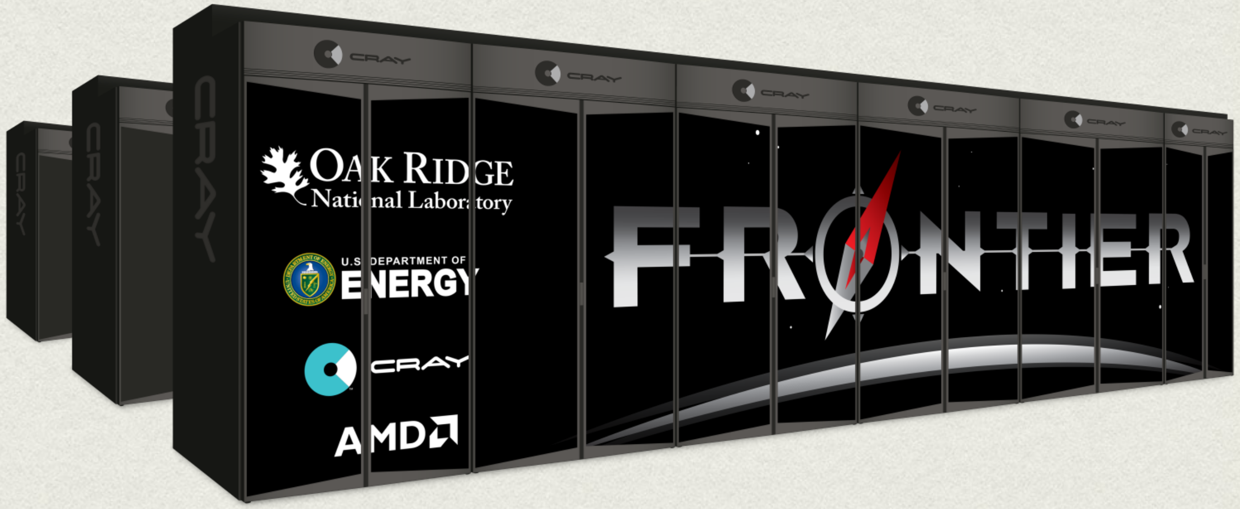
Some topics to ponder about…….

- SOFTWARE MIGRATION
- BENCHMARKING EFFORT
- DATA ANALYTICS AND ML PIPELINES
- RSE

SOFTWARE MIGRATION (OR REWRITE?) FROM ONE SYSTEM TO ANOTHER!

CAAR
ORNL-PICONGPU-FRONTIER

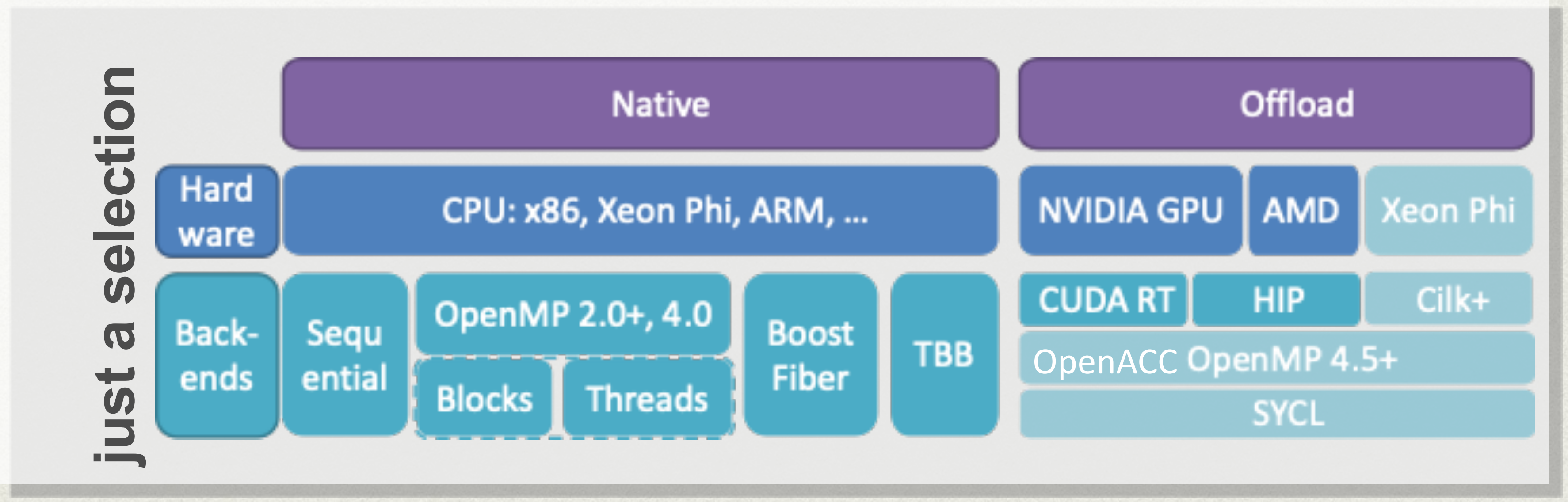
- Preparing PIConGPU, a plasma Physics application for the upcoming exascale system - Frontier



CAAR Project in Collaboration with COE (AMD + Cray) developers

Portability and maintainability

Performance Portability & maintainability on different architechtures



just a selection

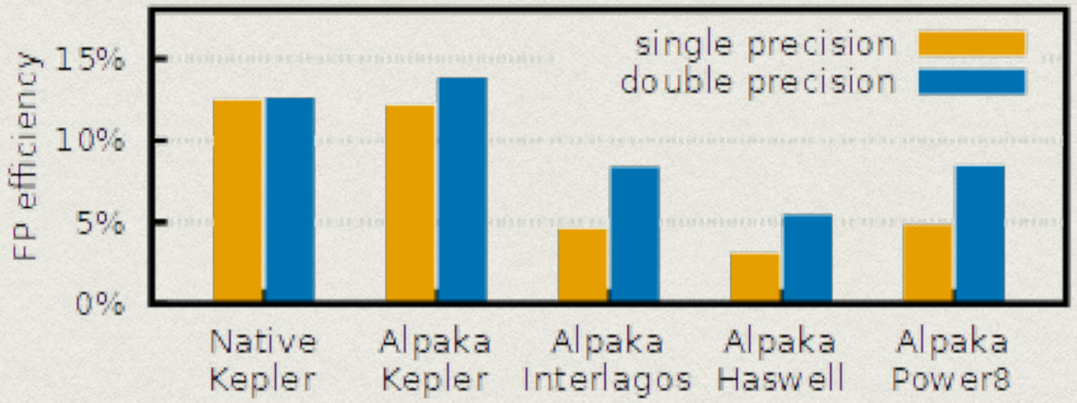
Native | Offload

Hard ware | CPU: x86, Xeon Phi, ARM, ... | NVIDIA GPU | AMD | Xeon Phi

Back-ends | Sequential | OpenMP 2.0+, 4.0 (Blocks, Threads) | Boost Fiber | TBB | CUDA RT | HIP | Cilk+ | OpenACC OpenMP 4.5+ | SYCL

C++ solutions:
Alpaka & cupla
single-source

```
template< typename T_Acc >
ALPAKA_FN_ACC void operator()(
    T_Acc const & acc,
    // ...
) const
{
    // ...
}
```



- **Maintainability**
- PIConGPU + PMacc code lines

- **Before:** 80k LOC (20k in kernels) **After:** 50k LOC (1 year)

E. Zenker et al., ISC (2016), 10.1007/978-3-319-46079-6_21
A. Matthes et al., ISC (2017), 10.1007/978-3-319-67630-2_36

A number of opportunities

- Create **mini test codes** to stress test compilers and hardware architectures
- Seek help from **profilers** to learn about the (new) architecture
- Explore and express **parallelism** at multiple levels (hardware, software and at the application level)
- Develop **synergies** between programming model communities to avoid reinventing the wheel
- Leverage **alpaka and OMP offloading V&V suite efforts**
- For CAAR-PIConGPU
  - Create a HIP and OMP5 offloading back-ends for Alpaka
  - Work closely with the Center of Excellence team to report bugs (aomp/clang, HIP, GNU, PGI, IBM XL)
  - Open new tickets/issues for newer feature requests with programming models

REPORT BUGS!

- **Code Review - Rule of Thumb:** Author of a PR cannot merge his/her own PR)
- **Report bugs:** Help improve compilers
- **Reproducible code:** Useful to debug
- **REPORT:** Workarounds OK but ”REPORT” bugs
- Report bugs via a ticket system (say Trac – wiki + issue tracker) and not via email – PLEASE! ☺
  - The bug and its fix got to be recorded
  - Documented
  - Code changes to be tracked
- **Time critical bugs** - Communicate with the developers directly

BENCHMARKING EFFORT

Stress testing hardware & software

- Work in collaboration with SPEC HPG
- Need a benchmark suite representative of scientific applications to stress test hardware and software
  - Mini-applications are OK as long as they reflect the physics of real apps
  - Consider various workloads
  - Consider weak/strong scaling applications
  - Must have mechanisms in place for validation (accuracy)
  - Performance modeling to draw insights into hardware and its impact on applications
  - Push limits of compilers and tools along with hardware architecture

APPLICABILITY OF DATA ANALYTICS AND ML ON REAL DATASET

Data Analytics ML-Based Pipeline for Omics + EHR data

- Challenges from the application of ML techniques on real subject (patient) dataset
- Real dataset are complex, disproportionate, heterogeneous, small and skewed
- Traditional ML techniques/pipelines cannot be directly applied
- Need for an end-end solution for data analysis of such complex dataset
    - Data preparation, cleaning, feature selection, classification, validation
- NGS technologies are improving as we speak; dataset will grow, are we ready with scalable data analytics pipeline for subject dataset?

RESEARCH SOFTWARE ENGINEERS (RSE)

Who is behind research software?

RSEs

- Software is critical to research success
- Sustainability and Reproducibility of software as part of research methods
- Driving the need of RSEs within academia
    - Need the push from labs and industries
- RSE as an essential core competence among young scientists

de RSE

US RSE

US RESEARCH
SOFTWARE ENGINEER
ASSOCIATION

https://de-rse.org/en/index.html

Best Practices - 7 of many!

- Best Practice #1 Profiling
- Best Practice #2: Systematic Testing
- Best Practice #3: Report bugs
- Best Practice #4: Automate
- Best Practice #5: Document
- Best Practice #6: Pair Programming
- Best Practice #7: Open Source but…

My group in action :-)

CRPL
Sunita Chandrasekaran

Computational Research and Programming Lab