AceCAST GPU-Accelerated WRF Model Overview

TempoQuest Inc.

AceCAST - Improving Modeling Capabilities

- Computational expense is a major limitation for all WRF users
- Advances in computing can enable better modeling through
  - Improved resolution
  - Larger spatial and temporal extents
  - More ensemble members
  - More complex physics
- These benefits are only realized if WRF can run on modern HPC architectures
- Limitation - WRF does not support execution on the GPU architectures that make up most of the HPC resources available today

AceCAST is a modified version of WRF developed with this sole purpose of enabling the model to run on modern GPU-based HPC systems

TempoQuest
WEATHER TRANSCENDED

AceCAST – GPU-Accelerated WRF Model

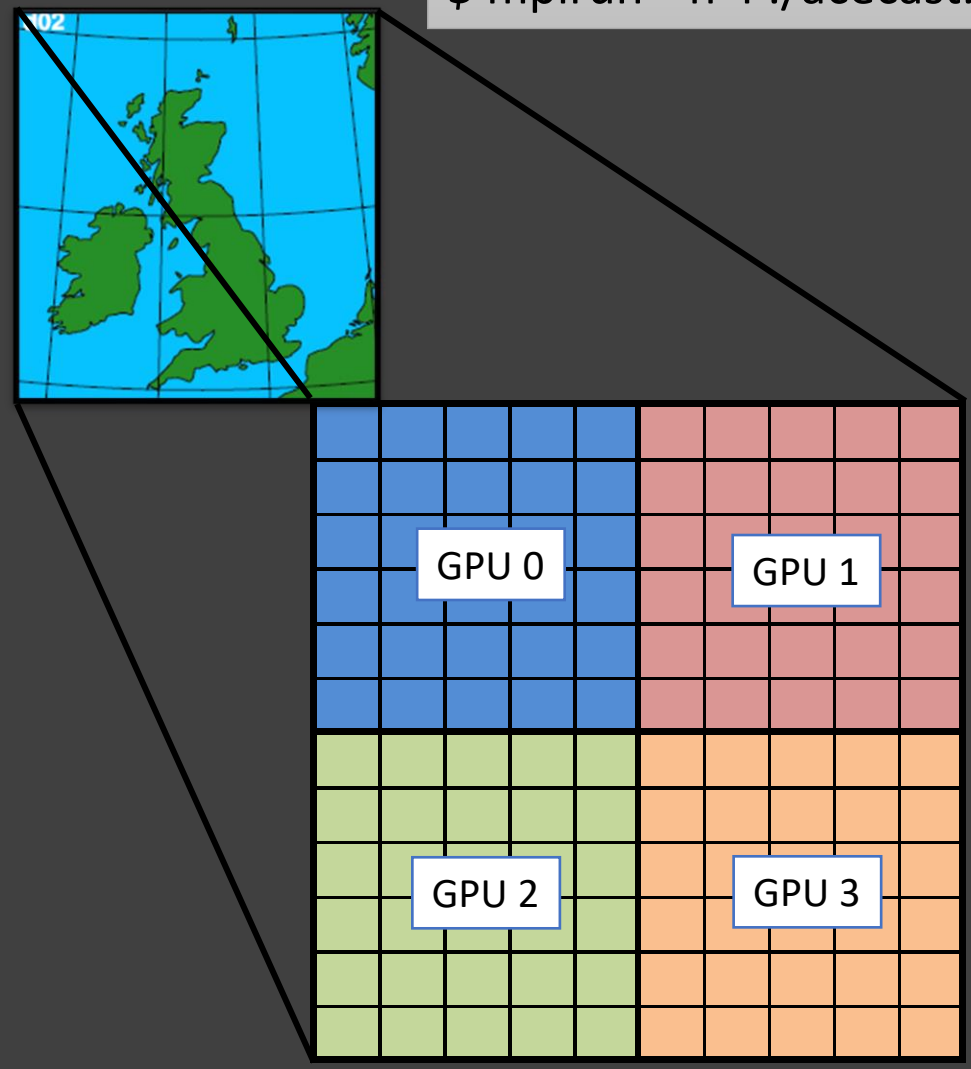
- AceCAST is an OpenACC/CUDA-based implementation of the CPU-based WRF model
  - OpenACC and CUDA are extensions to the C/C++/Fortran languages that enable execution on NVIDIA GPUs
- AceCAST is designed to be a drop-in replacement for CPU-WRF:
  - ***Same input/output files***
    - namelist.input, wrfinput*, wrfbdy*, wrfout*, etc.
  - Provides ***identical results*** to its CPU-counterpart
  - Implements the ***same model*** with highly optimized algorithms for running on GPU

In short, AceCAST does exactly what CPU-WRF does but much faster since it takes advantage of the superior performance of modern GPU architectures

Multi-GPU Execution with MPI

- Both AceCAST and WRF use MPI to enable multi-CPU/multi-GPU execution
- The main domain grid is separated into *patches*
- The data and computations for each *patch* is then assigned to a specific CPU core (WRF) or GPU (AceCAST)

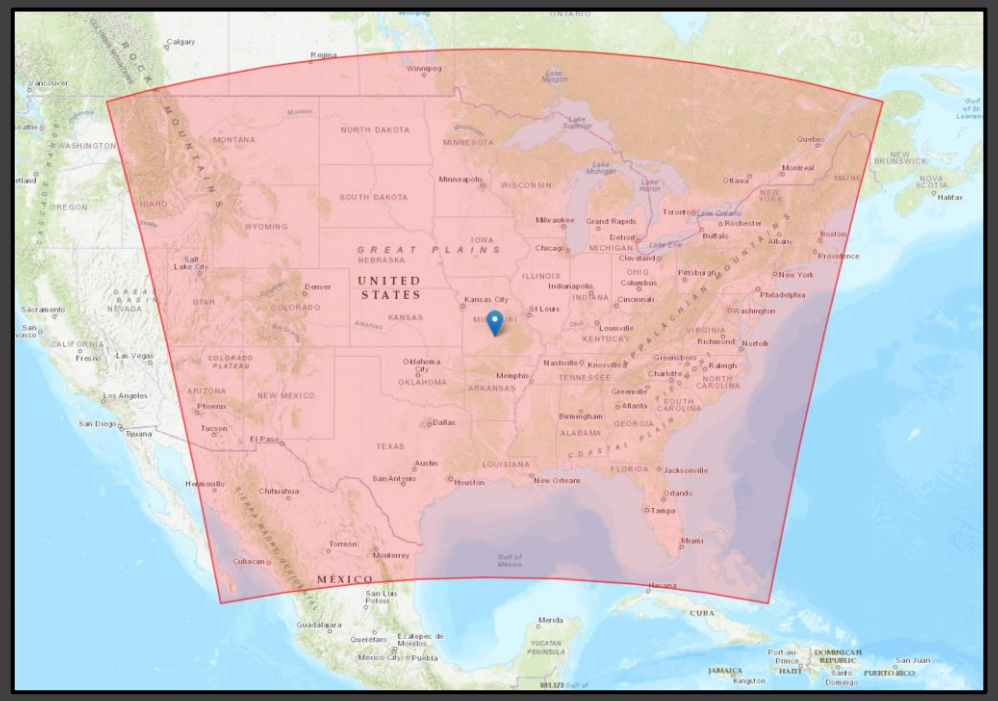
Running AceCAST with 4 GPUs:
$ mpirun --n 4 ./acecast.exe



AceCAST Performance

Benchmark – CONUS 2.5km

- 1501×1201×50 grid
- CONUS physics suite
- 3h simulation



System Specifications

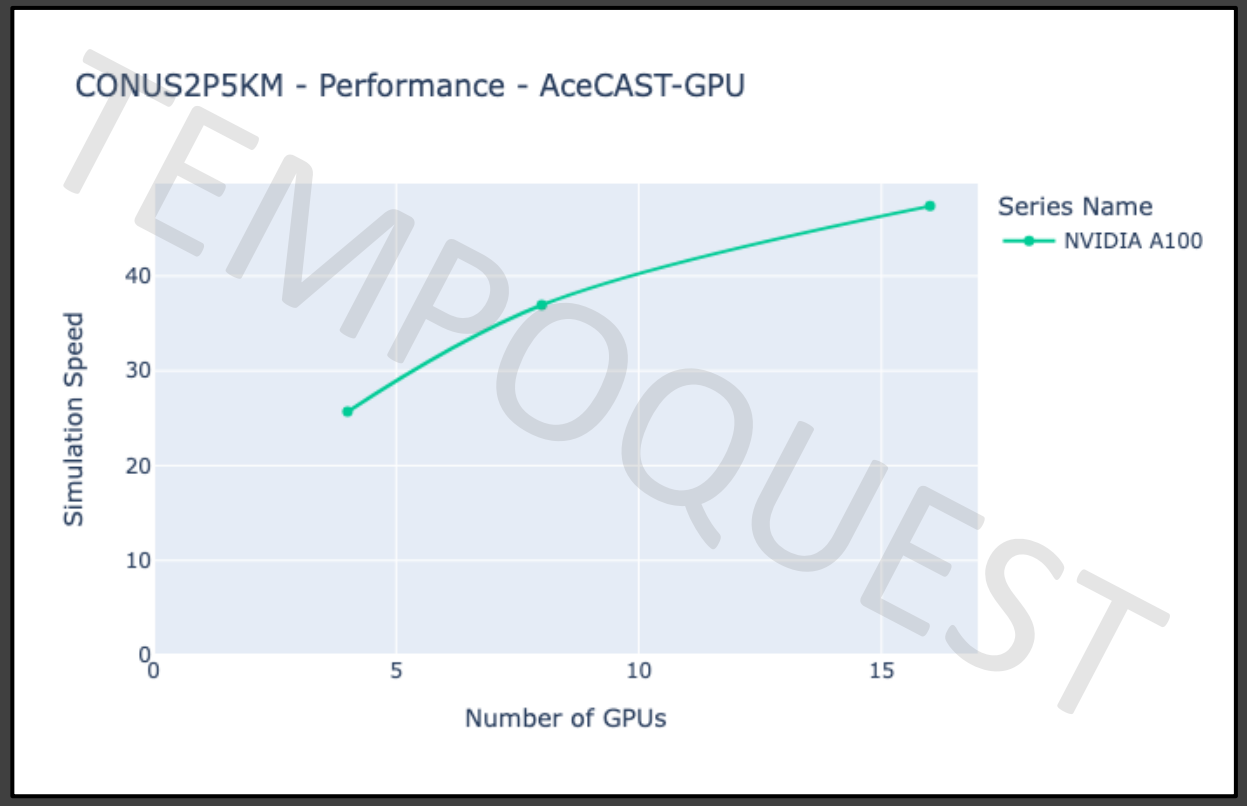
CPU Configuration

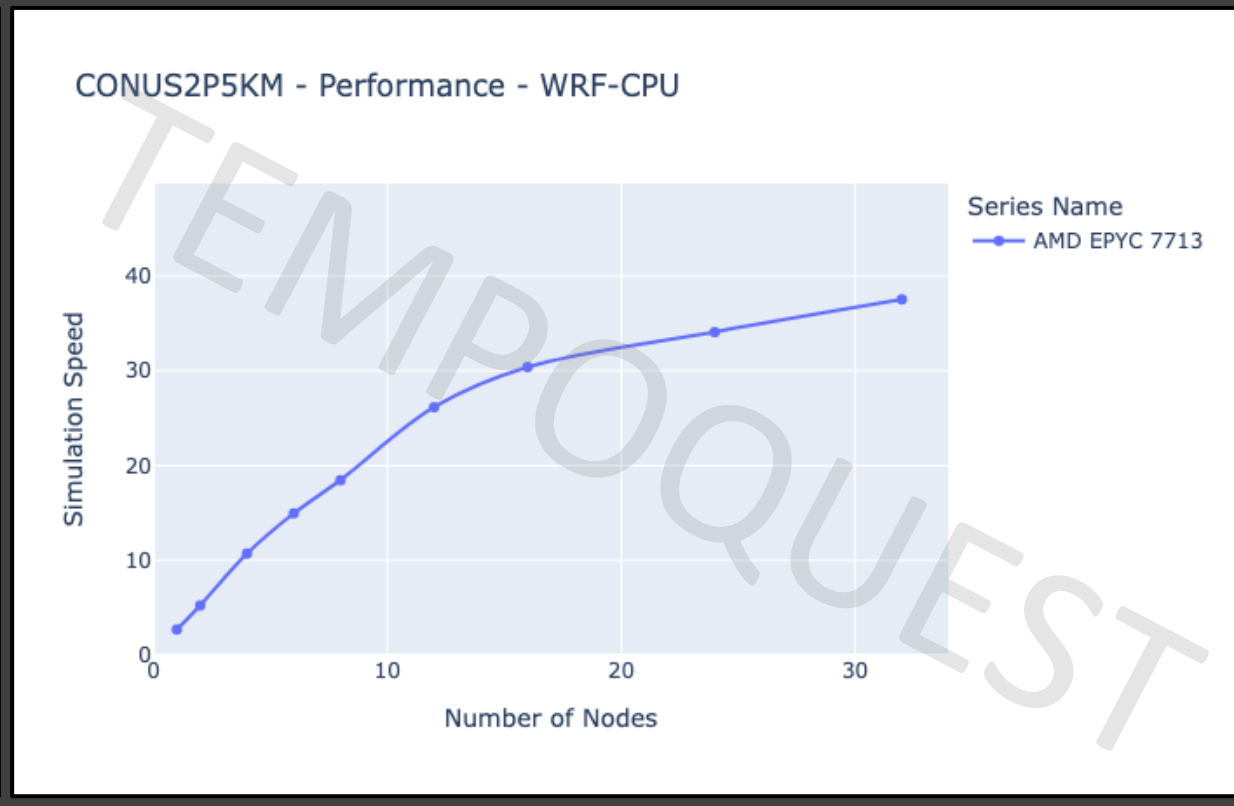
- Dell PowerEdge C6525
- 2X AMD EPYC 7713 CPU (128 total cores)
- 256 GB memory

GPU Configuration

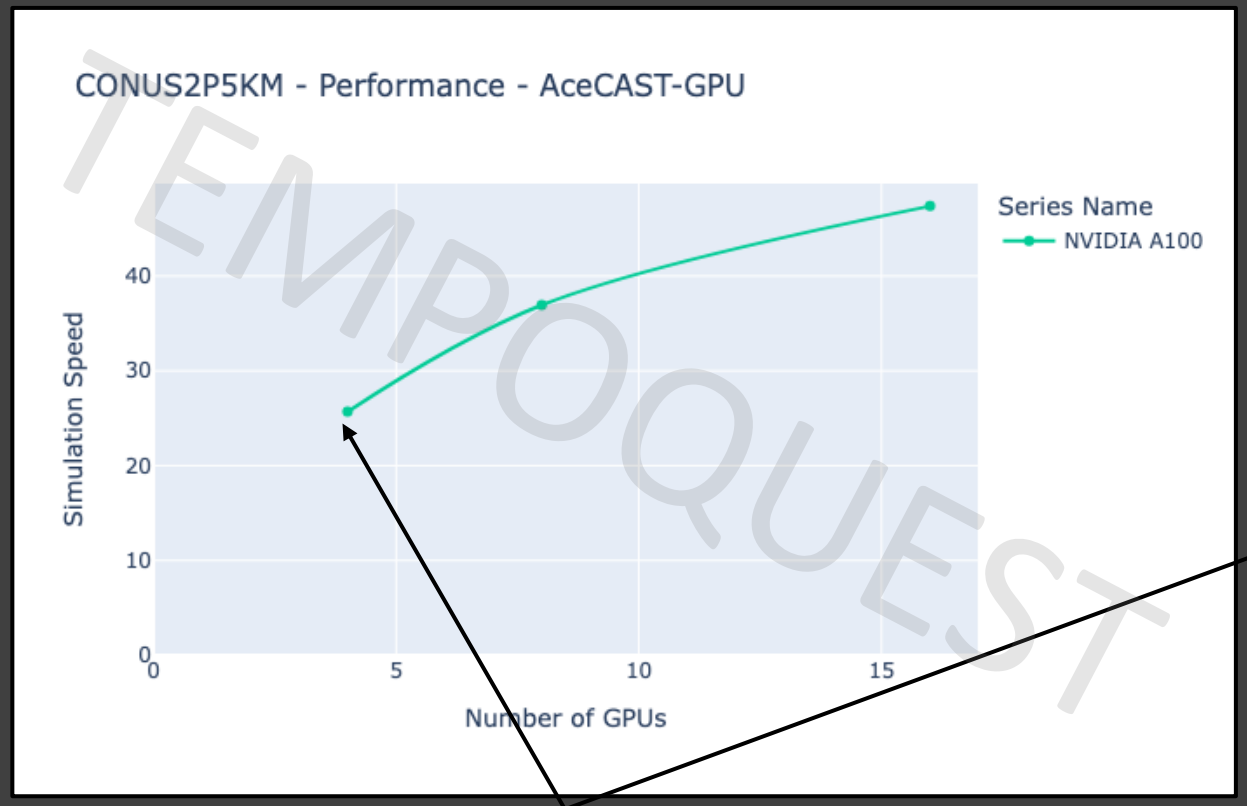
- Dell PowerEdge XE8545
- 2X AMD EPYC 7713 CPU (128 total cores)
- 4X NVIDIA A100-SXM4-80GB
- 512 GB memory

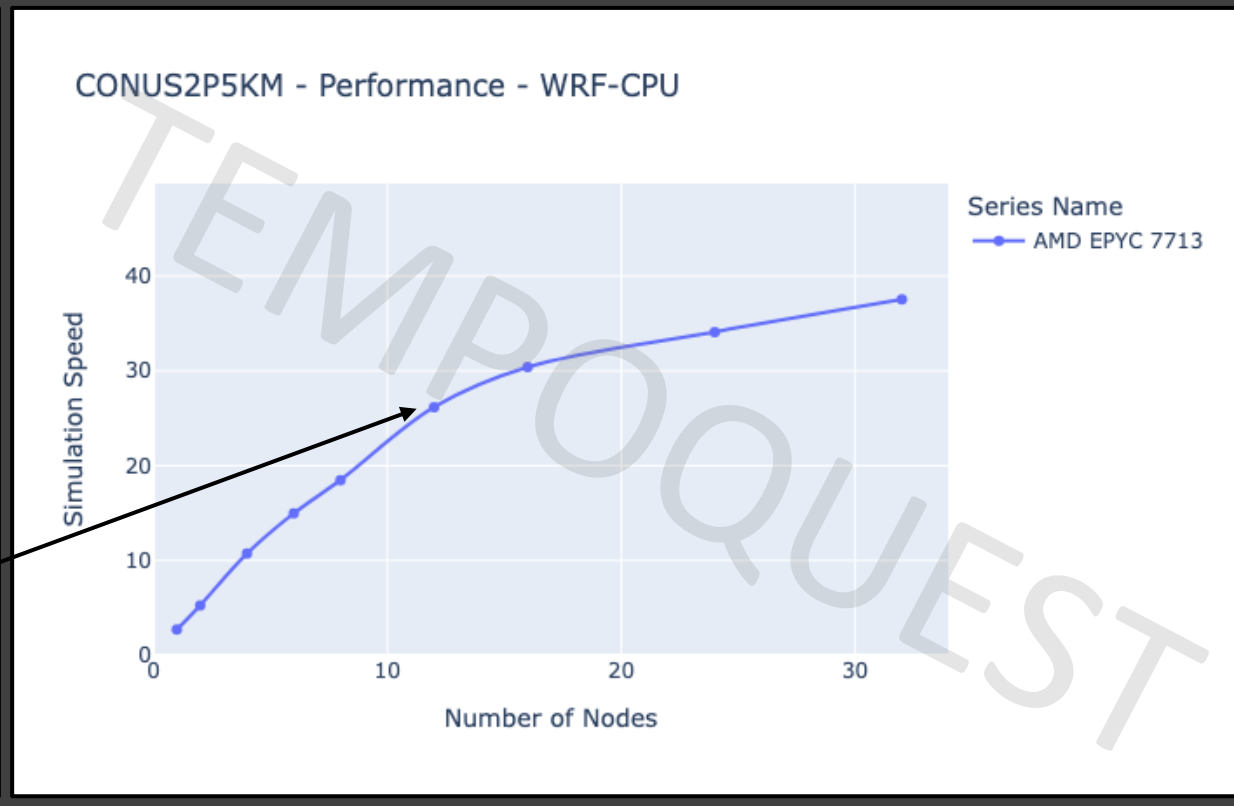
CONUS 2.5 Performance





CONUS 2.5 Performance

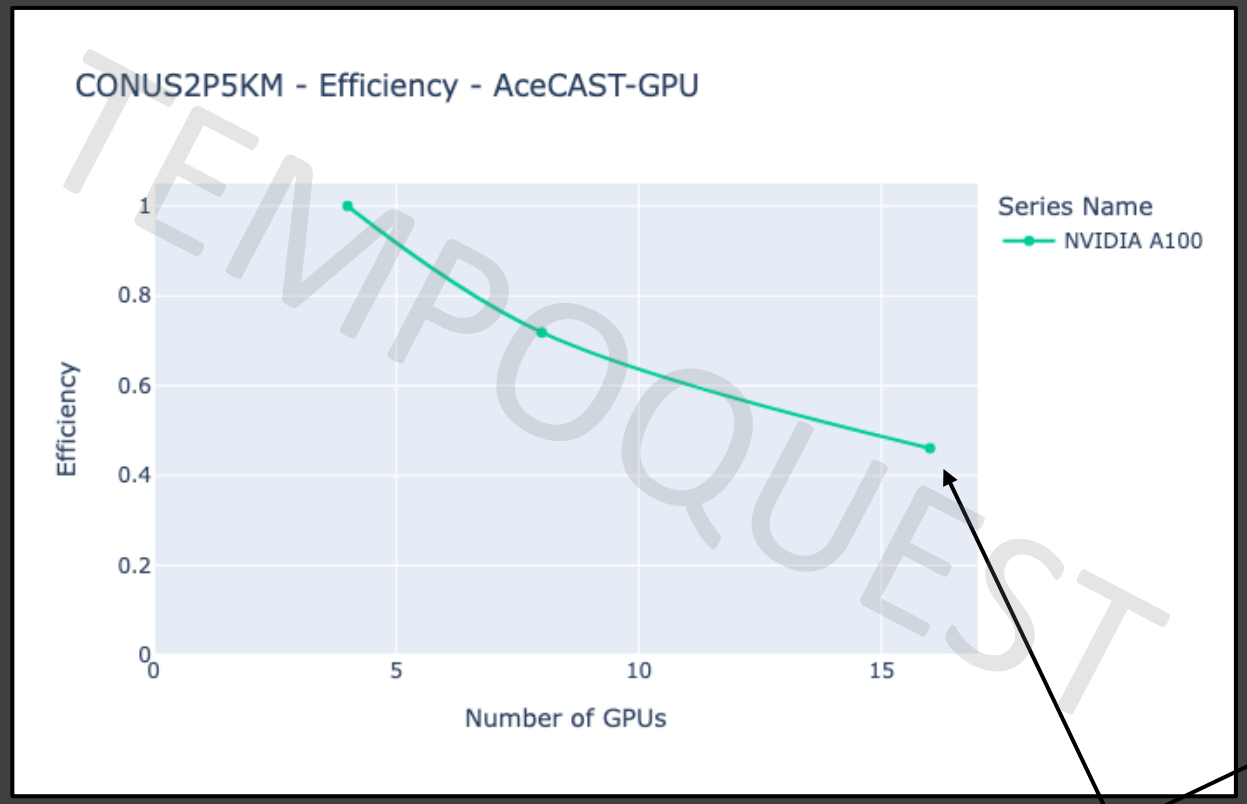


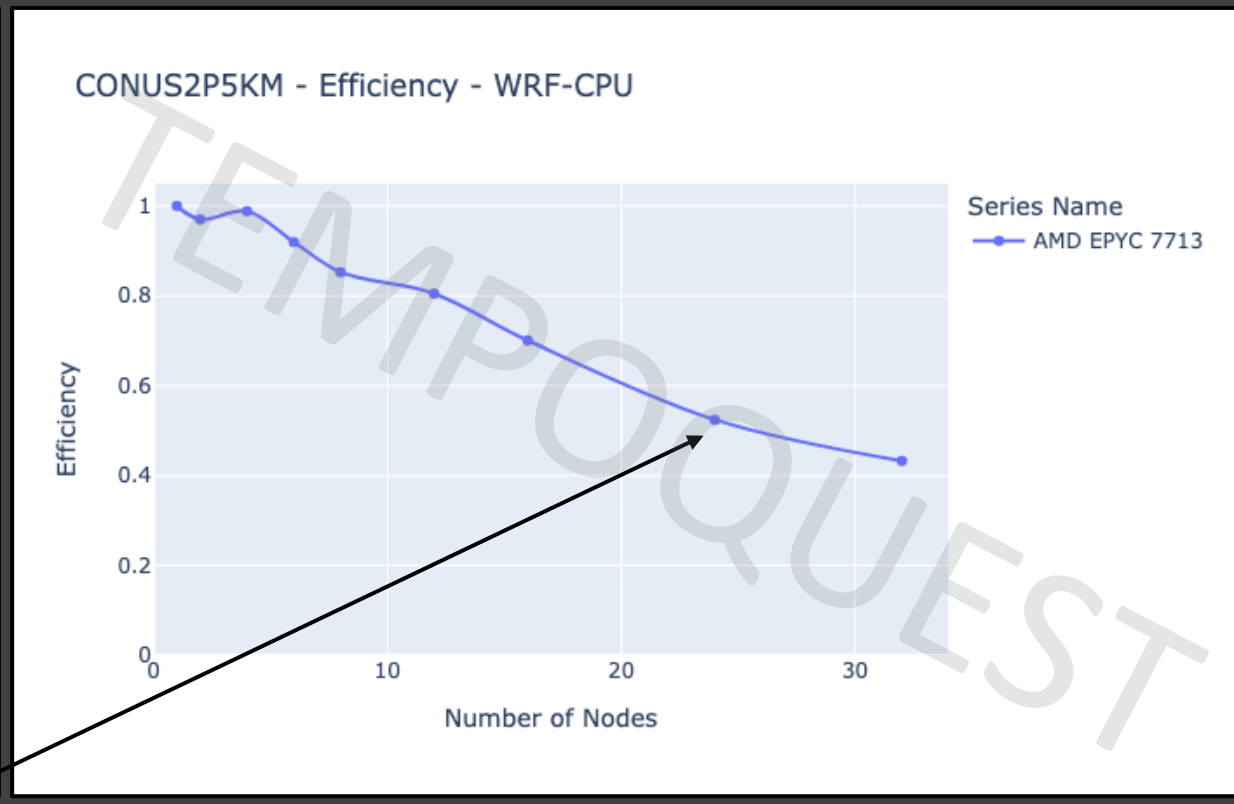


26X simulation speed reached at

- 1 GPU Node (4 NVIDIA A100 GPUS)
- 12 CPU Nodes

CONUS 2.5 Efficiency

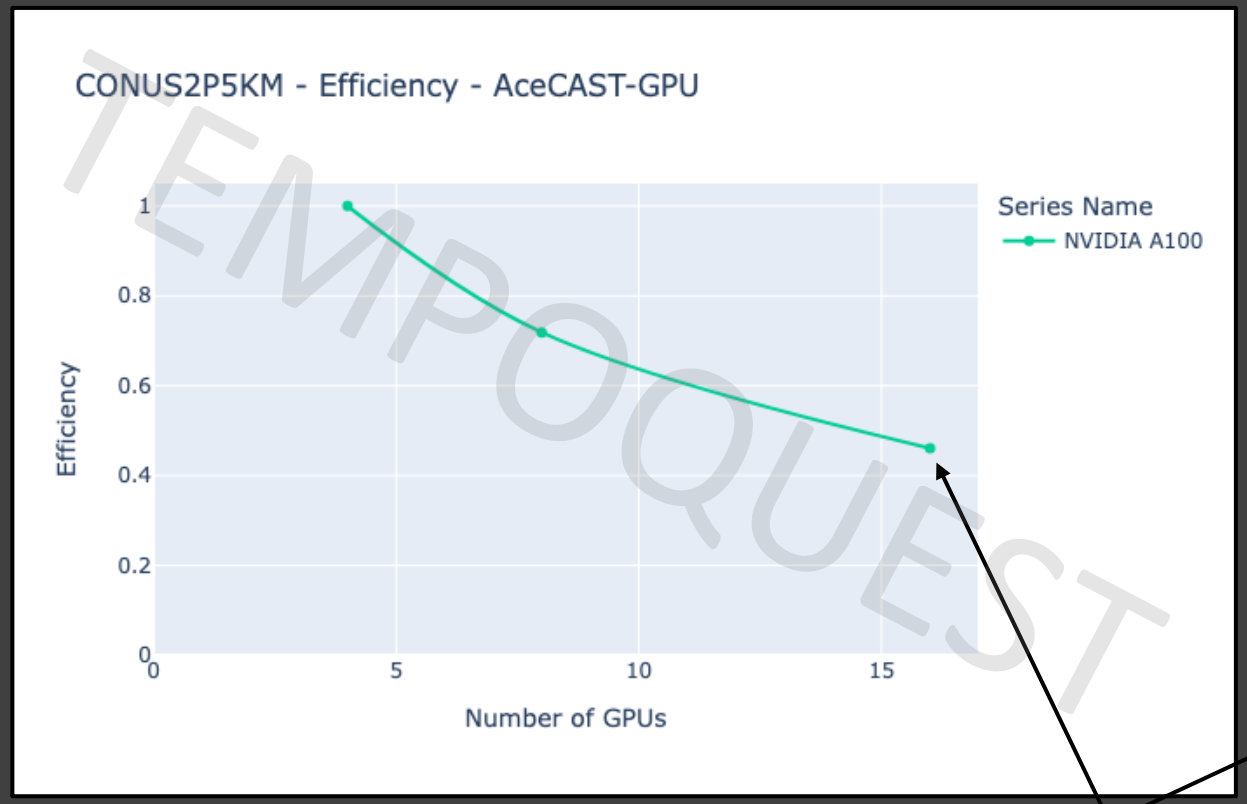


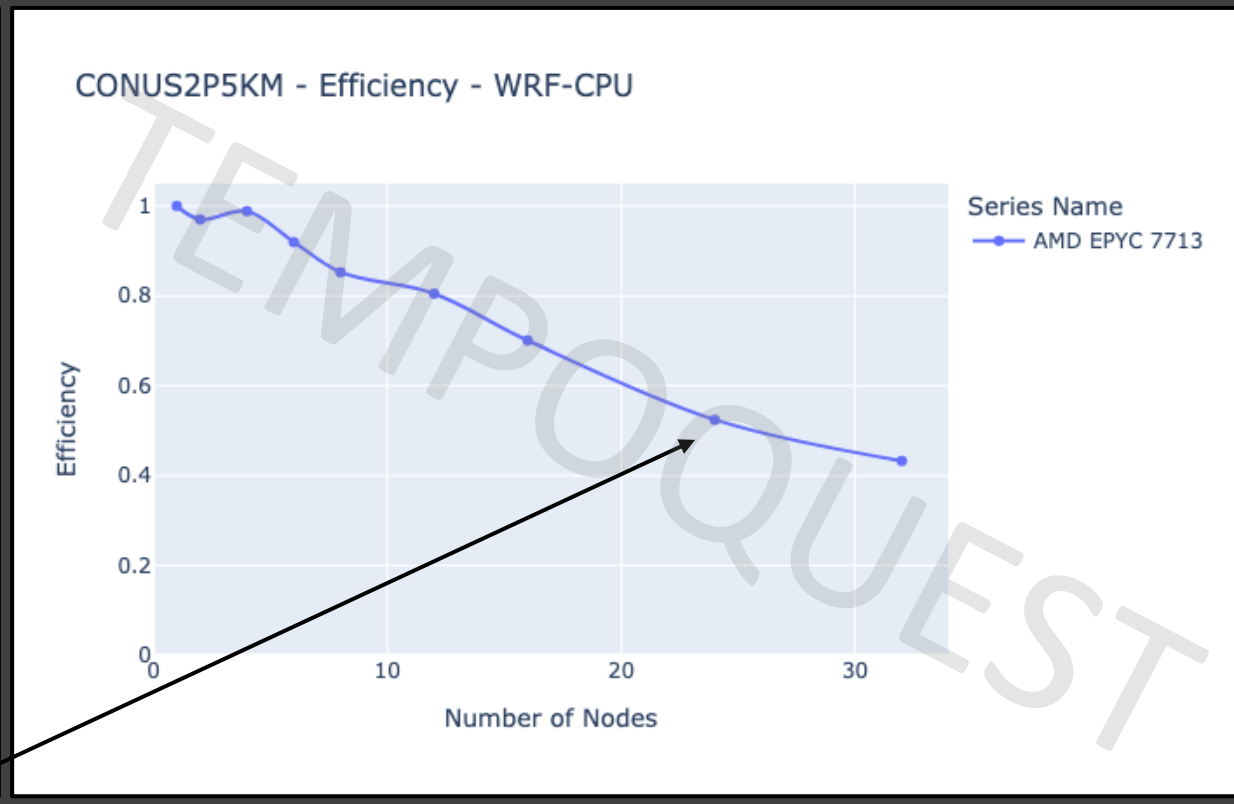


50% efficiency reached around

- 4 GPU Nodes
- 24 CPU Nodes

CONUS 2.5 Efficiency





GPU efficiency drops are primarily attributed to underutilization of the GPUs

- Patch sizes don't provide enough work for the thousands of GPU compute cores

50% efficiency reached around

- 4 GPU Nodes
- 24 CPU Nodes

CPU efficiency drops are primarily due to increasing MPI communication overhead

- This can be mitigated to further improve strong scaling with hybrid OpenMP/MPI execution

Validation

- Objective – Ensure that the differences between AceCAST and WRF results are *exclusively* due to floating-point error propagation rather than incorrect model implementation
- Standard validation process for any given WRF configuration:
  - Compile two builds of CPU-WRF, one using `-O0` and one using `-O3`
    - The higher optimization level will trigger things like FMA and different implementations of math functions
  - Generate two “baseline” WRF runs by running the simulation with both builds of CPU-WRF
    - The differences between these two baselines can be attributed to FP error propagation and therefore can be used to determine acceptable differences when validating AceCAST
  - Run the simulation with AceCAST on GPUs and compare the results to the baseline runs
  - If the AceCAST vs. CPU-baseline differences are consistent with the “acceptable differences” that we see when comparing the two baseline WRF runs, then we can conclude with reasonable certainty that AceCAST is implemented correctly

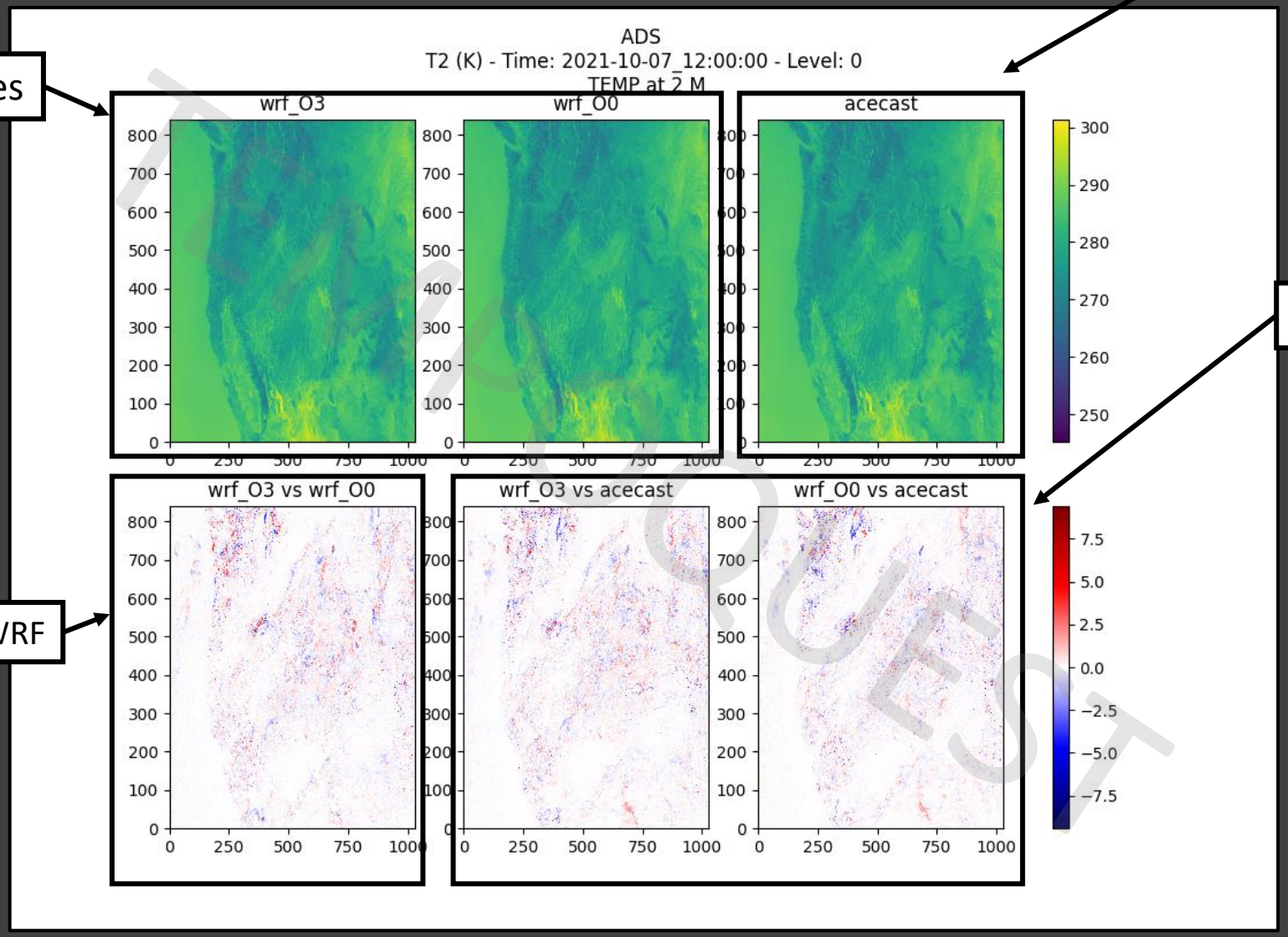
Validation – T2 at t=24 hours

AceCAST

CPU-WRF baselines

CPU-WRF vs. AceCAST

CPU-WRF vs. CPU-WRF



Validation – RAINNC at t=24 hours

AceCAST

CPU-WRF baselines

CPU-WRF vs. AceCAST

CPU-WRF vs. CPU-WRF