Scientific Software Productivity – Case studies, Challenges, Opportunities and Potential Solutions
Sunita Chandrasekaran, Assistant Professor, University of Delaware, USA

Migrate a scientific application from one architecture to another

(Case study: ORNL CAAR project, migration of plasma physics application from ORNL's Summit to Frontier. Work in collaboration with ORNL, COE and HZDR, Germany)

- Challenges with the migration process
  - Need to address programmability, performance and portability gaps emerging during the migration process of code from one architecture to another
  - How robust and usable are the compiler/tools for the new system?
    - Need tests, validation, verification and acceptance of tools before they can be used for real code
    - Need mini test codes for evaluation purposes, to provide feedback to vendors, and identify steps to incrementally improve performance on the main code under study
    - An optimization or a porting strategy that worked for System A may not be the same for System B due to architecture A being different to architecture B
  - Need profilers for emerging systems that can probe into the scientific application and recommend optimizations/improvements to the compiler and tools
  - Software must be maintainable, incrementally improved as science advances, and where possible - performant portable
  - Challenges not only on the programmability aspects but also on the I/O, visualization and networking aspects
  - What next? After frontier, yet another migration process? ☺
  - While it is absolutely important for the architectures to evolve, the software stack takes a long while before they can adapt themselves to evolving hardware – this has been the trend
- Opportunities during the migration process
  - Develop novel R&D strategies to express parallelism at multiple levels i.e. software, hardware and at the application level
  - May be the newer system is creating opportunities for scientific advancements
    - E.g. Could the upcoming systems facilitate an improved MRI machine via advancements in science and technology that will no longer require a patient undergoing shooting pain to remain still for several minutes together
- Potential solutions for the migration process
  - Develop synergy between different working programming model communities to bridge programmability gaps by taking the best of the worlds and not by reinventing the wheel and creating yet another new programming model, compiler and runtime components

    - Create better usability (infrastructure? Pipeline?) of hardware/software stack for domain scientists
    - Develop next-generation workforce – we need MORE RSEs!!!!
        - Encourage universities to build RSE pipeline
        - Need accountability from funding organizations to help build RSE pipeline (RSE needs to be a line item in budgets)

Develop a data analytics scalable pipeline/workflow for big data

(Case Study: Classification of patients cohort for datasets, e.g. Pediatric Cancer Dataset. Work in collaboration with Nemours duPont I Hospital for Children, Delaware)

- Challenges
    - Exponential increase in data availability and also at times, lack of adequate data availability (biology and rare type diseases)
    - Data quality is questionable
    - Can the ML/DL techniques developed for synthetic data be applicable for real data?
    - How to handle missing data which is common in biology dataset
    - Need for data analysis and classification at scale
    - Need for an end-to-end solution
        - Data gathering, cleaning, processing, feature selection, classification, voting for the best classifier
        - Accuracy, Sensitivity and Specificity metrics cannot be compromised
- Opportunities
    - Work closely with domain scientists to develop techniques for data cleaning, manipulation for their dataset.
        - Note: every dataset is different. Steps applied to a bio dataset cannot be directly translated to a sun's simulation dataset
    - Explore different techniques for data analysis and its suitability for the particular dataset under study
    - Identify gaps in current techniques for feature selection, analysis and classification
    - Explore data classification techniques, suitability to not just synthetic data but also to real-datasets
- Potential Solutions
    - Create a data analytics scalable pipeline of data cleaning, analysis and classification steps. Each step can have multiple techniques. Refresh and repeat the pipeline. Identify steps that can run in parallel and scale. Put in place a model validation and model voting step to filter out the best combination of techniques per step

Disruptive hardware is fantastic for exploration, however a study/report on the suitability of architectures for applications from different scientific domains is needed. How do we enable this action item?

(Case Study: Benchmarking Effort. Work in collaboration with SPEC HPG)

- Challenges with choosing benchmark candidate
    - Derive representative benchmarks; while creating mini-applications is a great approach, a word of caution is to ensure the mini-app reflects the physics of the real application
    - Full application could be encouraged provided they can be packaged and not contain too many dependencies
    - Candidate is capable of solving problems of various sizes (workloads) scaling single to multiple nodes
    - Accuracy is not lost in the benchmark development process; must have mechanisms in place for validation
    - Highly parallel, computationally intensive, of course
    - Spans more than a few application domains
    - Supports more than just 1-2 programming models
- Opportunities while developing a benchmark suite
    - Creating and developing application-facing benchmark suites and analyzing their impact on hardware and software
    - Develop performance modeling of applications to draw insights into the hardware and software the application targets
    - Analyze scalability studies of the application on more than one hardware
    - Systematic analysis and reporting of software bugs and re-testing the fixes to enable software robustness
    - Push limits of software (compiler and tools) and hardware boundaries from a scientific standpoint
- Potential Solution
    - Work in progress, SPEC HPG HPC2020 benchmark creation effort started in 2017. https://dev-www.spec.org/hpg/hpc/
    - Team includes AMD, HPE, Intel, IBM, Lenovo, NVIDIA, HZDR, IU, LBNL, ORNL, RWTH Aachen, UBasel, UDelaware
    - First Workshop Report https://scholarworks.iu.edu/dspace/bitstream/handle/2022/25344/Application_Benchmarking_Workshop_Report.pdf?sequence=3&isAllowed=y
    - Tentative Benchmark Release – November 2020