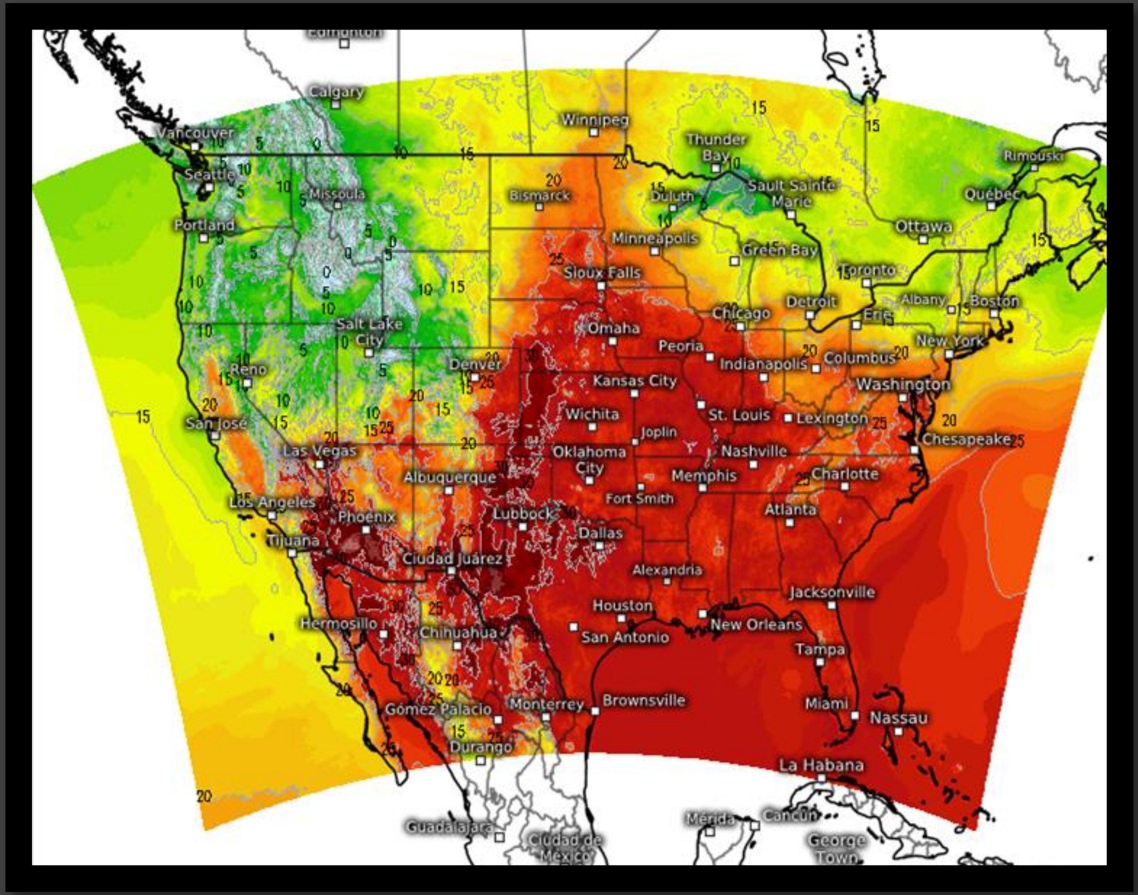
AceCAST GPU-Accelerated WRF

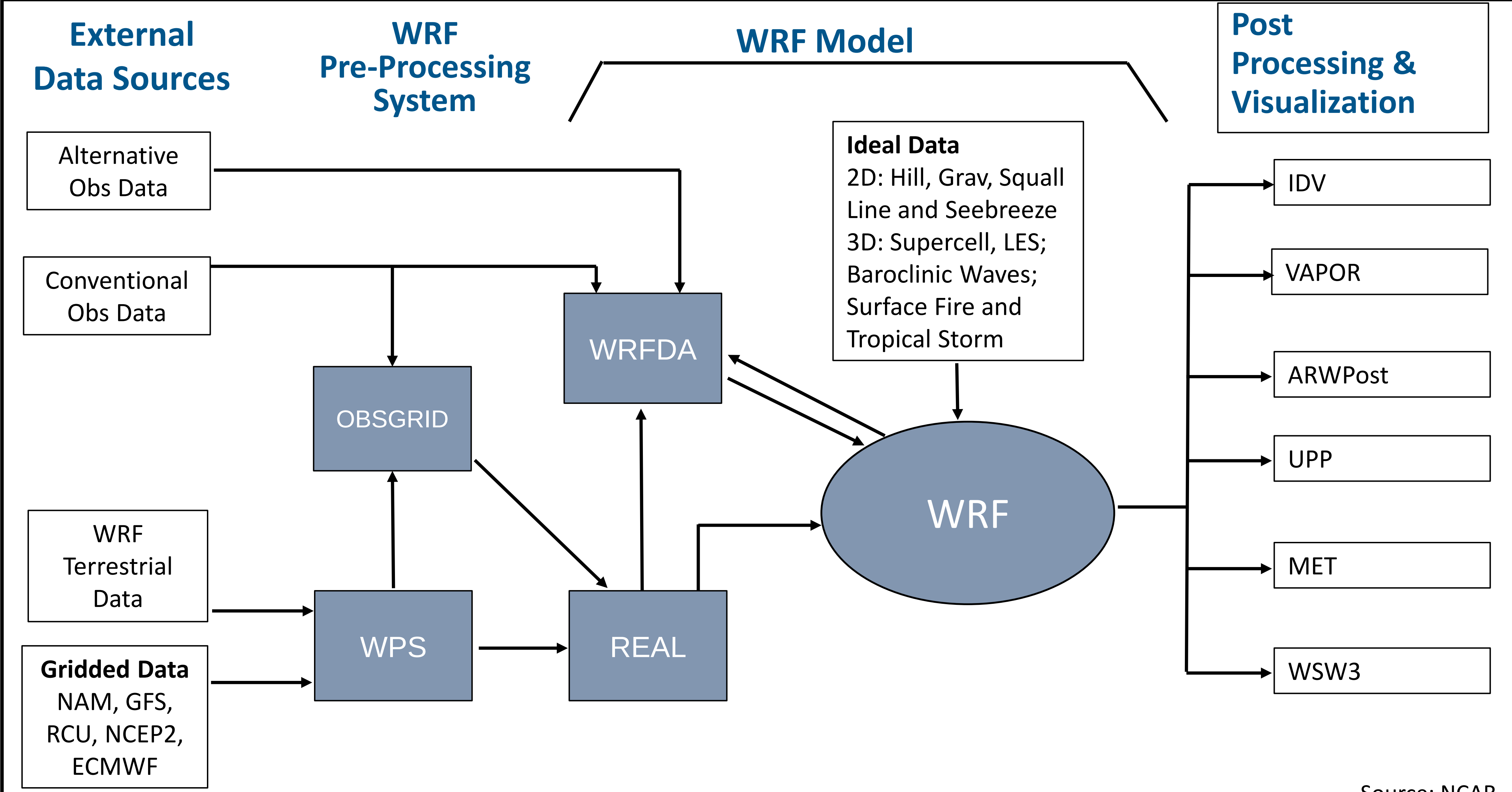
Model, Model Description, Performance, Validation and Impact

Weather Research and Forecasting (WRF) Model

- NWP simulation software designed for both research and operational forecasting
  - > 30,000 users worldwide
  - > 180 countries
- Developed, maintained and distributed primarily by the National Center for Atmospheric Research (NCAR)
- Extremely flexible code
- Well understood and documented (thousands of publications over multiple decades of widespread use)

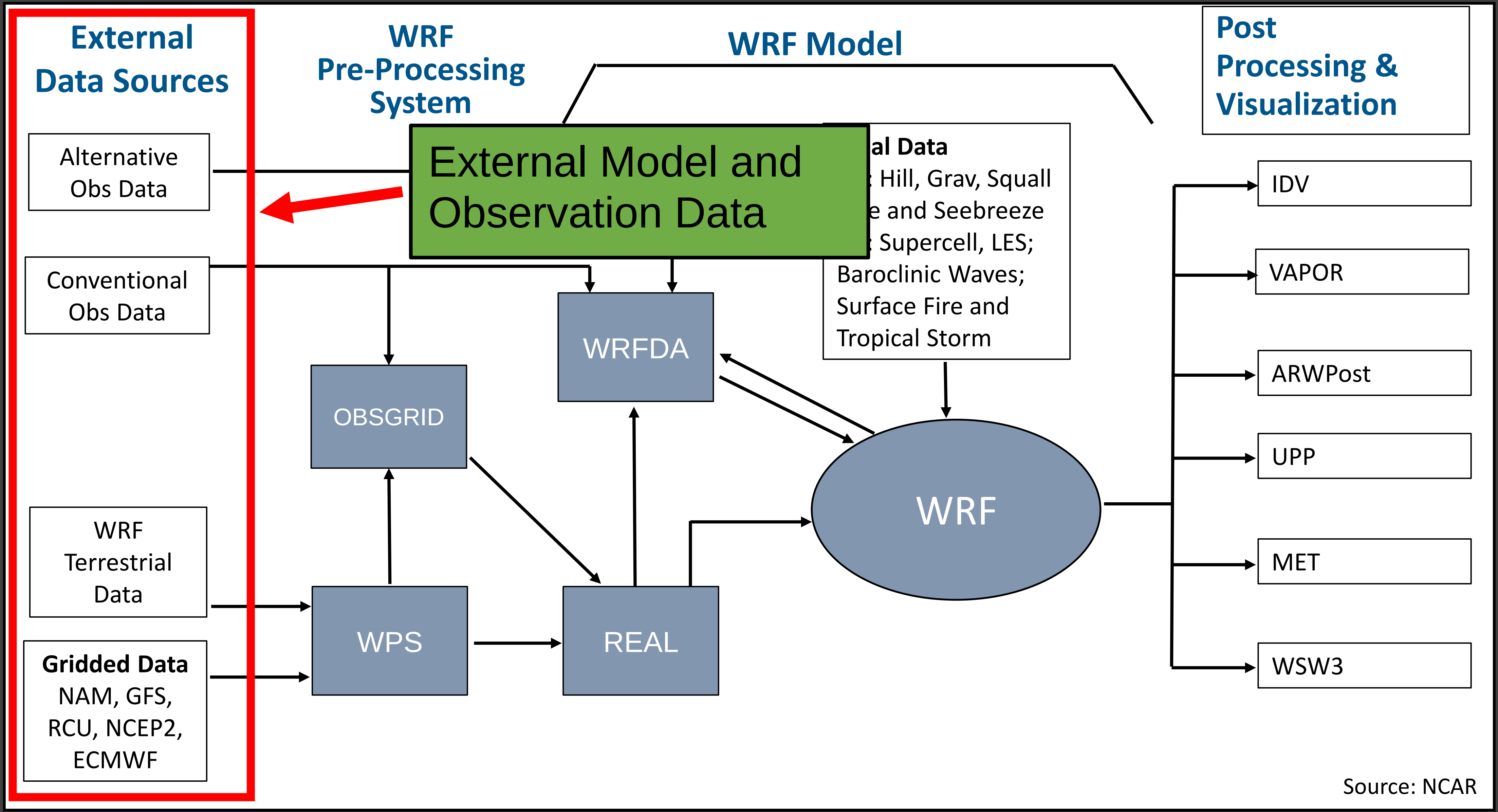


WRF Modeling Workflow

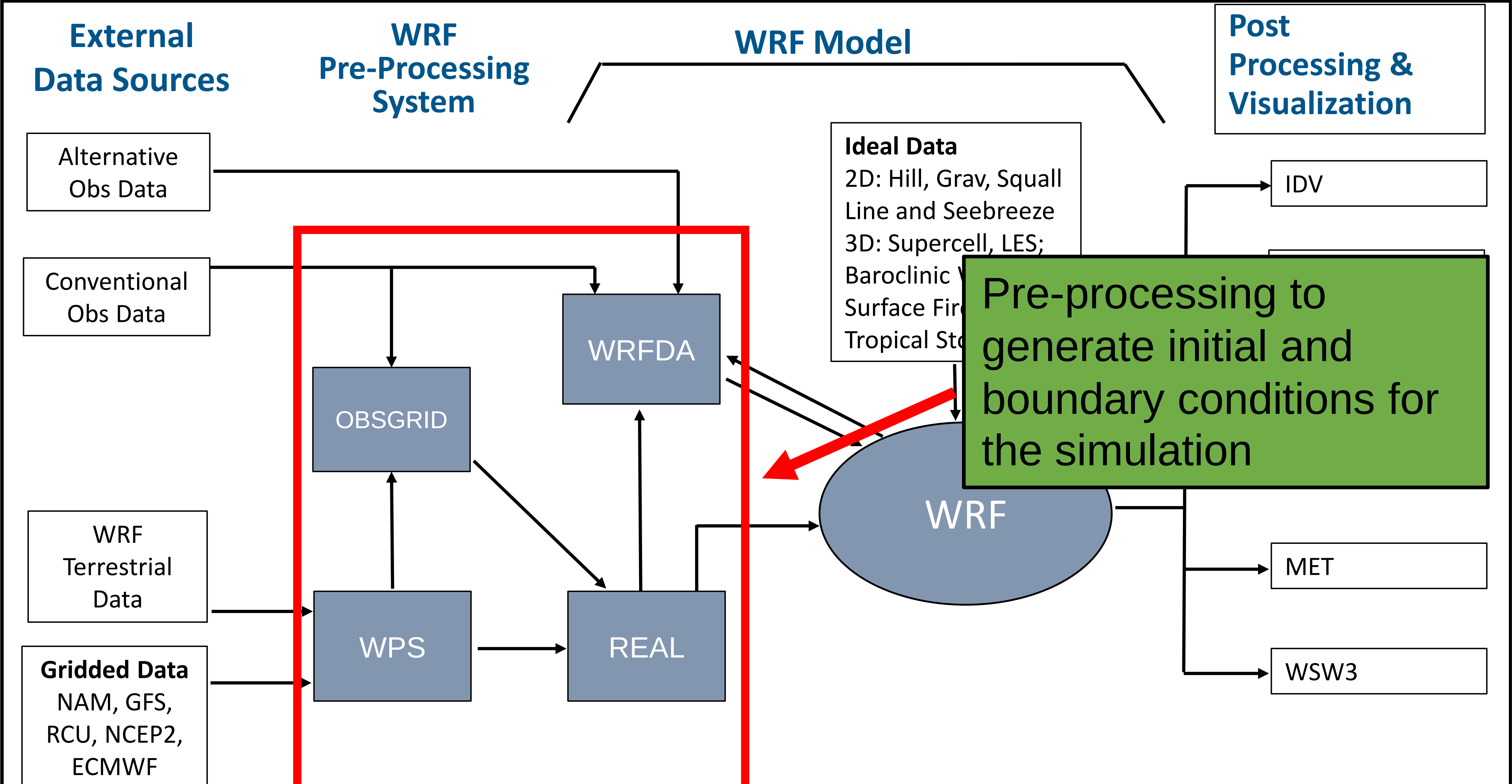


Source: NCAR

WRF Modeling Workflow

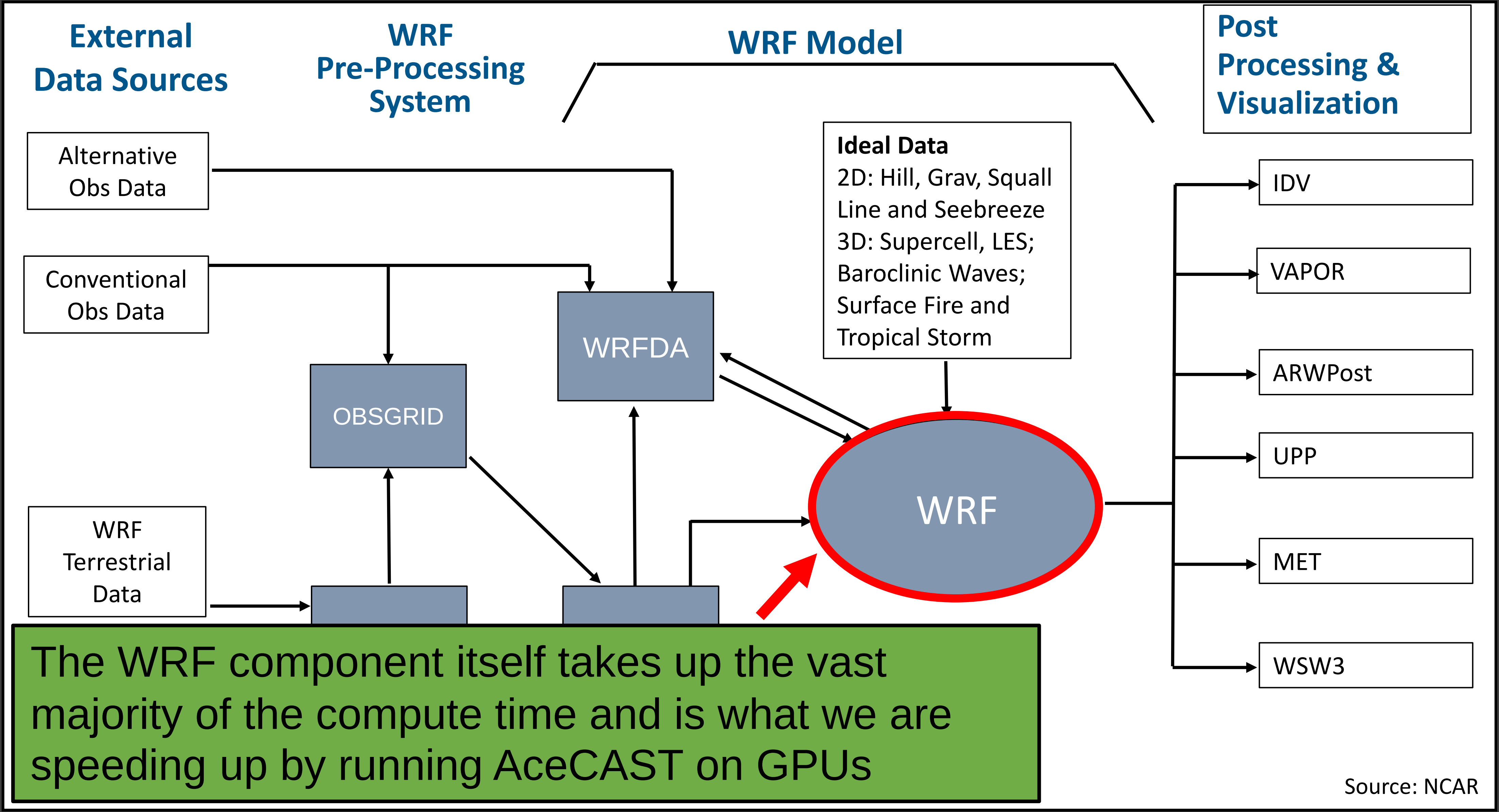


WRF Modeling Workflow



Source: NCAR

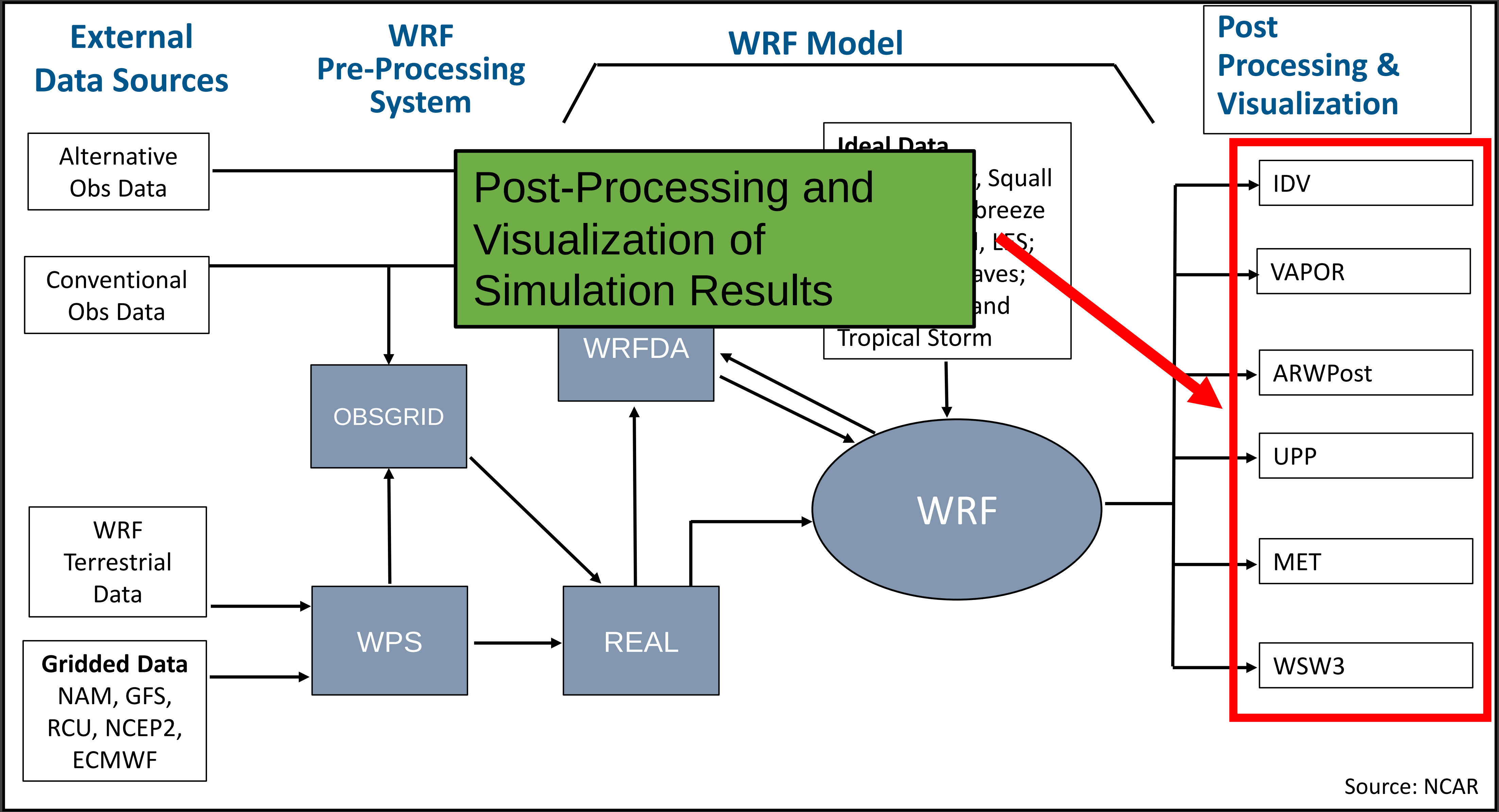
WRF Modeling Workflow



The WRF component itself takes up the vast majority of the compute time and is what we are speeding up by running AceCAST on GPUs

Source: NCAR

WRF Modeling Workflow



WRF Limitations

- Computational Expense
  - Compute resources are always a major limitation for users
  - Only implemented for execution on CPU
  - Can’t take advantage of modern HPC GPU architectures
- High barrier to entry
  - Steep learning curve
  - Requires specialized HPC skillsets

TQI’s AceCAST software addresses these limitations, which consequently enables significantly better modeling and operational forecasting capabilities

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

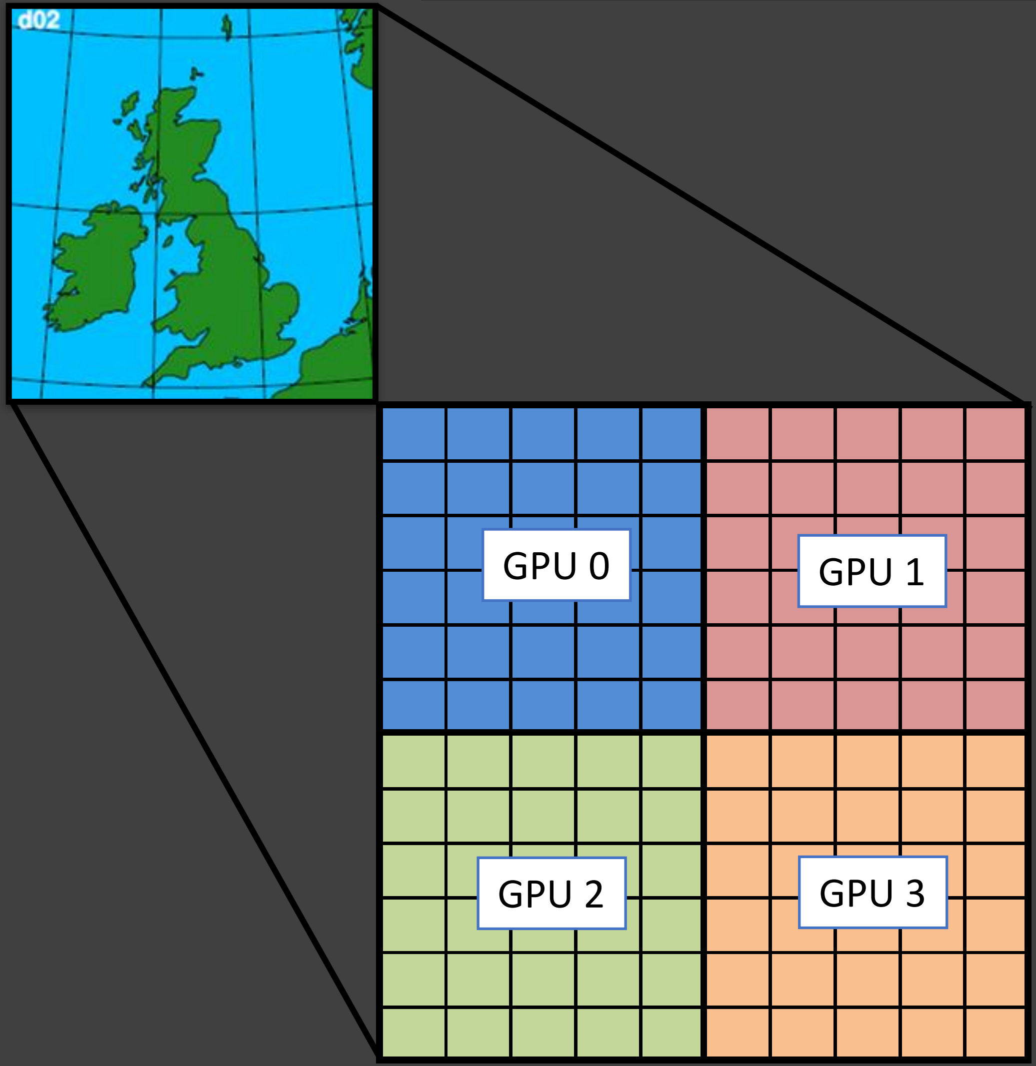
In short, AceCAST does exactly what CPU-WRF does but much faster since it takes advantage of the superior performance of modern GPU architectures

Multi-GPU Execution with MPI

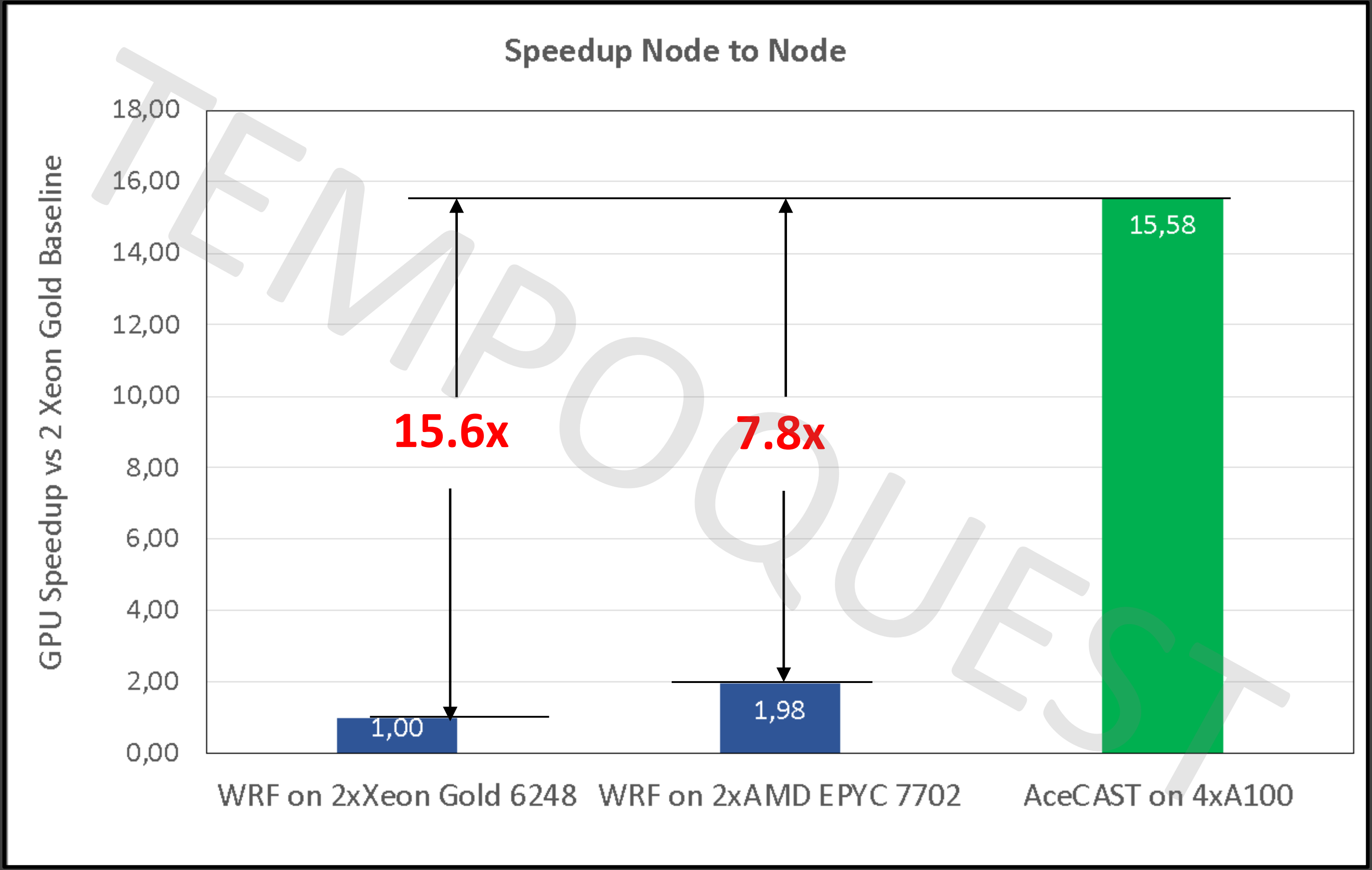
Running AceCAST with 4 GPUs:

```
$ mpirun --n 4 ./acecast.exe
```

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



Easter1500 Single-node Performance Benchmark

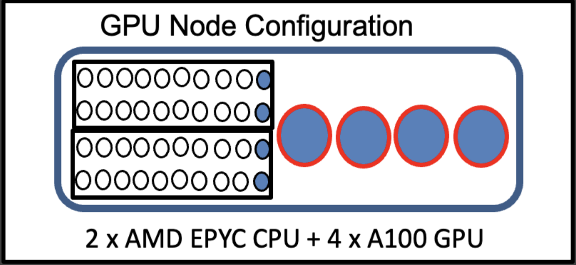


Model: Easter1500

- 1500x1500 grid
- 51 vertical levels
- 3 km resolution
- 1-hour forecast

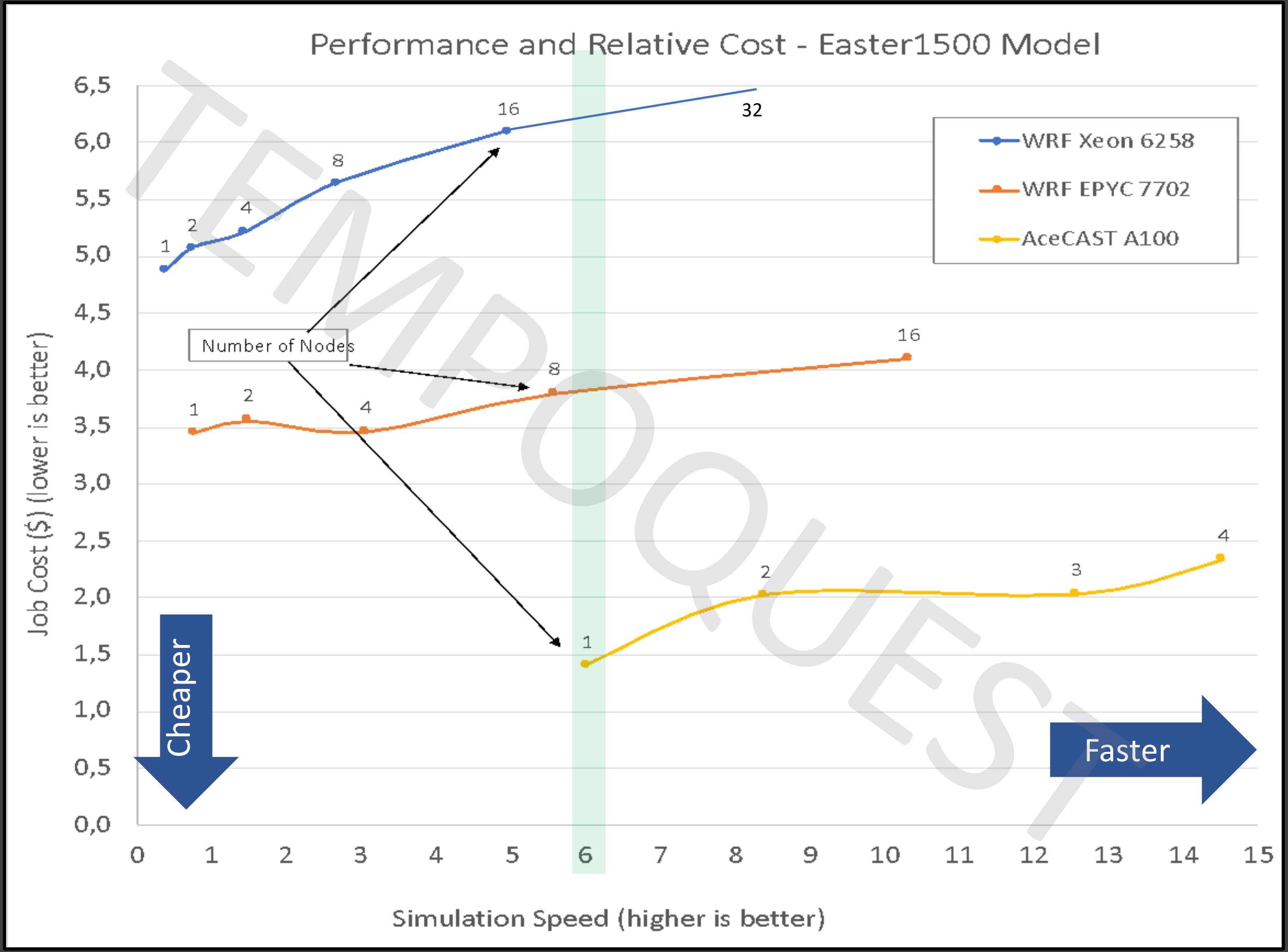
NAMELIST CONFIGURATION

&physics
mp_physics = 6
ra_lw_physics = 4
ra_sw_physics = 4
radt = 3
bl_pbl_physics = 1
sf_surface_physics = 2
sf_sfclay_physics = 1
cu_physics = 0



- AceCAST runs on single-node w/ 4xA100 GPU ***15.6x*** faster than WRF on single-node w/ 2x Xeon Gold 6248
- AceCAST runs on single-node w/ 4xA100 GPU ***7.8x*** faster than WRF on single-node w/ 2x AMD EPYC 770

Performance-cost analysis for systems with A100 GPUs



Node Replacement Factor at the same simulation sped = 6 :

- 1 node x4 A100 GPU
- 8 nodes x2 AMD EPYC CPU
- 18 nodes x2 Intel CPU

Cost of the job on 1 node A100 GPU is:

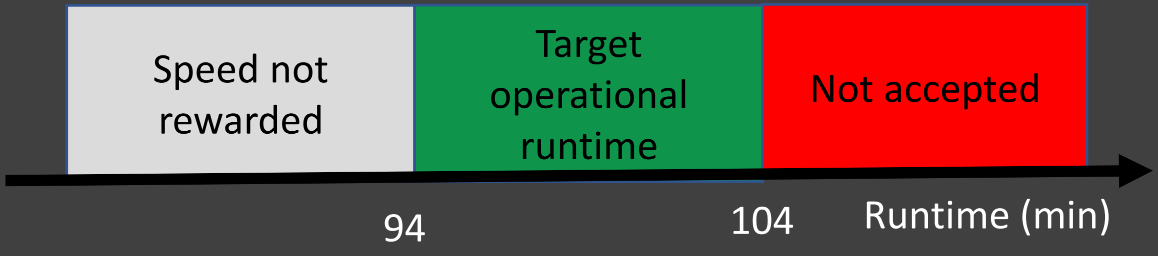
- **4x** cheaper than on Intel CPU WRF
- **2.5x** cheaper than on AMD CPU WRF

TCO Assumptions:

- 3 year depreciation
- 220 days /year
- 66% utilization

Capacity Benchmark for Operational Forecast

- **RFP operational requirement:**
  - **10-member ensemble**
  - **Model: 1250 x 1100 x 42, 10km res, 72-hour**
  - **Run time between 94 -104 minutes**
  - **Slower than 104 min is not accepted**

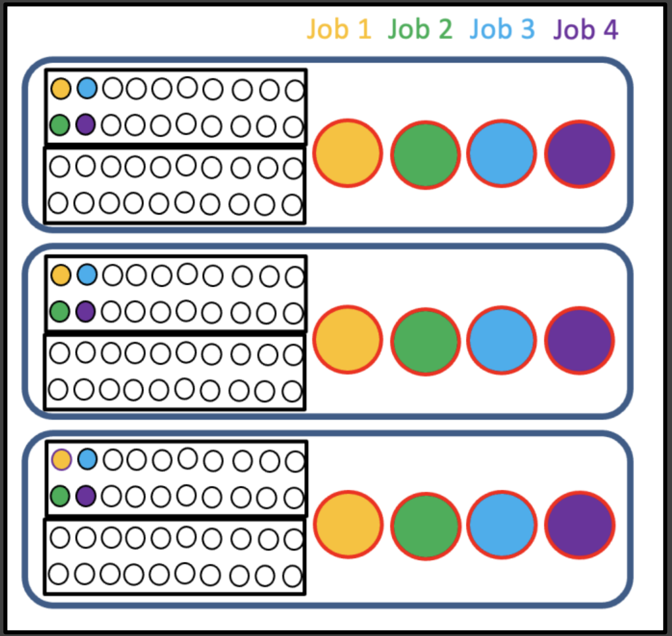


For Each Member:

- **WRF runs on 6 CPU nodes in 90 min**
- **AceCAST runs on 3 GPUs in 91 min and on 4 GPU in 75 min**

Entire Ensemble:

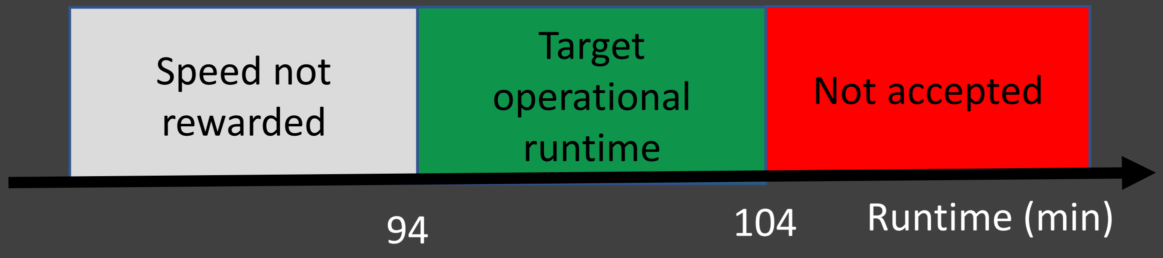
- **Configuration for WRF: Cluster with 60 CPU nodes, each with 2xAMD CPUs**
- **Configuration for AceCAST: Cluster with 8 GPU-nodes, each with 4x A100 GPU**

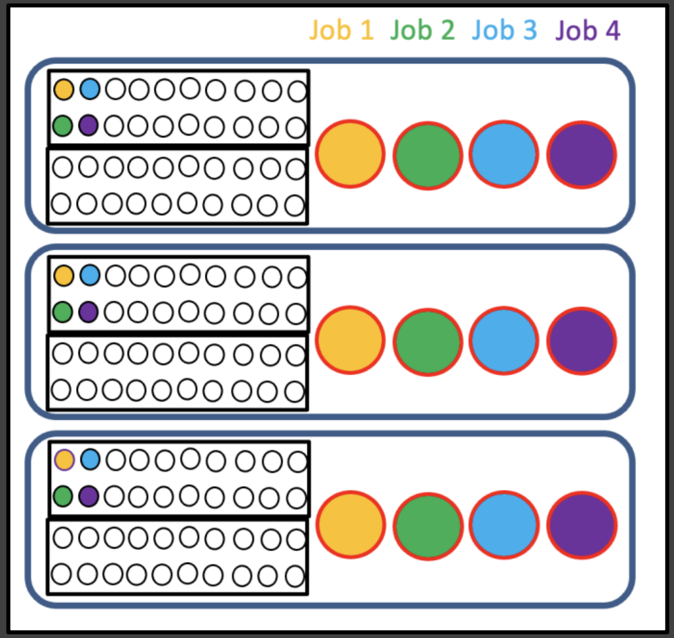


Source: TQI benchmark on HPC Lab systems

Capacity Benchmark for Operational Forecast

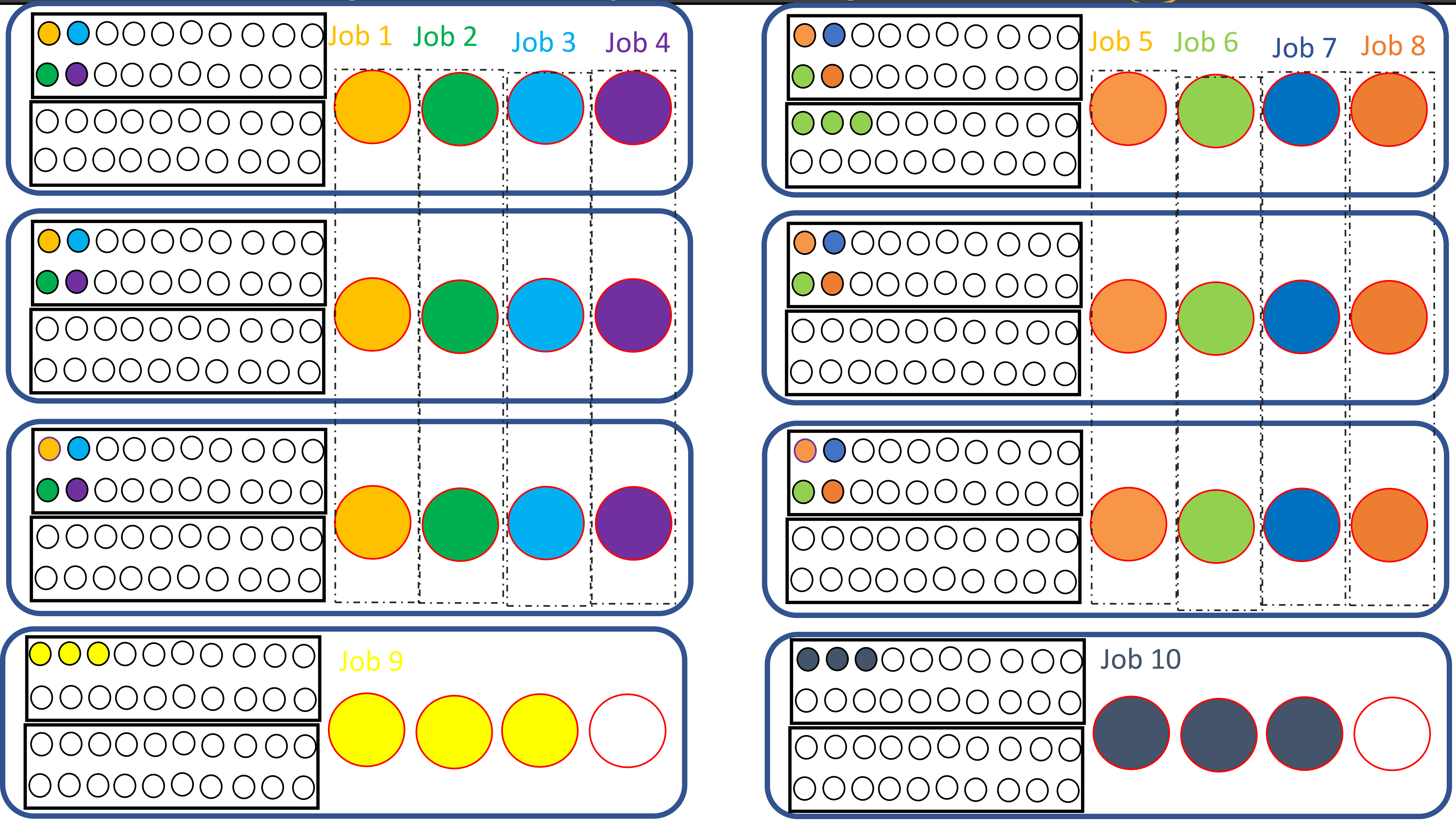
- AceCAST runs entirely on GPU and uses one core per GPU for job control and IO
- A throughput of 4 Jobs (each using 3xA100 GPU) runs on 3 nodes in 96 min
- 5% overhead vs single job running in 91 min



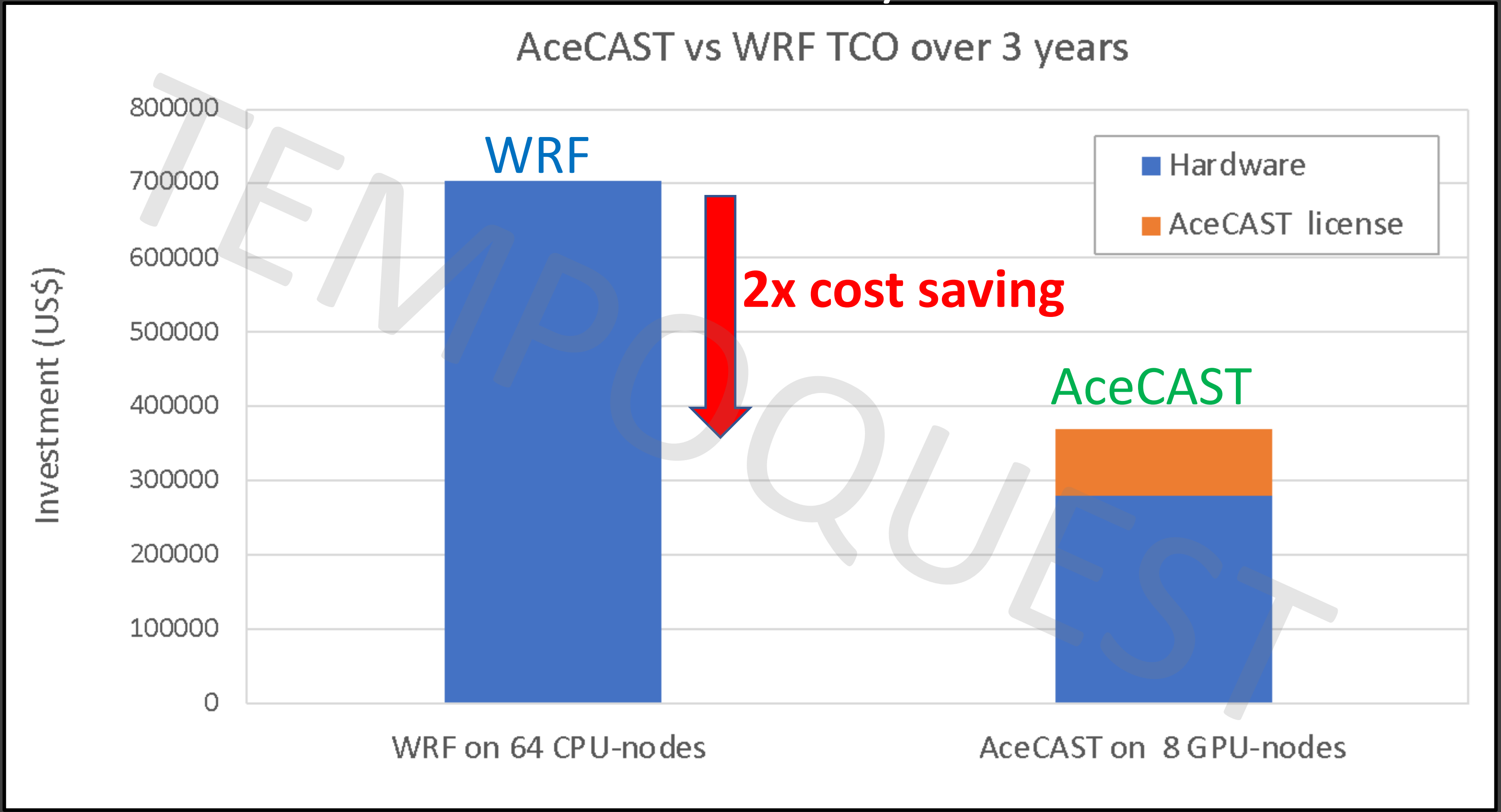


Source: TQI benchmark on HPC Lab systems

Ensemble Configuration - 10 jobs running on 8 nodes



TCO for AceCAST vs WRF for 3-year utilization



The cost reduction is actually up to **3x** when we add:

- Interconnect, storage, rack infrastructure costs
- Power consumption: one GPU node consumes the same power as 2.5 CPU nodes, saving energy for 44 CPU nodes

AceCAST Validation

Why is validation necessary?

- Floating point errors cause simulation results with identical configurations (namelist, wrfinput, wrfbdy, etc.) to differ between multiple runs due to
  - Differing compute architectures
  - Differing compilers/compiler optimizations
  - Runtime-specific aggregate operations
- Errors due to these reasons are acceptable and are to be expected
- Errors due to flawed implementation of the underlying mathematics of the model are not acceptable
- How do we determine if the error between simulations run with CPU-WRF and GPU-AceCAST is acceptable or not?

Methodology

Objective – Ensure that AceCAST simulation results are statistically equivalent to their CPU-WRF counterpart

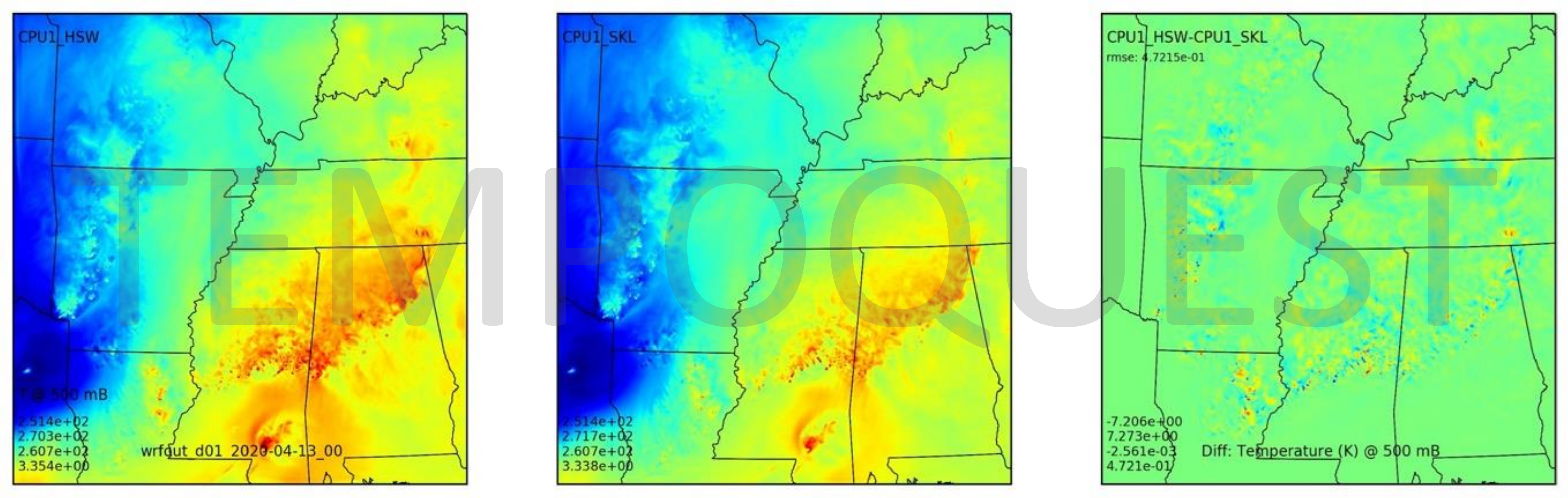
- For any given simulation configuration (namelist, input files):
  - Run the simulation with AceCAST as well as with two or more differing CPU-WRF setups (different architectures, different compilers, etc.)
  - Compare the differing CPU-WRF simulation results to determine approximate acceptable error tolerances
  - Determine if the errors between the CPU-WRF simulations and the AceCAST ones are within the acceptable error tolerances
    - If they are – we can assume the AceCAST implementation is correct
      - Or at least that the effect of any bugs in the code have negligible effects on the results
    - If they aren't – The AceCAST implementation likely has one or more bugs that need to be addressed by the developers prior to distribution

TempoQuest
WEATHER TRANSCENDED

CPU vs. CPU – Determining Acceptable Error Tolerances

Temperature (K) at 500 mb at = 24h

WRF run 1 | WRF run 2 | Error

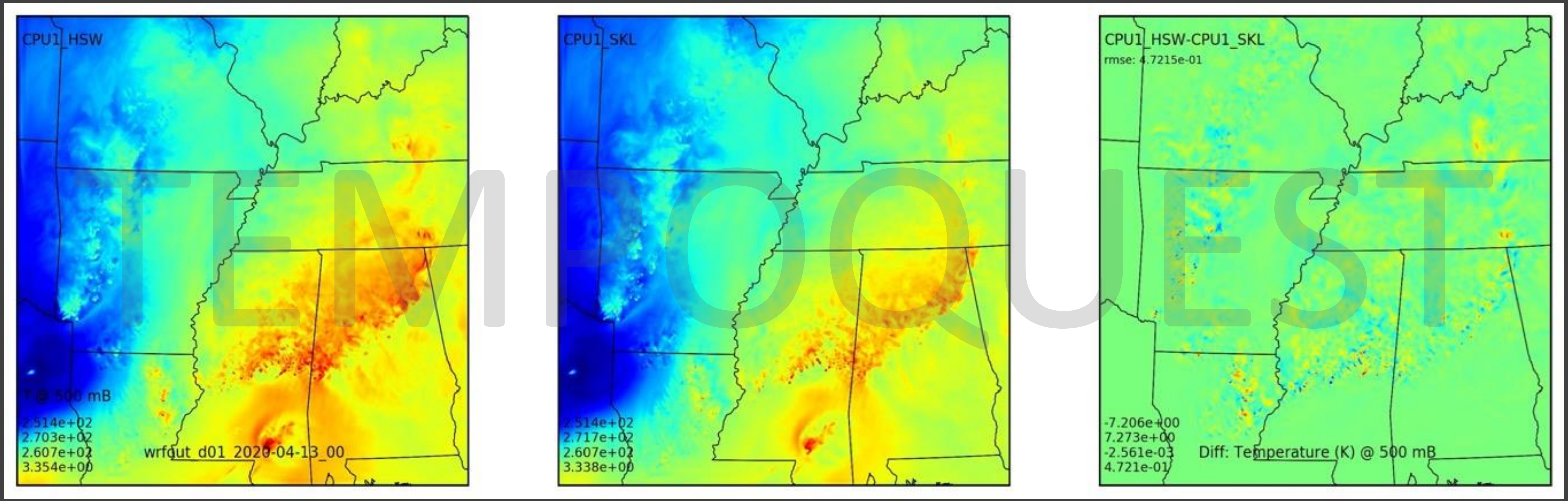


A useful metric for the comparisons is the Root Mean Squared Error (RMSE) which in this case is $4.72E^{-1}$

WRF vs. AceCAST – Validating Results

Temperature (K) at 500 mb at = 24h

WRF run 1 | WRF run 2 | Error



WRF vs. WRF RMSE: $4.72E^{-1}$
WRF vs. AceCAST RMSE: $4.76E^{-1}$

Can conclude that the AceCAST implementation is working for this field since the ***AceCAST error*** (WRF vs. AceCAST RMSE) is very similar to our ***acceptable error tolerance*** (WRF vs. WRF RMSE)