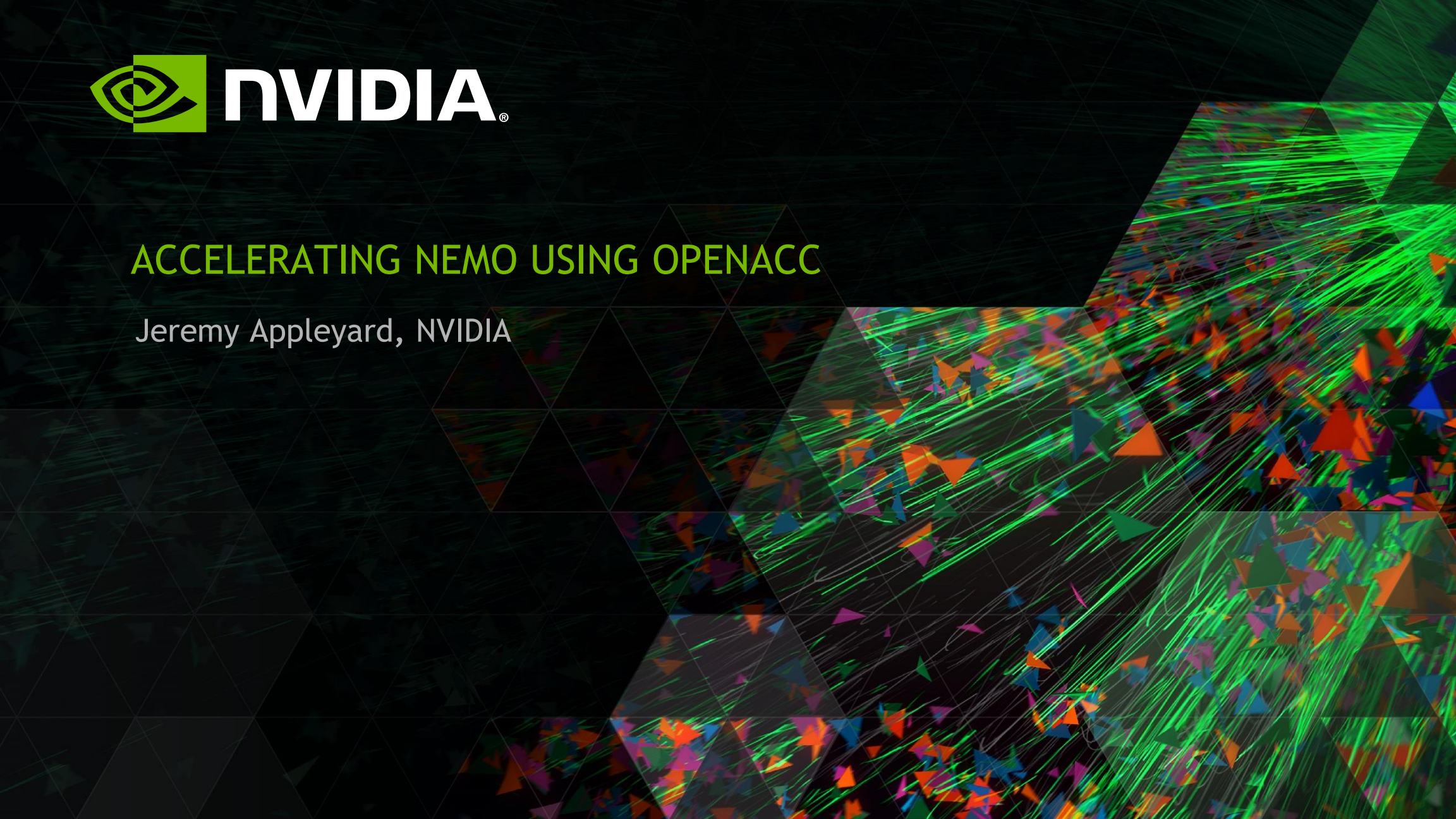


ACCELERATING NEMO USING OPENACC

Jeremy Appleyard, NVIDIA

NEMO

What is it?

- **N**ucleus for **E**uropean **M**odelling of the **O**cean
- 5 major components
  - Blue Ocean
  - White Ocean
  - Green Ocean
  - Adaptive mesh refinement
  - Assimilation

NEMO

What is it?

- **N**ucleus for **E**uropean **M**odelling of the **O**cean
- 5 major components
  - **Blue Ocean**
  - **White Ocean**
  - Green Ocean
  - Adaptive mesh refinement
  - Assimilation

NEMO

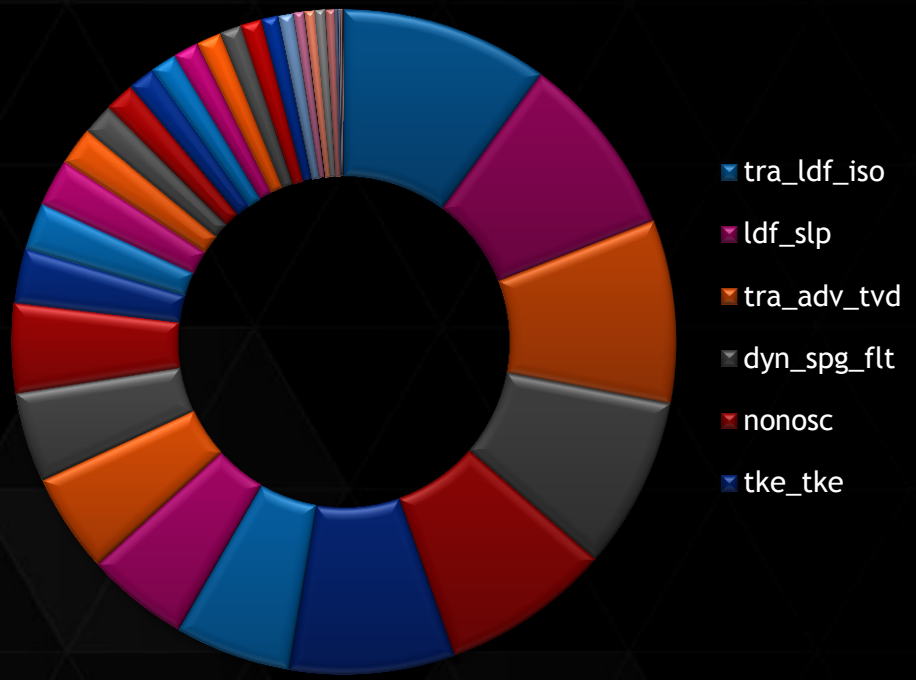
Vital Statistics

- Available with a public license
  - www.nemo-ocean.eu
- Used by “240 projects in 27 countries”
- 170,000 lines of FORTRAN 90
- Mostly small stencil/single element calculations

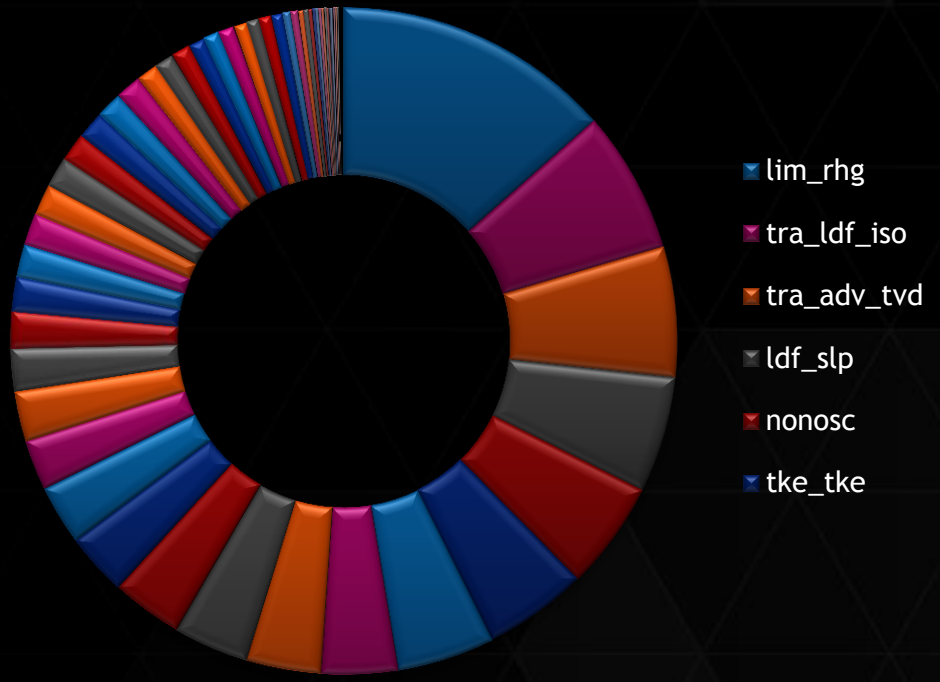
CODE PROFILE

Very Flat

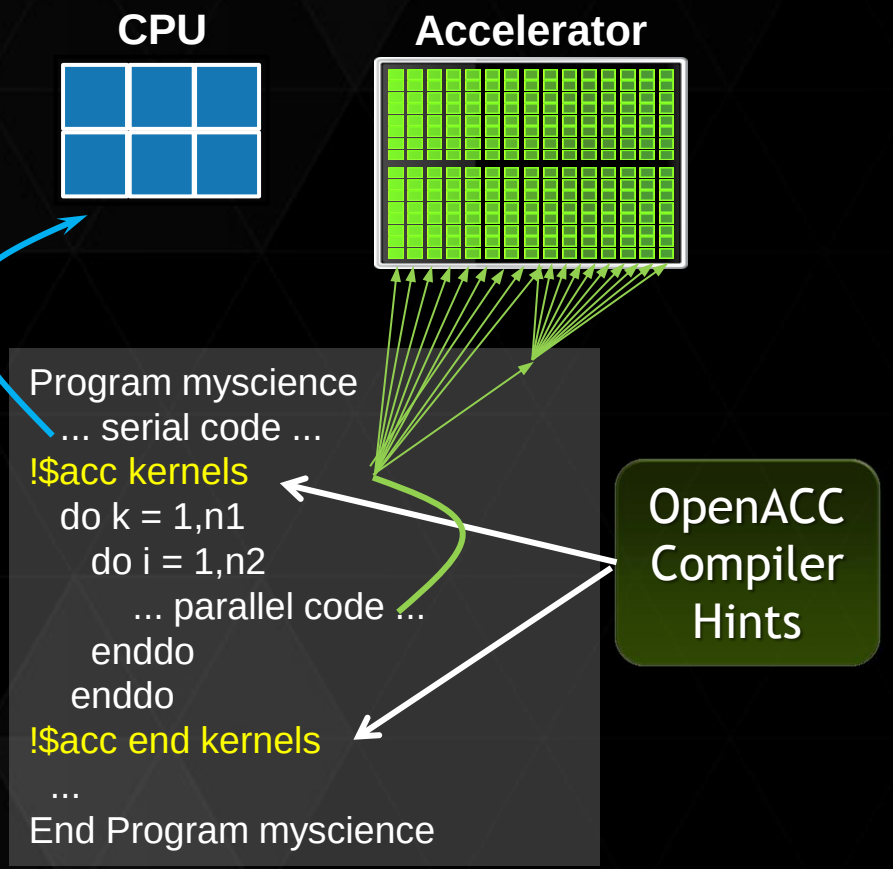
GYRE25



ORCA025



OPENACC DIRECTIVES



Familiar

Insert compiler hints into Fortran & C code
Preserves legacy code portability

Powerful

Compiler parallelizes code with less developer effort

Open

Supports accelerators from NVIDIA, AMD, Intel

Open specification driven by HPC industry vendors

OPENACC

Natural Solution

- Directives are a good solution for flat profile
- No massive gains to be made in any single place
- More maintainable, quicker to implement

NEMO: ADDING OPENACC

Minimal Impact

- Almost entirely insertion of directives
- Small changes to MPI routines:
  - Multiple sequential calls “batched”
  - Also beneficial to CPU code
- Code runs on CPU if compiled without OpenACC
  - As if no changes have been made

NEMO: ADDING OPENACC

Challenges

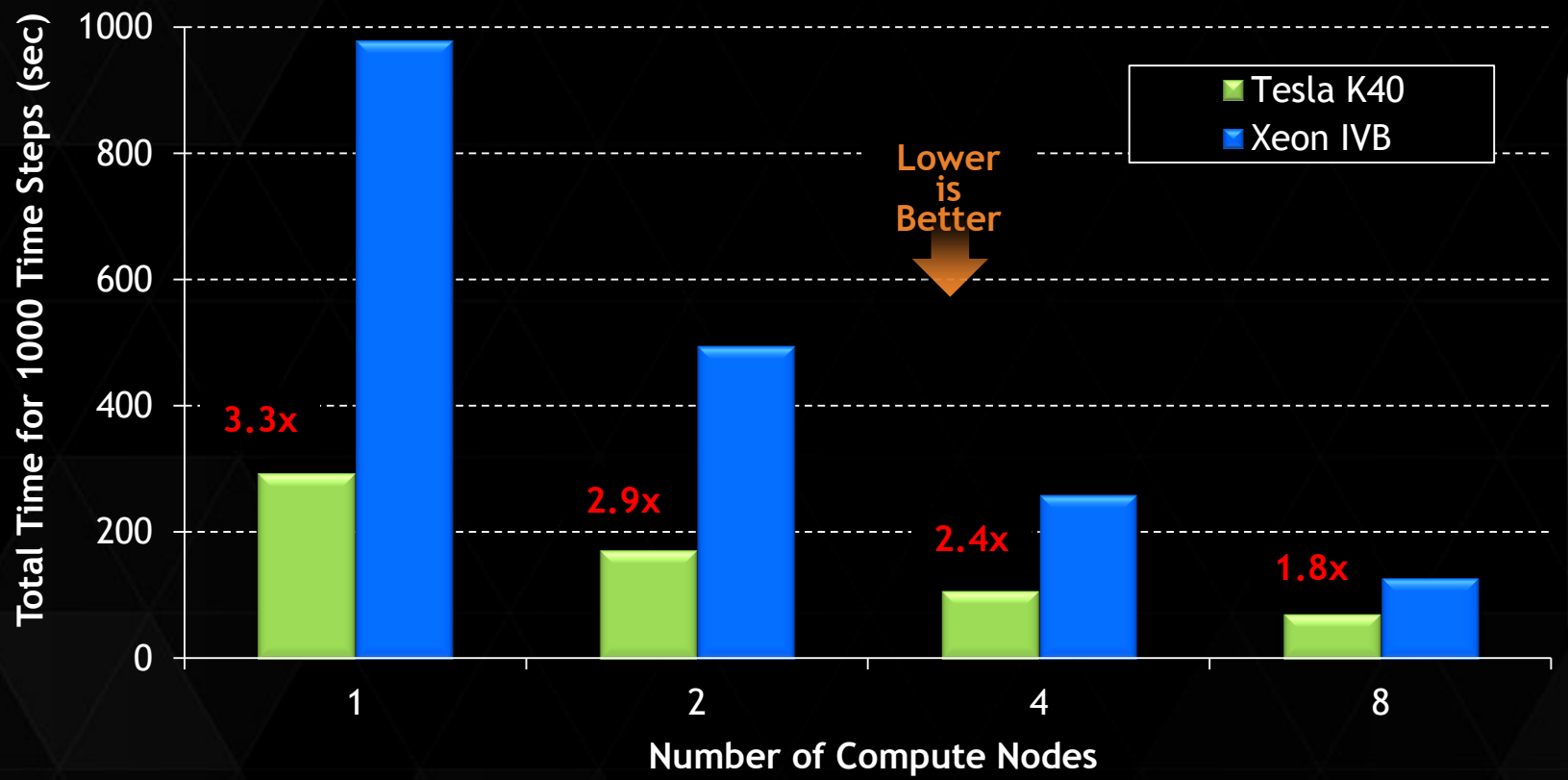
- Memory heterogeniaility
  - Not a complete port
  - Code can update a CPU version of the data without updating GPU version
  - Hard to debug
- Solution?
  - Rely on implicit copies and incrementally enable data regions (pcopy)
  - Throw errors in known incomplete areas

TEST CASE 1

GYRE25

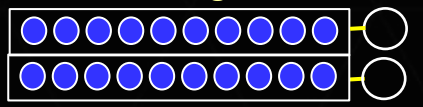
- Idealised **Blue Ocean** test case
- Used for simple benchmarks/validation
- Equivalent to global ½° horizontal resolution (752x502)
- 31 vertical levels
- Approximately 7GB RAM required

GYRE25 TIMING

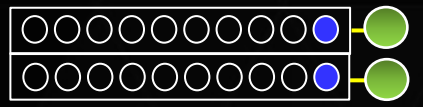


Node utilization:
2 x IVB + 2 x K40

Without using GPUs



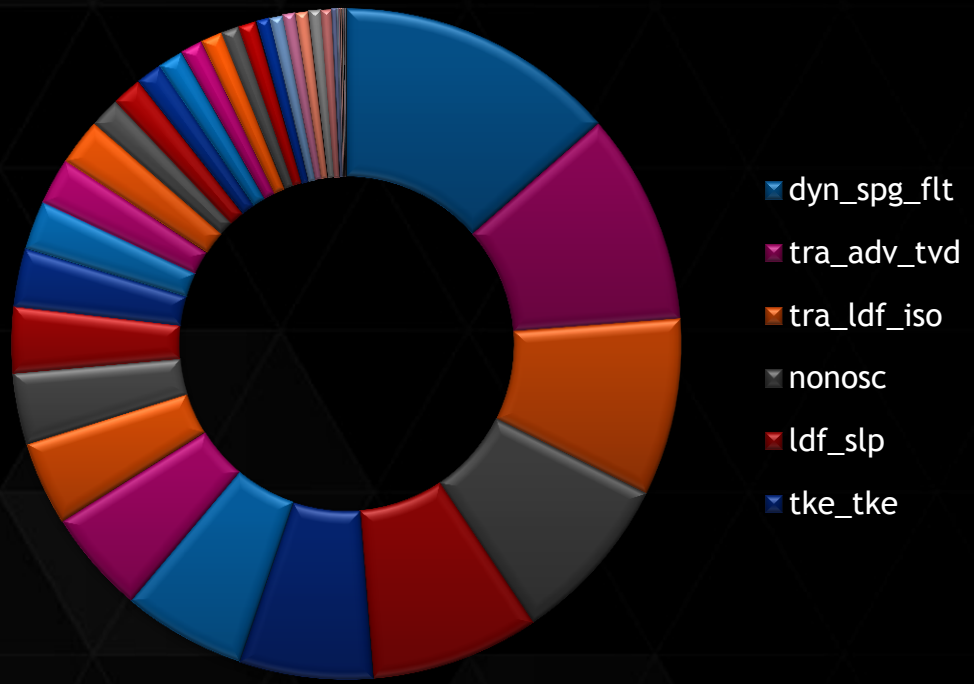
Use of GPUs



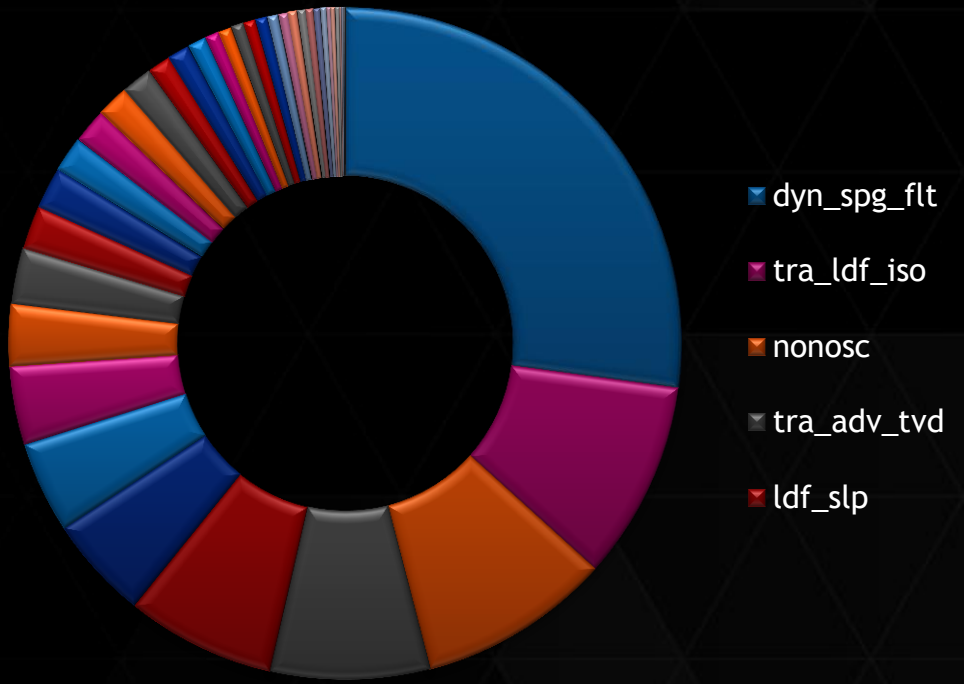
STRONG SCALING PROFILE

8 Nodes

CPU Strong Scaling



GPU Strong Scaling



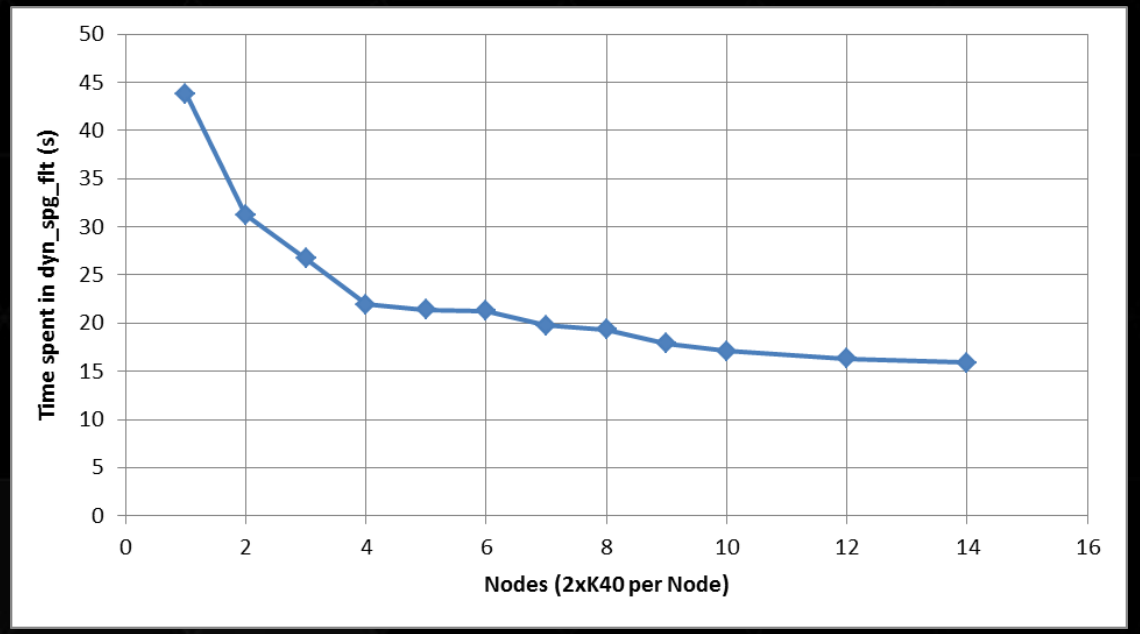
STRONG SCALING

What is the limiter?

- Linear solver: SOR method
  - Lots of communication. Lots of very small kernels
- sol_sor iterates 200 times per time step
  - 1000 time steps => 200,000 calls.
  - ~6 kernels/call => 1,200,000 kernel launches
  - ~10μs latency => ~12 seconds + MPI latency

STRONG SCALING

Solver performance



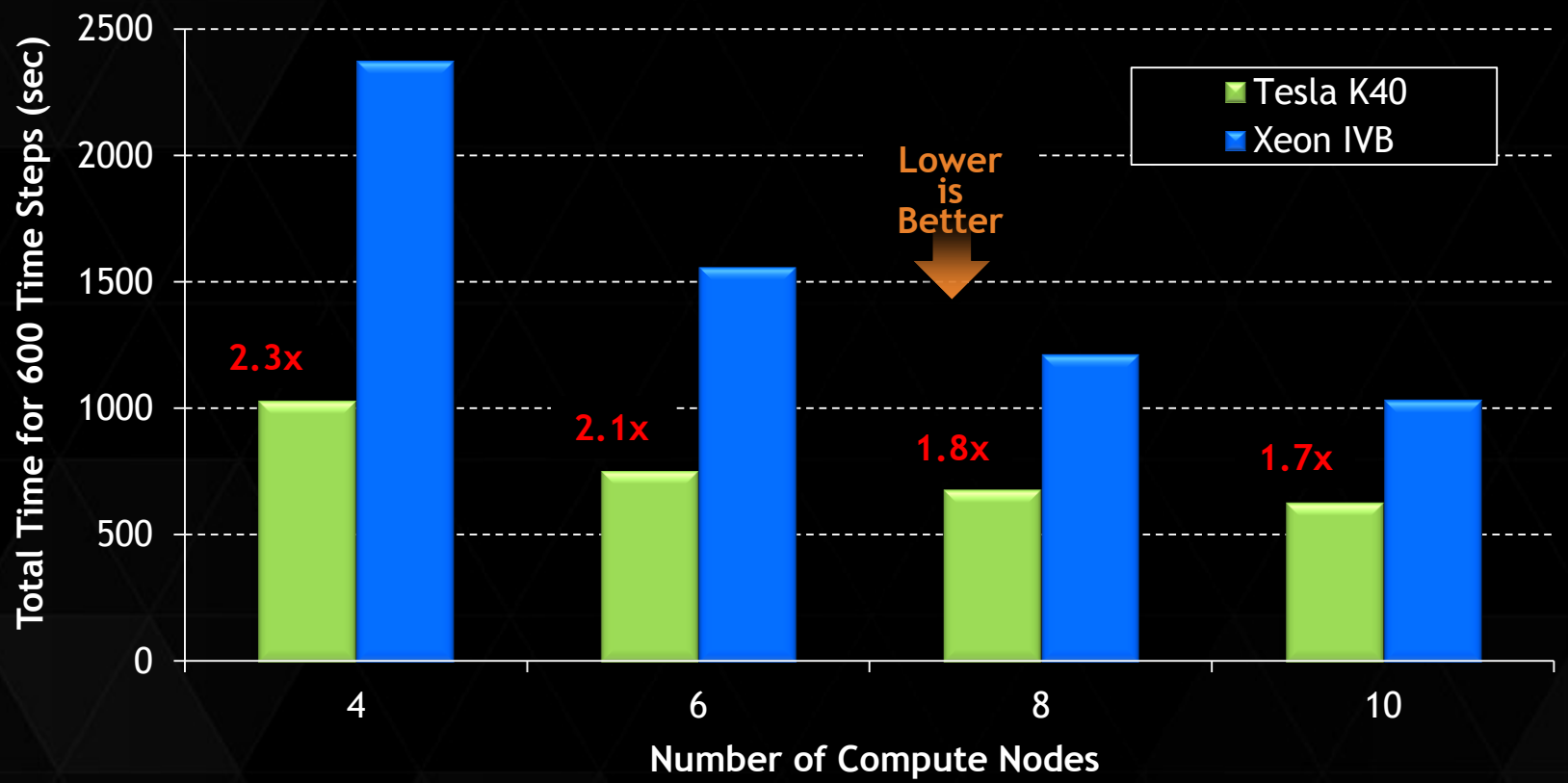
- Solver does not scale well on GPU
- Solution?
  - More efficient MPI packing/unpacking
  - Reduced solver communications
  - Not implemented yet

TEST CASE 2

ORCA025

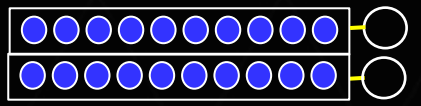
- Builds on GYRE. **Blue Ocean** and **White Ocean**
- LIM2 Ice Model
- Regular horizontal grid
  - Global ¼° horizontal resolution (1442x1021)
- Variable vertical grid
  - 75 vertical levels
- ~90 GB RAM

ORCA025 TIMING

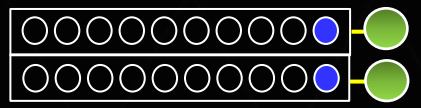


Node utilization:
2 x IVB + 2 x K40

Without using GPUs



Use of GPUs

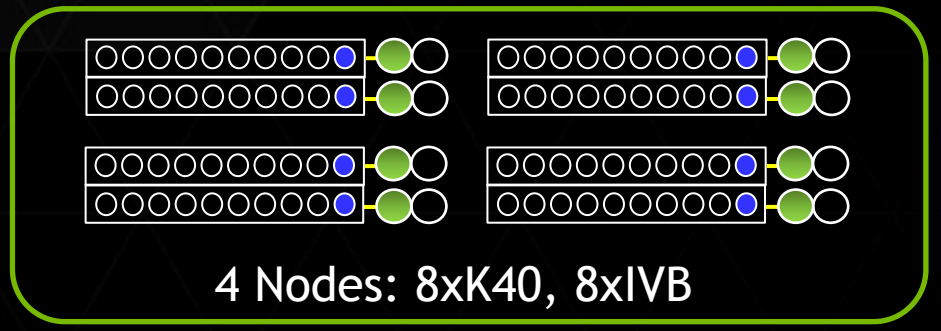


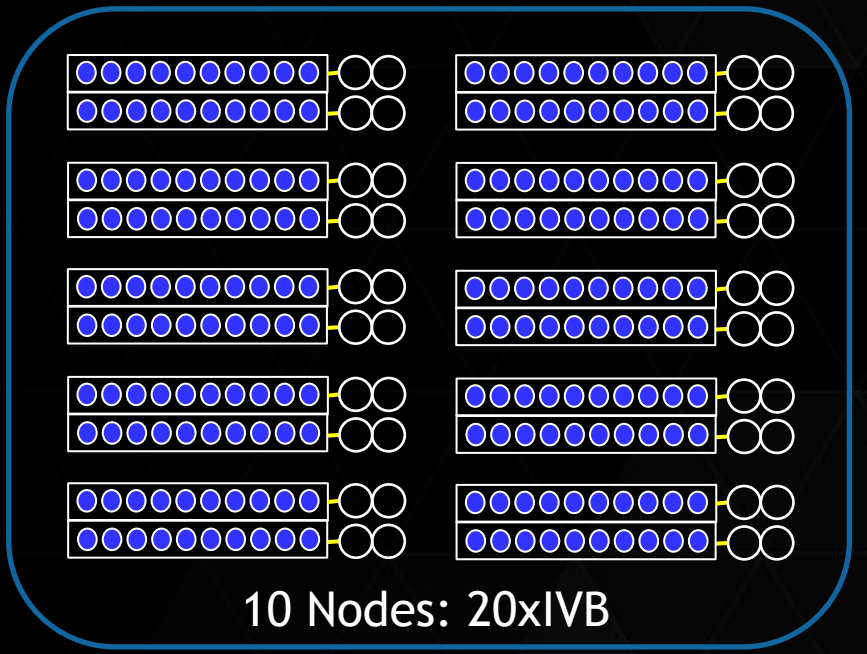
ORCA025 settings:

- **Output every 5 days**
- **Total run: 10 days**
- **Time steps: 600**

NODE CONFIGURATIONS

Configurations give similar performance

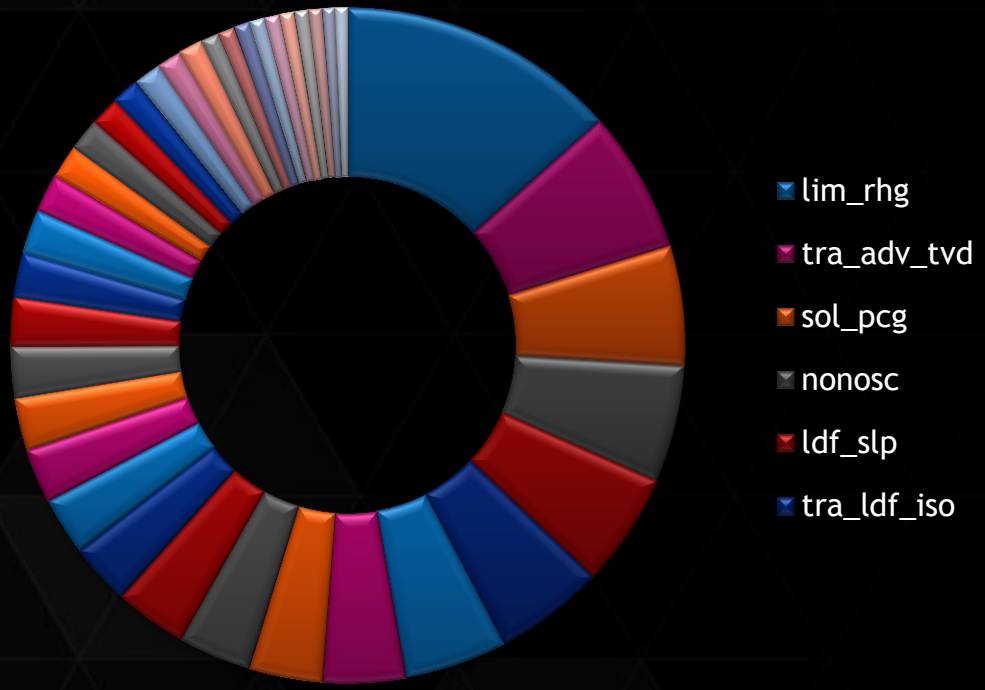




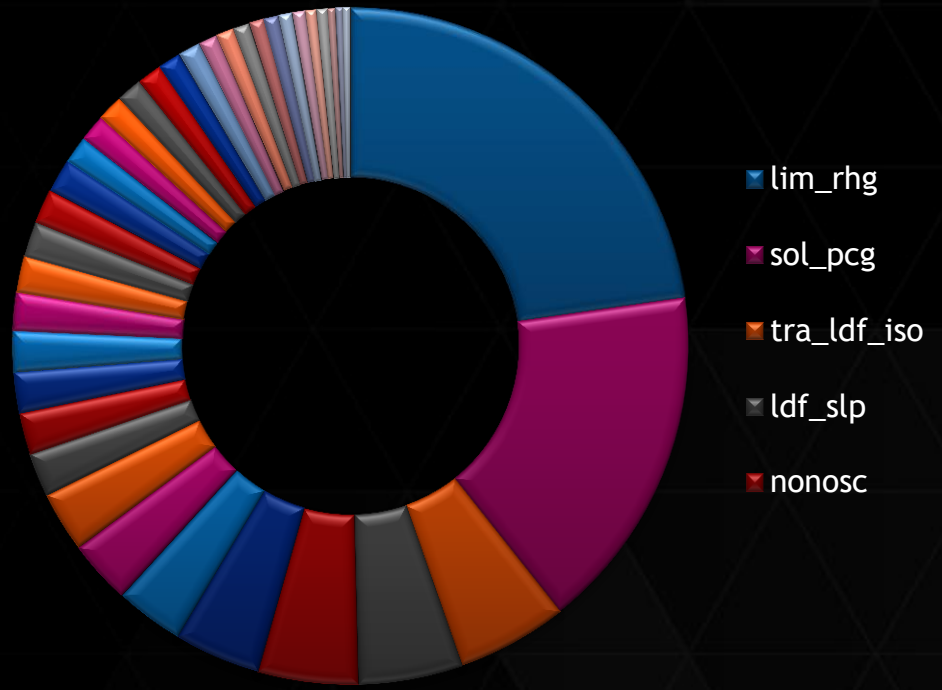
STRONG SCALING PROFILE

8 Nodes

CPU Strong Scaling



GPU Strong Scaling



PERFORMANCE

Strong Scaling

- As with GYRE, strong scaling on the GPU needs to be improved
- Same problems as before:
  - Too many communications
  - Inefficient use of the GPU in these communications

PERFORMANCE

Codebase

- OpenACC not heavily tuned:
  - No worker/vector/gang clauses
  - No restructuring of the original source
- Compiler-side optimisations still possible
  - Loop fusion (worth ~10%)
  - Shared memory

NEXT STEPS

NEMO 3.6

- Replaces LIM2 ice model with LIM3 ice model
  - Better physics
  - Fewer communications
- Streamlined MPI
- Several other fixes/improvements

NEXT STEPS

NEMO 3.6

- Replaces LIM2 ice model with LIM3 ice model
  - Better physics
  - Fewer communications
- Streamlined MPI
- Several other fixes/improvements

NEXT STEPS

Beyond NEMO 3.6

- Possible changes to structure to improve MPI comms
  - Larger halo allowing for reduced comms
  - Reordering of MPI routines to improving packing/unpacking performance
- In isolation: 15-20% application speedup on 8xK40 case
  - Projected increase from 2.3x to 2.7x speedup vs 8xIVB
  - Significantly better strong scaling performance

LONGER TERM

GOcean

- Collaboration between STFC, NERC and University of Manchester
- Evaluating GungHo separation of concerns for NEMO
  - Separate Kernel, Physics and Algorithm layer
- 4-5 years from now NEMO may look quite different
  - Directives probably still the solution

THANK YOU

Questions?

Jeremy Appleyard, NVIDIA - jappleyard@nvidia.com

With thanks to:

Mike Ashworth (STFC, UK)

Andrew Coward (NOC, UK)