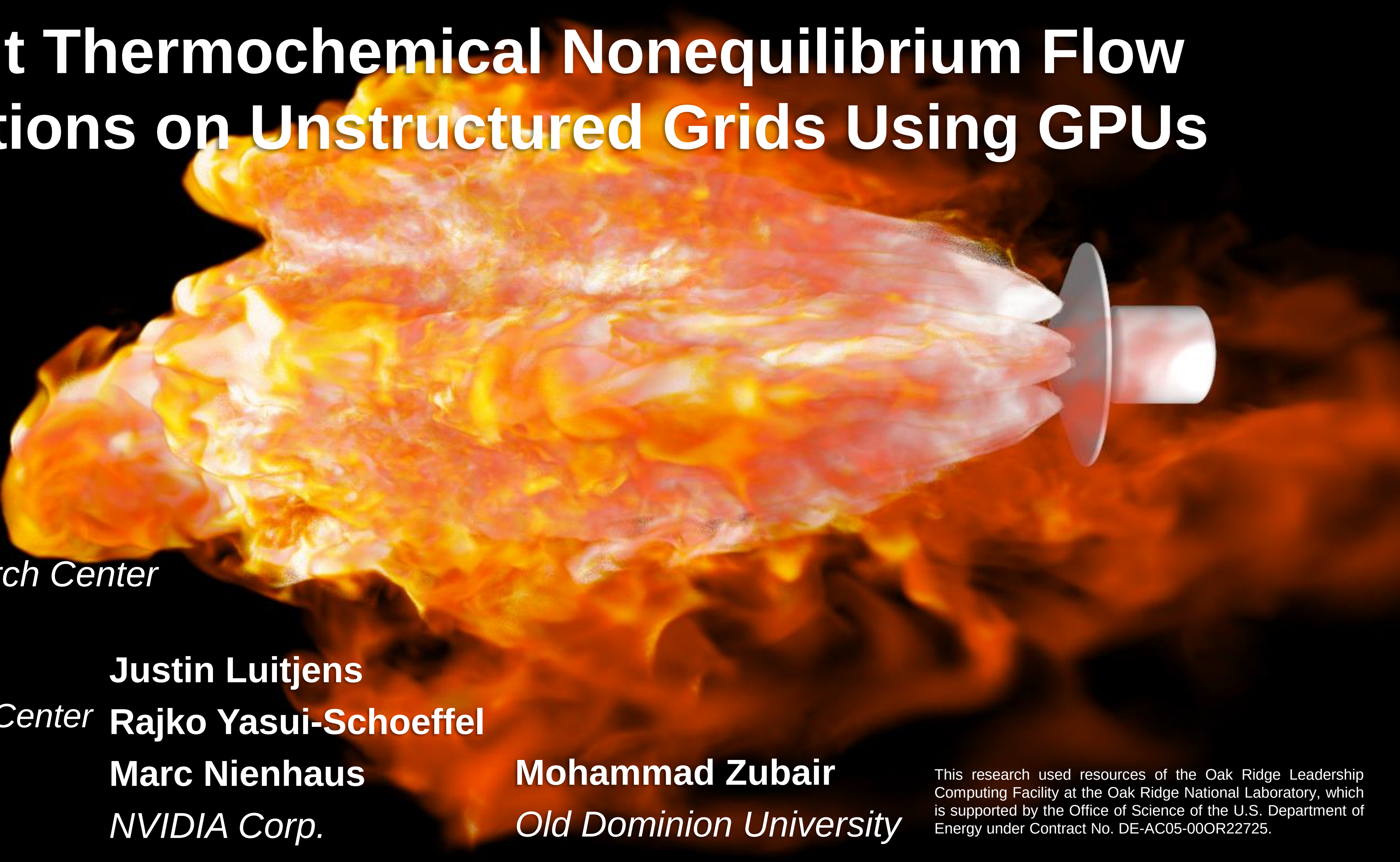
Implicit Thermochemical Nonequilibrium Flow Simulations on Unstructured Grids Using GPUs



Gabriel Nastac

Aaron Walden

Eric Nielsen

Ashley Korzun

Bill Jones

Li Wang

NASA Langley Research Center

Patrick Moran

NASA Ames Research Center

Justin Luitjens

Rajko Yasui-Schoeffel

Marc Nienhaus

NVIDIA Corp.

Paul Kolano

Inu Teq, LLC

Mohammad Zubair

Old Dominion University

This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

Motivation

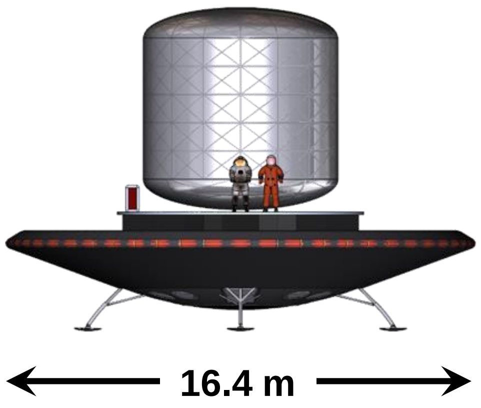
- Current plans for U.S. exascale systems rely on GPU acceleration
  - 130 out of 500 fastest supercomputers (6 out of top 10) utilize GPU hardware
- Port to GPU architectures positions FUN3D, an unstructured-grid CFD solver, for this emerging landscape
  - Dramatically reduced run times enable early penetration of high-fidelity modeling
  - Ability to elucidate unprecedented physics – temporal, spatial, physical models
- The perfect gas path of FUN3D has been previously ported to NVIDIA Tesla GPUs
- Here we port the generic gas path of FUN3D, which models thermochemical nonequilibrium flows including atmospheric entry, hypersonics, and combustion
- Initial Mars atmospheric entry retropropulsion simulations will also be presented

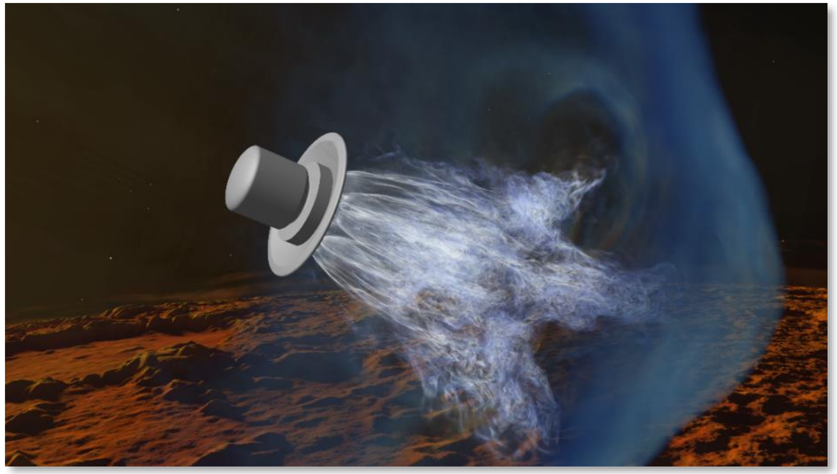
Current HPC Landscape
2. ORNL Summit (149 PF)
46. NASA Pleiades (6 PF)
53. NASA Electra (5 PF)
71. NASA Aitken 2 (4 PF)
168. NASA Aitken (2 PF)

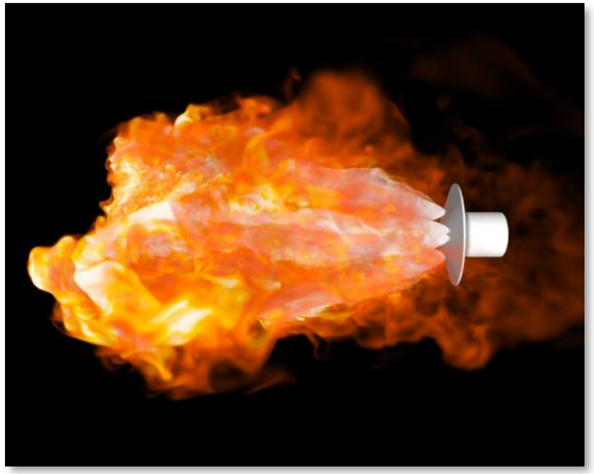
New US Systems in 2021-2023
ANL Aurora (1000 PF)
ORNL Frontier (1500 PF)
LLNL El Capitan (2000 PF)

Architecture: CPU / GPU

PF: PetaFLOPS, or 10^{15} Floating-Point Operations Per Second







Governing Equations and Numerical Implementation

- Conservation of species, momentum, energies, and turbulence variables
- Two-temperature model available for thermal nonequilibrium
- Spalart-Allmaras turbulence model with Catris-Aupoix compressibility correction; DES option
- Variable species, energies, and turbulence equations
- Node-based finite-volume approach on general unstructured grids
- Fully implicit formulations are used to integrate the equations in time
  - Sparse block linear system: $A\boldsymbol{x} = \boldsymbol{b}$
  - Matrix A composed of diagonal and off-diagonal $N_{eq} \times N_{eq}$ blocks
  - Memory and solution time increases as $O\left(N_{eq}^2\right)$
- System solved with multicolor point-implicit approach

$$\frac{\partial}{\partial t}(\rho y_s) + \frac{\partial}{\partial x_j}(\rho y_s u_j) - \frac{\partial}{\partial x_j}(J_{sj}) = \dot{\omega}_s$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_i u_j + p\delta_{ij}) - \frac{\partial}{\partial x_j}(\tau_{ij}) = 0$$

$$\frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_j}\left((\rho E + p)u_j\right) - \frac{\partial}{\partial x_j}\left(u_k\tau_{kj} + \dot{q}_j + \sum_{s=1}^{N_s} h_s J_{sj}\right) = 0$$

$$\frac{\partial}{\partial t}(\rho E_v) + \frac{\partial}{\partial x_j}(\rho E_v u_j) - \frac{\partial}{\partial x_j}\left(\dot{q_V}_j + \sum_{s=1}^{N_s} h_{V\,s} J_{sj}\right) = S_v$$

$$\frac{\partial}{\partial t}(\rho\tilde{\nu}) + \frac{\partial}{\partial x_j}(\rho\tilde{\nu}u_j) - \frac{\partial}{\partial x_j}\left(\frac{1}{\sigma}\left(\mu\frac{\partial\tilde{\nu}}{\partial x_j} + \sqrt{\rho}\tilde{\nu}\frac{\partial\sqrt{\rho}\tilde{\nu}}{\partial x_j}\right)\right) = S_{\tilde{\nu}}$$

$$\boldsymbol{q} = [\rho\vec{y}_s, \rho\vec{u}, \rho E, \rho E_v, \rho\tilde{\nu}]^T$$

$$\int_V \frac{\partial \boldsymbol{q}}{\partial t} dV + \oint_S (\boldsymbol{F}\cdot\boldsymbol{n})dS - \int_V \boldsymbol{S}dV = \boldsymbol{0}$$

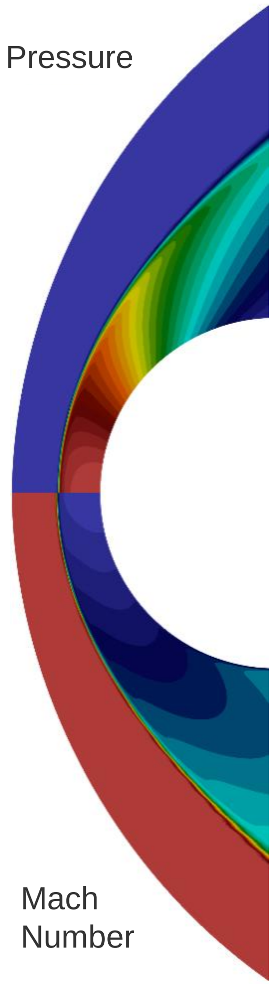
$$\left[\frac{V}{\Delta\tau}\boldsymbol{I} + \frac{V}{\Delta t}\boldsymbol{I} + \frac{\partial\widehat{\boldsymbol{R}}}{\partial\boldsymbol{q}}\right]\Delta\mathbf{q} = -\mathbf{R}(\mathbf{q}^{n+1,m}) - \frac{V}{\Delta t}(\mathbf{q}^{n+1,m} - \mathbf{q}^{n})$$

$$\mathbf{q}^{n+1,m} = \mathbf{q}^{n+1,m} + \Delta\boldsymbol{q}$$

GPU Design Approach

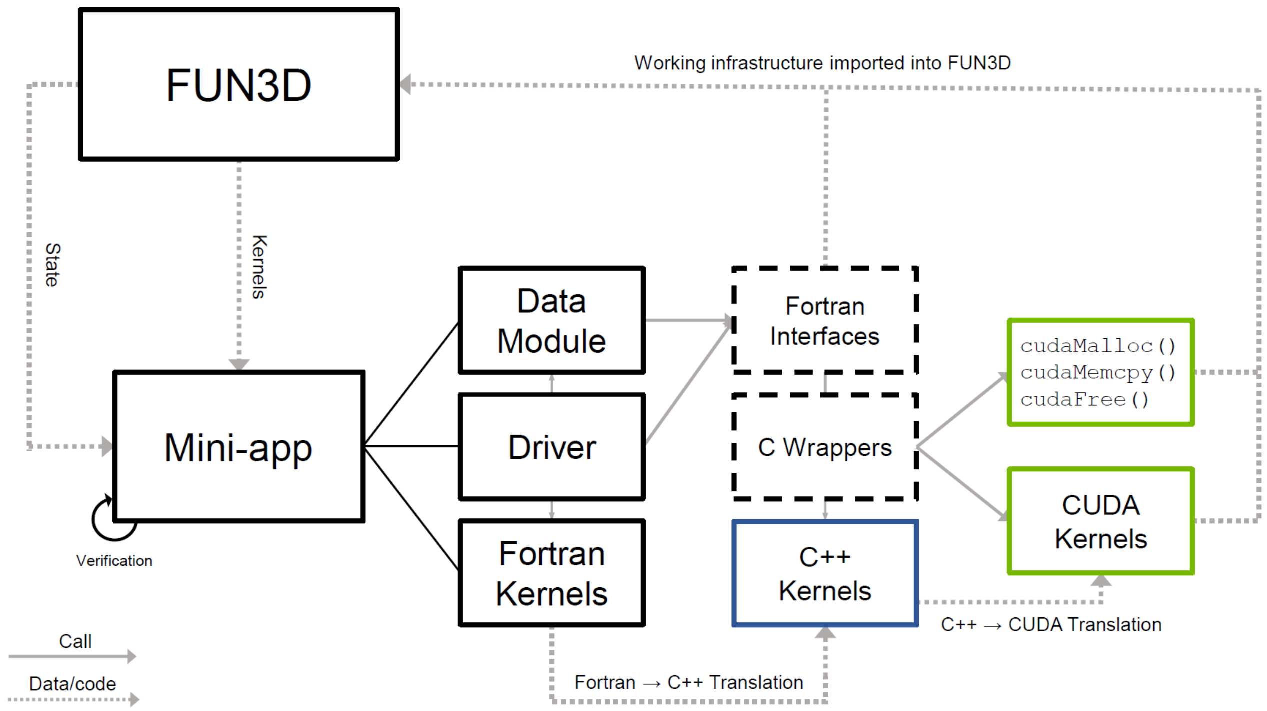
- Nomenclature: Host = CPU, Device = GPU
- FLUDA Library
  - CUDA C++ port of compute kernels in FUN3D
  - No external libraries are required
  - Use of library in FUN3D is controlled by a run-time parameter
- Preprocessing routines remain on the host
- All PDE kernels (~100) performed on the device
- Minimal data transfer between host/device (mainly scalars)
  - Large data motion only at user-specified frequencies (e.g., restarts, visualization support)
- Data structures are identical between CUDA and Fortran contexts
  - Column-major order array layouts
  - GPU “mirror” data structures that match CPU data structures
  - Variable precision is identical to CPU approach
    - FP64 for most variables, with mixed FP32/FP64 for linear algebra

Hypersonic Cylinder
$M_\infty = 8.7$, 5-species air



GPU Implementation

- Mini-app utilizes an entire state of FUN3D to perform a full iteration of the solve
- CPU and GPU kernels can be run at the same time and have outputs compared
- Once RMS norm of outputs is within specified tolerance (10^{-14} for FP64, 10^{-7} for FP32), kernels are integrated into FUN3D
    - Most kernels match to machine precision
    - Individual FP values generally do not match to machine precision due to order-of-operations; further complicated by asynchronous execution
    - Behavior not unique to GPUs; also observed on CPUs with random loop permutations



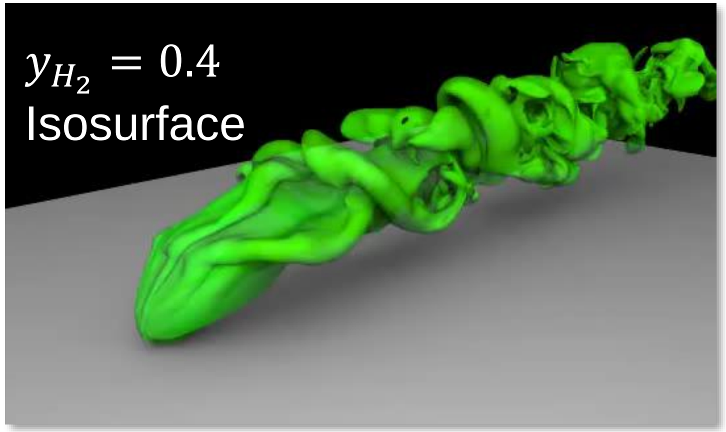
FUN3D mini-app structure and porting workflow

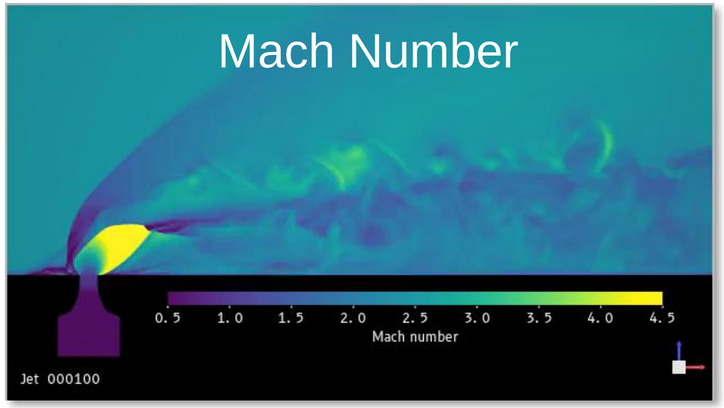
GPU Optimizations

- Reduction of kernel state
  - Fortran implementation utilizes variable-length arrays (VLAs) for workspace
    - Since VLAs do not exist for CUDA, templating is extensively used
  - Initial naive CUDA port resulted in stack frames so large that the GPU ran out of memory immediately
  - To remedy this, multiple threads are assigned to a work item (such as a Jacobian) which reduces 2D arrays to scalars in many cases
  - Registers and shared memory are heavily used
- Reduce thread divergence
- Coalesced memory accesses
- Kernel launch parameter optimization
- See paper for more details:

Gabriel Nastac, Aaron Walden, Eric J. Nielsen, and Kader Frendi. "Implicit Thermochemical Nonequilibrium Flow Simulations on Unstructured Grids using GPUs," AIAA 2021-0159. AIAA Scitech 2021 Forum. January 2021.

Transverse Hydrogen Jet in Supersonic Cross Flow: $M_\infty = 2.4$, 9 species, DES

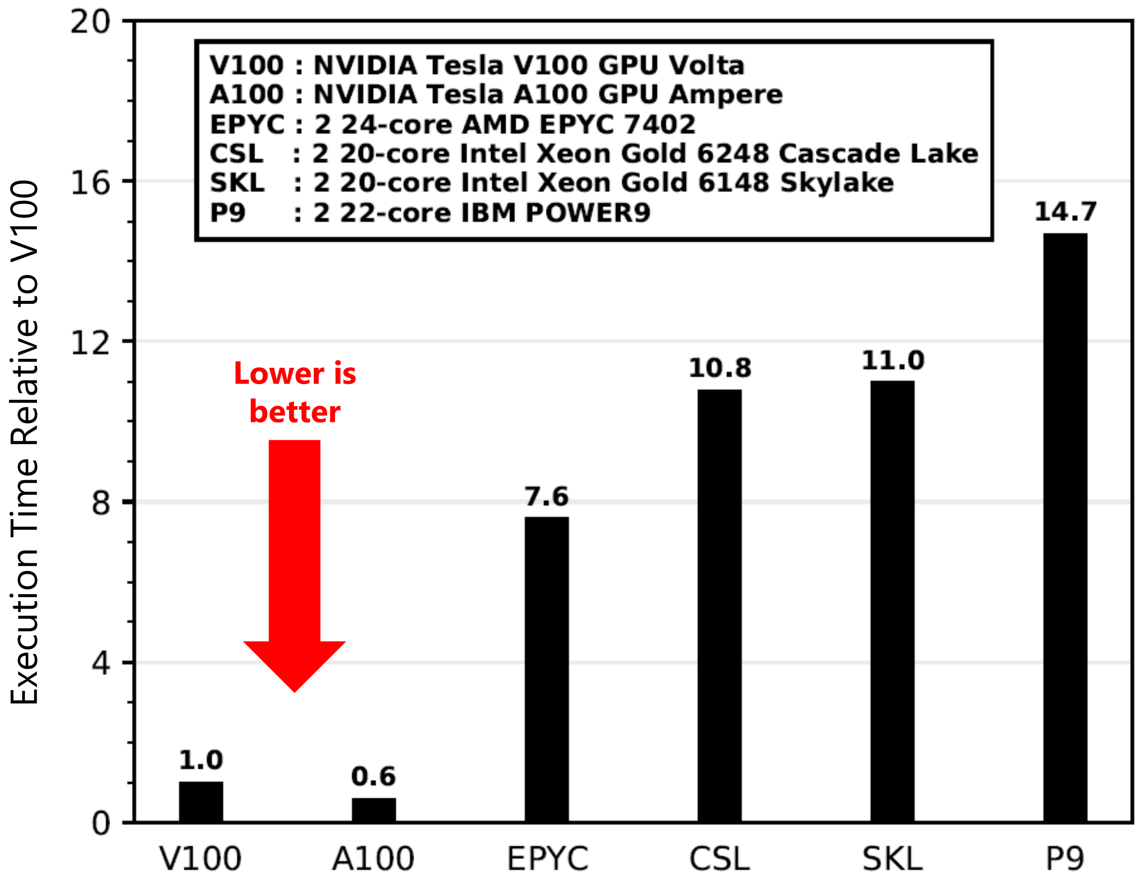




48 V100s ~ 20,000 Skylake Cores
1 V100 ≈ 417 Skylake cores

Device Level Performance

- Memory-bound applications should exhibit speedups commensurate with hardware memory bandwidth ratio
  - E.g., perfect gas FUN3D shows 4.5x speedup for NVIDIA Tesla V100 over dual-socket Intel Xeon
- Generic gas CPU implementation is not optimal
  - Templates are not natively available in Fortran
  - Optimizations (e.g., reduction of workspace, transpose) have not been performed on CPU

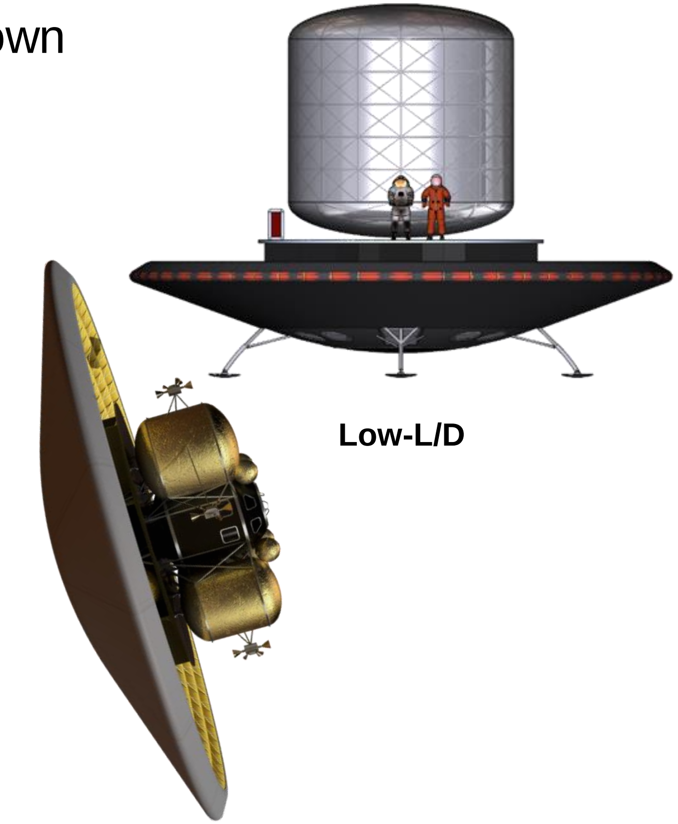


Retropropulsion for Human Mars Exploration

Human-scale Mars landers require new approaches to all phases of Entry, Descent, and Landing

- Cannot use heritage, low-L/D rigid capsules → deployable hypersonic decelerators or mid-L/D rigid aeroshells
- Cannot use parachutes → retropropulsion, from supersonic conditions to touchdown
- No viable alternative to an extended, retropropulsive phase of flight

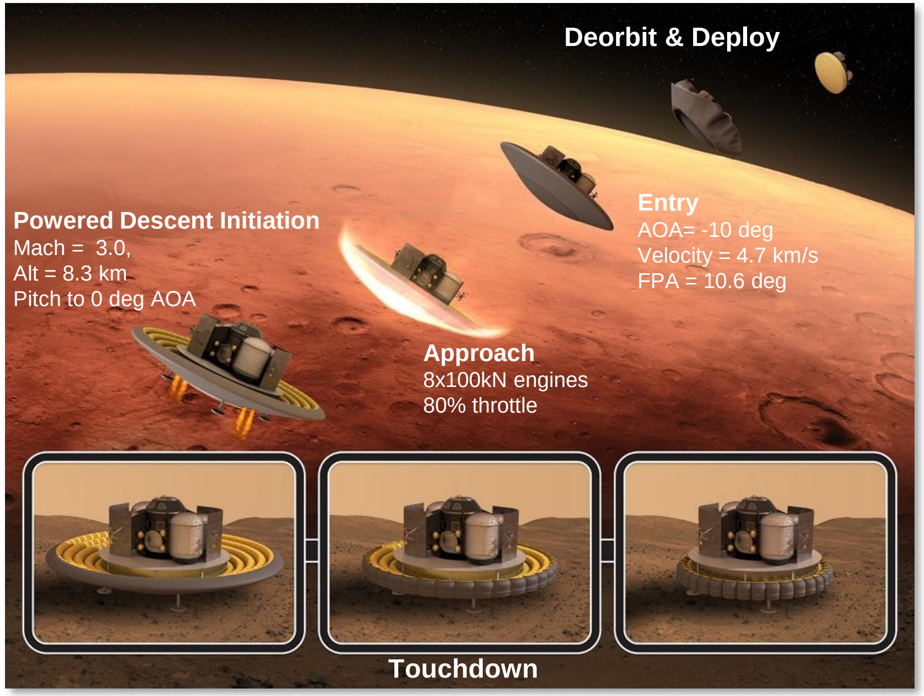
	Viking	Pathfinder	MERs	Phoenix	MSL	InSight	M2020 (2 weeks ago!)	Human-Scale Lander (Projected)
Entry Capsule (to scale)								
Diameter (m)	3.505	2.65	2.65	2.65	4.52	2.65	4.5	16 - 19
Entry Mass (t)	0.930	0.584	0.832	0.573	3.153	0.608	3.440	40 - 65
Parachute Diameter (m)	16.0	12.5	14.0	11.8	19.7	11.8	21.5	N/A
Parachute Deploy (Mach)	1.1	1.57	1.77	1.65	2.2	1.66	1.75	N/A
Landed Mass (t)	0.603	0.360	0.539	0.364	0.899	0.375	1.050	26 - 36
Landing Altitude (km)	-3.5	-2.5	-1.4	-4.1	-4.4	-2.6	-2.5	+/- 2.0
Landing Technology	Retro-propulsion	Airbags	Airbags	Retro-propulsion	Skycrane	Retro-propulsion	Skycrane	Retro-propulsion



Steady progression of "in family" EDL → New EDL Paradigm

Early Science 2018, INCITE 2019/2021 Efforts

"Aero-Propulsive Real Gas Effects for Human-Scale Mars Entry"



Our previous Summit campaigns focused on perfect gas retropropulsion simulations

- Limited experiments on Earth are perfect gas

Our current efforts are exploring effects of reacting gas chemistry on these retropropulsion flows across the flight trajectory

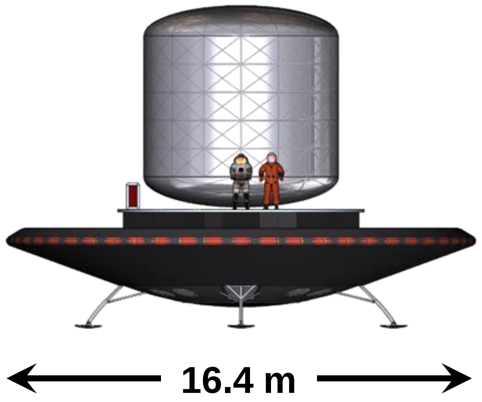
- Methane combustion in Martian CO_2 atmosphere
- ~10x more expensive computationally

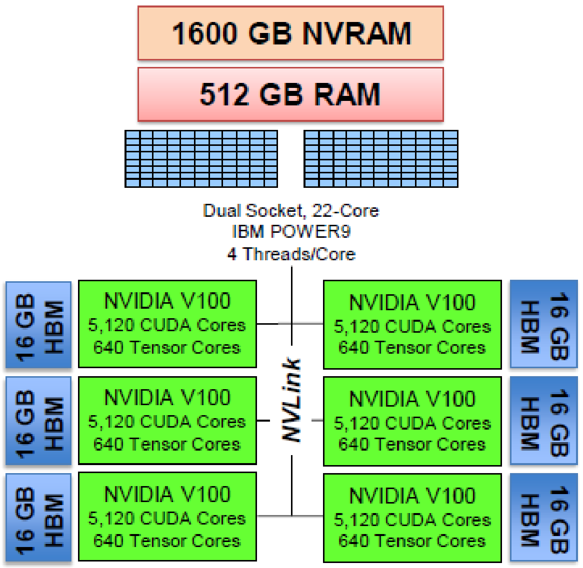
Campaign Goals

- ***Science***: Advance the understanding of retropropulsion flow physics during Mars EDL of a human-scale lander
- ***Computational***: Demonstrate production readiness and efficiency advantages of GPU implementation of the FUN3D CFD code at scale

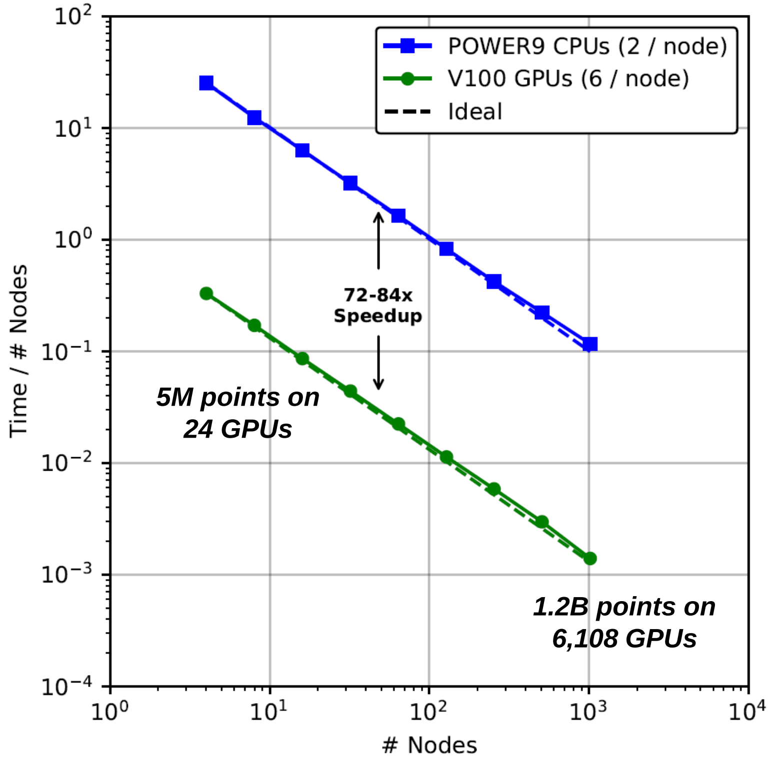
Performance at Scale

- Weak scaling evaluated on Oak Ridge Summit system
- Node consists of 2 22-core IBM POWER9 CPUs, 6 NVIDIA Tesla V100 GPUs
- Each run places 1.2M grid points/node, or 200K grid points/GPU
- CPU- and GPU-only executions scale linearly
- GPUs retain ~75x node-level speedup at scale
- 1.2 billion points on 1,018 nodes
  - One physical time step with BDF2 takes about one second
  - Performance equivalent to several million CPU cores



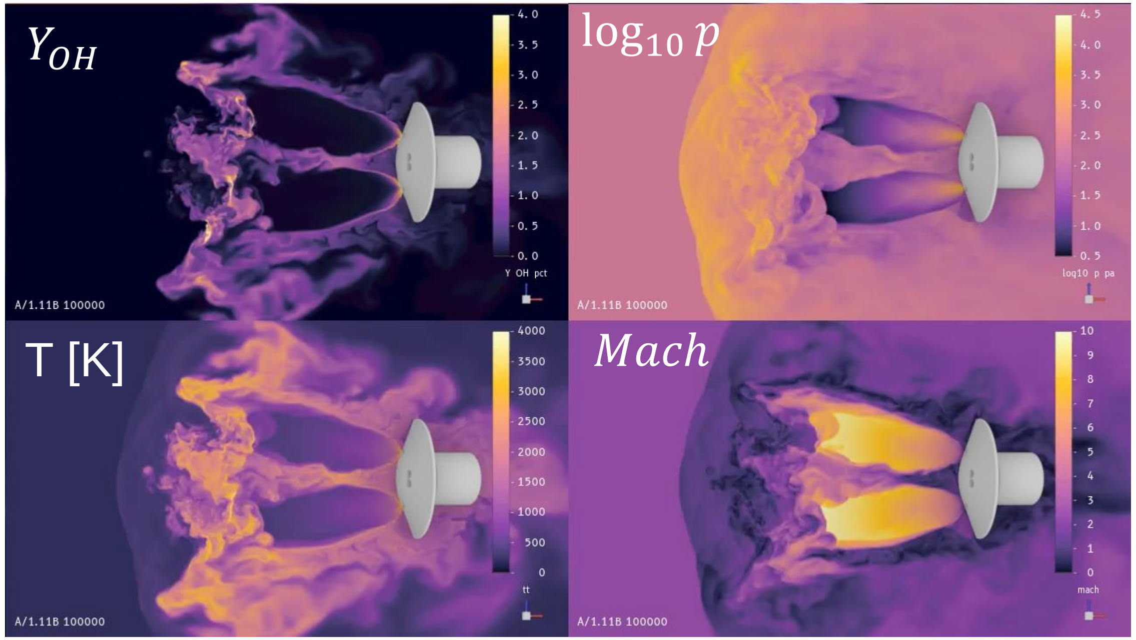


Summit Node



Langley Research Center

Summit Simulation Overview

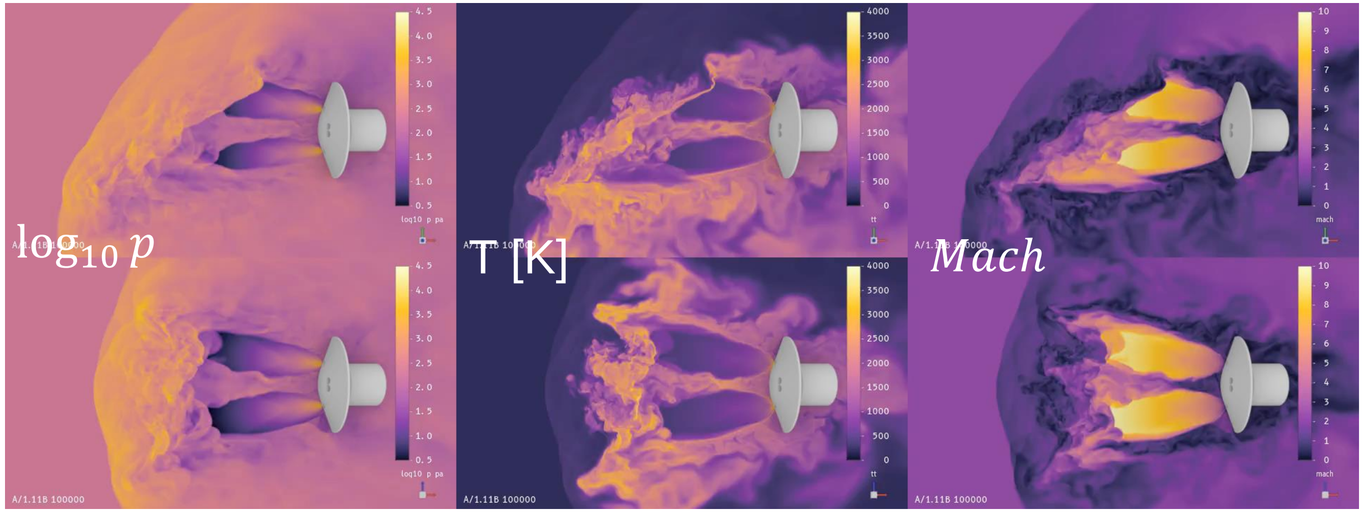


- $M_\infty = 2.4$, Martian atmosphere freestream
- Eight Engine Plena:
  - Products of methane combustion
  - $T_0 \sim 3600\ K, p_0 \sim 80\ bar$
- 10-species SA-Catris DES
- 6B cells, 1.1B nodes (~94% tetrahedra)
- 300k timesteps (5 subiterations per timestep)
  - Total integration time of ~2.5 seconds real-time
- 922 Nodes of Summit (5532 NVIDIA V100 GPUs)
  - 1.2~ seconds per timestep nominally
  - Estimated about 2.3 million Xeon Skylake cores to match performance
- Data Output
  - Saving 20 variables every 50 timesteps, 90+ GB per minute for the entire simulation of 100~ hours
  - Asynchronous I/O, less than 1% overhead (effectively free)
  - About 600 TB of data generated, data migration is nontrivial at this scale (50 TB/day to NASA from DOE)
  - Another goal of the campaign is to demonstrate in situ visualization with NVIDIA at this scale for unstructured grids

Effect of Generic Gas on Retropropulsion

Preliminary: Plume Asymmetry Investigation

Mach 2.4, Generic Gas, Z-Plane



Mach 2.4, Generic Gas, Y-Plane

Preliminary

Parameter	Generic Gas / Perfect Gas[1]
Axial Force Mean	1.7
Axial Force Fluctuation	1.5
Asymmetric Moment Mean	0.7
Asymmetric Moment Fluctuation	1.6

[1]Ashley M. Korzun, Eric Nielsen, Aaron Walden, William Jones, Jan-Reneé Carlson, Patrick Moran, Christopher Henze, and Timothy Sandstrom. "Computational Investigation of Retropropulsion Operating Environments with a Massively Parallel Detached Eddy Simulation Approach," AIAA 2020-4228. ASCEND 2020. November 2020.

Summary and Future Work

- Generic gas path of FUN3D has been successfully ported, optimized, and verified for NVIDIA Tesla GPUs
- One NVIDIA Tesla V100 equivalent to ~400 Intel Xeon Skylake cores
- Benchmarks have been performed using over 6,000 GPUs with grids containing several billion elements
    - Performance equivalent to several million CPU cores
- Retropropulsion reacting flow simulations on grids of several billion elements are being performed at scale on Summit
    - Performance is maintained at scale with asynchronous I/O of full volume every minute
    - 90+ GB/min for 100 hours

- Other Summit campaign simulations at other points in descent trajectory
- Algorithmic improvements to solver to increase performance and lower memory requirements
- Adaptation of code base to other architectures used in future exascale machines