Sandia National Laboratories

Exceptional service in the national interest

Revolutionary Speedups in SIERRA Structural Dynamics Enhance Mission Impact

SIERRA Structural Dynamics Code Team

Presented by: Johnathan Vo

October 26, 2022

SAND2022-14882 PE

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

U.S. DEPARTMENT OF ENERGY

NNSA

BLUF: 10x speedups expand analysis capabilities

Speedups were achieved through a combination of:

- **Software**
  - Sierra Structural Dynamics (SD) code developers enabled GPUs to parallelize computations
    - Traditional machines CTS-1 (Commodity Technology Systems 1) use CPUs only
    - GPUs are another level of parallelism
  - Defaults in the code allow the same simulation files to run with or without GPUs
- **Hardware**
  - LLNL's Sierra machine is the world's 5th fastest supercomputer
    - Built for GPU computations
    - The platform is called ATS-2 (Advanced Technology System 2)

10-20x speedups have been recorded for real analysis problems

- Sub-assembly simulation times have been reduced from hours to minutes
- Complex analyses can be run overnight
- **Extremely large and complex analyses are now possible**
- **A step towards faster design cycles**

LLNL's Sierra machine (ATS-2)

CPU: Central Processing Unit
GPU: Graphical Processing Unit

Structural Dynamics: Use Cases

SNL will demonstrate milestone completion with the SIERRA/SD structural dynamics application

- SD has been in active development for 25 years under the ASC program.
- SD is part of the SIERRA Engineering Mechanics code suite for mechanical, fluid, and thermal modeling for design and qualification.
- SD is extensively used for normal environment nuclear deterrence analysis
  - Response of systems to high-energy vibration environments such as reentry or flight
  - Mechanical shock response
  - Fatigue life predictions
  - Component environment specification
  - Multiphysics structural-thermal-fluid coupling

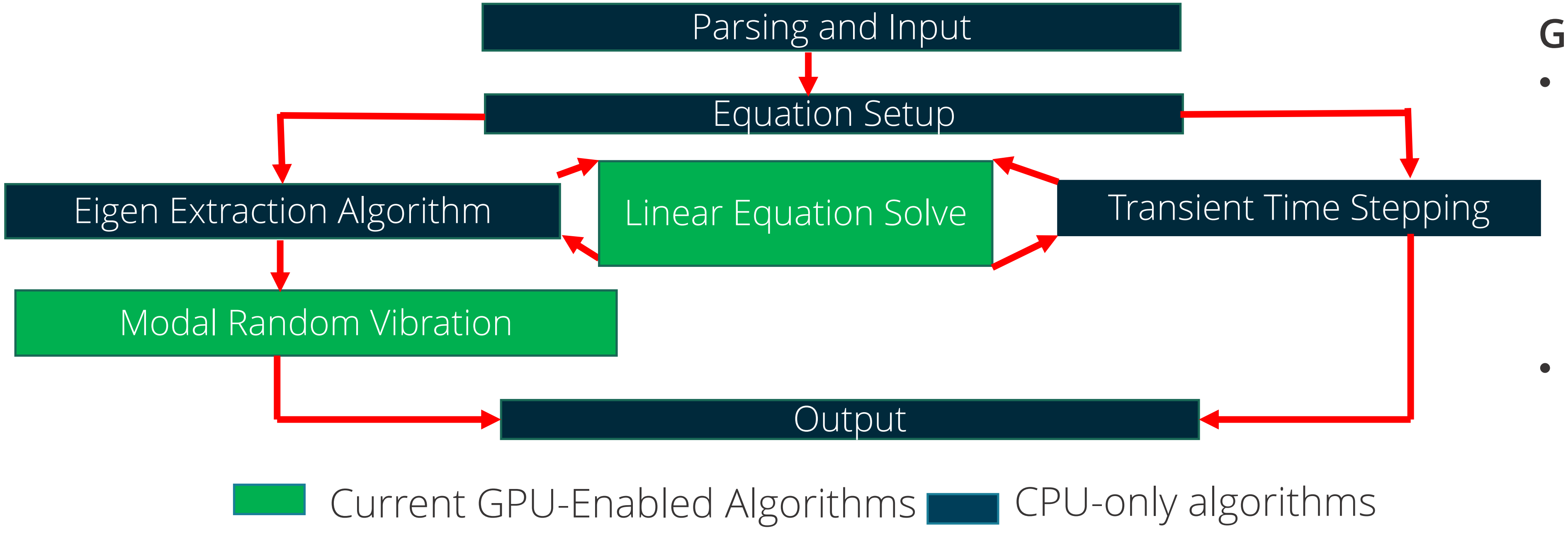
COMPSIM
STRUCTURAL DYNAMICS

We are a massively parallel structural dynamics FEA code used for system-level analysis and design

Structural Dynamics: Mathematics

SD is primarily a linear code. Most use cases require solving the same linear system many times with different right-hand sides. For example:

- Eigen vector/value extraction (up to tens of thousands of modes)
- Linear transient/shock response
- Statistical response to random vibration loads



Goal:

- **Take models analysts are running right now on CTS-1, make them run efficiently on Sierra with no input modification**
- **Focus GPU conversion on the algorithms with long runtime and low code volume**

Processes and Tools: TRILINOS

- SD uses:
  - Tpetra: parallel communication and linear algebra tools
  - STK: mesh database
  - Tacho: GPU-focused linear solver
  - Teuchos: parameters and parsing
  - Kokkos: Performance portability
  - Kokkos Kernels: GPU ready implementation of common algorithms such as linear algebra, graph algorithms, sorting, etc. Wraps cuBLAS.

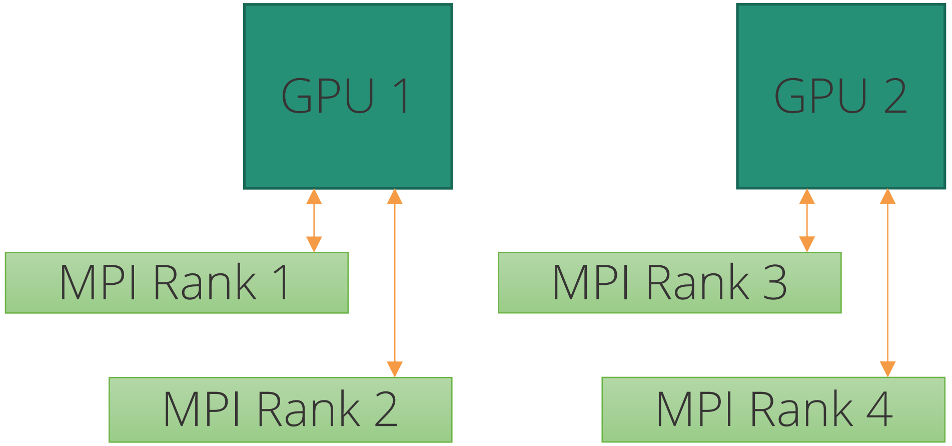
- Trilinos packages implement GPU-ready operations via Kokkos. The GPU-related complexity and maintenance is largely hidden from the SD application
- Most SD performance-critical operations are built on Trilinos objects
- The Tacho solver is especially key for GPU performance

Processes and Tools: Runtime

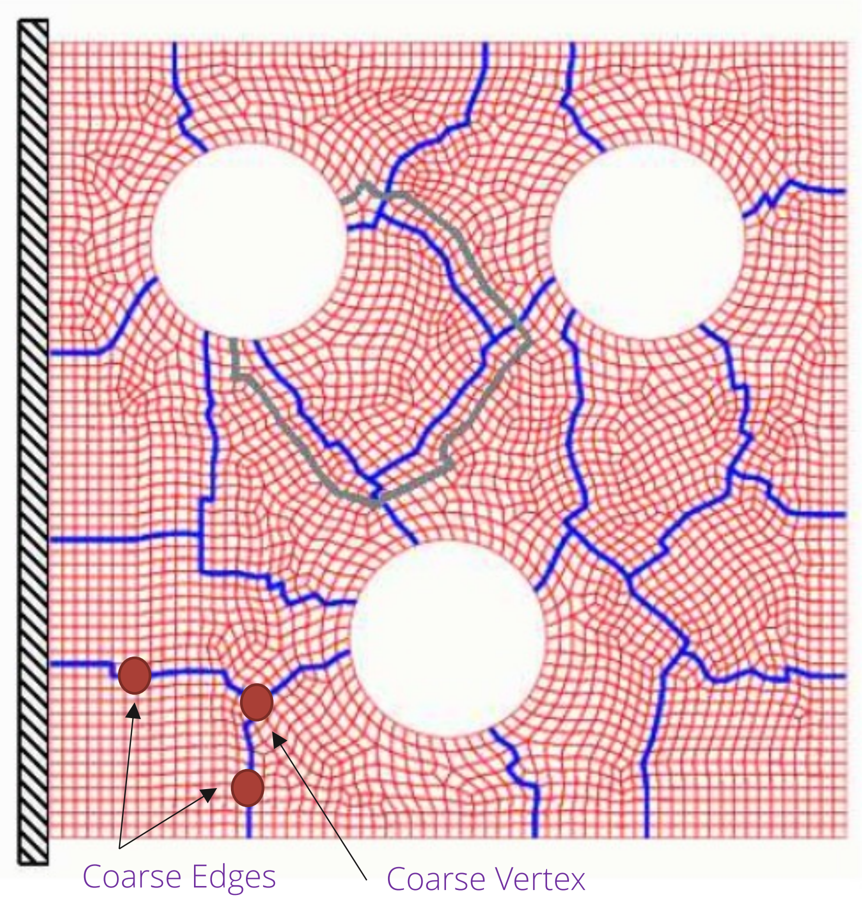
Hybrid Execution:

- Flexible MPI Parallelism. Allow between 1 and 10 MPI ranks to share each GPU (5 often works best)
  - Use available CPU resources in algorithms not yet converted to GPU (forming of matrices, load application, postprocessing)
  - Enable optimal subdomain sizing
  - Multi-process-service (MPS) allows concurrent execution and is key to hide latency and use full throughput of GPU (~3X overall speed improvement with MPS on)

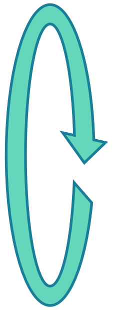
- Challenges
  - Each rank independently loads GPU drivers+executables (~900 Mb) and this consumes GPU memory
  - Balancing GPU vs. CPU has been major focus of GPU optimization and usage guidelines



A Brief Overview of Domain Decomposition Preconditioners



1) Mesh is decomposed into N subdomains. N always equals the number of MPI ranks being used
2) A matrix linear combination of stiffness, mass, damping is set up for each subdomain and factored
3) Many subsequent linear solution steps are performed:
   a) Previously-saved search directions are used to provide a good initial guess of the next solution and provide a high-power preconditioner (orthogonalization)
   b) Iterative domain decomposition solve:
      - Each subdomain is solved independently
      - Single coarse problem is solved involving unknowns at subdomain interfaces (faces, edges and vertices)
      - Continue Krylov solver iterations until acceptable residual tolerance reached

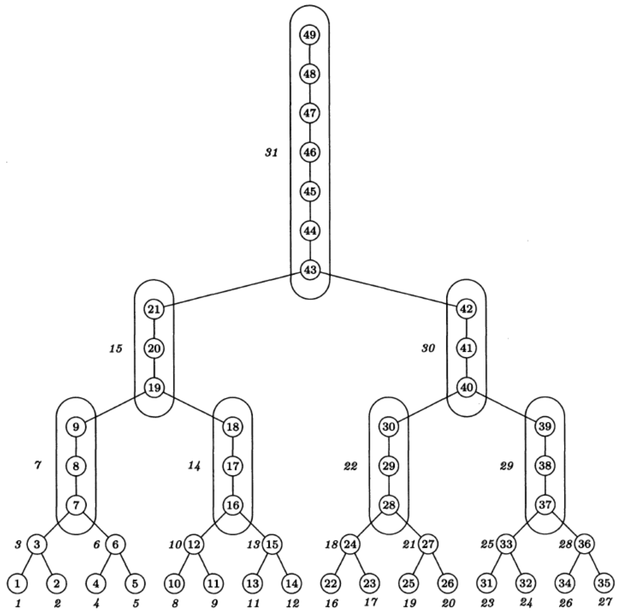


Clark R. Dohrmann and Olof B. Widlund, "An overlapping Schwarz algorithm for almost incompressible elasticity," *SIAM Journal on Numerical Analysis*, 47(4), 2897-2923 (2009).

Clark R. Dohrmann and Olof B. Widlund, "Hybrid domain decomposition algorithms for compressible and almost incompressible elasticity," *International Journal for Numerical Methods in Engineering*, 82, 157-183 (2010).

GPU use in Solver Kernels

- Sparse-direct linear solvers (Tacho for GPU)
  - Each MPI process requires linear solver for two different subdomain problems
  - Coarse problem (solved once for all domains) requires linear solve
  - Focus on speeding up the "solve" phase
    - Initialization costs amortized over several solves since matrices remain the same
  - Level-scheduling algorithm used for on-node parallelism
    - Matrix columns grouped into supernodes, which are then partitioned into different levels
    - Computational work at each level can be done concurrently
    - Kokkos-kernels provides performance portability with Cuda backend and cuBLAS wrappers



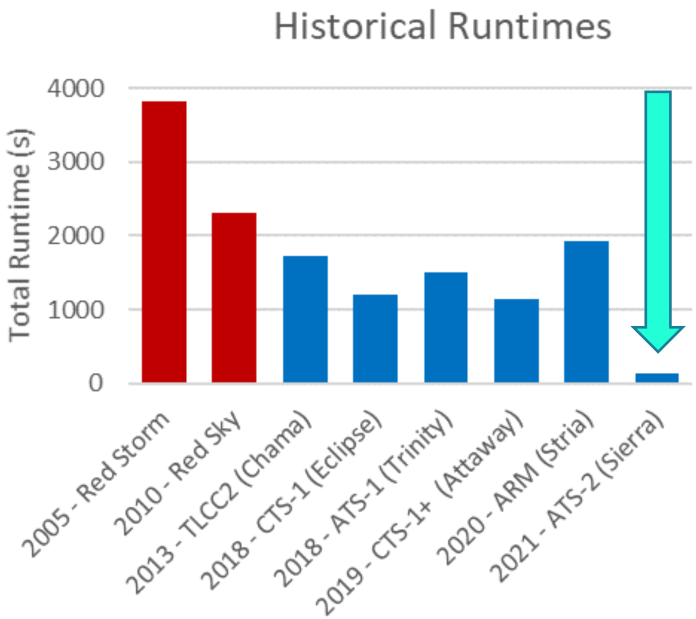
Supernodes and elimination tree for toy problem

- Orthogonalization computations
  - Involves dense matrix-vector products (transpose and non-transpose with a tall and skinny dense matrix)
  - Computations done very fast on GPU using Kokkos-kernels wrapper to cuBLAS
  - Storing Krylov search spaces for subsequent solves can reduce iterations significantly

Speedups

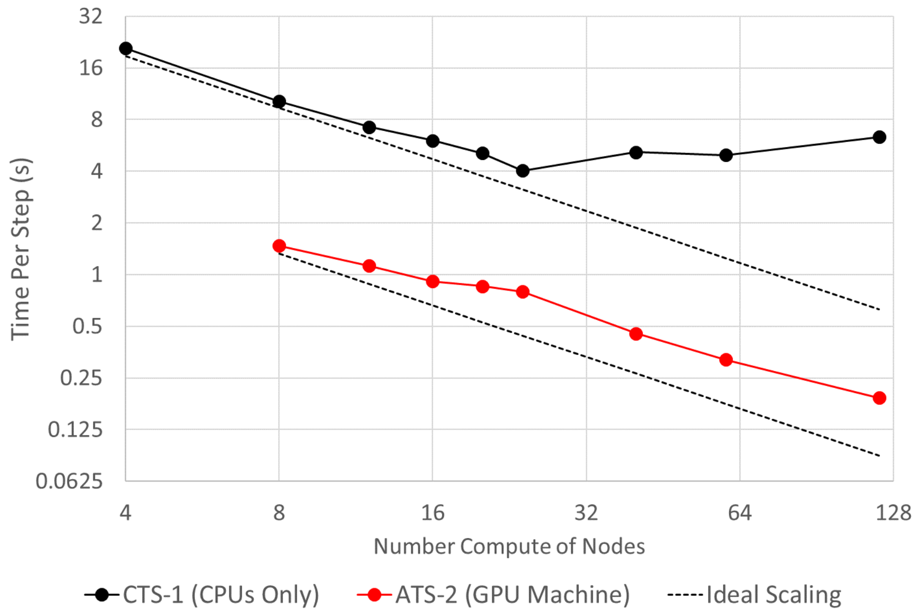
GPU Based Machines Yield a Revolution in Runtime Reduction

Acceptance Test Model



The multiyear GPU development yielded dramatic runtime reduction

Transient Dynamics Model

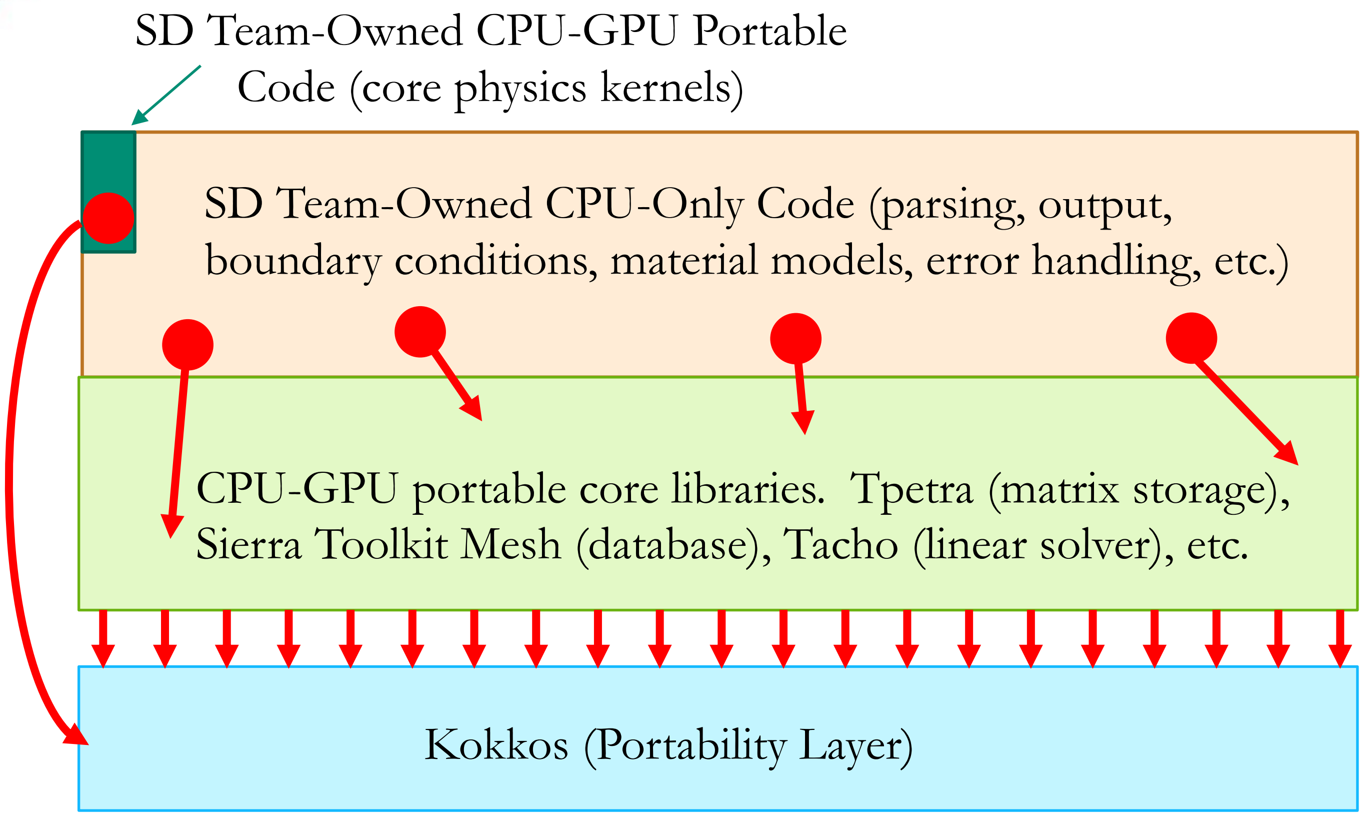


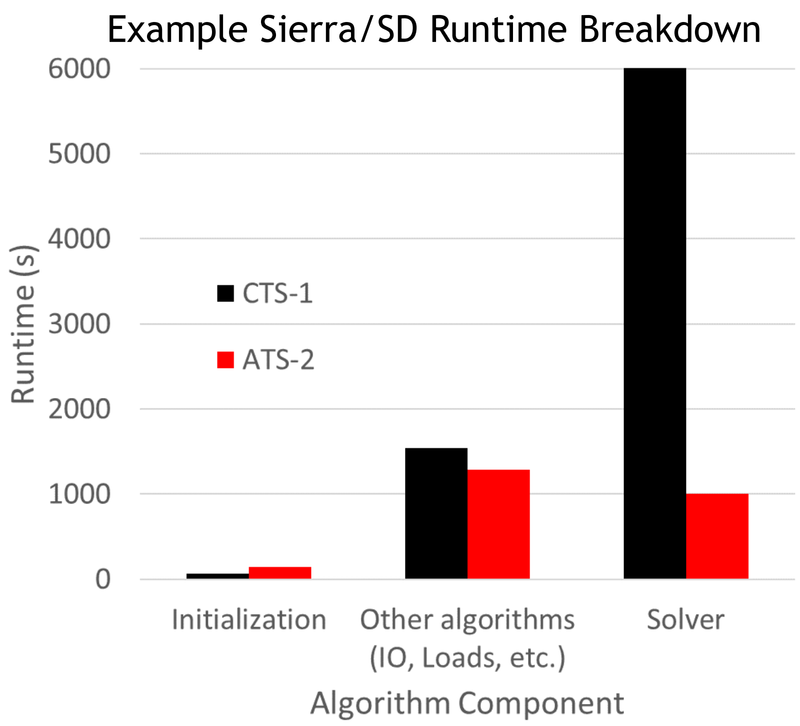
Faster, more scalable, higher throughput!

Acceptance test-driven development led to algorithmic optimizations that also produce speedups

- Make better use of memory
- Benefit analysts on traditional and GPU platforms
- Speedups compare between runs with the algorithmic optimizations

Sustainable Component Architecture





SD achieved success by

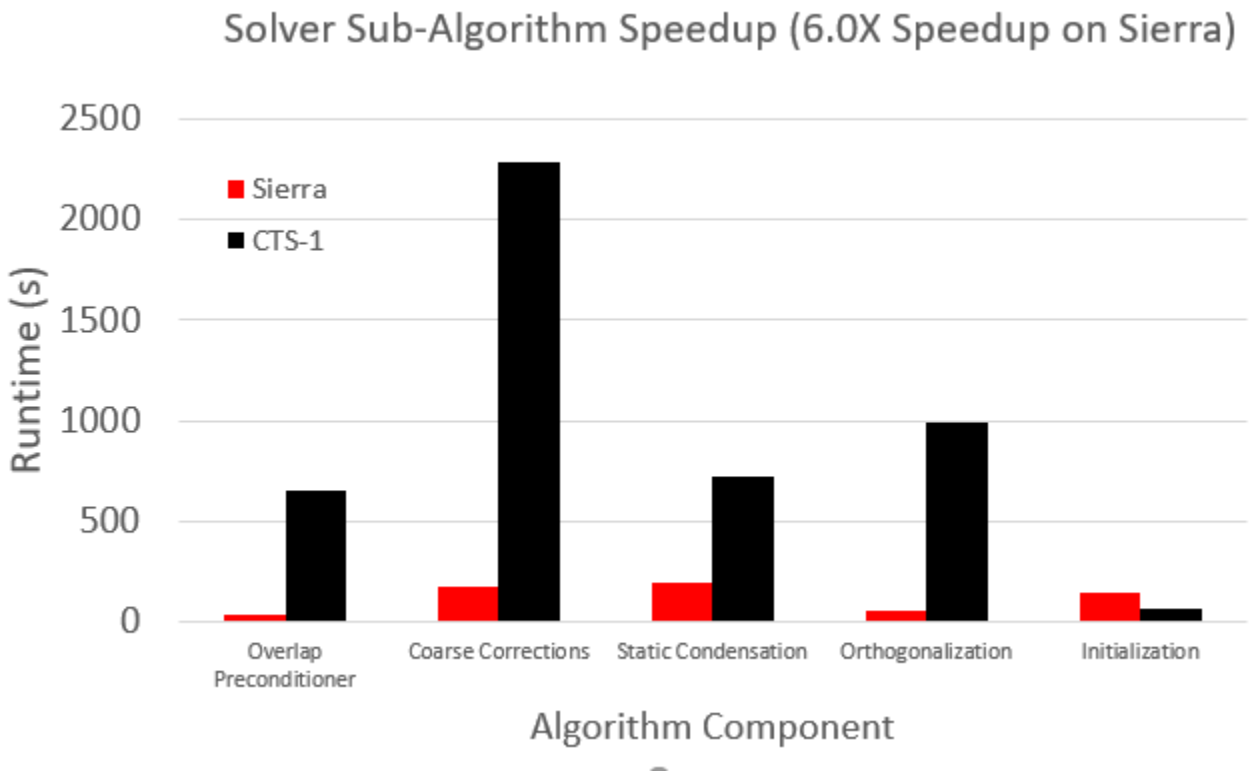
- leveraging many development efforts across Sandia
- focusing on the tall performance poles
- focusing on the big picture

Moving forward:

Expand GPU support to more algorithms

Migrate to next generation machines (ATS-4)

Deeper Dive on Solver



Overlap Preconditioner: Per-subdomain solve on overlapped region

Coarse Corrections: Global solve for coarse problem (plus restriction and prolongation)

Static Condensation: Per-subdomain solve to eliminate subdomain interior residuals

Orthogonalization: Use of previously-saved solutions to predict next solution and form a high-power preconditioner

Initialization: One-time cost to generate and factor matrix

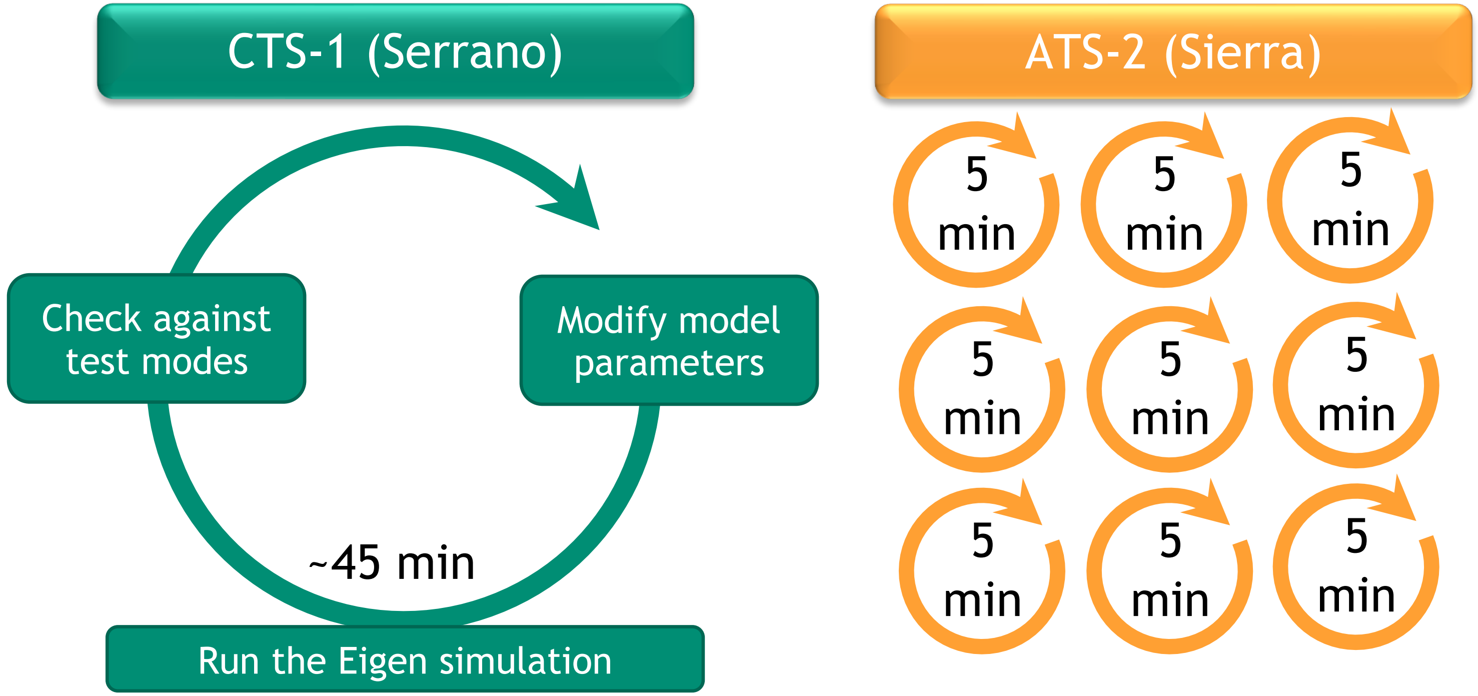
- All solver per-timestep operations are on GPU and show good speedup
- Most solver initialization steps are currently done on CPU. An additional step is needed for GPU for level scheduling. One time initialization cost can be a bottleneck for analyses such as statics where only a single solve is done.

Impact

Improving Existing Analyses: Component Model Updating

Component assembly
- 1.6 million nodes
- 100 modes

Update the model to match measured test modes:
- Compare frequencies and mode shapes between test and simulation
- Make incremental changes until the simulation responds similarly to the measured response



- Contact definition
- Joint properties
- Geometry simplifications
- Mass and stiffness

Impact:
- Closer to real-time feedback about parameter changes
- Faster results provided more freedom to “see what happens”

Speedup: 45 min to 5 min
Program: ND

High-Fidelity Experimental Test Support

Experimental design support

- Impedance-matched multi-axis testing (IMMAT)
  - Better replicate reentry random vibration
  - Complex test setup

- High fidelity system models were used to inform test design
  - **~ 1 week from request to results**, including a simulation setup modification
  - Simulation **speedups from ~20 hours to ~3 hours** runtime

New Capability

- Previously unobtainable turnaround times to support experimental test design

High Frequency Margin Assessment Support

Sub-assembly model with soft components

- Compute modes to support environments margin assessment
- High modal density from soft components
- Required ~1500 modes for the frequency range of interest
  - For reference, generally compute 100-200 modes

Expanded frequency range enabled

- Computed **1500 modes in 1.5 hours with no restarts**
  - Traditional runs on the CPU machines would require restarts to get around runtime limits
  - **Not feasible to run to the required frequency range on CPU machines**

New Capability

- More computational power means complex models can be run to frequency ranges that weren't previously feasible

Routine "Heroic" Simulations

Heroic: Long, complex, high fidelity simulations that are rarely run

High fidelity full system model

- Sub-component analysis raised questions about the full-system response
- Access to ATS-2 enabled an overnight run of the high fidelity, full system dynamics
  - **Simulation ran in 10 hours on ATS-2**
  - With queue times and restarts, **runs could take weeks**
- Provided better information to customers in a timely manner
  - More informed decision-making

New Capability

- High-fidelity system dynamic runs produce more conclusive results overnight

Closing Comments

Closing Comments

10x speedups with Sierra SD has enabled simulations that weren’t previously possible

The impact shown was possible because of the Sierra SD team:

- Prioritized user experience so that no changes were required to run on GPUs
- From an analyst perspective, choose a different machine and get results 10x faster
- Ease of use led many analysts to become early adopters
- Impact will continue to grow

Questions?