optimization and uncertainty quantification at exascale

Abstract

We have developed and implemented a comprehensive, rigorous and algorithmic framework capable of utilizing all available information to rigorously predict the impact of high-impact rare events, where our predictors are multiply-nested global optimizations over all possible valid scenarios. Such optimizations are high-dimensional, highly-constrained, non-convex, and generally impossible to solve with current optimization technology; however, by addressing optimization constraints as quantum operators on a probability distribution, our software converts highly-nonlinear statistical calculations to those that are nearly embarrasingly parallel. By utilizing abstractions on programming models and global distributed caching of data and results, we can scale up from desktop calculations to petascale and larger with little burden on the programmer, where the upscaling depends on the complexity of the information at hand (constraints, restraints, theoretical models, datasets, etc).

Optimal Systems Design From a broad perspective, our ongoing work is to build an algorithmic framework for uncertainty quantification (UQ), which can rigorously calculate critical system design elements such as *value at risk* and *hedge*. Our framework can address model uncertainty and optimal design questions such as: How "good" is my model? How can I best improve my model? Given the uncertainty on the system performance, does it make sense to push certain design boundaries? The infrastructure described below supports Caltech's Predictive Science Academic Alliance Program Center (PSAAP) and the Exascale Co-design Center for Materials in Extreme Environments (ExMatEx), and leverages the optimal uncertainty quantification (OUQ) framework [5]. This work is in close collaboration with LLNL, LANL, and BNL, and is supported by the AFOSR under award number FA9550- 12-1-0389 (Scientific Computation of Optimal Statistical Estimators).

UQ in the Co-Design Loop We are deploying the *mystic* and *pathos* [3, 1] frameworks on high-performance computing resources targeted by ExMatEx and other efforts, to help address the question of exascale system design and performance optimization. The *pathos* framework provides an abstraction layer on programming models for heterogeneous parallel and distributed computing. Currently *pathos* has multi-core, multi-node, multi-thread, and multi-cluster launching capabilities, and we will extend the abstraction layer to include GPU-based and SMT-enabled platforms. Exascale platforms are expected to have strongly probabilistic behavior, and thus we will focus on extending *pathos* to provide dynamic load balancing, scheduling, and communication strategies that leverage developments in OUQ. This will enable more complex and highly efficient parallel map algorithms to be developed, including those that provide robust fault-tolerant behavior. Using *mystic* and *pathos* to drive ExMatEx's system emulator, GREMLIN, could enable rigorous performance measures for the stress-testing of models of exascale system design to be developed.

The mathematics of optimal systems design As shown in [4], when an unknown distribution π' is generating sample data, the accuracy of a statistical estimator can still be expressed as the solution of a well posed optimization problem (with optimization variables corresponding to measures over the admissible set). For instance, if the accuracy of a statistical estimator θ is expressed in a mean squared sense, then it can be written as

$$\sup_{\pi'} \mathbb{E}_{(g,\nu,data)\sim\pi'}\left[\Big(\theta(data) - \Phi(g,\nu)\Big)^2\right] \tag{0.1}$$

where the supremum in π' is taken over all distributions π' than can generate the data (and the admissible scenario (g, ν) for the unknown reality $(G, \mathbb{P})$). An optimal statistical estimator is then the solution of a min-max optimization problem written as

$$\inf_{\theta} \sup_{\pi'} \mathbb{E}_{(g,\nu,data)\sim\pi'}\left[\Big(\theta(data) - \Phi(g,\nu)\Big)^2\right] \tag{0.2}$$

where the infimum is taken over all functions θ of the data.

A new paradigm for constrained global optimizations While equation (0.2) can be simply expressed, it carries a large computational burden. The root of this equation is a global optimization over all possible scenarios, an OUQ calculation, while the exterior layer is an additional global optimization over all functions of the data. OUQ is implemented through a scale-bridging paradigm, where the optimization over scenarios dynamically spawns constraints solvers that ensure the constraints are respected. These constraints solvers are often global optimizations themselves, and can be coupled to other dynamically deployed solvers or sources of information. The UQ software framework, *mystic* [2], provides optimization algorithms that leverage parallel computing at several levels – population-based solvers [6] that can launch function evaluations in parallel, and ensemble solvers that can launch multiple optimizers in parallel [3]. The complexity required by hierarchial optimizations is managed by the underlying graph execution and management framework, *pathos* [1]. With computations of this size and complexity, resource failures will not only occur, they will be commonplace. The framework must be robust against failure, and both robust and reactive in the presence of dynamically shifting resources. New tools for dynamic workload balancing and efficient parallelism in the face of resource failure must be developed.

Scalability through asynchronous parallel computing OUQ calculations cast as global optimizations with dynamically spawned constrains solvers are possible because *mystic* provides a functional abstract programming interface (API) for constraints solvers and optimization algorithms. The OUQ calculations drive the execution logic, and through the functional API, stage and launch several thousand optimizer and constraints solver instances.

A full suite of blocking, iterative non-blocking, and asynchronous maps and pipes were developed for multiprocessing, MPI, and IPC-based distributed computing. With a unified API for programming models, we have taken the first steps toward dynamic workload balancing across heterogeneous resources. We have also implemented better and more stable bindings between graph nodes and cluster schedulers, as well as more stable workload management strategies.

All of *mystic's* optimizers were adapted for asynchronous computing, where each optimizer is now capable of saving its full state at each iteration in the optimization. While most optimization frameworks implement optimization algorithms that block until complete, *mystic's* new asynchronous solvers enable optimizations to proceed step-by-step, with full restart capabilities at each iteration. Because *mystic's* solvers are also serializable, optimizer state can be shipped off to a database simply by logging the optimizer itself in the database. Additionally, this serialization enables an optimizer to be stopped, saved, shipped to another resource, and then restarted without any loss of information, accuracy, or progress.

UQ in dynamic graph execution and management One of the core design requirements in moving to dynamic graph execution is the ability to measure and predict the availibility of heterogeneous resources. This is fundamentally nothing more than statistical estimation based on available data, models, and other information. In that light, system resource monitors and system programming models could be utilized in statistical estimators for dynamic decision-making in workload balancing and efficient parallel maps.

References

[1] M. McKerns and M. Aivazis. Pathos: A framework for heterogeneous computing, 2010. http://pythonhosted.org/pathos.

[2] M. McKerns, P. Hung, and M. Aivazis. Mystic: A simple model-independent inversion framework, 2009. http://pythonhosted.org/mystic.

[3] M. M. McKerns, L. Strand, T. J. Sullivan, A. Fang, and M. A. G. Aivazis. Building a framework for predictive science. In *Proceedings of the 10th Python in Science Conference (SciPy 2011)*, 2011.

[4] H. Owhadi, C. Scovel, and T. Sullivan. Bayesian Brittleness: Why no Bayesian model is "good enough". 2013. http://arxiv.org/pdf/1304.6772.

[5] H. Owhadi, C. Scovel, T. J. Sullivan, M. McKerns, and M. Ortiz. Optimal Uncertainty Quantification. *SIAM Review*, 2013. http://arxiv.org/pdf/1009.0679v2.

[6] R. M. Storn and K. V. Price. Differential evolution — a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.*, 11(4):341–359, 1997.