# GSL GPU Efforts

Mark Govett
NOAA Global Systems Laboratory

Jacques Middlecoff, Duane Rosenberg
Cooperative Institute for Research in the Atmosphere (CIRA)

Lynd Stringer, Yonggang Yu
Cooperative Institute for Research in Environmental Science (CIRES)

# Background: GPU development at GSL

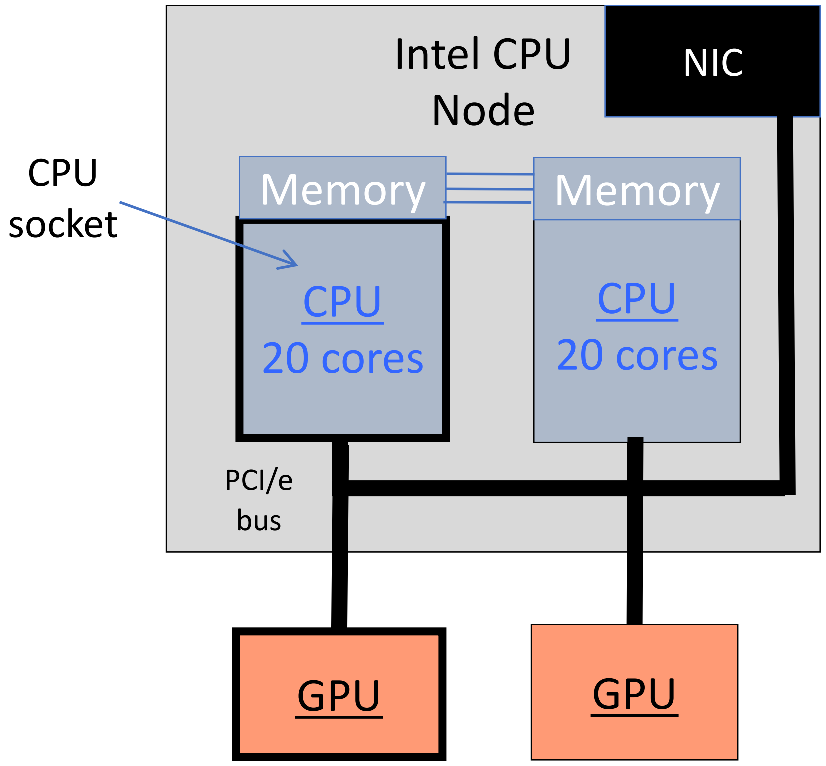
- 2008
  - GSL began exploring this technology shortly after CUDA was released by NVIDIA
- 2010 - 2012
  - GSL developed a GPU compiler called F2C-ACC to translate Fortran + directives into CUDA
  - Worked with Cray, NVIDIA and PGI to improve capabilities of openACC compilers
- 2010 – 2016
  - GSL developed the Non-hydrostatic Icosahedral Model (NIM) as an experimental dycore to explore fine-grain computing and non-hydrostatic modeling
  - GSL demonstrated performance, portability and scalability with a single Fortran source code using openACC, openMP and SMS directives
  - Runs on CPU, MIC, GPU, in serial and parallel with up to 10,000 CPUs, GPUs
- 2017
  - Govett, M., et al, Parallelization and Performance of the NIM Weather Model on CPU, GPU and MIC Processors, BAMS, October 2017

    DOI: [https://doi.org/10.1175/BAMS-D-15-00278.1](https://doi.org/10.1175/BAMS-D-15-00278.1)
  - Stopped development of the NIM model

# Understanding CPU and GPU Performance

- Single CPU core versus GPU
  - 30x – 100x speedups with GPUs frequently reported
- Single CPU socket versus GPU
- Single CPU node ( 2 sockets) versus GPU
  - **De facto standard for CPU – GPU comparisons**

- Examples of misleading comparisons in literature
  - 1 CPU core vs 1 GPU
  - 2012 CPU vs 2016 GPU
  - 2012 CPU vs two 2015 GPUs
- Fairest comparisons:
  - Same generation chips, dual-socket CPU, single GPU

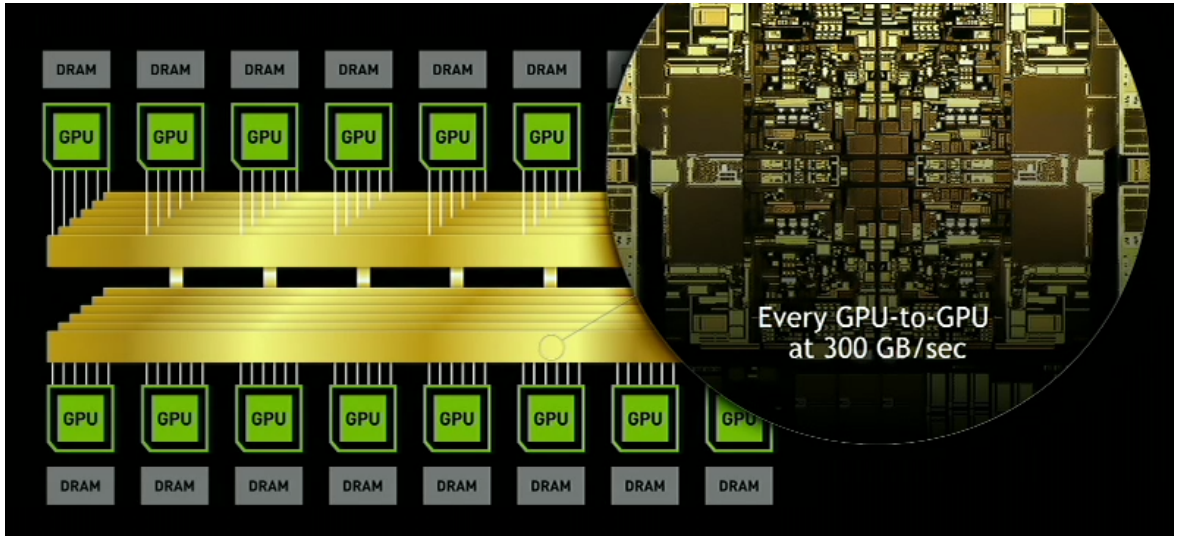



Standard 40 core dual-socket CPU node

- Cost, energy comparisons would be best but that's hard to do

# Advanced Node Technologies

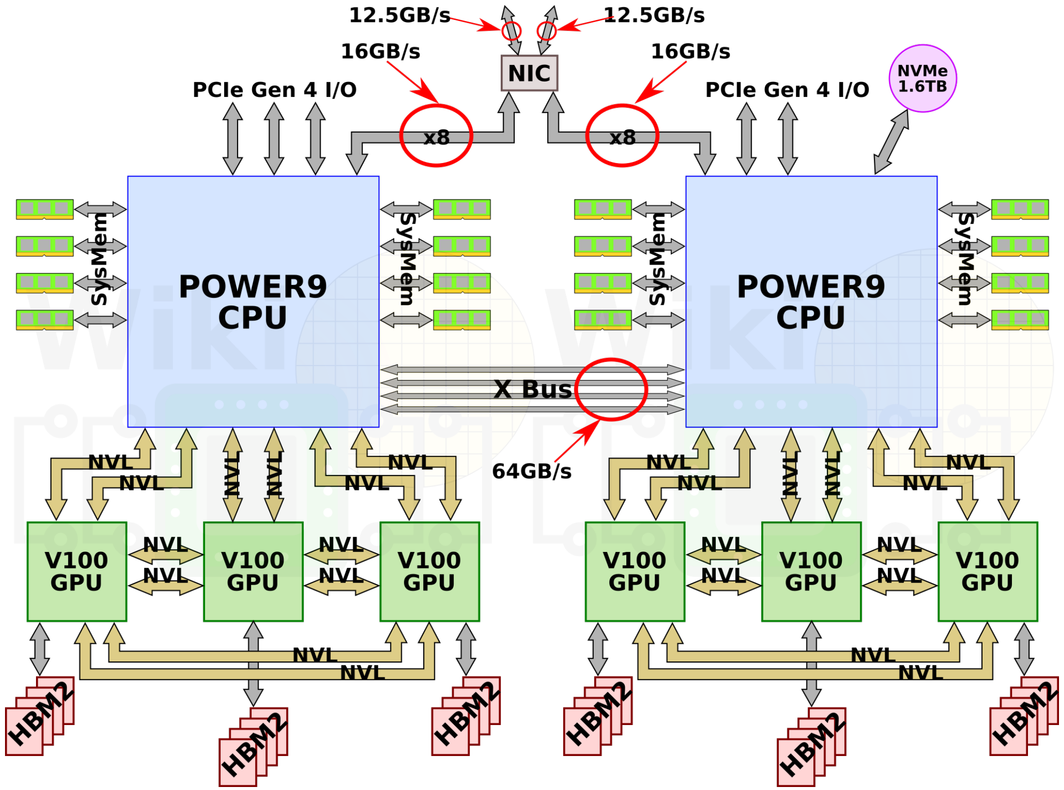
- Diversity
  - Performance, cost, power
- Complexity




**NVIDIA DGX-2**: 16 Tesla V100 GPUs, (81K GPU, 10K Tensor cores).

- 1.5 TB DDR4 RAM, 500 GB HBM2,   10kW power
- 300 GB/s NVLINK
- PCIe Gen3, 8x EDR IB / 100 Gigabit Ethernet

ORNL Summit Node



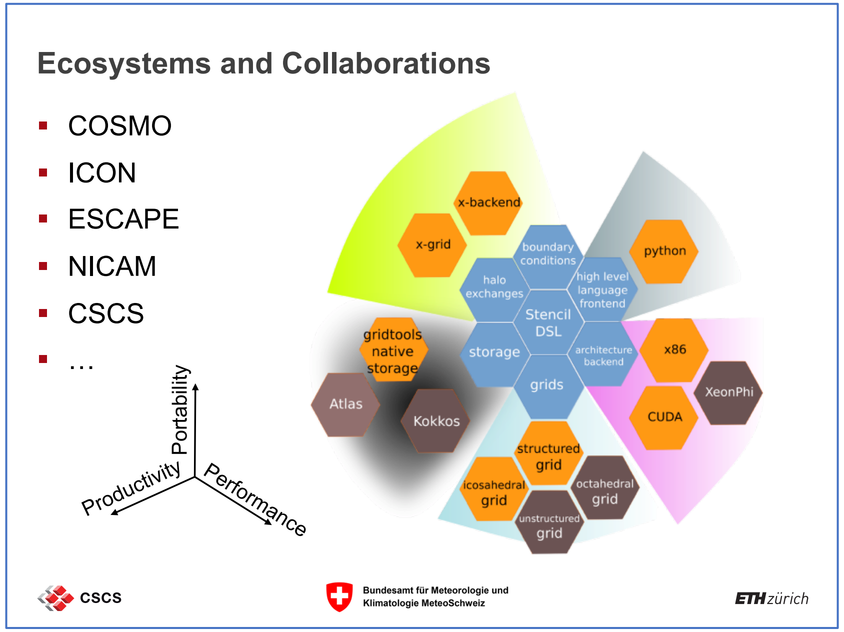

**DOE Summit node**:

- IBM Power9 CPU,  6 V100 GPUs, 30K GPU cores
- 512 GB DDR4 RAM, 96 GB HBM2
- NVLINK, 50GB/s bandwidth per link
- PCIe Gen 4 (16GB/s) for inter-node, I/O

**Summit System:  4600 nodes, 27K GPUs**

# Portability Approaches

- Preserve original Fortran, add directives
  - openACC, openMP
  - Least invasive for modeling teams
- Create a separate version for GPU
  - PGI-Fortran
  - CUDA
  - OpenCL
- Develop / use tools to transform Fortran
  - GridTools (CSCS), PSYclone (Ukmet)
  - Separation of concerns lead to black box solutions
  - Must get buy-in from modeling team

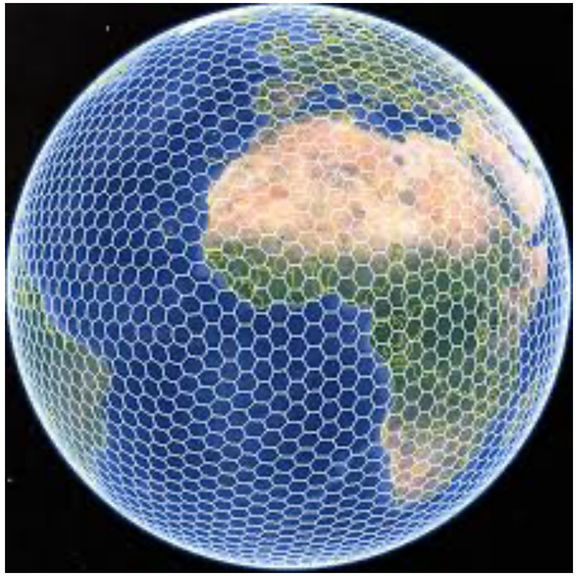



Courtesy of Oliver Fuhrer, CSCS

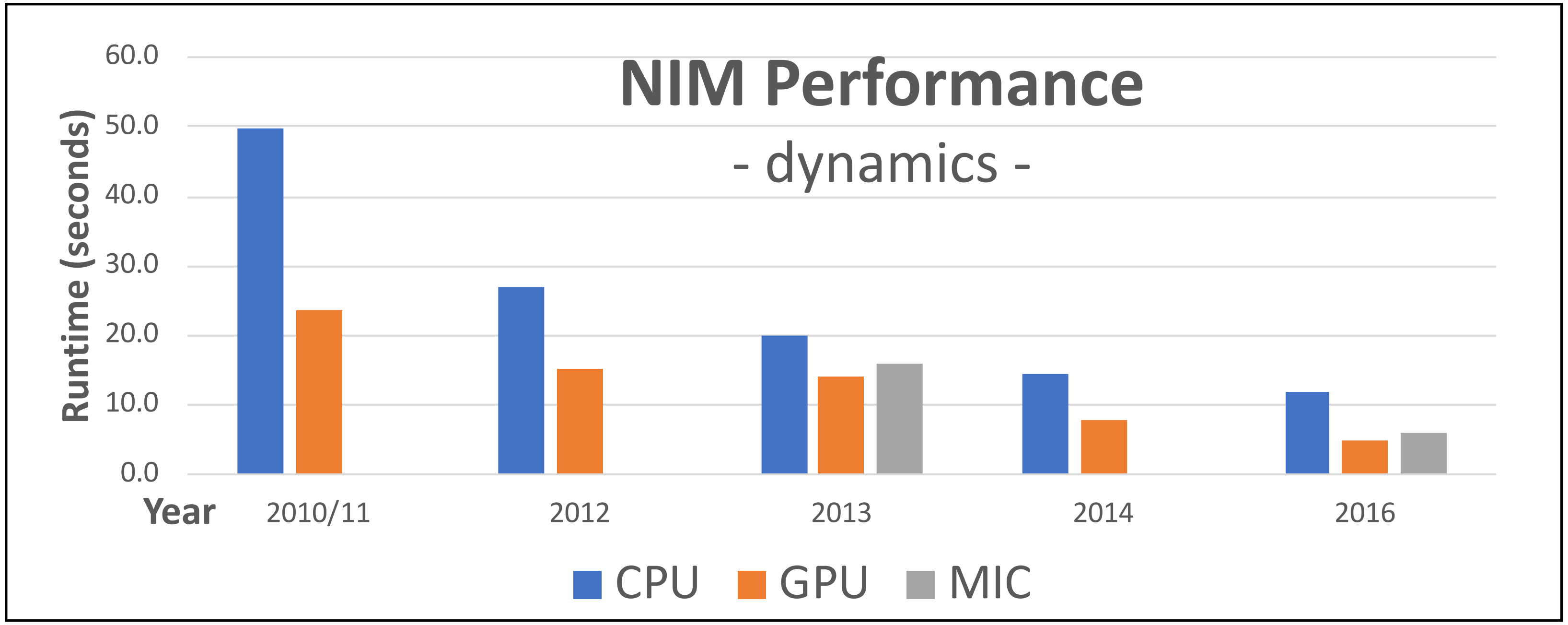
# Application Performance – Single Node

Research model developed by NOAA ESRL/GSD (2010—2016)

- **Directive-based (OpenACC, OpenMP, SMS), performance portable**
- GPU is 2-3 times faster than CPU (Fermi to Pascal generation GPUs)



Uniform Icosahedral Grid




M.Govett, et. al., Parallelization and Performance of the NIM Weather Model on CPU, GPU and MIC Processors, BAMS, October 2017

# GPU Parallelization (2016)

- Increased parallelism needed for GPU
  - Push vertical “k” dimension into routines

**Original: I – J**

```
do k = 1, npz
 call c_sw(a(:,:,k), )
enddo

subroutine c_sw (a, )
 real a(is:ie,js:je)

 do j
  do i
```

**Transformed: I – J – K**

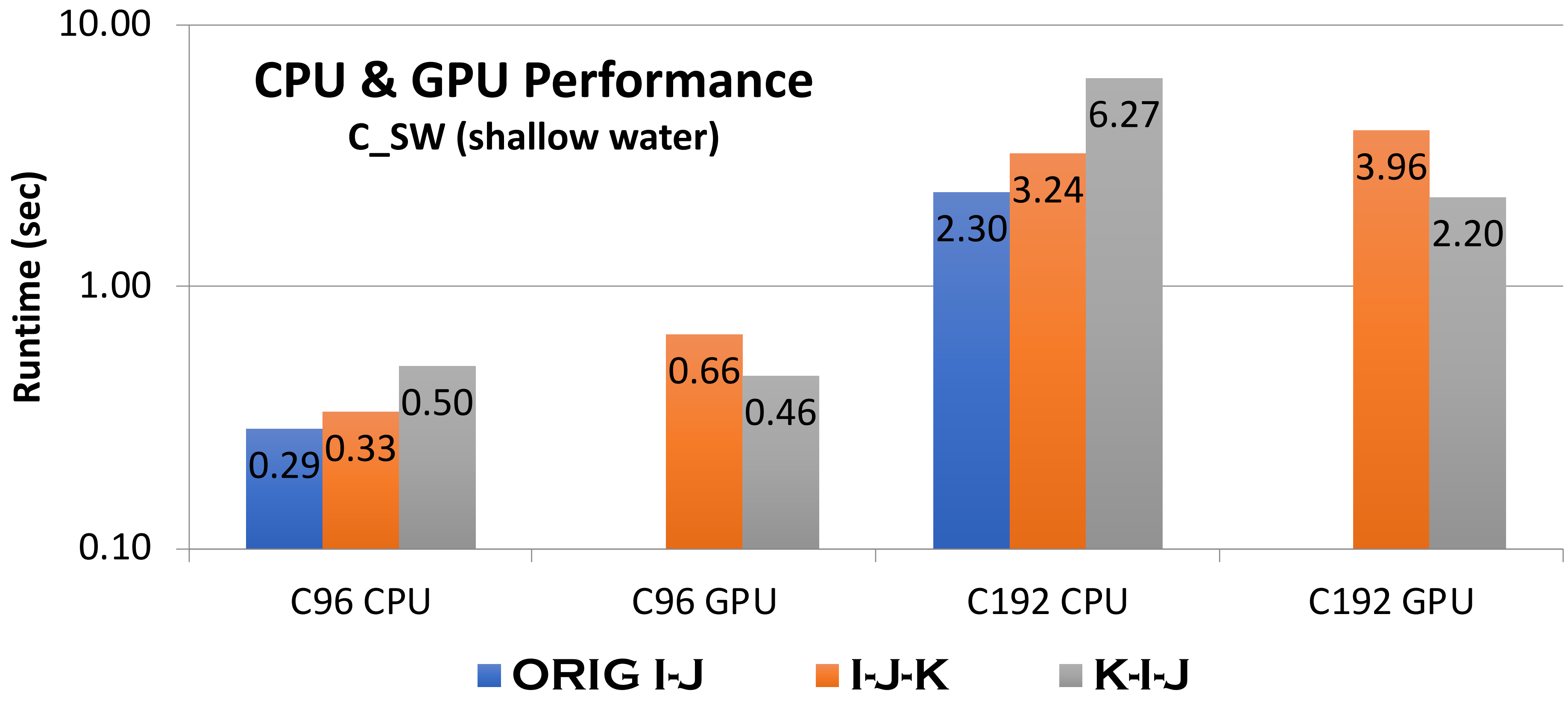
```
call c_sw(a(:,:,k), )

subroutine c_sw (a, )
 real a(is:ie,js:je,npz)

 do k
  do j
   do i
```

GPU Performance

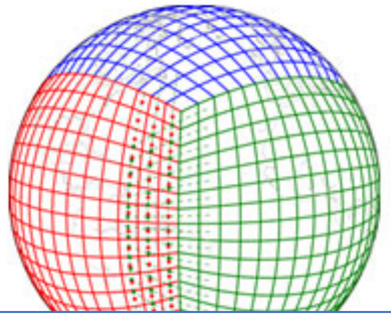
| Subroutine | I-J Runtime (sec) | I-J-K Runtime(sec) |
|---|---|---|
| c_sw | 105.3 | 0.81 |

# FV3 Performance (2016)

- CPU performance is best with I-J-K loops
- GPU performance is best with K-I-J loops

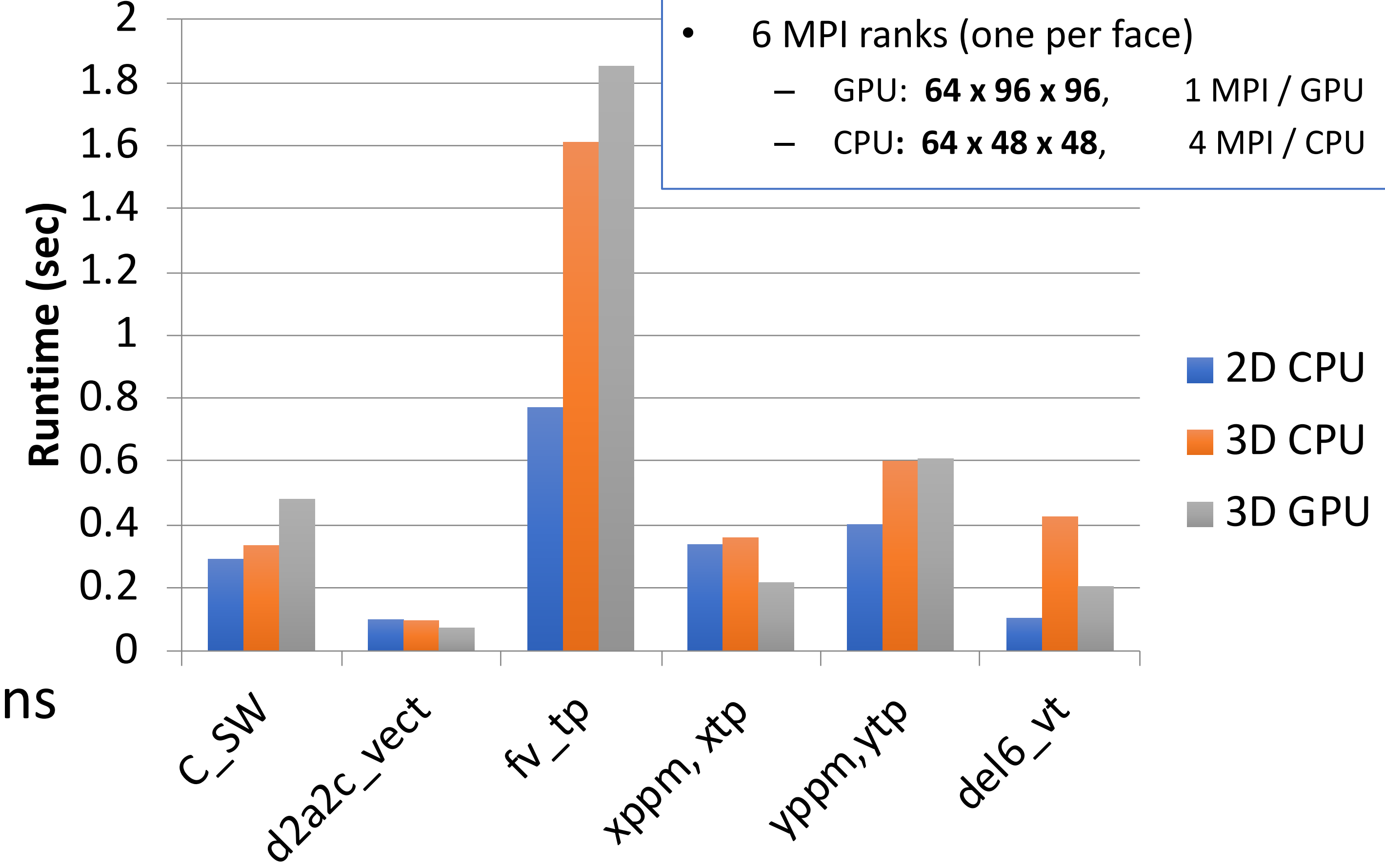


# FV3GFS GPU Performance (2017)

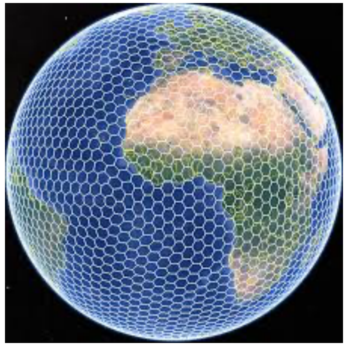


- Finite-Volume Cube-Sphere Model
  - Designed for CPU
  - Efficient use of cache memory
- Slower on GPU
  - Code changes slowed down CPU
  - Not performance portable
- Speculated inefficiencies
  - Limited parallelism on GPU
  - Non-uniform cube-sphere grid
  - Pervasive edge & corner calculations

- 6 MPI ranks (one per face)
  - GPU: **64 x 96 x 96**, 1 MPI / GPU
  - CPU: **64 x 48 x 48**, 4 MPI / CPU

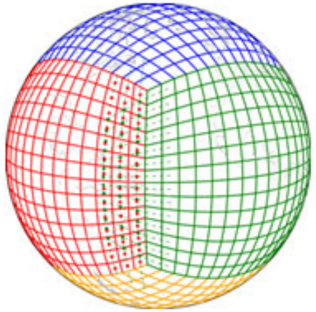


# Conclusions - Dycore Performance (2017)



NIM

- Performance portable to CPU, GPU with same source code
  - 3X faster on GPU than CPU
- i-j-k loop ordering enables portability
- Uniform grid exposed more parallelism for GPU
- Adopted by MPAS



FV3

- Designed for CPU, effective cache utilization k, i-j
- Parallelization to help GPU detrimental to CPU performance
- Performance portability is not possible using openACC
- Major changes would require approval by modeling team
- Unclear if major changes would give major improvement

# Final thoughts

- Performance portability remains a big challenging
  - Computing technologies are becoming increasingly diverse
- Simpler is always better
  - Easier to understand, develop, debug, maintain
- Computational and modeling teams need to be fully engaged
  - Agree on scientific accuracy, performance, portability
  - Software design increasingly important
- Running the UFS at cloud resolving scales will require exascale HPC to achieve an operational capability
  - Vulcan, NOAA, NASA efforts important to achieving this goal