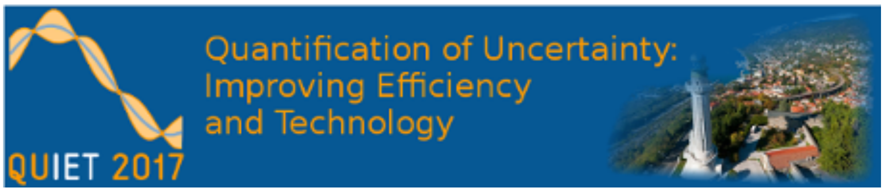

Tackling UQ in DARMA, a Programming Model for Task-Based Execution at Extreme-Scale

F. Rizzi[1], E. Phipps[1], D. Hollman[1], J. Lifflander[1], J. Wilke[1], A. Markosyan[1], H. Kona[1], N. Slattengren[1], K. Teranishi[1], and J. Bennett[1]

[1]Sandia National Laboratories, United States

This talk focuses on the advantages of task-based implementations and execution models for UQ problems. Task-based models show the potential to mitigate critical challenges posed by extreme-scale architectures such as exposing maximal parallelism, managing data locality and deep memory hierarchies, and hiding communication latency. Specifically, we demonstrate how to tackle UQ using DARMA (Distributed Asynchronous Resilient Models and Applications) (`http://www.sandia.gov/darma/`), an abstraction layer for asynchronous many-task (AMT) runtimes [2] that uses C++ template metaprogramming to facilitate the capture of data-task dependencies [1]. In this talk we (a) summarize how concurrency and parallelism are expressed using DARMA among independent UQ tasks, and (b) present an analysis of the benefits of reusing, within a pool of samples, tasks results to accelerate the execution time of other independent tasks. We demonstrate a set of basic UQ examples written in the DARMA, and then focus on a Multi-Level Monte Carlo test case.

References

[1] D. S. Hollman, J. C. Bennett, H. Kolla, J. Lifflander, N. Slattengren, and J. Wilke. Metaprogramming-enabled parallel execution of apparently sequential c++ code. In *Proceedings of the Second Internationsl Workshop on Extreme Scale Programming Models and Middleware*, pages 24–31, 2016.

[2] J. J. Wilke, J. C. Bennett, D. S. Hollman, N. L. Slattengren, H. Kolla, F. Rizzi, R. L. Clay, and K. Teranishi. The darma approach to asynchronous many-task programming. Technical report, Presented at ECP Review 2016, Sandia National Laboratories, 2016.