11th World Congress on Computational Mechanics (WCCM XI)
5th European Conference on Computational Mechanics (ECCM V)
6th European Conference on Computational Fluid Dynamics (ECFD VI)
July 20 - 25, 2014, Barcelona, Spain

Developing Software Frameworks for Petascale Computers and Beyond using Dynamic Graph-Based Approaches – Lessons and Achievements

Martin Berzins[1*], Alan Humphrey[2] Qingyu Meng[3] and John Schmidt[4]

[1] SCI Institute, University of Utah Salt Lake City UT 84112 USA, mb@sci.utah.edu
[2] SCI Institute, University of Utah Salt Lake City UT 84112 USA, ahumphrey@sci.utah.edu
[3] SCI Institute, University of Utah Salt Lake City UT 84112 USA, qymeng@sci.utah.edu
[4]SCI Institute, University of Utah Salt Lake City UT 84112 USA, jas@sci.utah.edu

Key Words: *Uintah software, software engineering, directed acyclic graphs, petascale computing, portability.*

The challenge of creating software that may be used to solve challenging engineering problems while making full use of large-scale parallel computers has always been considerable. Present trends in hardware however add additional complexity in that there is considerable uncertainty about the nature of the next generation or two of parallel computers, except that such machines will make use of accelerators and will attempt to reduce the power requirements of high performance computing. Both the complexity of the architectures and the complexity of the problems being solved make it challenging to create long-lived software without adopting a multi-disciplinary approach.

In this talk we will describe the Uintah software effort at the University of Utah and explain how the approach that we are using makes it possible to address some of these challenges while making it possible for the users of the software to solve challenging multi-scale engineering problems. This software originated in the DOE funded CSAFE activity at the University of Utah and has been further developed recently to run on a range of machines from laptops to some of the largest supercomputers in use today. The software is developed by a multi-disciplinary group using many software engineering approaches such as version control and automated testing while also using a software decomposition approach that enables applications developers to code applications and computer scientists to improve scalability and performance without each group disrupting the others efforts.

A central idea in the Uintah software is to create an abstract problem specification layer that deals only with the specification of the computational tasks but not their full implementation. The user writes code for an abstract hexahedral mesh patch that only specifies its connectivity via halo elements through a data warehouse. No explicit communications are specified.
The tasks are then executed by a runtime system that both maps the tasks to hardware and executes them in a dynamic and asynchronous and often indeed out-of-order way. We will show that it is not just the taskgraph approach that is responsible for the scalability of the

software, but that it is the implementation of the taskgraph is critical. A key feature of this approach is that it meets our desired goal of allowing application developers and computer scientists to work independently on their contributions to the software.

Although this model has currently proved to be very successful in allowing Uintah to scale to as many as 500K cores for challenging applications, it is important to show that the approach will work on broad architectures, for example those that include accelerators and GPUs and also to show if the approach will encompass multi-scale simulations that include not just continuum solvers but coarse-grained and standard molecular dynamics codes. The talk will provide examples that show that it is possible to achieve these goals within the Uintah framework.

Finally a major, but hidden, challenge lies in forming an effective multidisciplinary team that is capable of working together in such an activity. We will describe some of the challenges that have arisen and how they have been overcome. Throughout the talk, concrete examples will be used relating to realistic fluid-structure and multi-scale applications. The examples will range from fires and explosions through to materials by design in the context of electrochemistry applications and also describe current efforts underway in a recently funded DOE PSAAP2 Center that is tackling a problem related to clean-coal energy production.

Finally the talk will contrast the Uintah approach with other similar frameworks and discuss the strengths and weaknesses of the approach adopted for this framework.

REFERENCES

[1] M. Berzins, J. Schmidt, Q. Meng, A. Humphrey. **"Past, Present, and Future Scalability of the Uintah Software,"** In *Proceedings of the Blue Waters Extreme Scaling Workshop 2012*, pp. Article No.: 6. 2013.

[2] Q. Meng, A. Humphrey, J. Schmidt, M. Berzins. **"Investigating Applications Portability with the Uintah DAG-based Runtime System on PetaScale Supercomputers,"**to appear in the proceedings of SuperComputing 2013 ACM. See also SCI Technical Report, No. UUSCI-2013-003, *SCI Institute, University of Utah*, 2013.