FESOM2 - on the road to ExaScale

Natalja Rakowsky, Sven Harig, Vadym Aizinger,
Annika Fuchs, Stephan Frickenhaus, . . .

Computing Center, HPC & Data Processing,
Alfred Wegener Institute for Polar and Marine Research, Bremerhaven

FESOM Days, 7-8 December 2020

Looking back

Linear solvers

- History: Experience with other models, FoSSI
- From Petsc to pARMS, with own routines added (flavours of BiCGstab)

MPI: Domain decomposition, halo exchange

- load balancing: 2D and 3D nodes as Metis weights
- MPI datatypes instead of explicit buffers
- combine halo exchange of several variables
- asynchronous: exchange_halo_begin, compute, exchange_halo_end
- init routines: much faster, less memory, serial
- hierarchical partitioning (switches $\rightarrow$ nodes $\rightarrow$ sockets $\rightarrow$ cores)

Single processor optimization

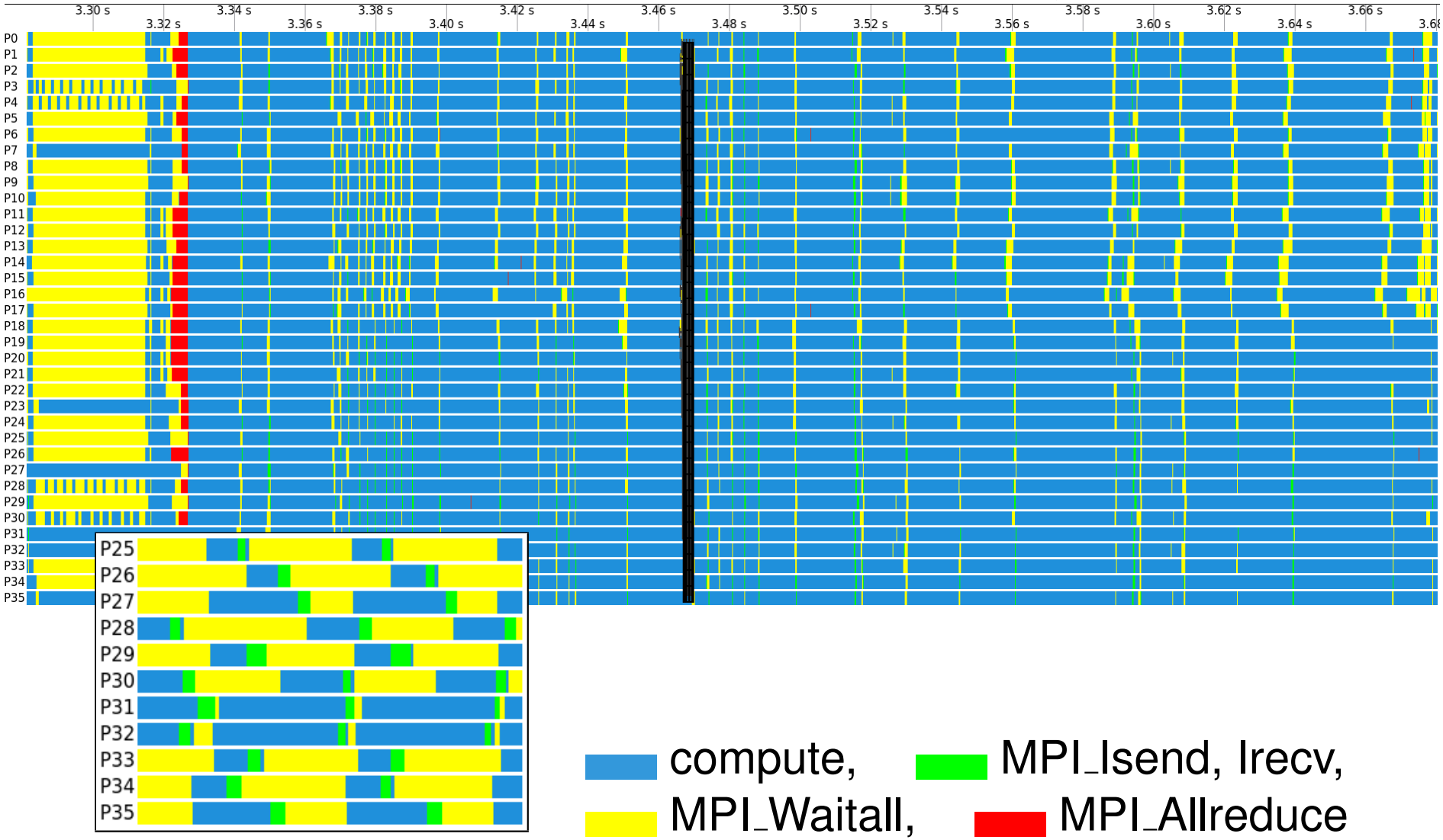
- vectorization: vertical as inner loop is crucial
- FESOM2 is memory bound: fuse loops
- loops: precompute constants

FESOM2 timestep

- **2D, iterative** Sea-ice dynamics
- **3D** Horizontal velocity
- **2D, iterative** Solve for sea surface height
- **3D** Horizontal velocity correction
- **3D** ALE (adjust vertical layers) and vertical velocity step
- **3D** Tracer advection and diffusion
- **Output** Diagnostics, restart - in some steps

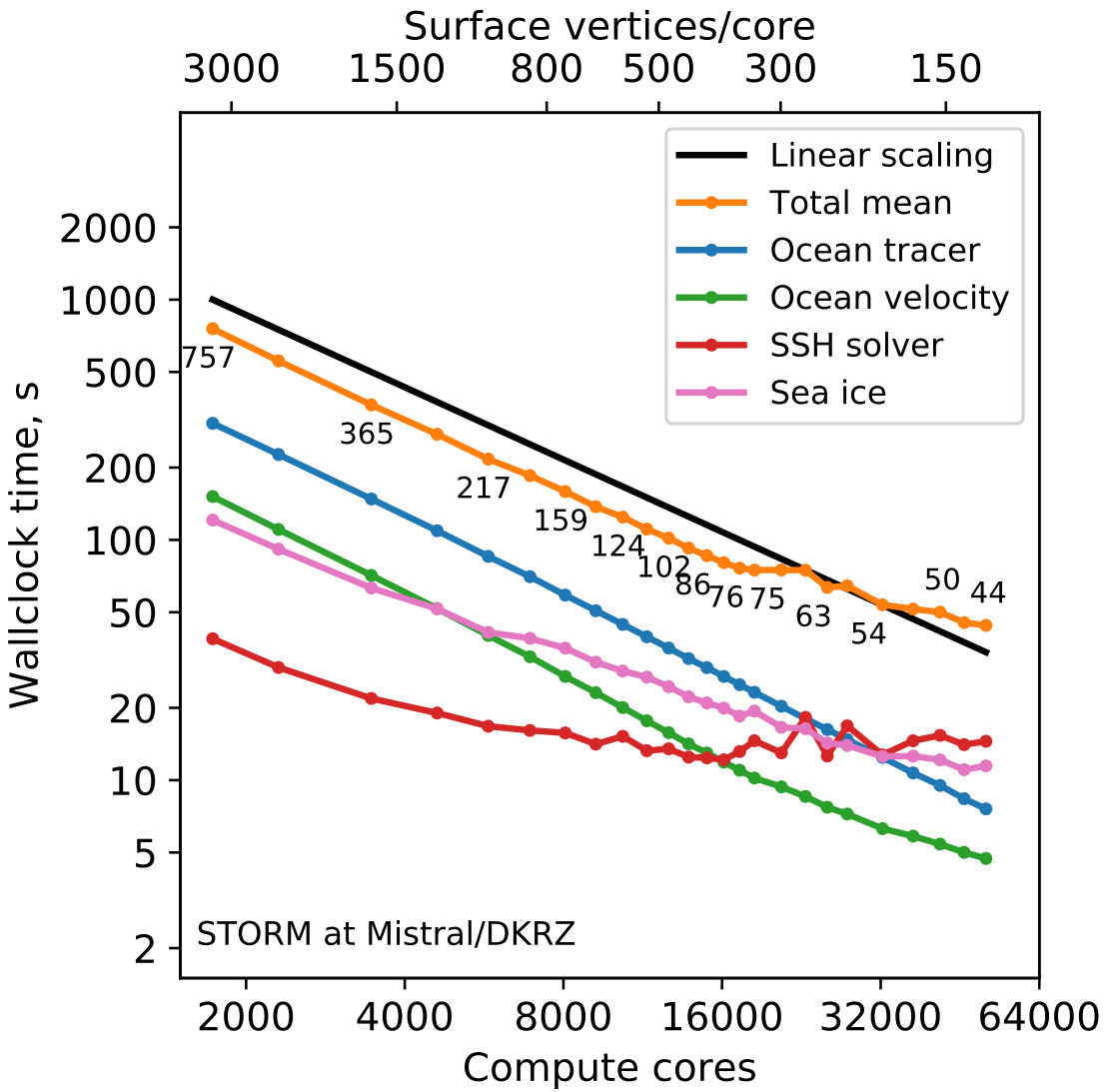
Timestep in Intel Trace Analyzer

Core2 mesh, one Ollie node, 2×18 cores Broadwell, 36 MPI tasks



HELMHOLTZ

FESOM2 Scaling for STORM mesh



Wallclock time

timesteps 1-1800

STORM (2D: 5.6M vert.)

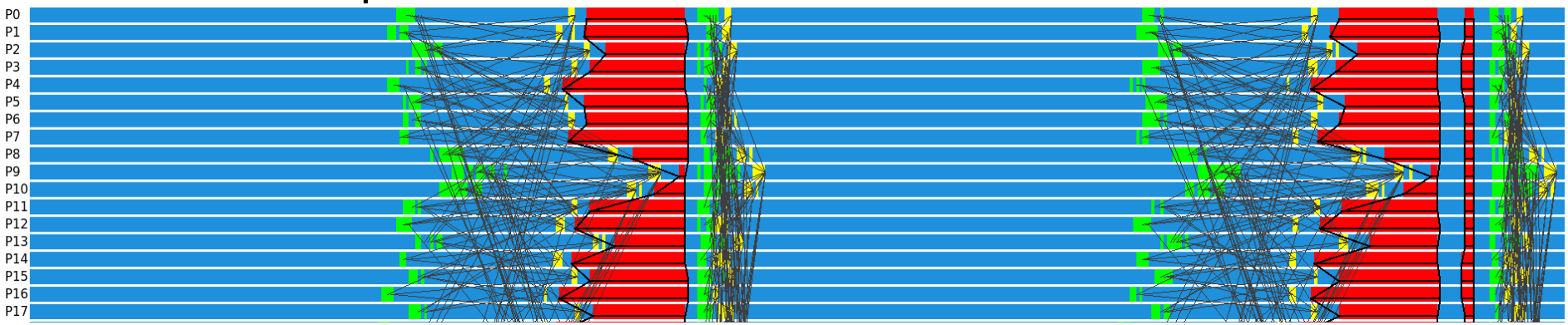
on Mistral

w/o output

HELMHOLTZ

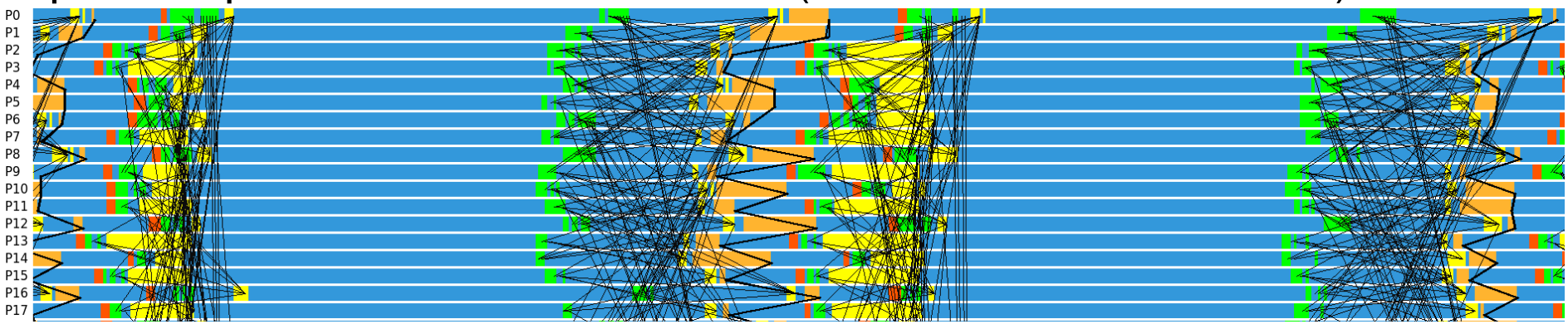
FESOM2 Scaling for STORM mesh

One iteration of preconditioned BiCGstab



compute, MPI_Isend, Irecv, MPI_Waitall, MPI_Allreduce

Pipelined preconditioned BiCGstab (Cools & Vanroose 2016)



MPI_Iallreduce, MPI_Wait for Iallreduce

HELMHOLTZ

ESiWACE2 Service-1: Port to GPUs

Intermediate Results

Gijs van den Oord, eScience-Center Amsterdam, and Atos.

- Tracer ported to CUDA with GCC
  Each loop: tuned kernel $\rightarrow$ performant, but invasive
- Tracer ported to OpenACC with NVIDIA (former PGI)
  Each loop: OpenACC directives $\rightarrow$ fast to develop
- Copy mesh information at init,
  time-dependent fields async. before tracer start
- Low compute complexity of kernels $\rightarrow$ no free lunch
- Irregular 2D access patterns, but regular accesses in vertical
  $\rightarrow$ kernels close to peak bandwidth

ESiWACE2 Service-1: Port to GPUs

Intermediate Results, Timings for Tracer

Core2 (126,000 2D vert.)

Baseline: Intel	603s	on 16 Core Sandy Bridge
GCC + CUDA	518s	on 2x NVIDIA K40
PGI + OpenACC	689s	on 2x NVIDIA K40

STORM (5.6M 2D vert.)

Baseline: Intel	326s	on 12x 16 Core Sandy Bridge
GCC + CUDA	290s	on 2x NVIDIA K40
PGI + OpenACC	430s	on 2x NVIDIA K40

Tracer account for approx. 40% of full baseline run on CPUs.

Outlook

- Sustain/extend GPU code (CUDA and/or OpenACC)
- Looking forward to test ARM processors (at DKRZ)
- Dwarfs and improved modularization (PilotLab ExaESM)

- One-sided MPI-2
- Resort the inner halo to better overlap MPI and compute
- Elaborate on hierarchical partitioning,
  employ shared memory on nodes/sockets

- DSL
- . . .