SAND2013-7007C

Realizing Exascale Performance for Uncertainty Quantification

Eric Phipps (etphipp@sandia.gov),
H. Carter Edwards, Jonathan Hu
Sandia National Laboratories
and
Clayton Webster
Oak Ridge National Laboratory

DOE Workshop on Applied Mathematics Research for Exascale Computing

August 21-22, 2013

SAND 2013-xxxx C

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Can Exascale Solve the UQ Challenge?

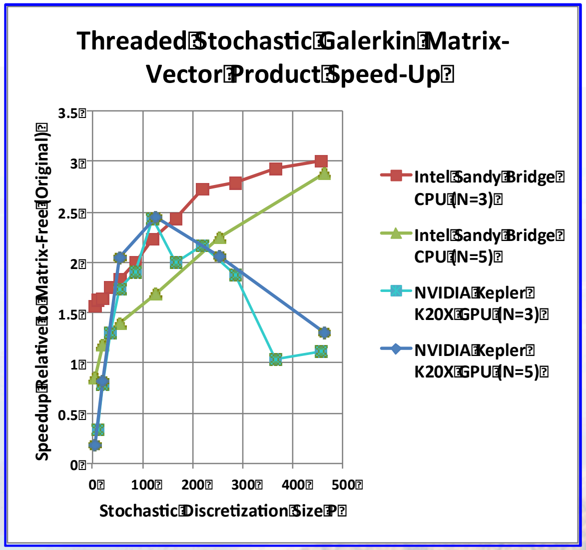
- UQ means many things
  - Best estimate + uncertainty, model validation, model calibration, …
- A key to many UQ tasks is forward uncertainty propagation
  - Given uncertainty model of input data (aleatory, epistemic, …)
  - Propagate uncertainty to output quantities of interest
- There are many forward uncertainty propagation approaches
  - Monte Carlo, stochastic collocation, polynomial chaos, stochastic Galerkin, …
- Key challenge:
  - Accurately quantifying rare events and localized behavior in high-dimensional uncertain input spaces
  - Can easily require $O(10^4\text{-}10^6)$ expensive forward simulations
  - Often can only afford $O(10^2)$ on today's petascale machines

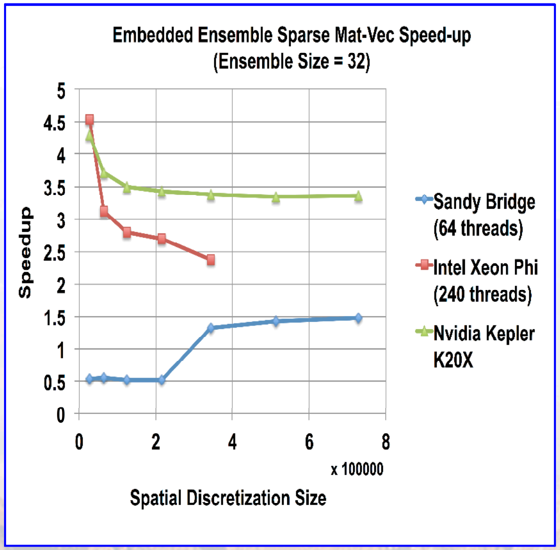
Achieving Exascale Performance Requires New Approaches

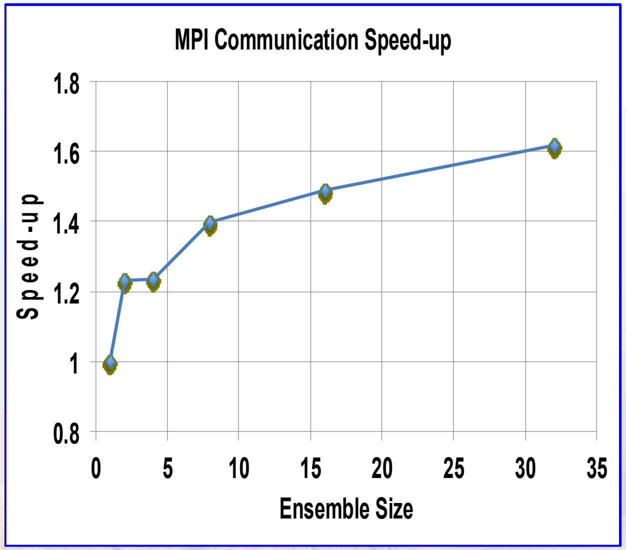
- UQ approaches usually implemented as an outer loop
  - Repeated calls of deterministic solver
  - Coarse-grained distributed memory parallelism over samples
  - How do we achieve a 1000-fold increase in available uncertainty propagation?
- No increase in clock-speed
  - Must increase parallelism
- No decrease in latency, latency hiding through instruction-level parallelism & out-of-order execution replaced by hardware multi-threading and vectorization
  - Simulations must exhibit good data locality and expose sufficient fine-grained parallelism
  - Extremely challenging for many simulation algorithms, e.g., sparse linear algebra on complex (unstructured) domains
- Little increase in total node count, dramatic increase in node-level parallelism
  - Must evaluate multiple samples in parallel *on each node*
- Node memory increase of 0.1-0.01 of floating-point capacity
  - Parallel sample evaluations must share data when possible (threads)
- UQ is a highly structured calculation
  - Add new dimensions of fine-grained parallelism through *embedded* approaches

Scalar/Core-level Uncertainty Propagation

- Propagate collections of uncertainty information at scalar-level of calculation
  - scalar -> array (e.g., samples, PCE coefficients)
  - random memory access -> contiguous array access
  - scalar arithmetic -> parallel_for







Benefits

- Improved data locality
- Amortize latency/communication across UQ array
- New dimension for fine-grained parallelism
  – Vectorization and hardware multi-threading
- Data reuse
  – Mesh/graph data structures
- Solver/preconditioner reuse/acceleration
  – Single preconditioner for UQ array
  – Recycle Krylov bases
  – Reuse multi-grid hierarchy/aggregrates
  – Accelerate solver by interpolating between samples

Challenges and Opportunities

- Significant effort to refactor simulation codes
  - Introduce abstraction at scalar level
  - *Template-based generic programming*
- Increased cache pressure
  - Can't make UQ array too big
- Propagating samples together requires commonality in solution process
  - Often need to refine UQ discretization near localized behavior/discontinuities/bifurcations
  - How to group samples to exploit commonality when you have it, and separate samples when you don't?
- Improve flop/byte ratios
  - Embedded sample propagation doesn't change flops/byte
  - Stochastic Galerkin increases flops/byte
- Solvers/preconditioners optimized for embedded uncertainty propagation
  - Kronecker product structure
- Partitioning, balancing, reordering of higher-order tensors