www.bsc.es

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Optimizing an Earth Science Atmospheric Application with the OmpSs Programming Model

Workshop on High performance computing in meteorology

George S. Markomanolis, Jesus Labarta, Oriol Jorba

EXCELENCIA SEVERO OCHOA

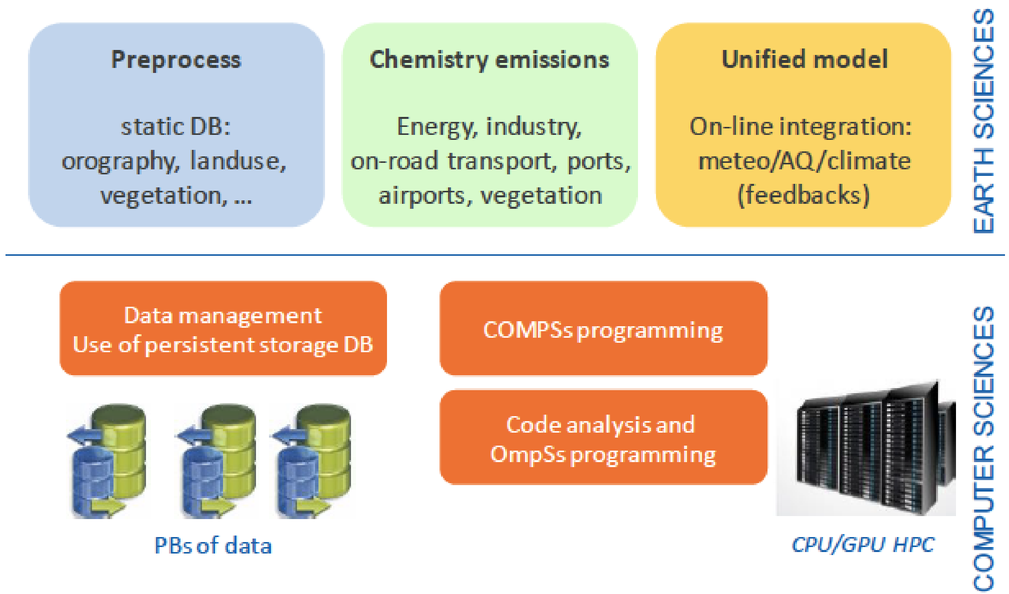
ECMWF, Reading, UK, 31 October 2014

Outline

- Introduction to NMMB/BSC-CTM
- Performance overview of NMMB/BSC-CTM
- Experiences with OmpSs
- Future work

Severo-Ochoa Earth Sciences Application

- Development of a Unified Meteorology/Air Quality/Climate model
  - Towards a global high-resolution system for global to local assessments



- Extending NMMB/BSC-CTM from coarse regional scales to global high-resolution configurations

- Coupling with a Data Assimilation System for Aerosols

- International collaborations:

Meteorology

National Centers for Environmental Predictions

Climate
Global aerosols

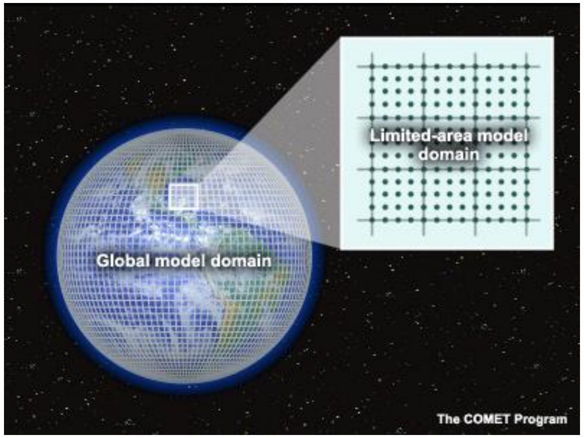
Goddard Institute Space Studies

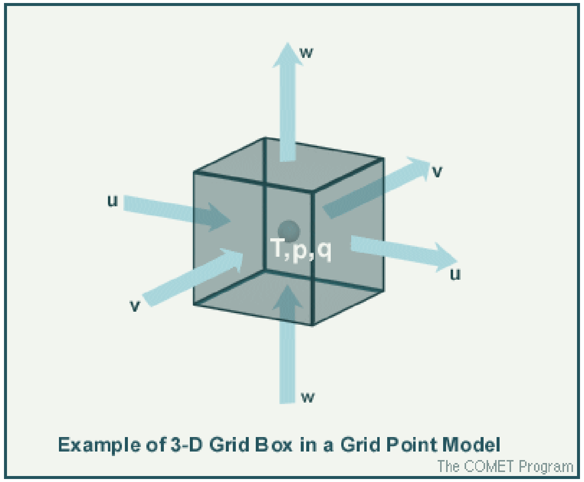
UCIrvine Air Quality

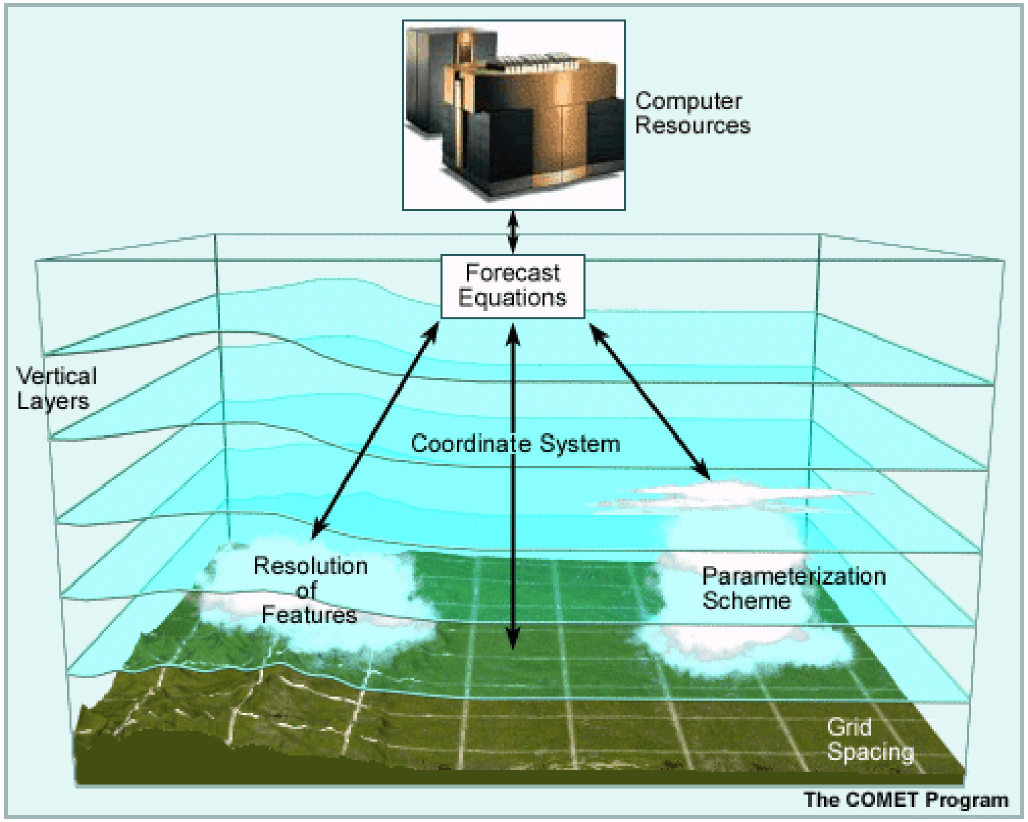
Uni. of California Irvine

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Where do we solve the primitive equations? Grid discretization







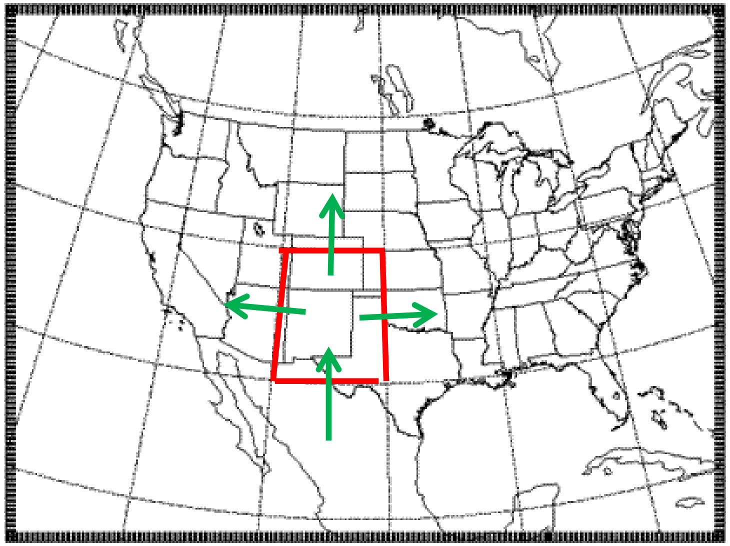
High performance computing resources:
If we plan to solve small scale features we need higher resolution in the mesh and so more HPC resources are required.

Parallelizing Atmospheric Models

- We need to be able to run this models in Multi-core architectures.

- Model domain is decomposed in patches

- Patch: portion of the model domain allocated to a distributed/shared memory node.

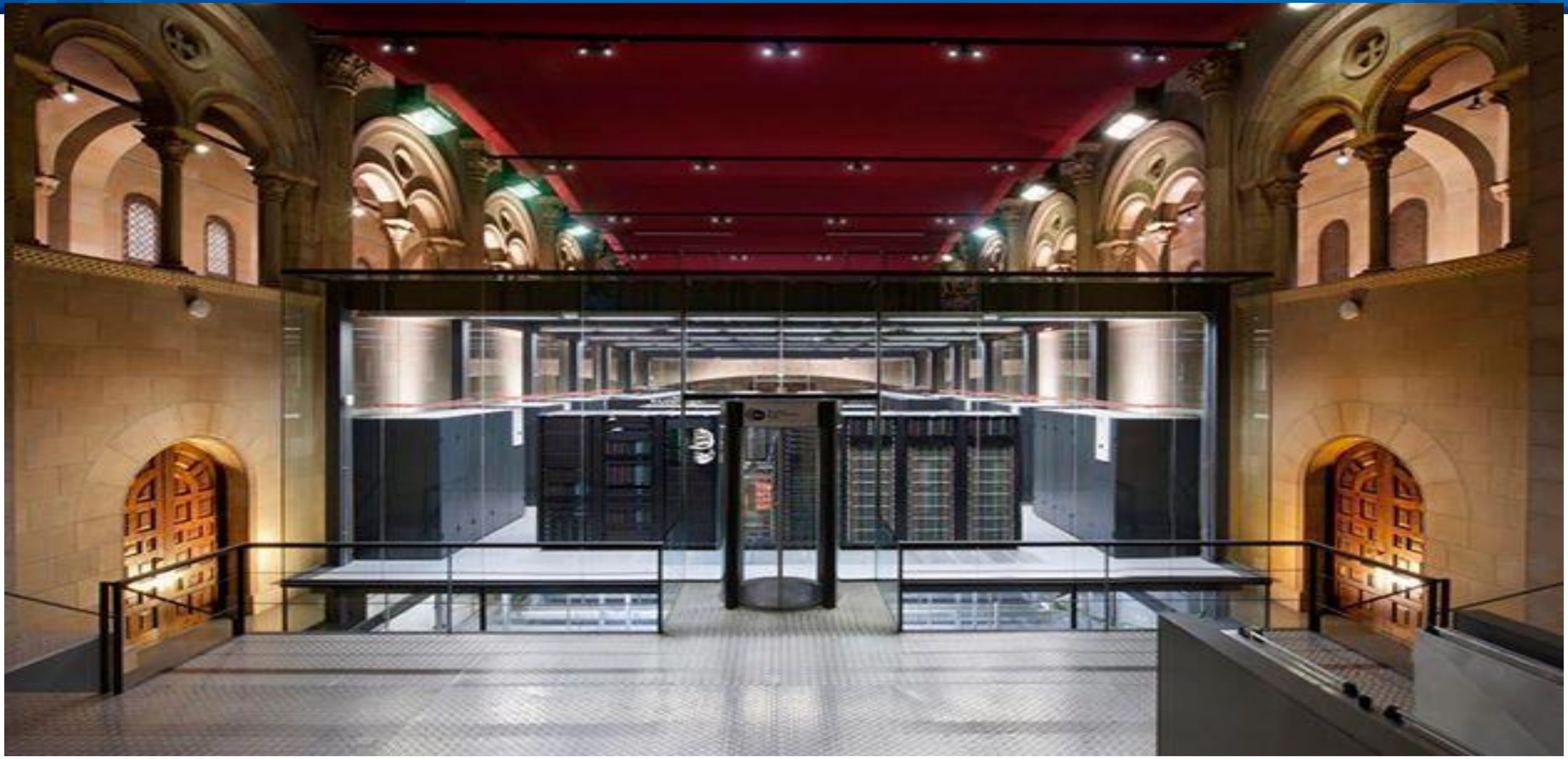


NMMB/BSC-CTM

- NMMB/BSC-CTM is used operationally for the dust forecast center in Barcelona
  - NMMB is the operational model of NCEP

- The general purpose is to improve its scalability and the simulation resolution

MarenoStrum III



- 3,056 compute nodes
- 2x Intel SandyBridge-EP E5-2670 2.6 GHz
- 32 GB memory per node

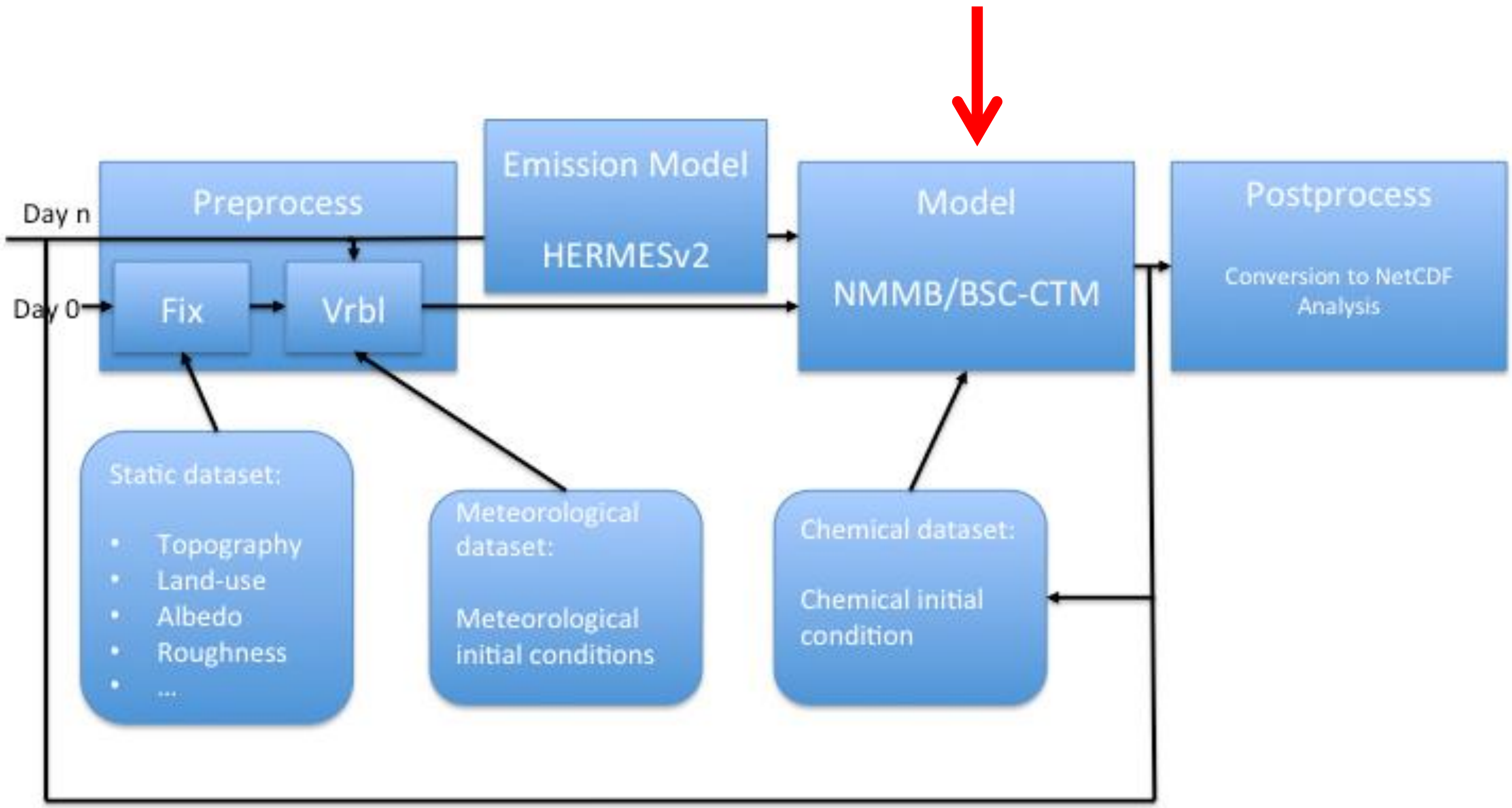
- Infiniband FDR10
- OpenMPI 1.8.1
- ifort 13.0.1

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

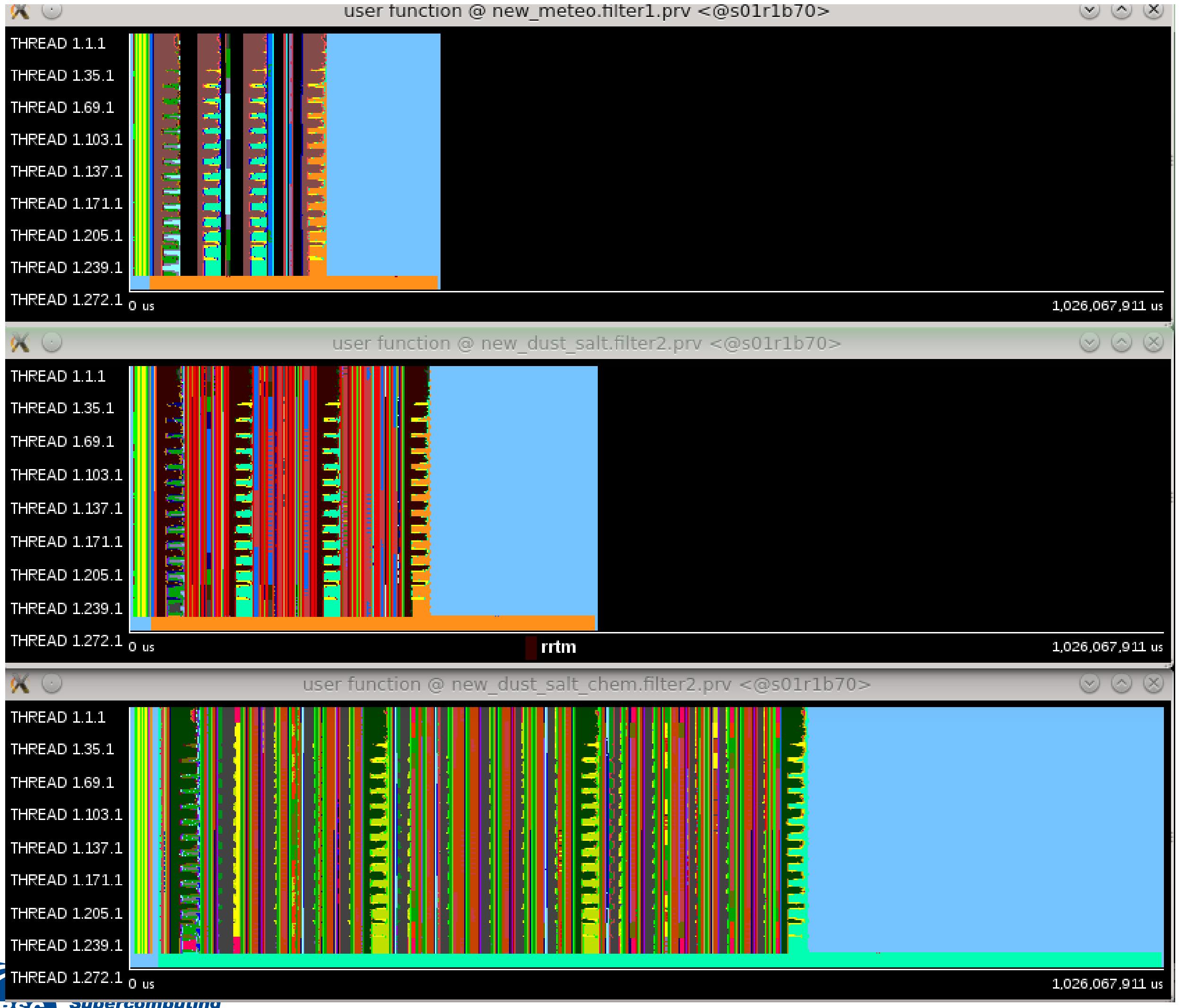
Performance Overview of NMMB/BSC-CTM Model

Execution diagram – Focus on the Model



Paraver

- One hour simulation of NMMB/BSC-CTM, global, 24km, 64 layers



meteo: 9 tracers

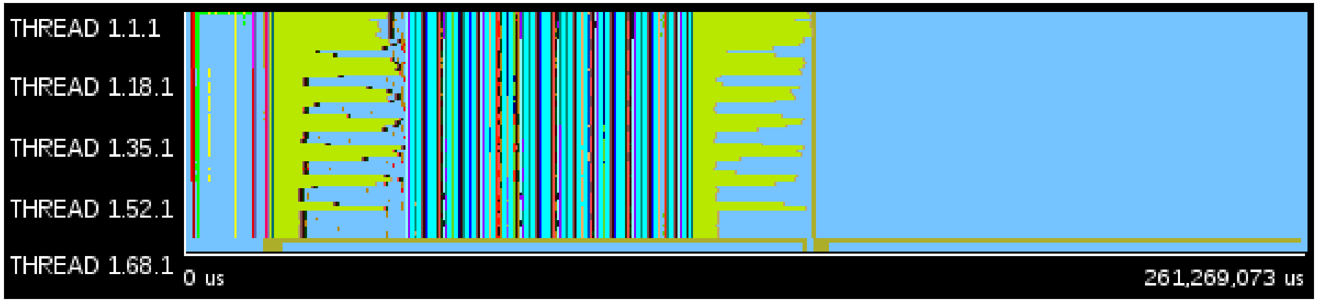
meteo + aerosols:
9 + 16 tracers

meteo + aerosols + gases: 9 + 16 + 53

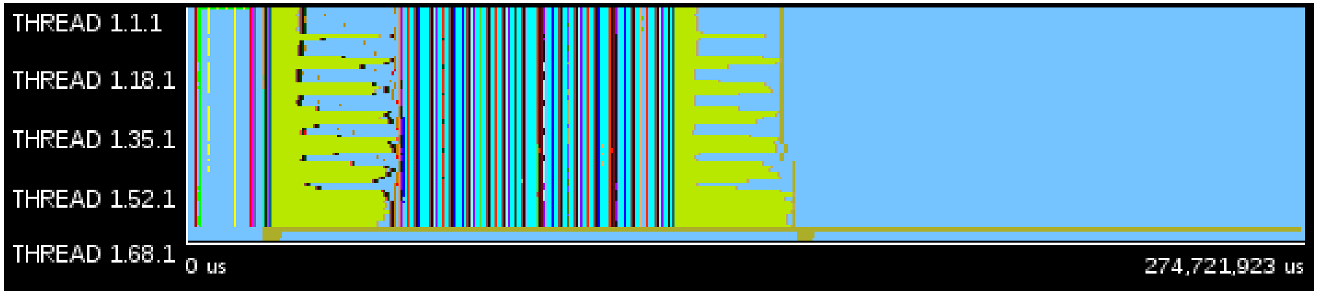
Dynamic load balancing

- Different simulation dates cause different load balance issue (useful functions, global 24km meteo configuration)

20/12/2005



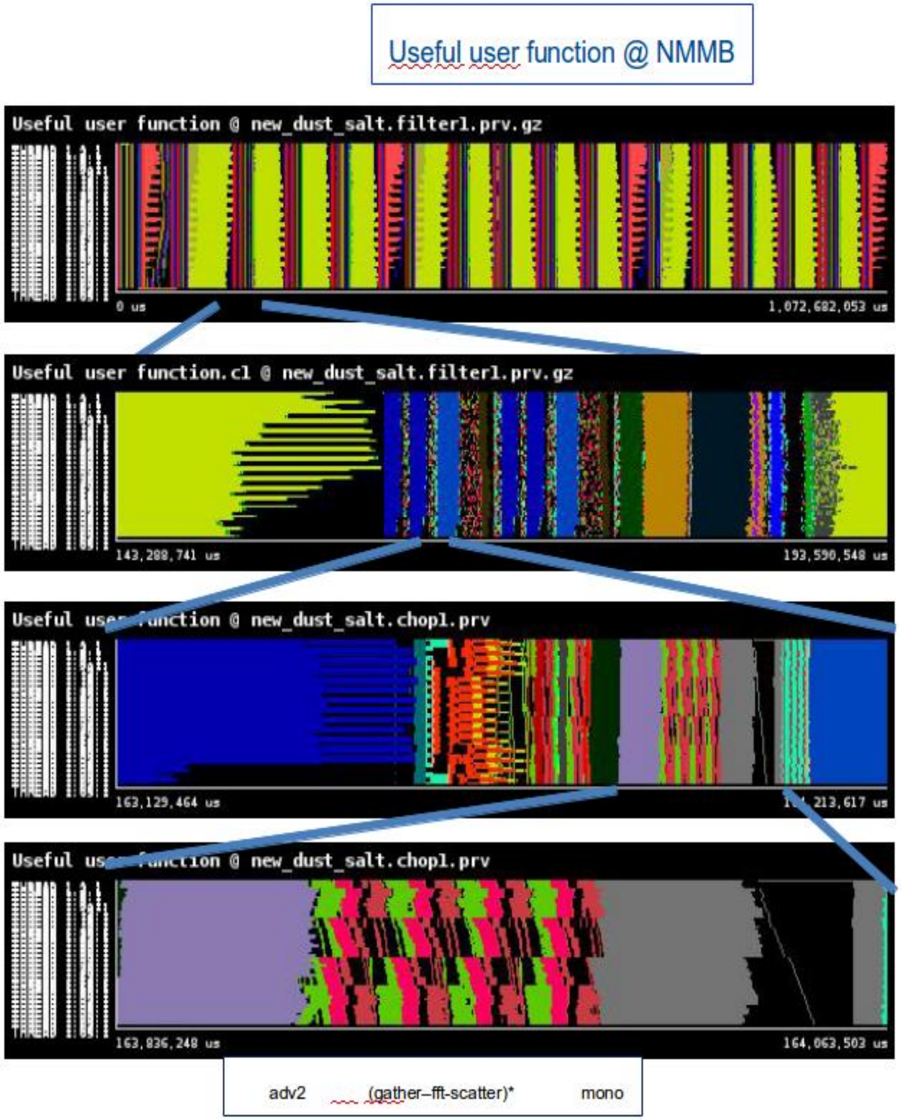
20/07/2005



A "different" view point

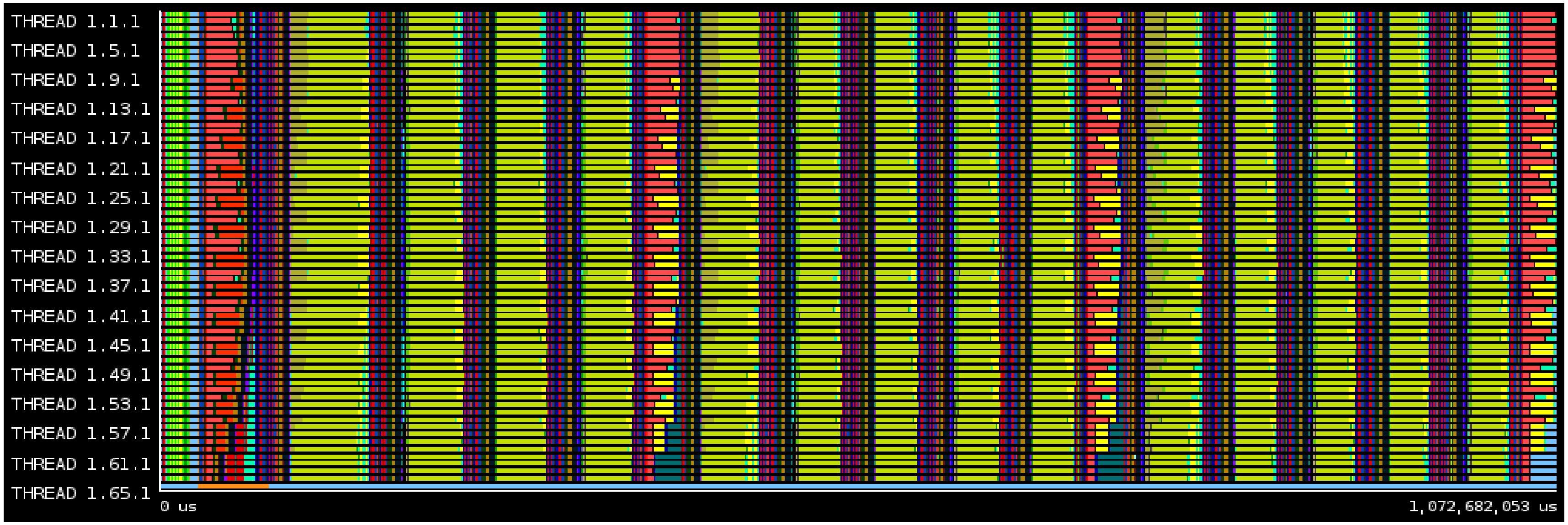
eff.= LB * Ser * Trf

LB	Ser	Trf	Eff
0.83	0.97		0.80
0.87	0.9		0.78
0.88	0.97	0.84	0.73
0.88	0.96	0.75	0.61



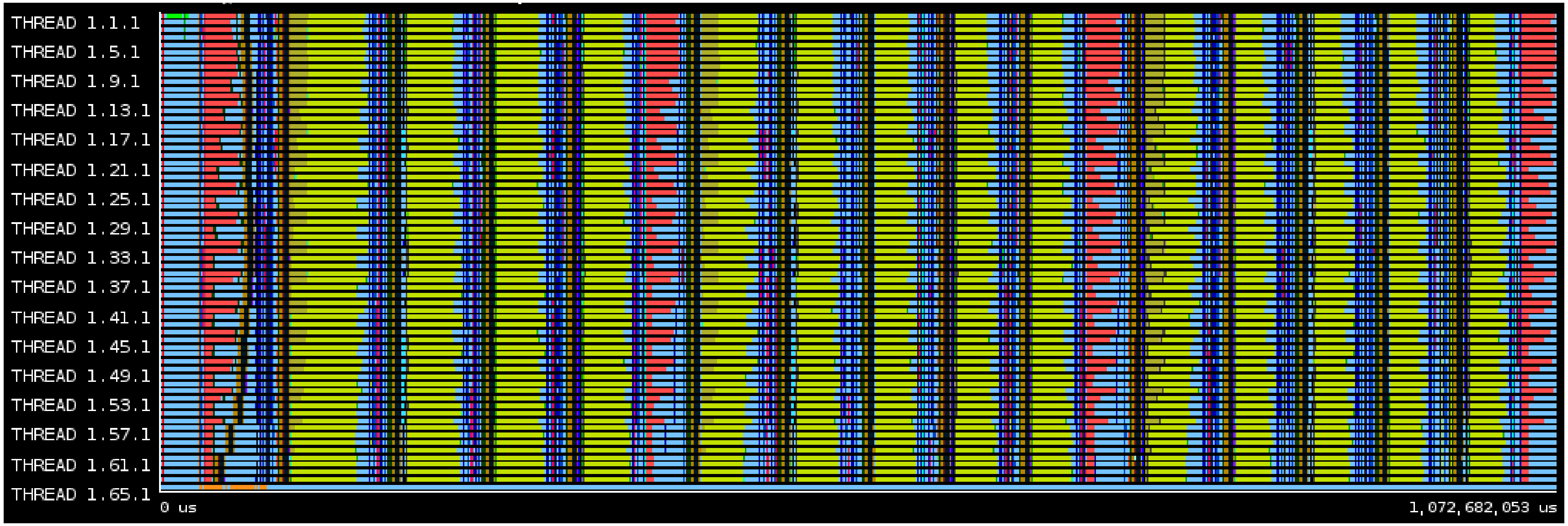
One hour simulation

- One hour simulation, chemistry configuration for global model, 24 km resolution



One hour simulation

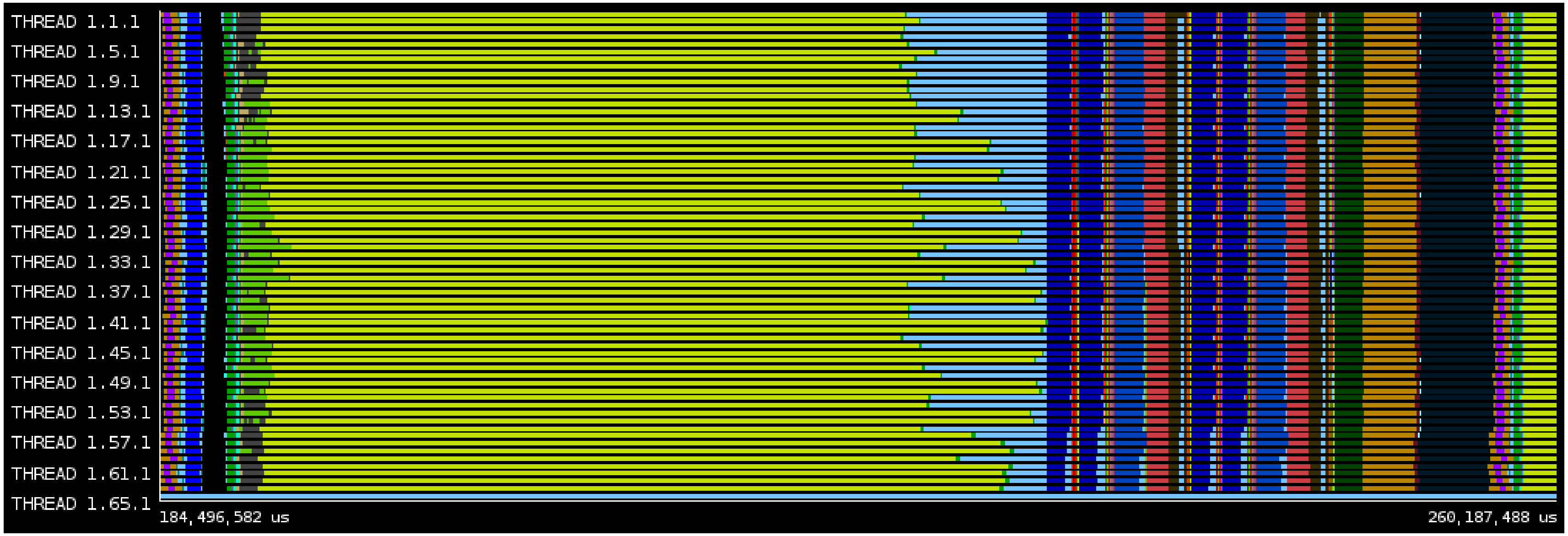
- Useful functions



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Zoom on EBI solver and next calls

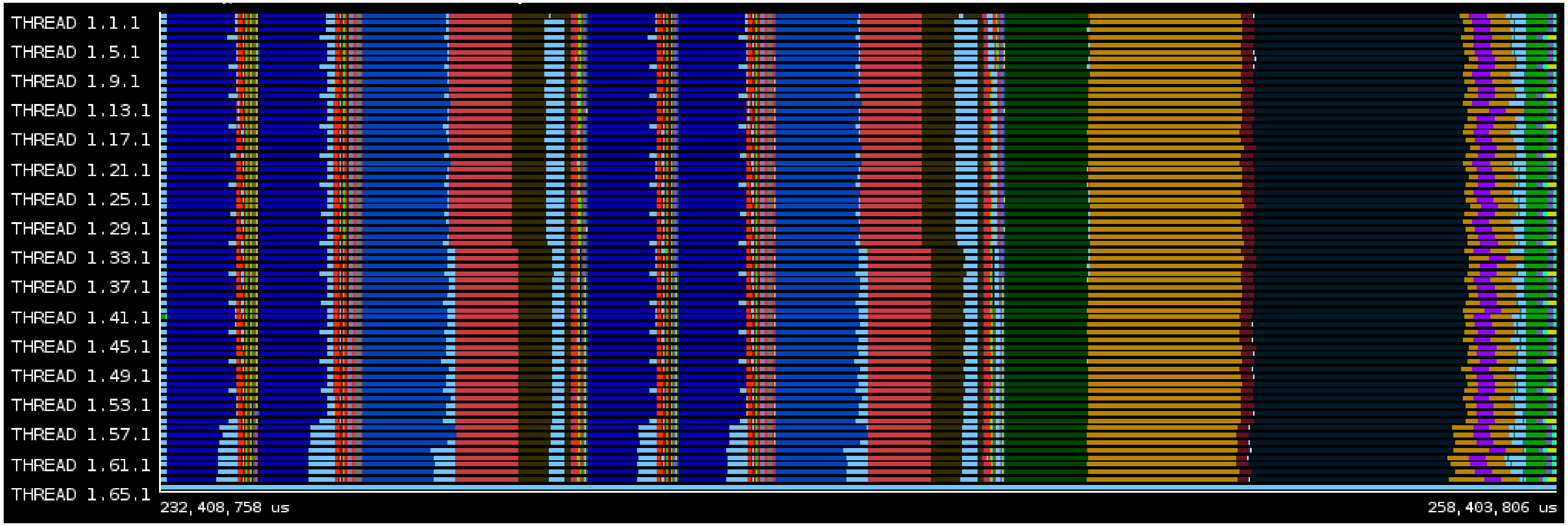
- EBI solver (run_ebi) and the calls till the next call to the solver



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Zoom between EBI solvers

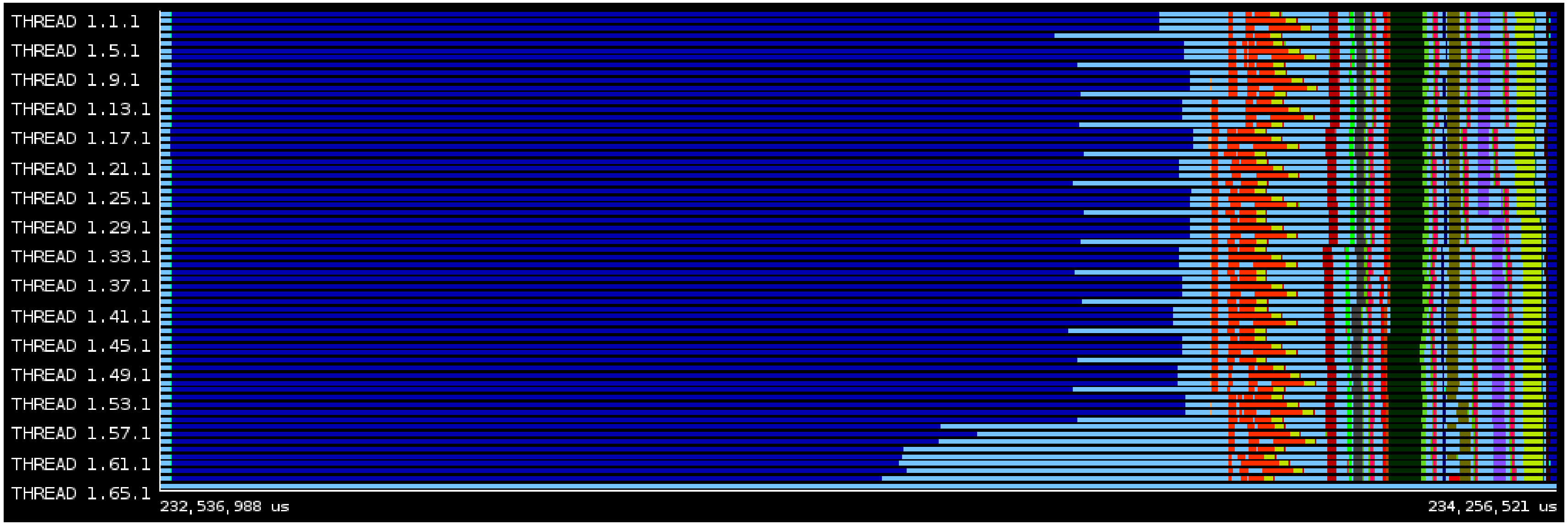
- The useful functions call between two EBI solvers
- The first two dark blue areas are horizontal diffusion calls and the light dark is advection chemistry.



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Horizontal diffusion

- We zoom on horizontal diffusion and the calls that follow
- Horizontal diffusion (blue colour) has load imbalance



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

BSC Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Experiences with OmpSs

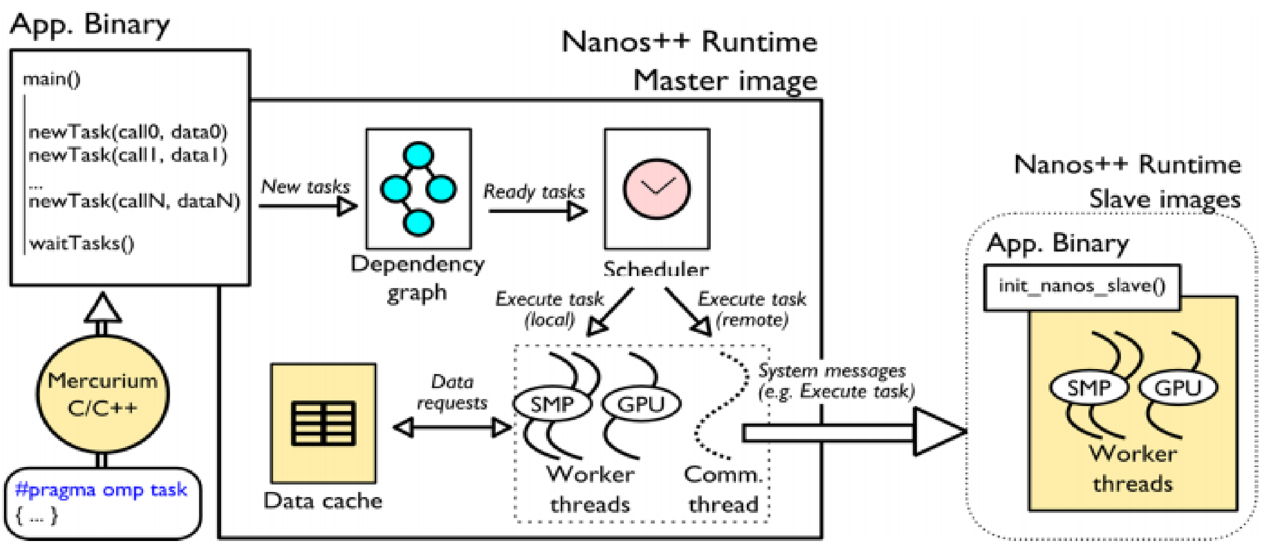
Objectives

- Trying to apply OmpSs on a real application
- Applying incremental methodology
- Identify opportunities
- Exploring difficulties

OmpSs introduction

- Parallel Programming Model
  - Build on existing standard: OpenMP
  - Directive based to keep a serial version
  - Targeting: SMP, clusters, and accelerator devices
  - Developed in Barcelona Supercomputing Center (BSC)
    - Mercurium source-to-source compiler
    - Nanos++ runtime system

https://pm.bsc.es/ompss



Studying cases

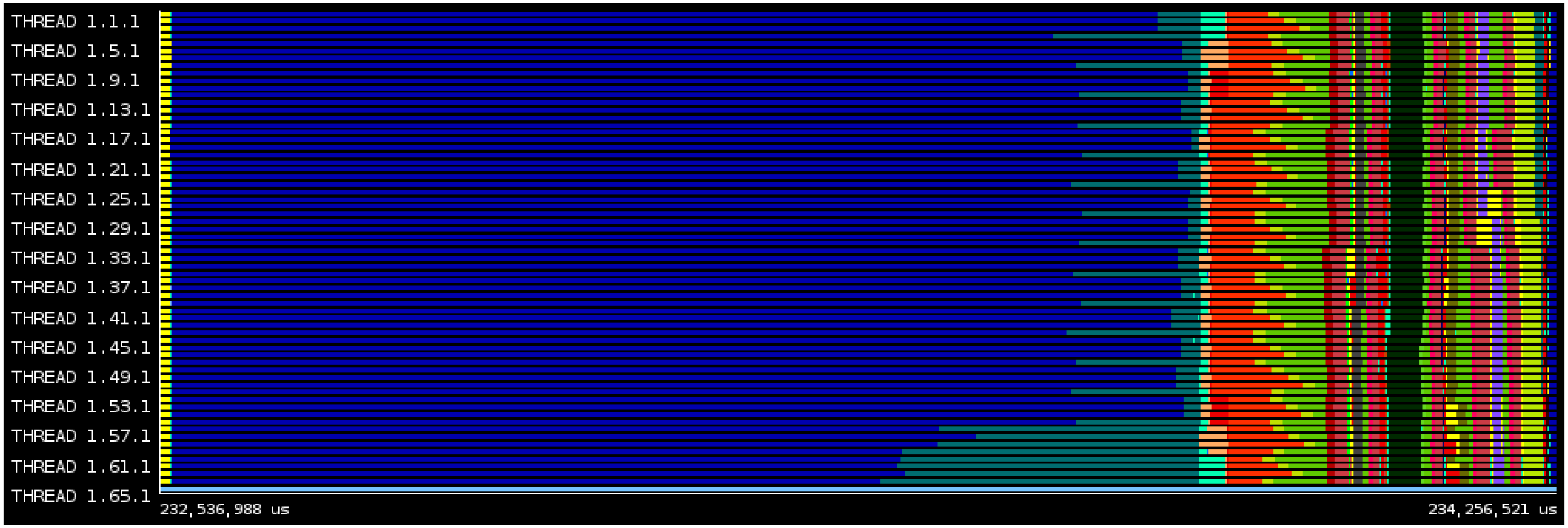
- Taskify a computation routine and investigate potential improvements and overlap between computations of different granularities

- Overlap communication with packing and unpacking costs

- Overlap independent coarse grain operations composed of communication and computation phases

Horizontal diffusion + communication

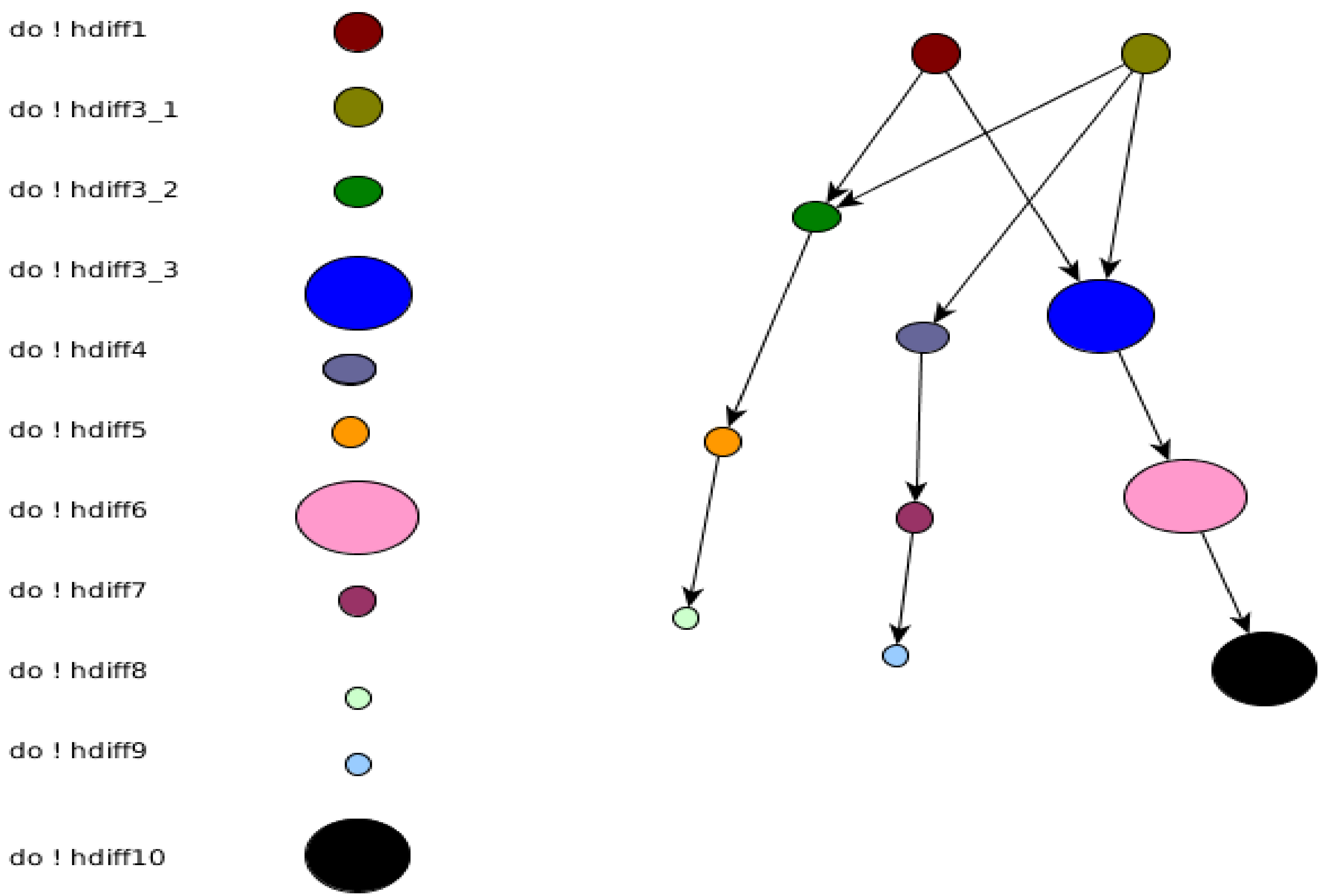
- The horizontal diffusion has some load imbalance(blue code)
- There is some computation about packing/unpacking data for the communication buffers (red area)
- Gather (green colour) and scatter for the FFTs



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

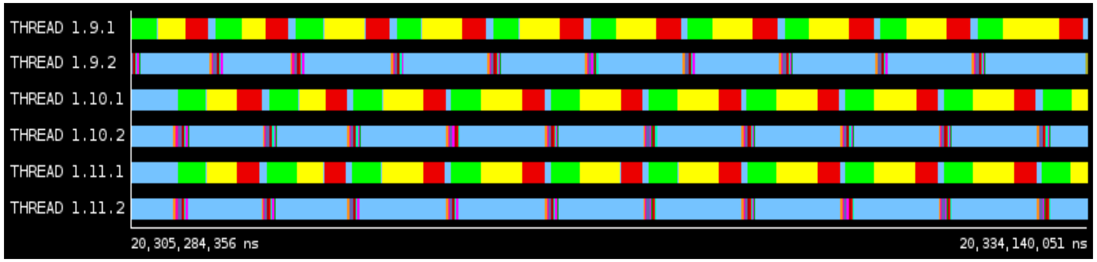
Horizontal diffusion skeleton code

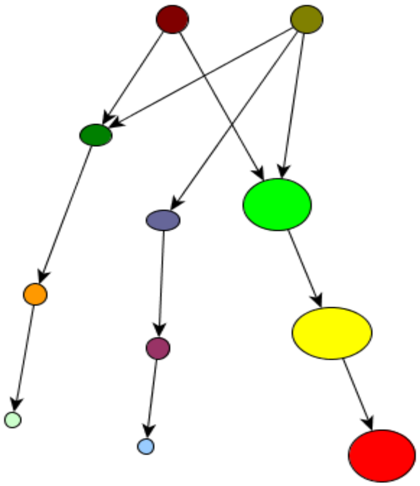
- The hdiff subroutine has the following loops and dependencies



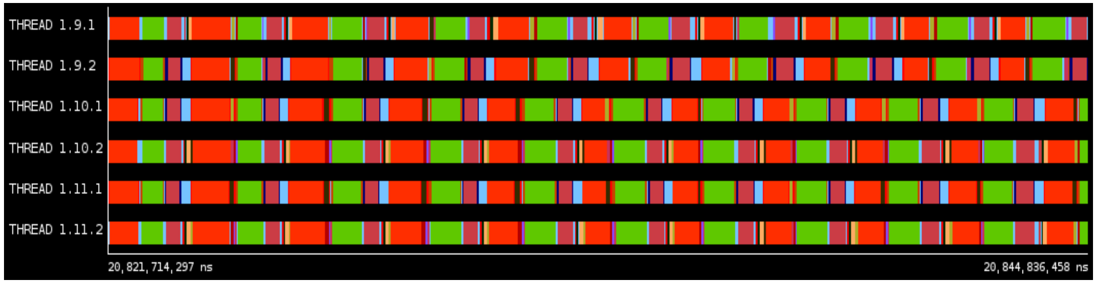
Parallelizing loops

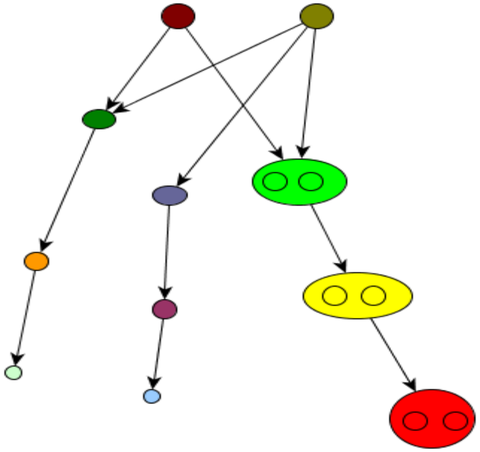
- Part of hdiff with 2 threads





- Parallelizing the most important loops

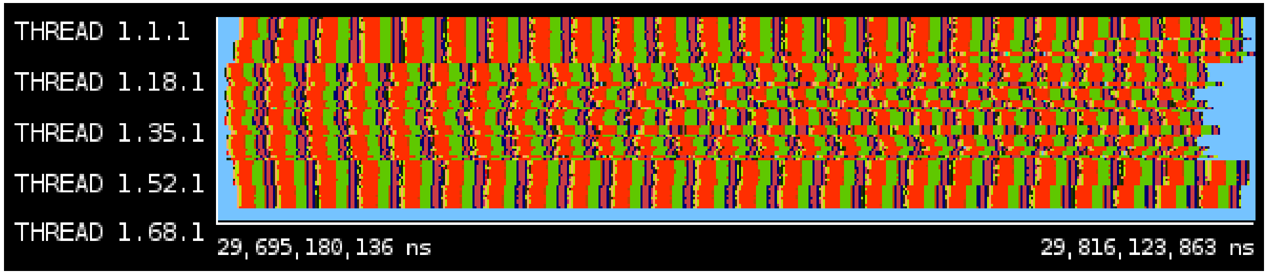




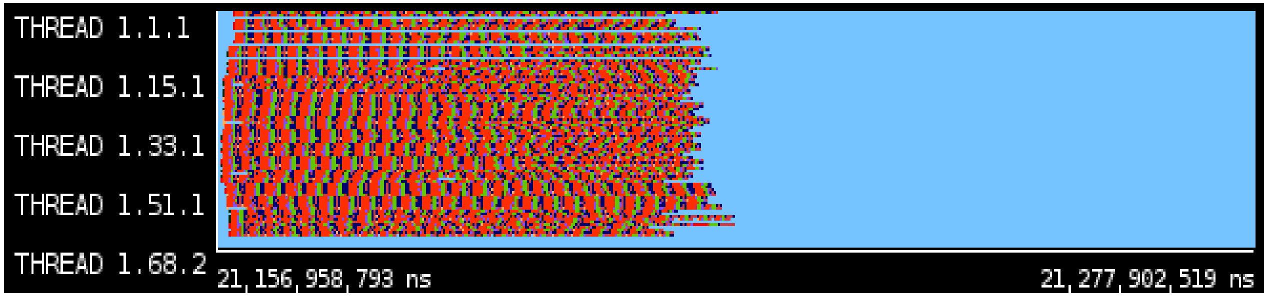
- We have a speedup of 1.3 by using worksharing

Comparison

- The execution of hdiff subroutine with 1 thread takes 120 ms



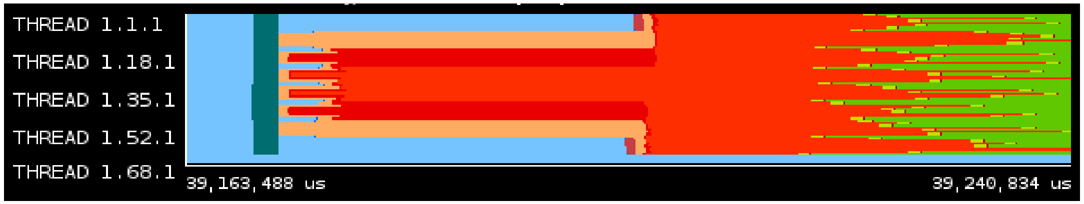
- The execution of hdiff subroutine with 2 threads takes 56 ms, the speedup is 2.14



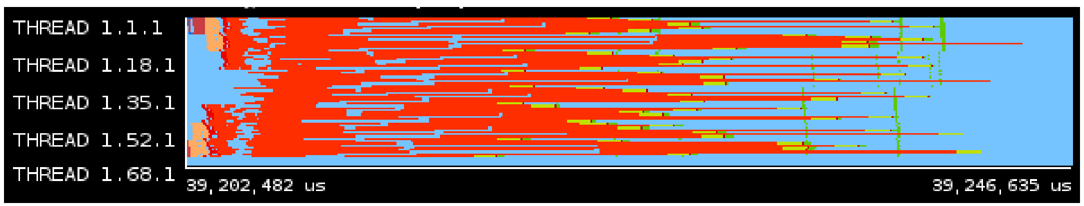
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Issues related to communication

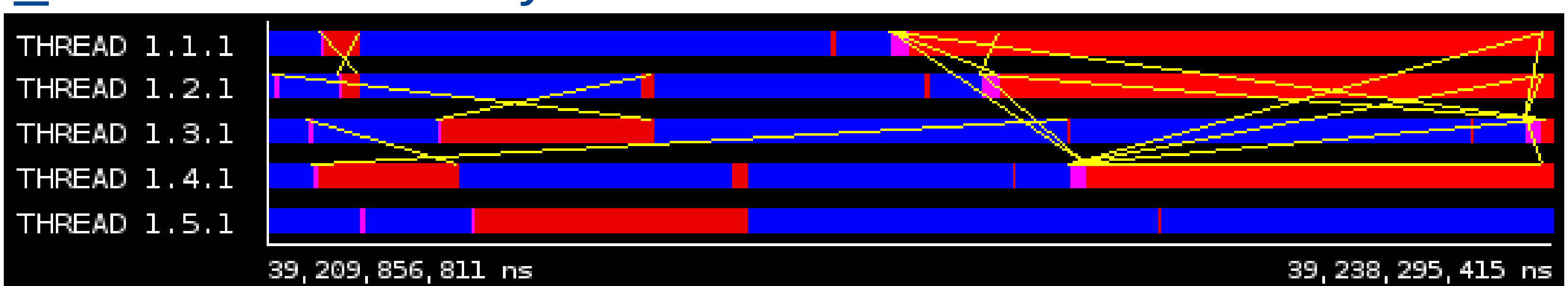
- We study the exch4 subroutine (red colour)



- The useful function of exch4 has some computation

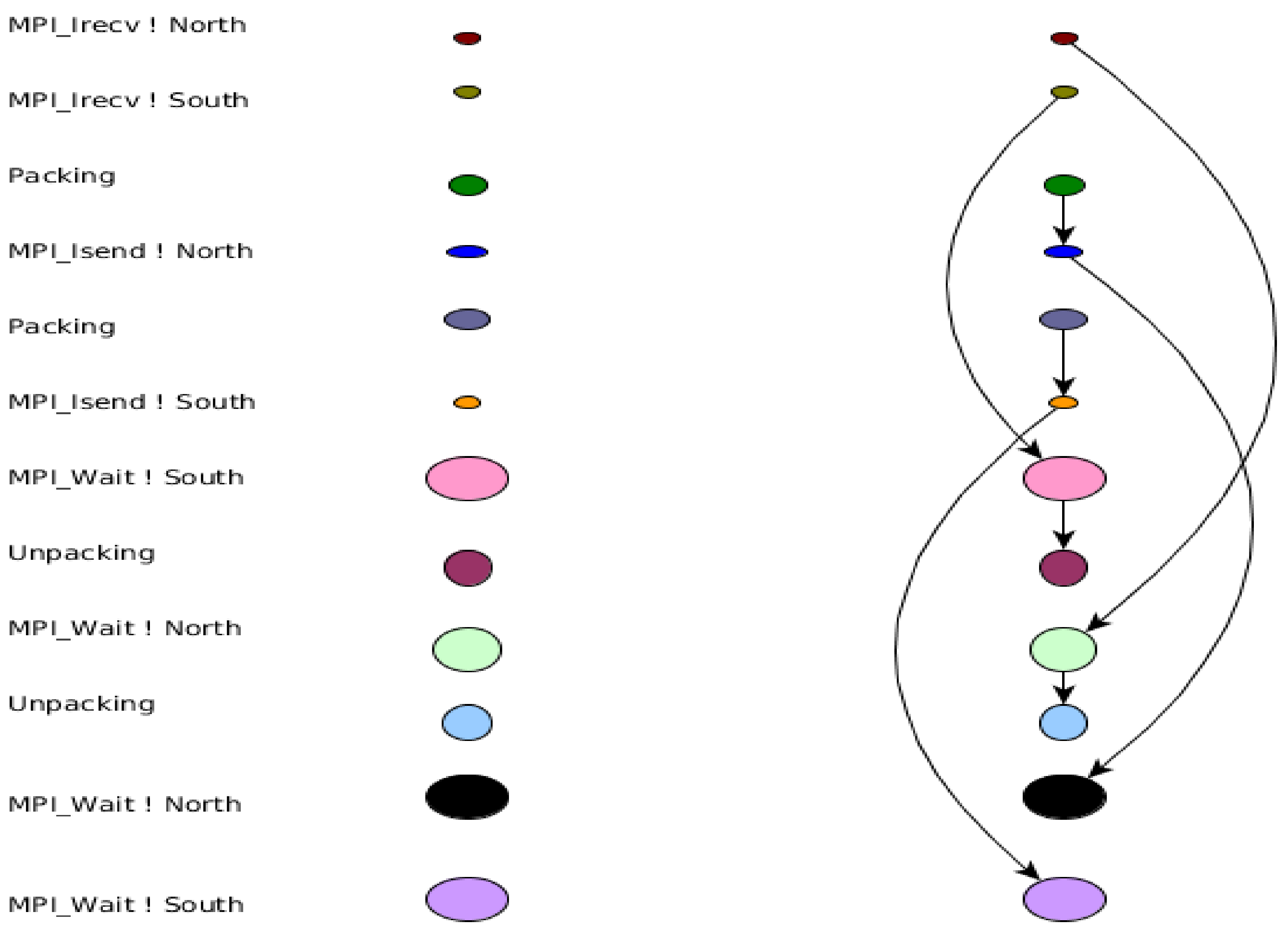


- The communication creates a pattern and the duration of the MPI_Wait calls can vary



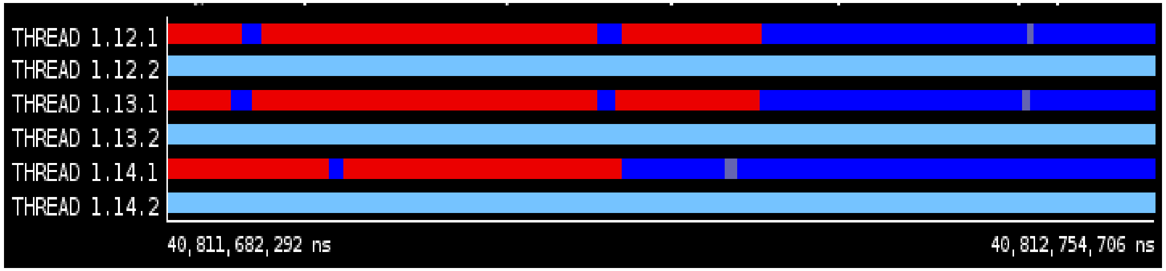
Issues related to communication

- Big load imbalance because message order
- There is also some computation

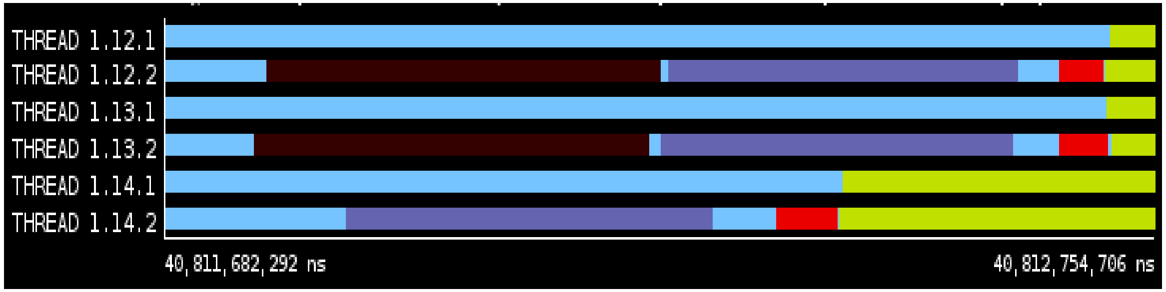


Taskify subroutine exch4

- We observe the MPI_Wait calls in the first thread

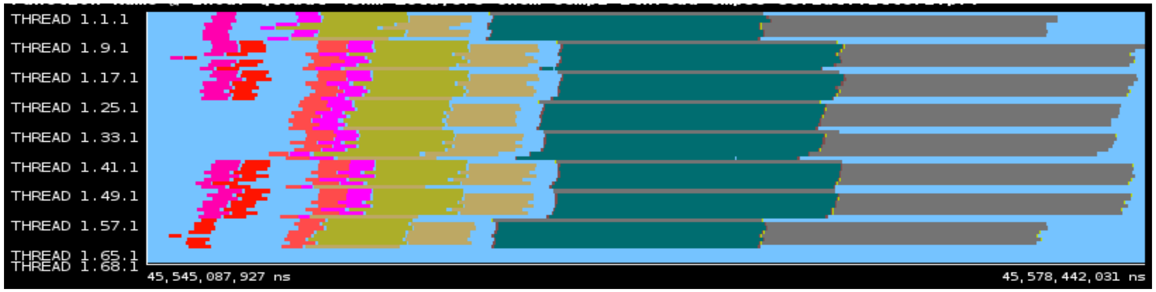


- In the same moment the second thread does the necessary computation and overlaps the communication

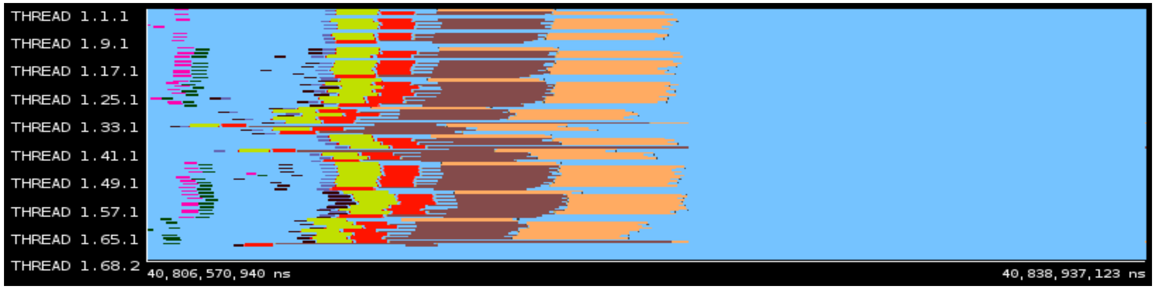


Taskify subroutine exch4

- The total execution of exch4 subrouting with 1 thread



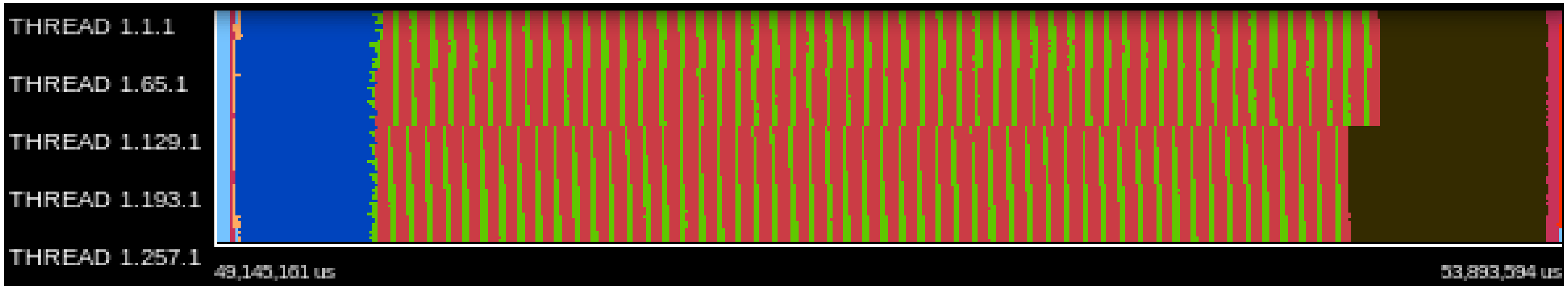
- The total execution of exch4 subroutine with 2 threads



- With 2 threads the speedup is 1.76 (more improvements have been identified)

Advection chemistry and FFT

- Advection chemistry (blue color) and 54 calls to gather/FFT/scatter till monotonization chemistry (brown color)

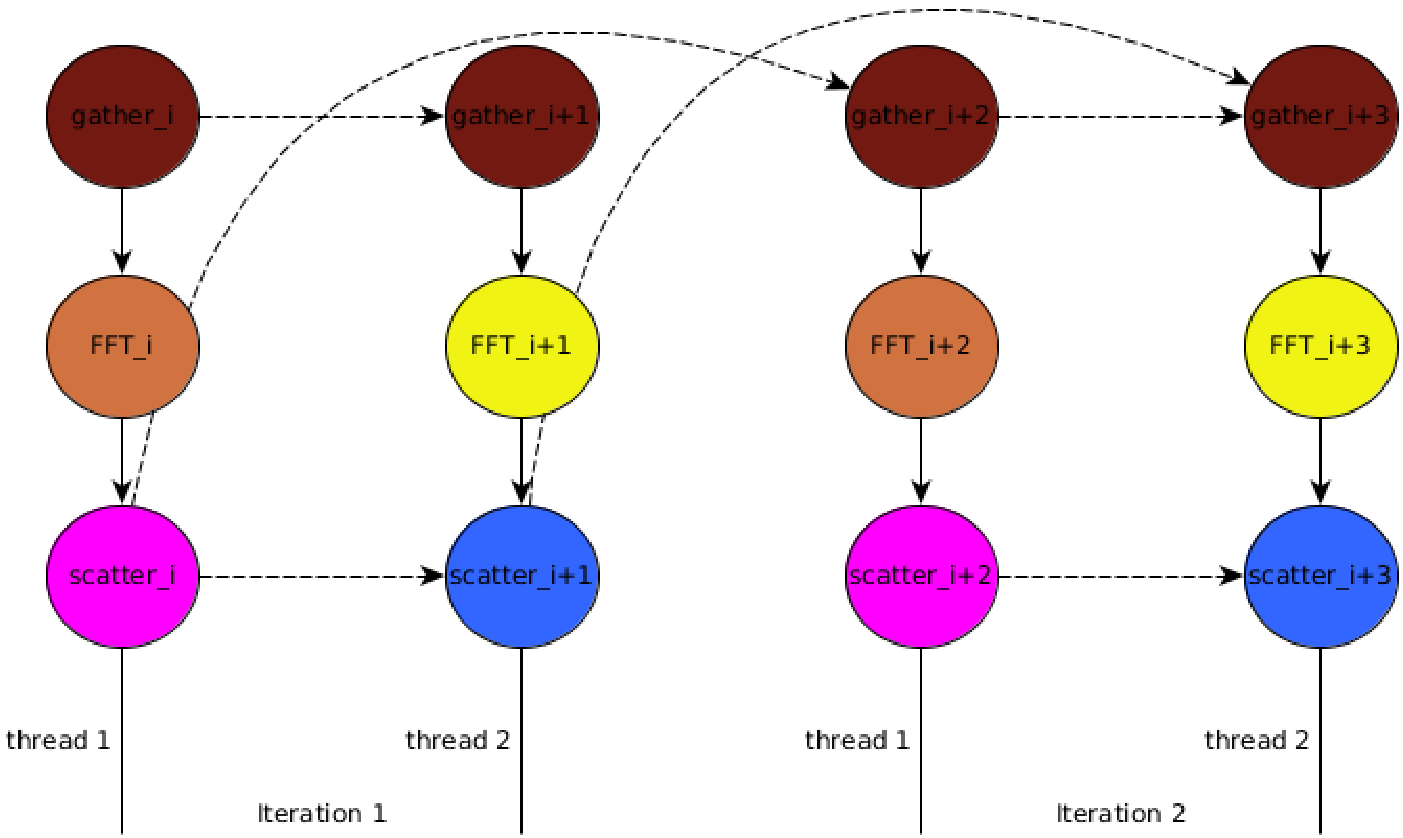


- Initial study to test the improvements of the execution with the OmpSs programming model

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

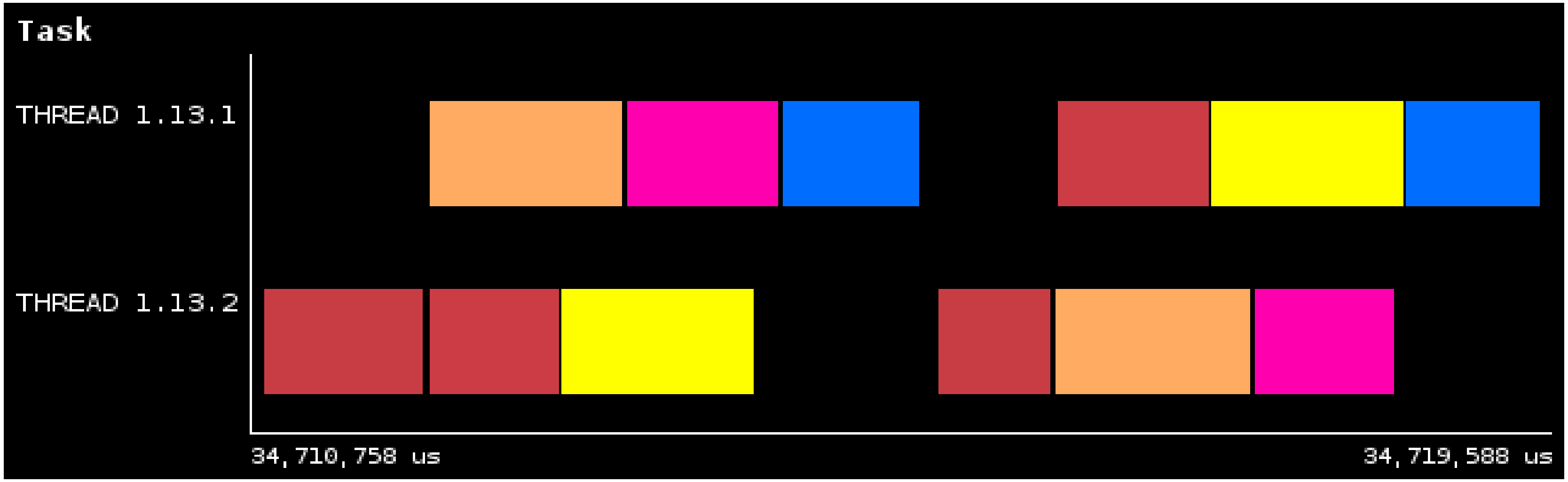
Study case: gather/FFT/scatter

- Workflow, two iterations, using two threads, declaring dependencies



Study case: gather/FFT/scatter

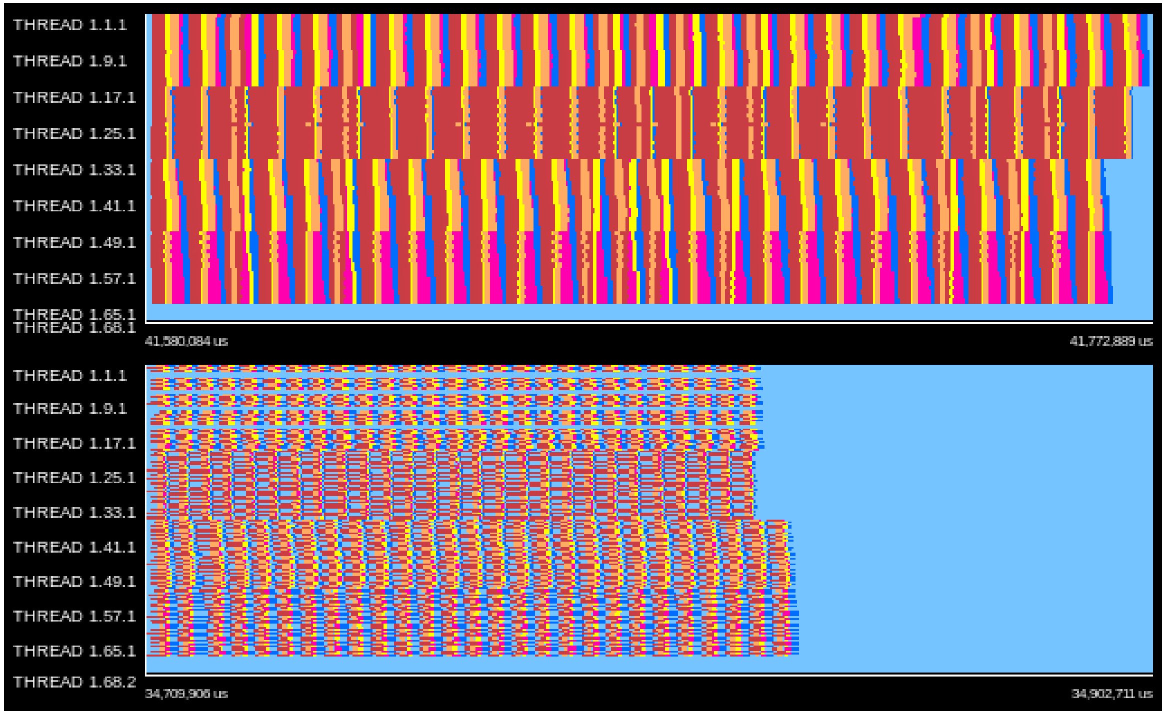
- Paraver view, two iterations, four tracers totally



- Thread 1:
  - Iteration 1: FFT_1, scatter_1, scatter_2
  - Iteration 2: gather_4, FFT_4, scatter_4
- Thread 2:
  - Iteration 1: gather_1, gather_2, fft_2
  - Iteration 2: gather_3, FFT_3, scatter_3

Study case: gather/FFT/scatter - Performance

- Comparing the execution time of 54 calls to gather/FFT/scatter with one and two threads

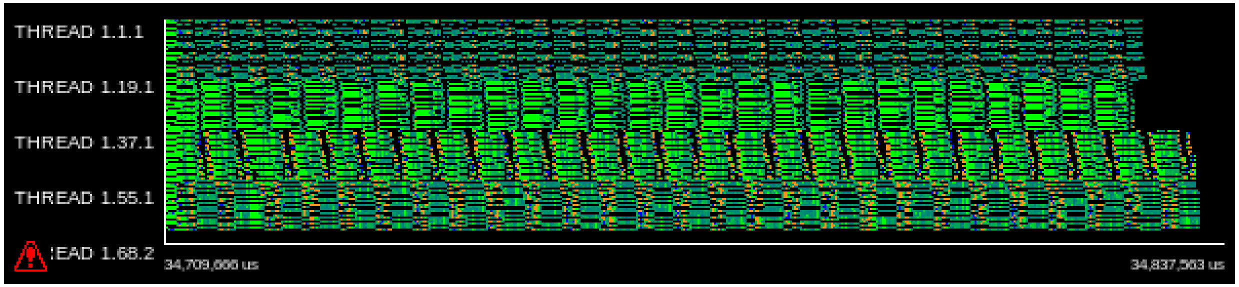


- The speedup with two threads is 1.56 and we have identified potential improvements

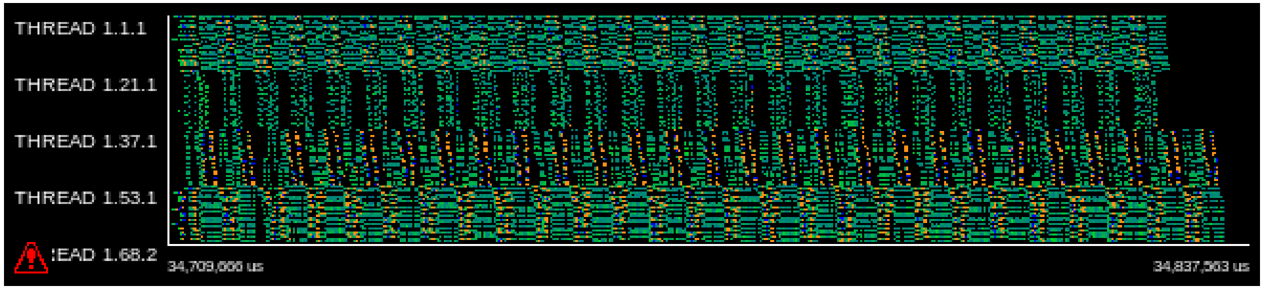
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

MPI Bandwidth

- MPI bandwidth for gather/scatter

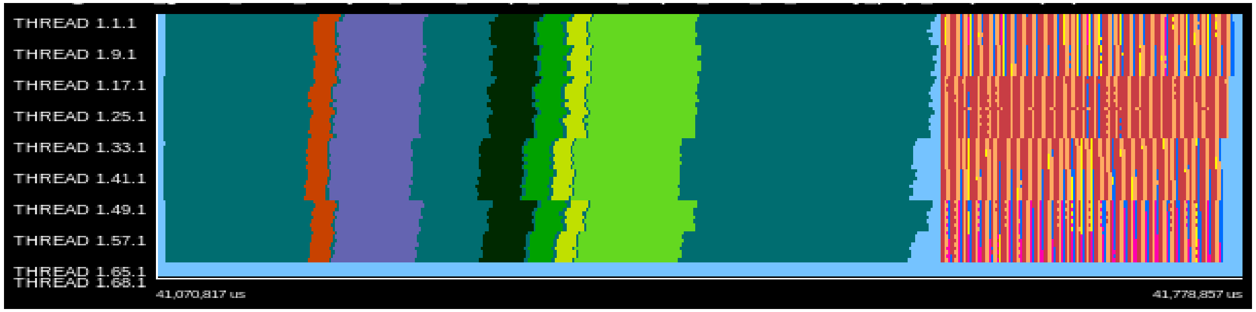


- MPI bandwidth over 1GB/s for gather/scatter

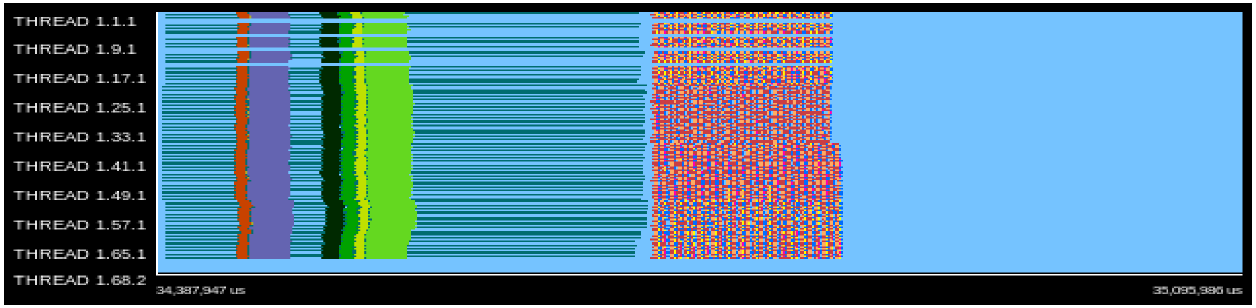


Combination of advection chemistry and FFT

- Advection chemistry with worksharing (not for all the loops), FFTs for one thread



- Similar but with two threads

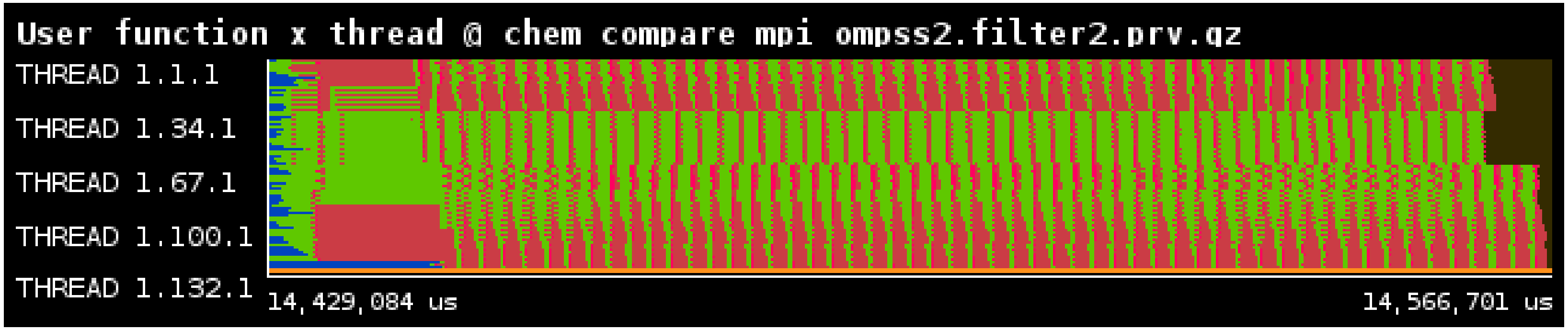


- The speedup for the advection chemistry routine is 1.6 and overall is 1.58

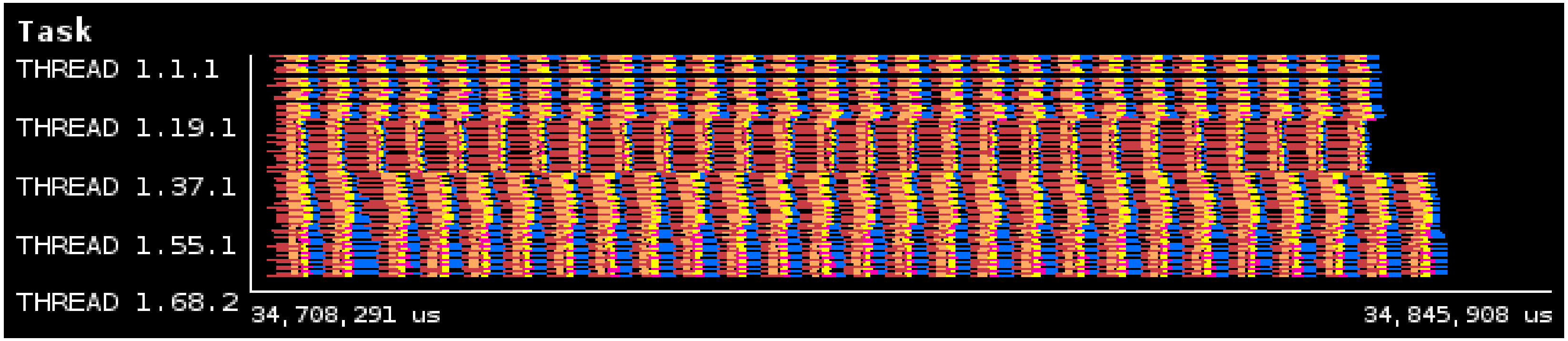
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Comparison between MPI and MPI+OmpSs

- Pure MPI, 128 computation processes and 4 I/O



- MPI + OmpSs: 64 MPI processes + 64 threads + 4 I/O



- The load imbalance for the FFT with pure MPI is 28% while with MPI+OmpSs is 54%

Incremental methodology with OmpSs

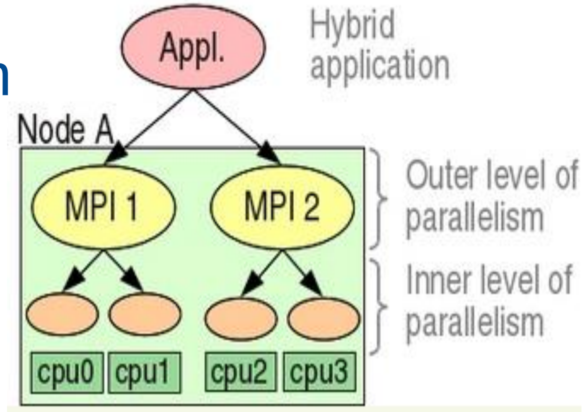
- Taskify the loops
- Start with 1 thread, use if(0) for serializing tasks
- Test that dependencies are correct (usually trial and error)
- Imagine an application crashing after adding 20+ new pragmas (true story)
- Do not parallelize loops that do not contain significant computation

Conclusions

- The incremental methodology is important for less overhead in the application
- OmpSs can be applied on a real application but is not straightforward
- It can achieve pretty good speedup, depending on the case
- Overlapping communication with computation is a really interesting topic
- We are still in the beginning but OmpSs seems promising

Future improvements

- Investigate the usage of multithreaded MPI
- One of the main functions of the application is the EBI solver (run_ebi). There is a problem with global variables that make the function not reentrant. Refactoring of the code is needed.
- Porting more code to OmpSs and investigate MPI calls as tasks
- Some computation is independent to the model's layers or to tracers. OpenCL kernels are going to be developed to test the performance on accelerators
- Testing versioning scheduler
- The dynamic load balancing library should be studied further (http://pm.bsc.es/dlb)
- Apply OmpSs for a data assimilation simulation



Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

Thank you!

For further information please contact

georgios.markomanolis@bsc.es

"Work funded by the SEV-2011-00067 grant of the Severo Ochoa Program, awarded by the Spanish Government."

Acknowledgements:
Rosa M. Badia, Judit Gimenez, Roger Ferrer Ibáñez, Julian Morillo, Victor López, Xavier Teruel, Harald Servat, BSC support