www.bsc.es

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Performance Analysis of an Earth Science Application

George S. Markomanolis, Oriol Jorba, Kim Serradell, Enza Di Tomaso

University of Athens, Department of Physics

Athens, 27 March 2014

EXCELENCIA SEVERO OCHOA

Outline

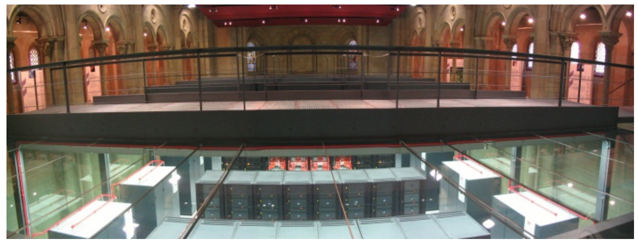
- Overview of BSC
- Introduction to Earth Sciences Modeling
- Preprocess
- Performance Analysis of NMMB/BSC-CTM Model
- OmpSs Programming Model
- Data Assimilation
- Future work

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Overview of BSC

BSC-CNS

- Barcelona Supercomputing Center – Centro Nacional de Supercomputación (BSC-CNS) is the Spanish National Laboratory in supercomputing.



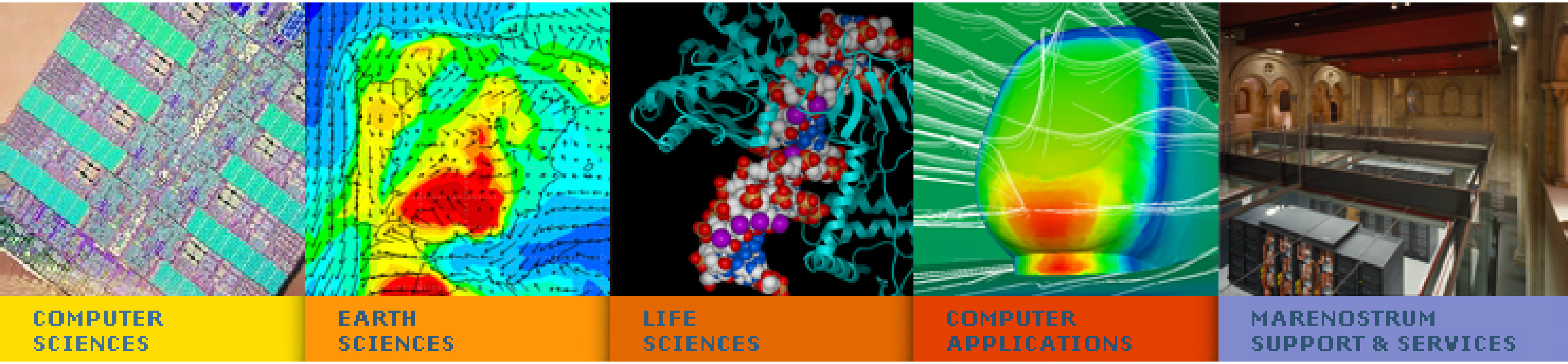
- The BSC mission:
  - To investigate, develop and manage technology to facilitate the advancement of science.

- The BSC objectives:
  - To perform R&D in Computer Sciences and e-Sciences
  - To provide Supercomputing support to external research.

- BSC is a consortium that includes:
  - the Spanish Government – 51%
  - the Catalan Government – 37%
  - the Technical University of Catalonia – 12%

BSC Scientific & Technical Departments

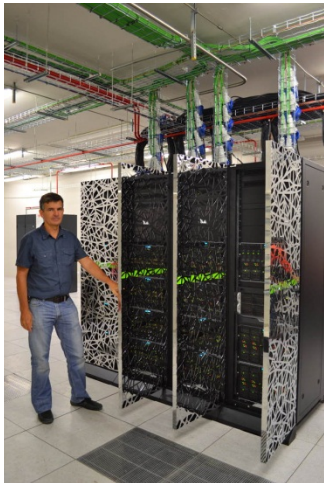
www.bsc.es



BSC Current Resources

- MareNostrum 2013
  - 48448 Intel SandyBridge-EP cores
  - 1 PFlops

- MinoTauro 2011
  - 128 compute nodes
  - 182 TFlops

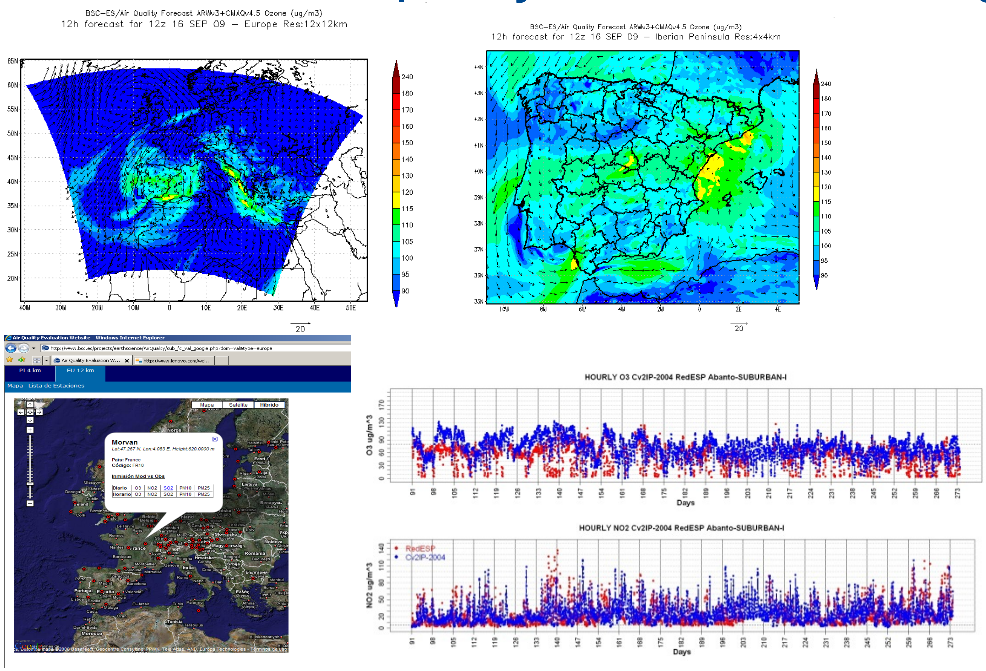


- HPC Storage and Backup:
  - 2.5 PB disk
  - 6.0 PB tapes Robot

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Introduction to Earth Sciences Modeling

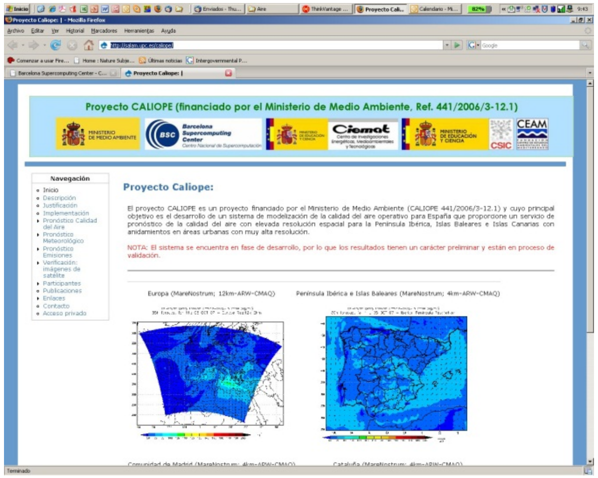
- Research in the Earth Sciences area is devoted to the development and implementation of regional and global state-of-the-art models for short-term air quality forecast and long-term climate applications.



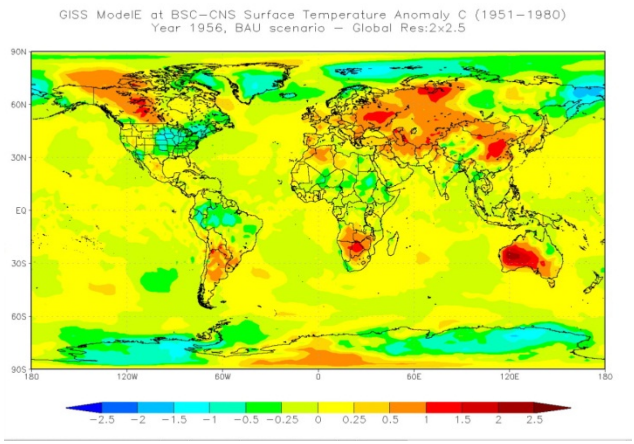
- ES maintains two daily operational systems: AQF CALIOPE and MD forecasts: BSC-DREAM8b and NMMB/BSC-CTM.

Earth Sciences research lines

Air Quality Forecast

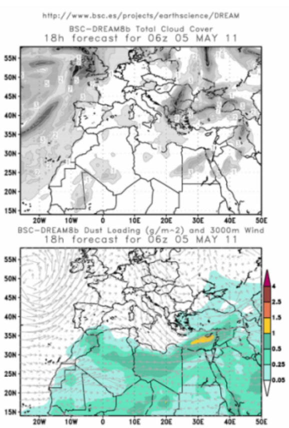


Climate change modelling

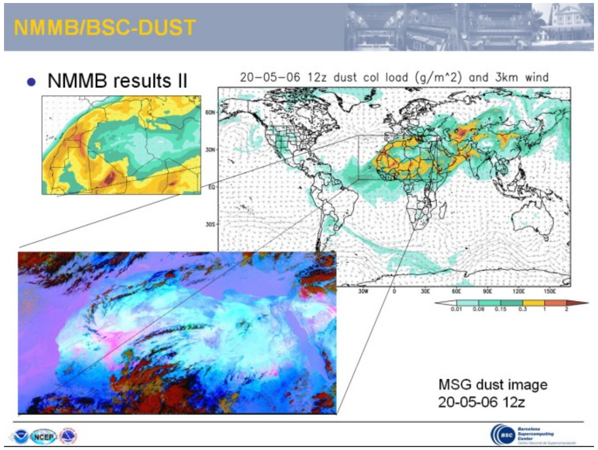


Transfer technology (EIA and AQ studies)

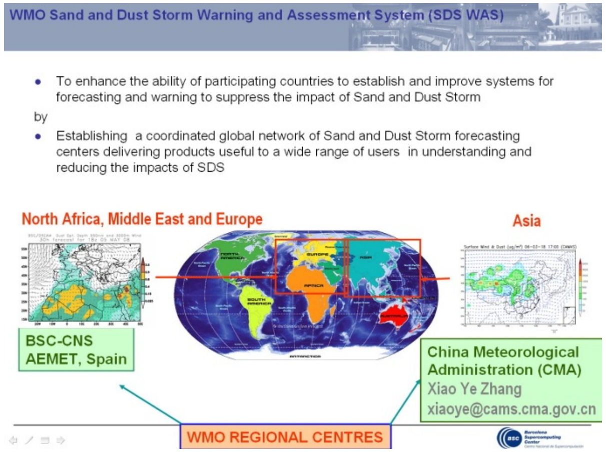
Mineral dust transport: BSC-DREAM8b



Atmospheric modelling: development of NMMB/BSC-CTM



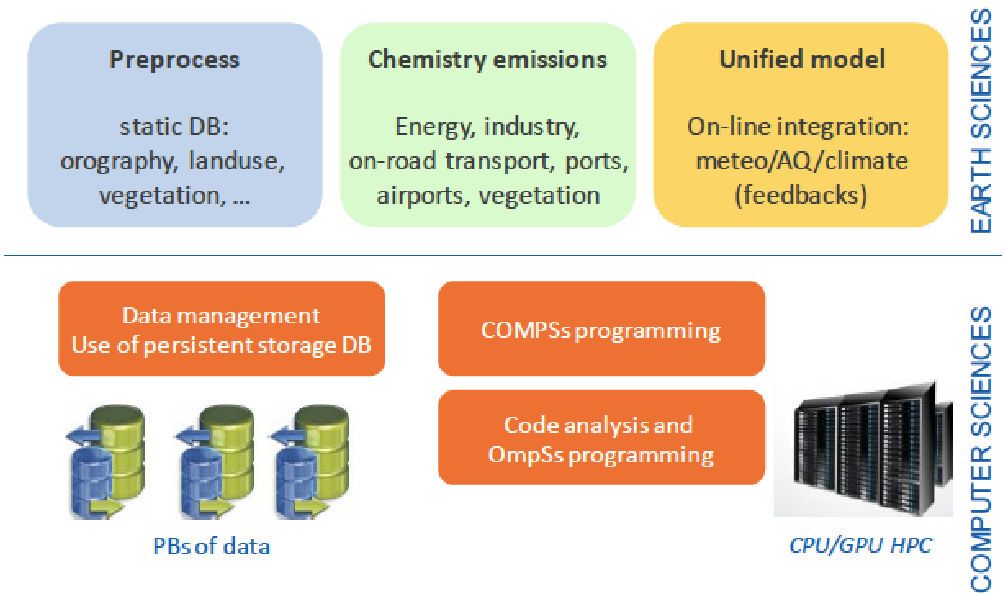
WMO SDS WAS [AEMET-BSC]



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

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

Air Quality

Uni. of California Irvine

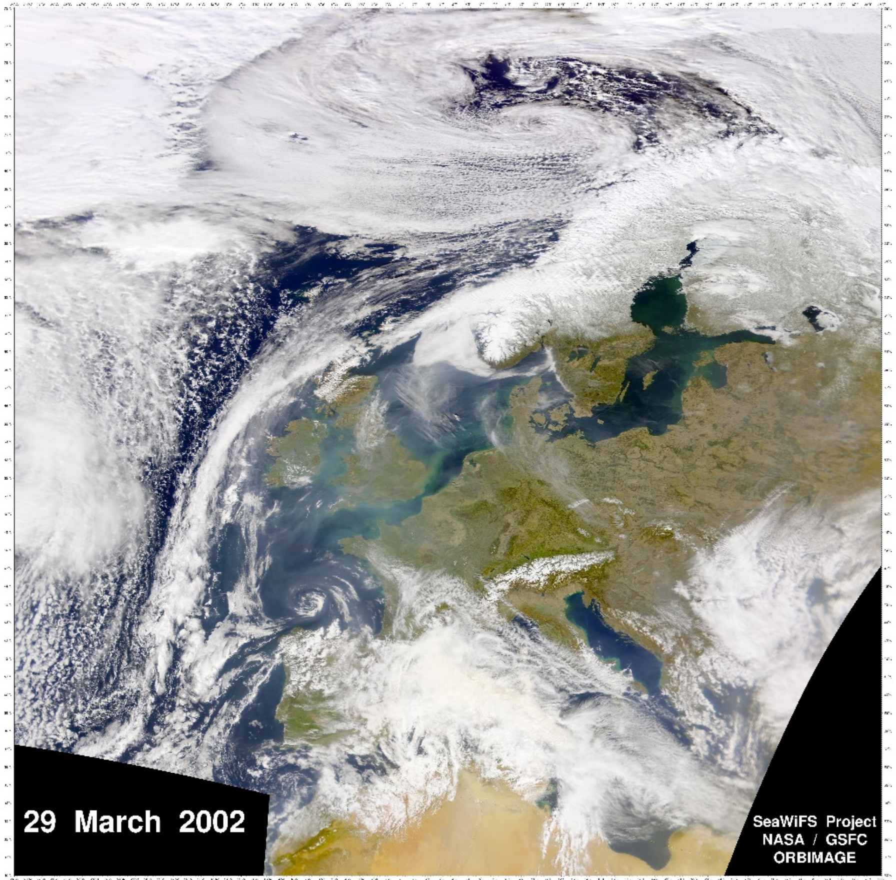
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

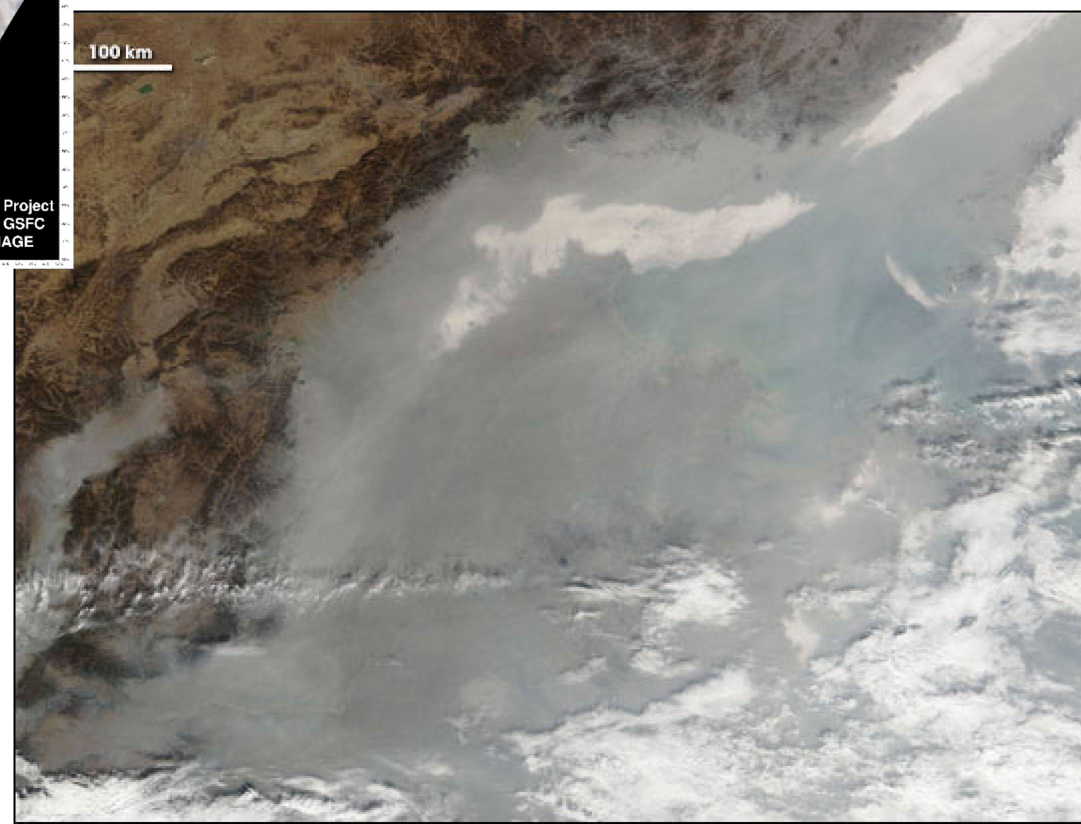
Is it a new problem?

- Not a new problem:
  - As far back as the 13 th century, people started complaining about coal dust and soot in the air over London, England.
  - As industry spread across the globe, so did air pollution.
  - The worst air pollution happened in London when dense smog (a mixture of smoke and fog) formed in December of 1952 and lasted until March of 1953. 4,000 people died in one week. 8,000 more died within six months.

- A picture is worth a thousand words

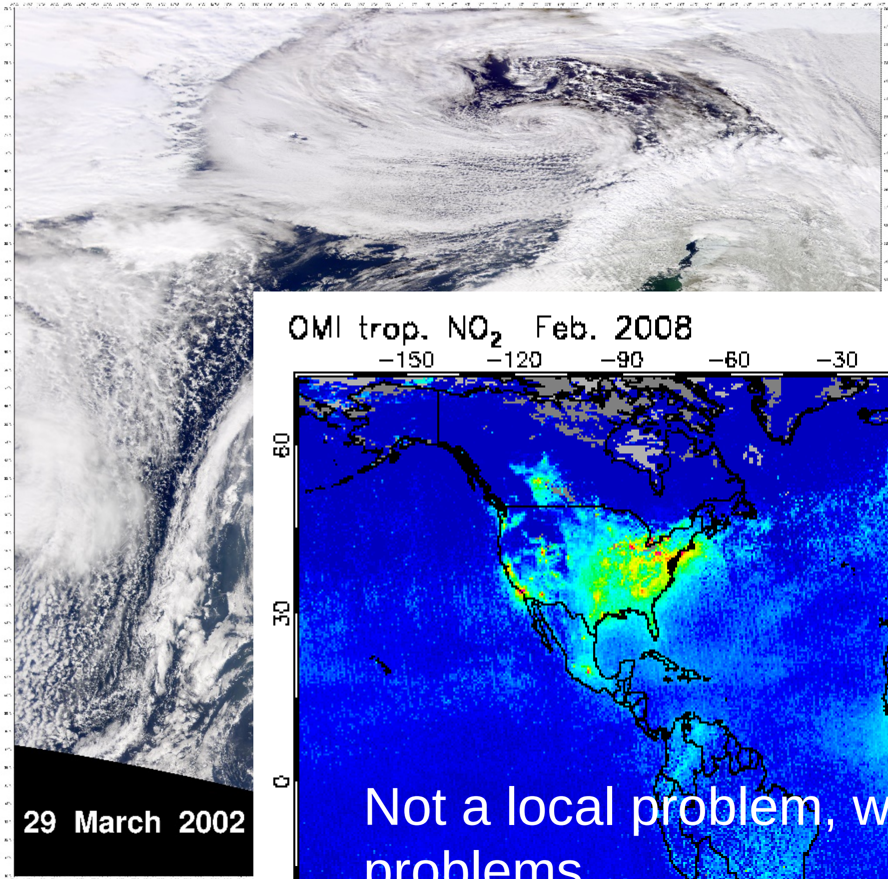
Air Pollution: Europe, South China, the Earth

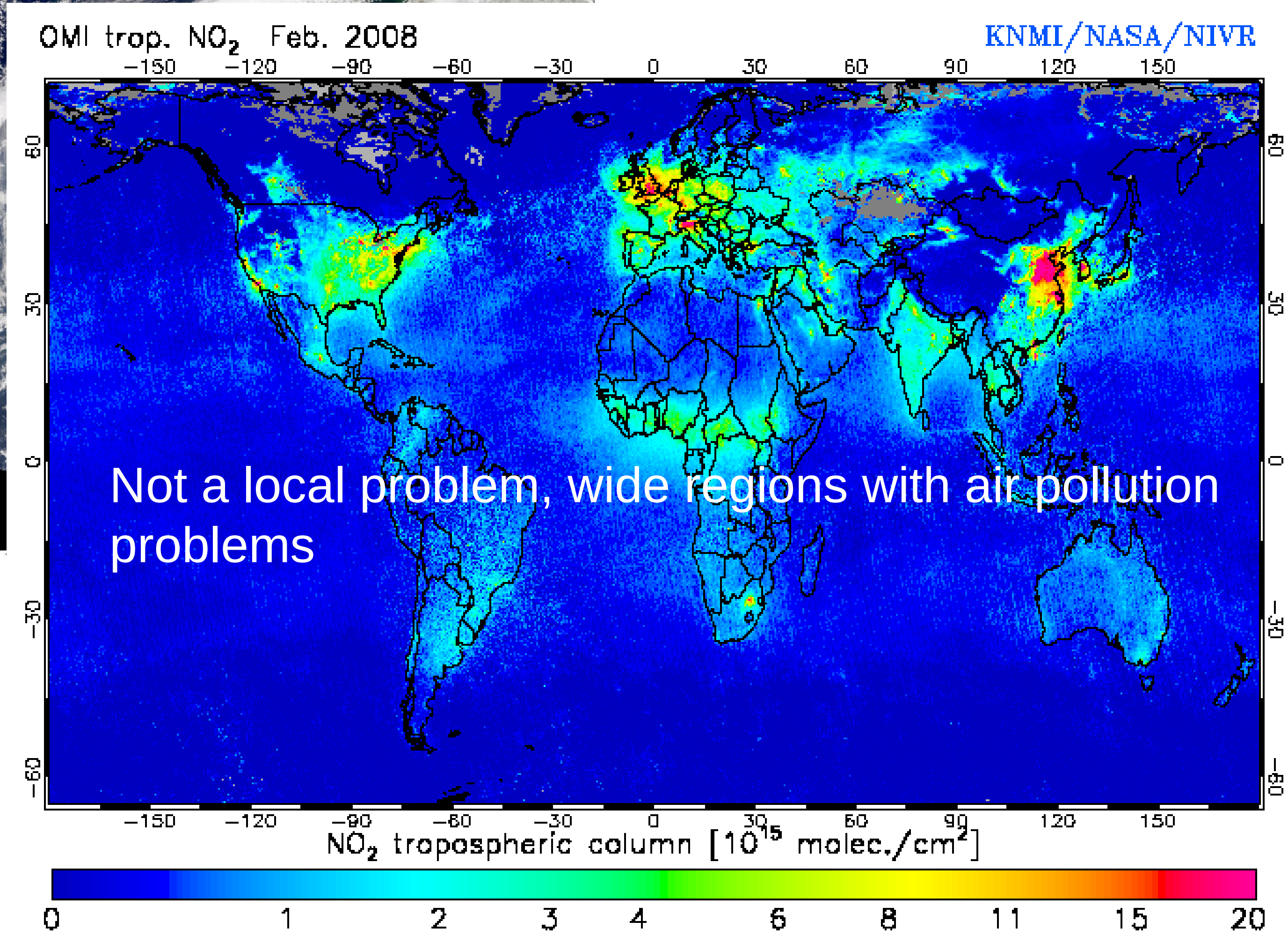




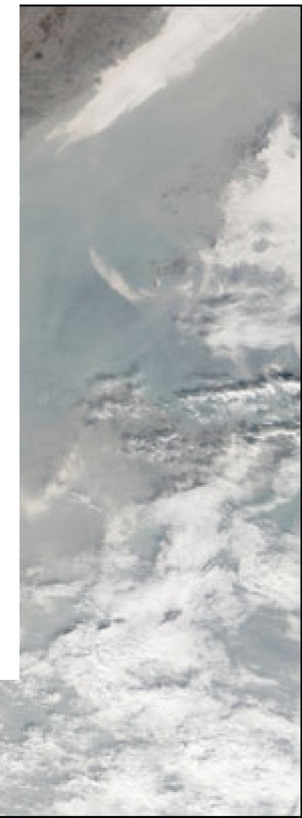
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Air Pollution: Europe, South China, the Earth



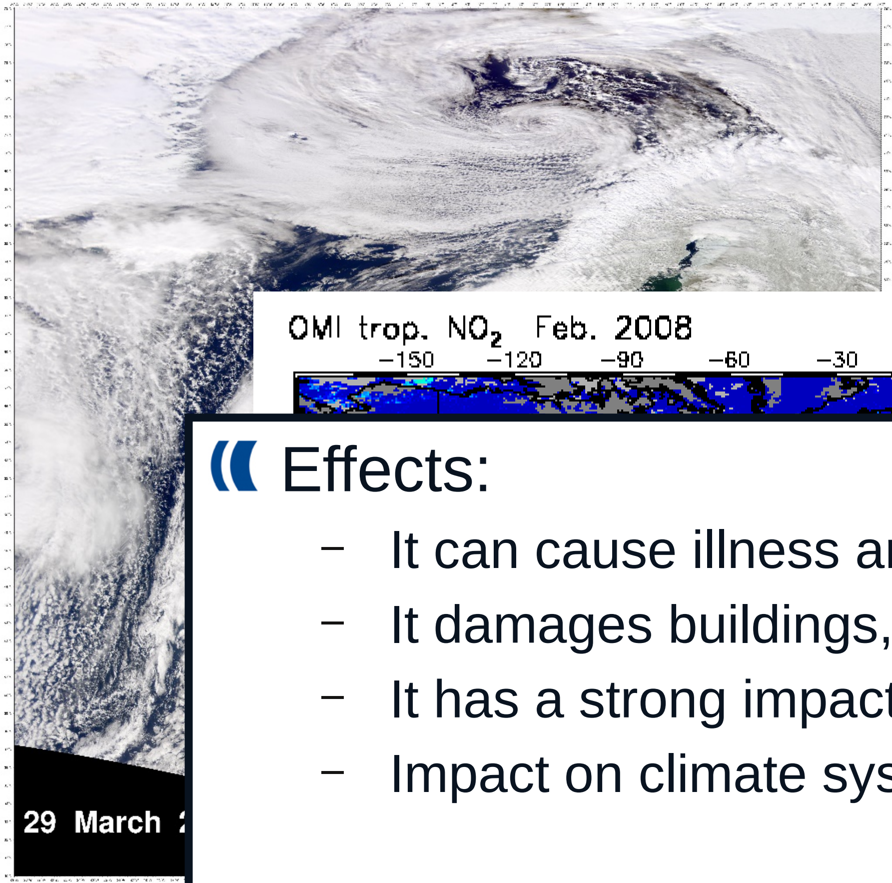


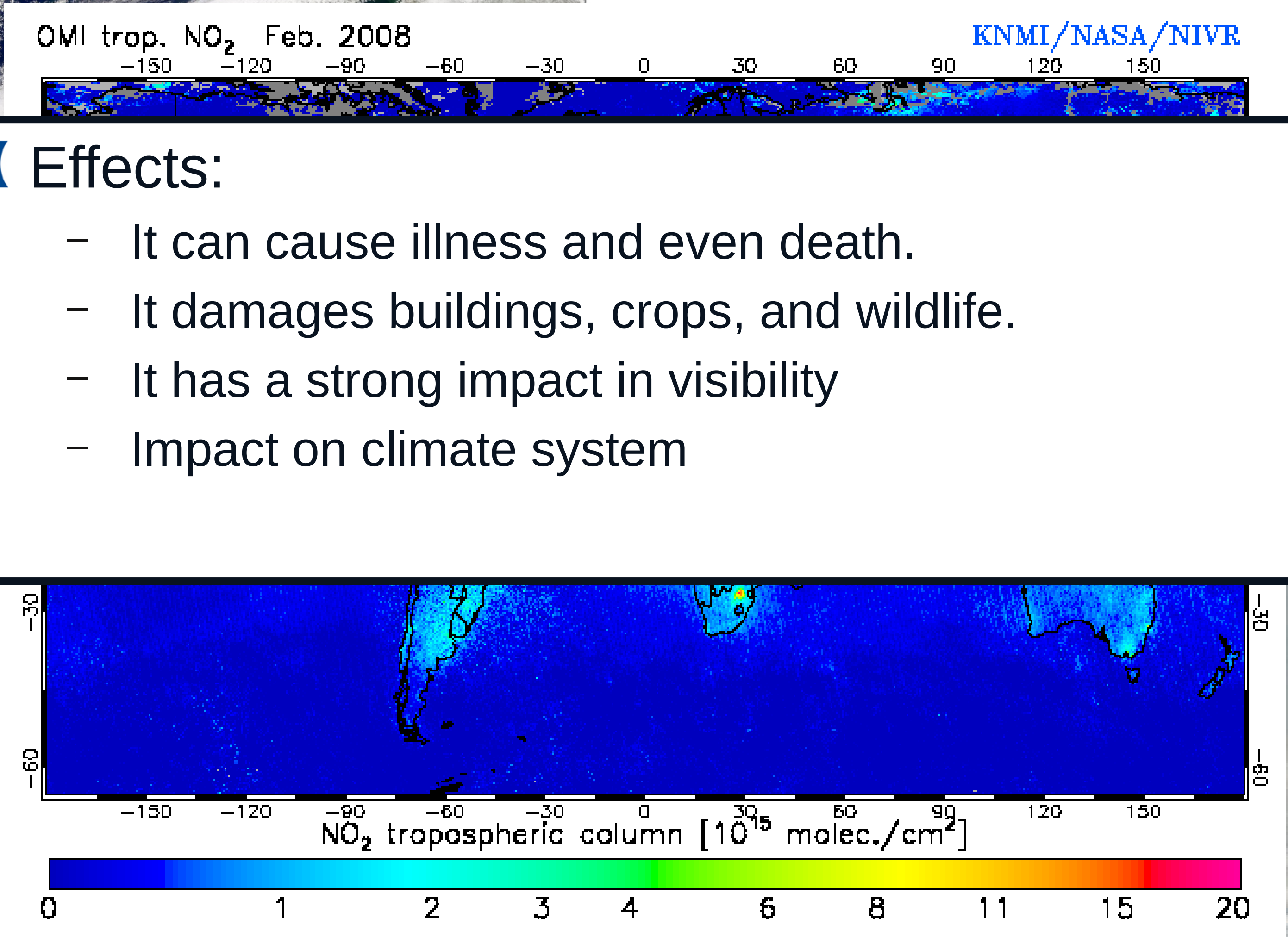
Not a local problem, wide regions with air pollution problems



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

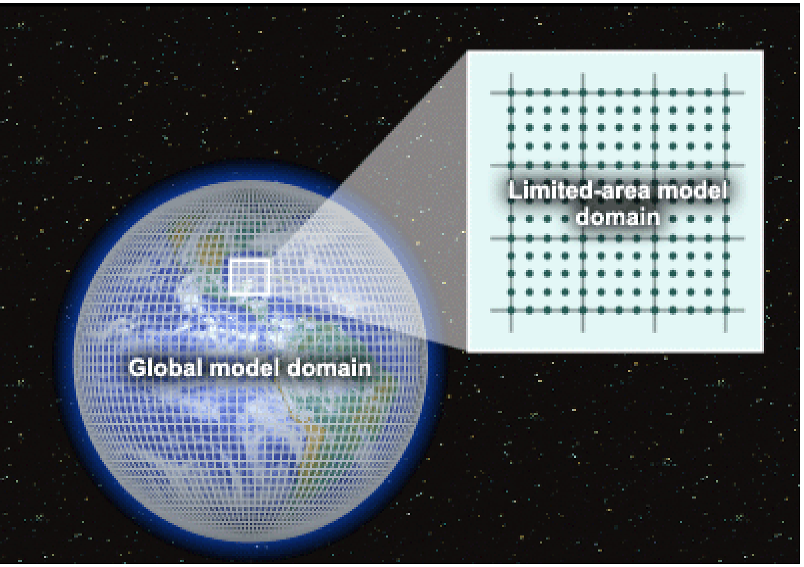
Air Pollution: Europe, South China, the Earth

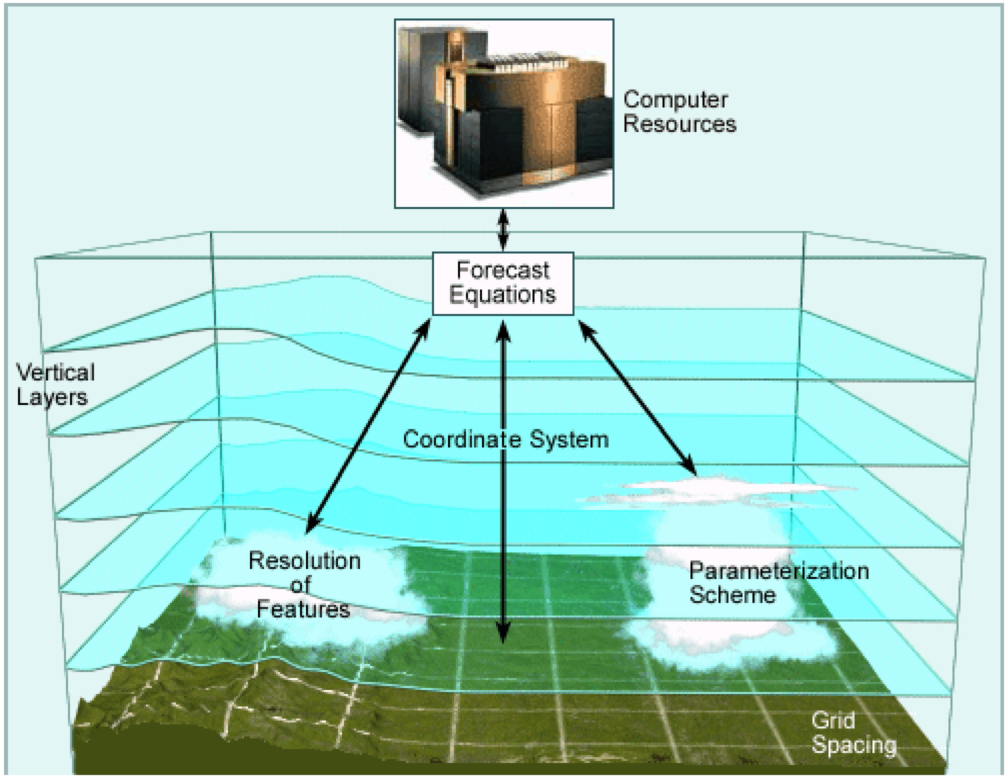


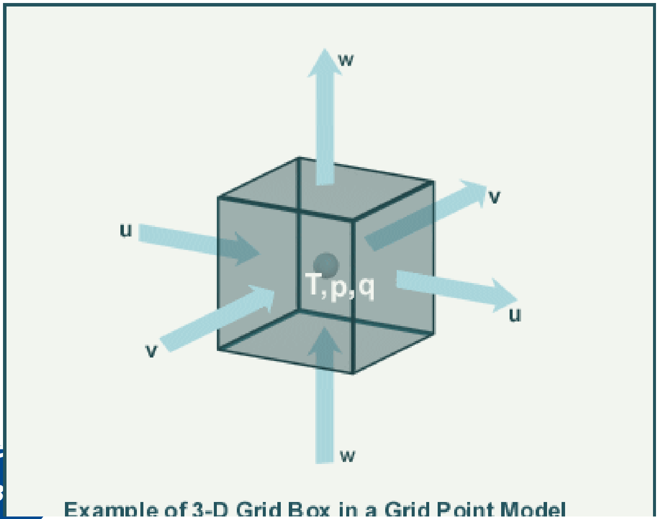


- Effects:
  - It can cause illness and even death.
  - It damages buildings, crops, and wildlife.
  - It has a strong impact in visibility
  - Impact on climate system

Where do we solve the primitive equations? Grid discretization



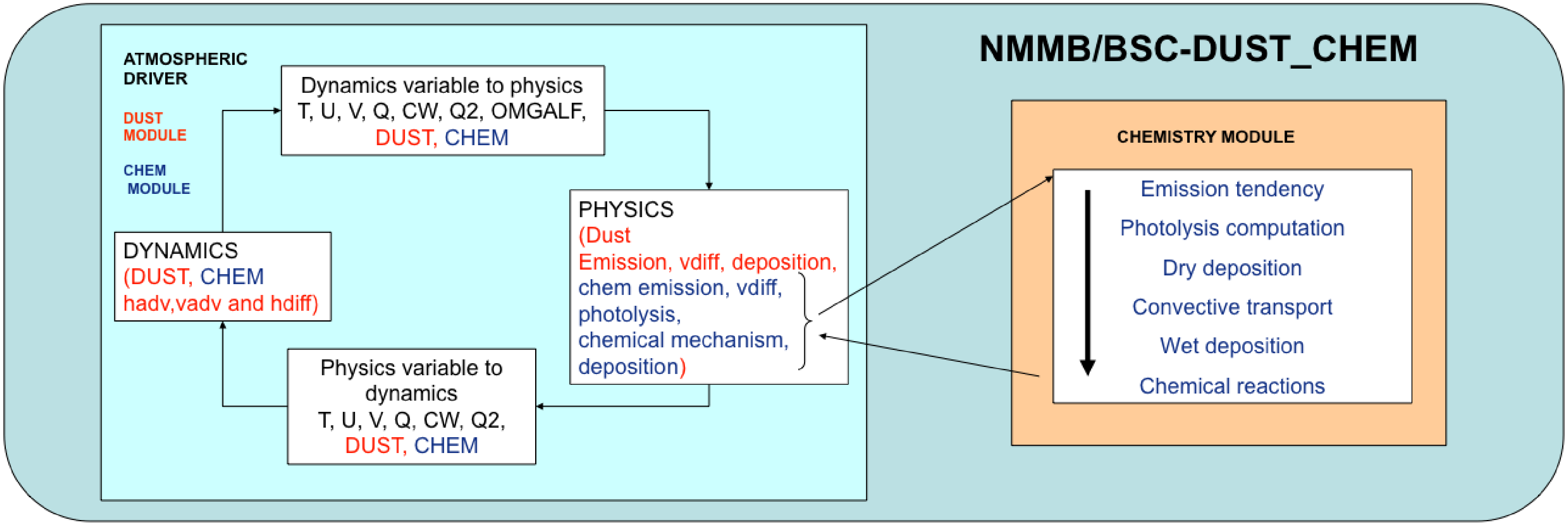


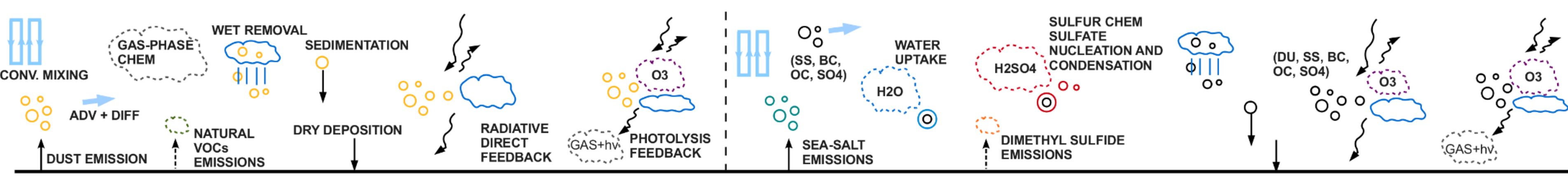


High performance computing resources:

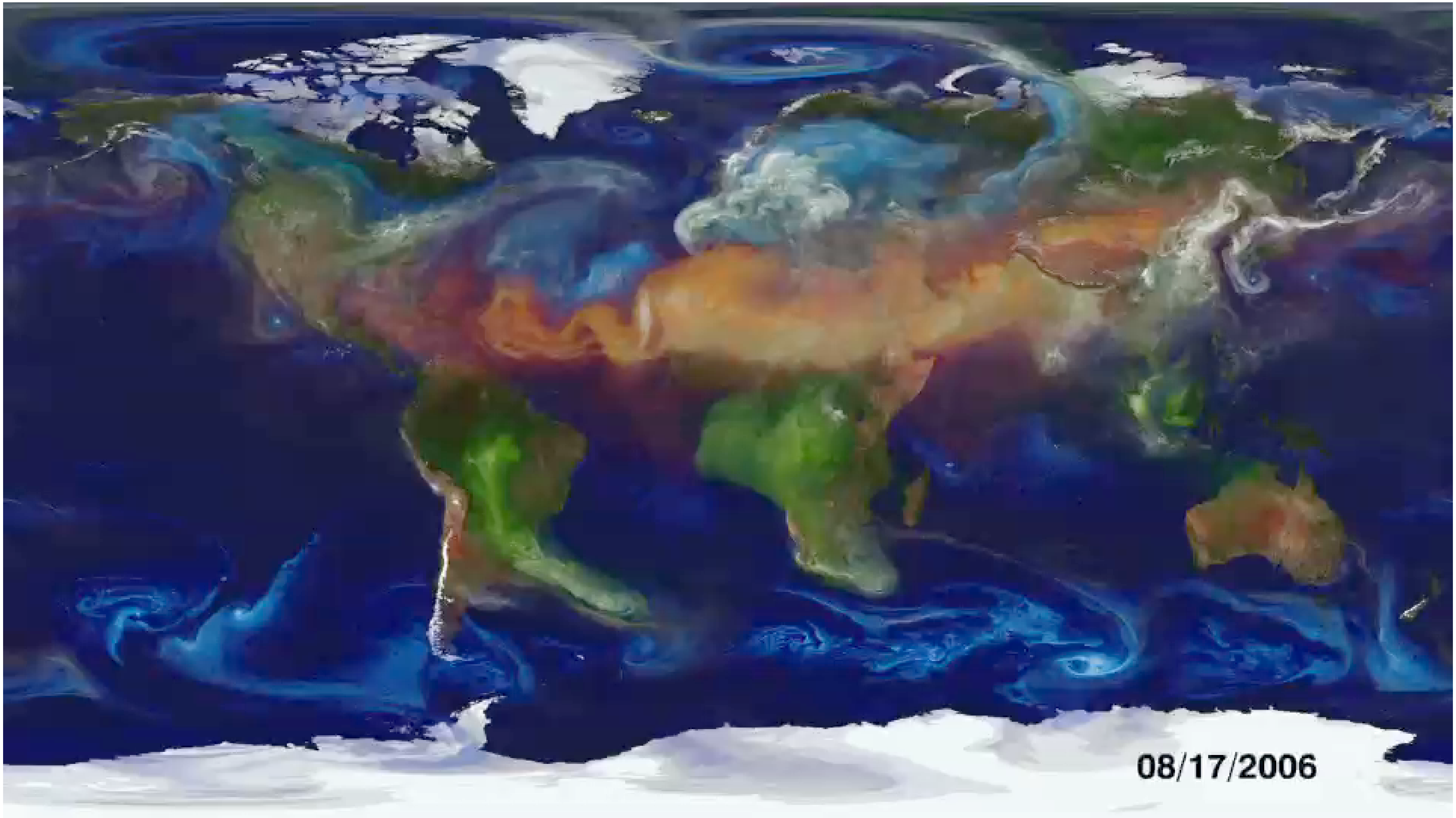
If we plan to solve small scale features we need higher resolution in the mesh and so more HPC resources are required.

Embedding chemistry processes within a meteorological core driver





Global aerosol simulation



Source: NASA GSFC

Types of simulations

- Climate Simulations
  - Global scale
  - Large periods
  - Huge amount of data created
  - Execution time is not a critical constraint
  - Example: EC-EARTH model for 1900 to 2100, year simulation

- Operational Simulations
  - Global/Regional Scale
  - Small periods
  - Data created is smaller but postprocess products are more important
  - Execution time and reliabilty are very critical
  - Example: Daily weather forecast

Setting up a model

- A model is a collection of source codes
- We need to compile to build an executable
- The executable will run and produce results

- Usually, models have a building producedure
  - Configure
  - Makefiles
  - Scripting…

Computational demands

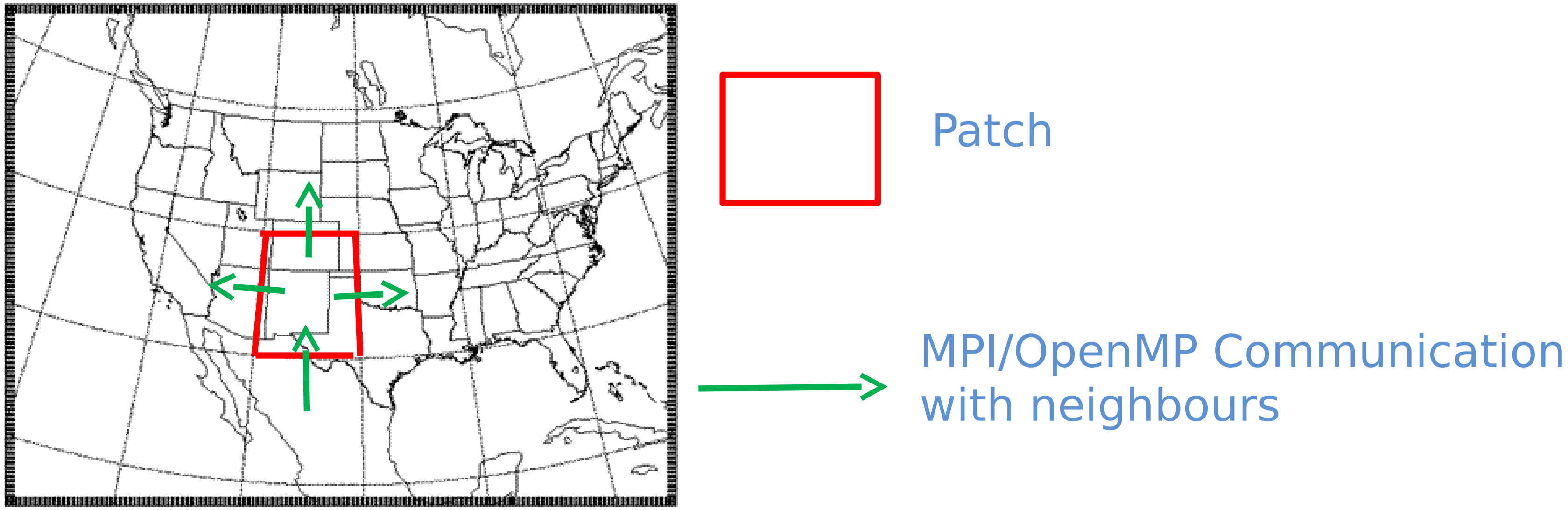
- Which domains are we simulating ¿?
  - Barcelona
  - Spain
  - World
- Which resolution ¿?
  - 1 km2
  - 4 km2
  - 12 km2
  - 50 km2
- How many variables we want to compute ¿?
  - T2
  - U10, V10
  - QRAIN, QVAPOR
- Increasing this parameters, increases the system constraints
  - Computation Needs (CPU’s, Memory Bandwith…)
  - Data Storage
- Define this parameters in function of your hardware and time to serve forecast.

Parallelizing Atmospheric Models

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

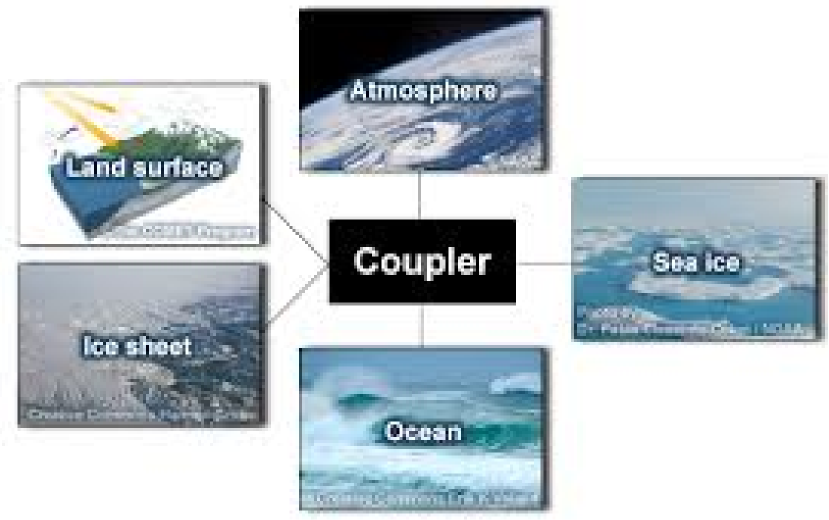
- Model domain is decomposed in patches

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



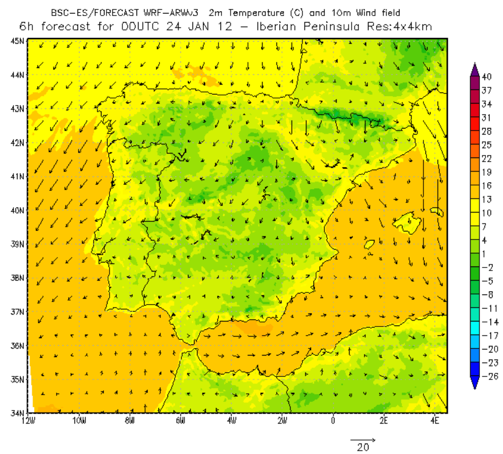
Couplers

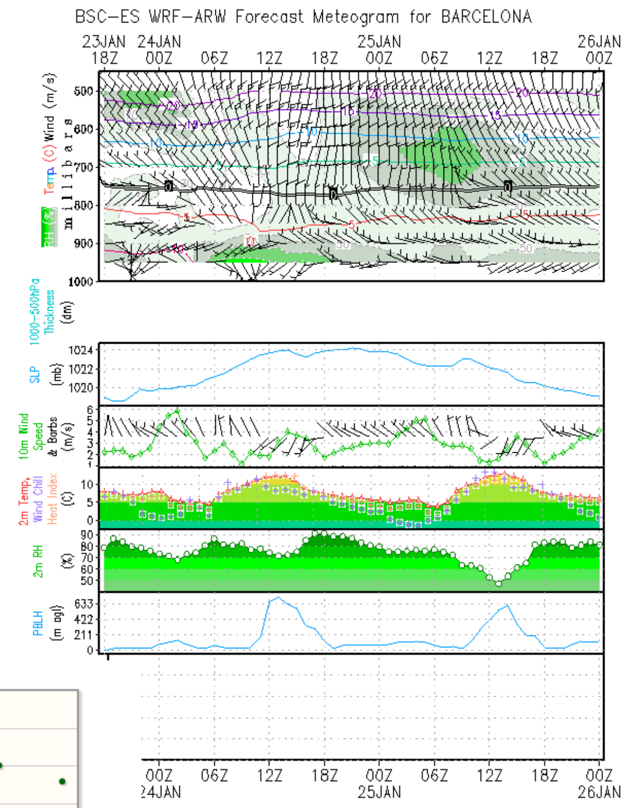
- What is the role of a coupler ?
  - Exchange and transform information through two or more diferent models.
  - Manage the execution and synchronization of the codes.
  - Example: couple an ocean model and atmosphere.

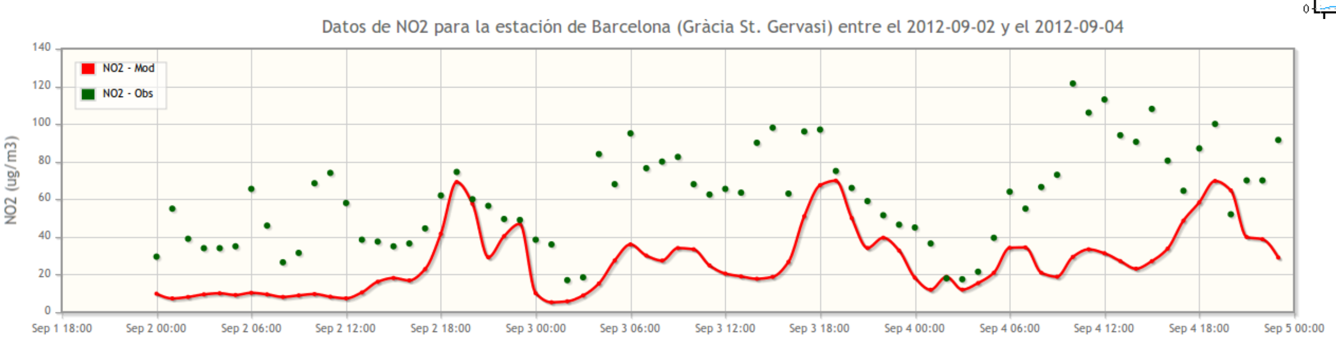


Post-processing

- Once the model is run successfully, we need to post-process results to visualize data
  - Maps
  - Plots
  - Text files
  - 3D Animations



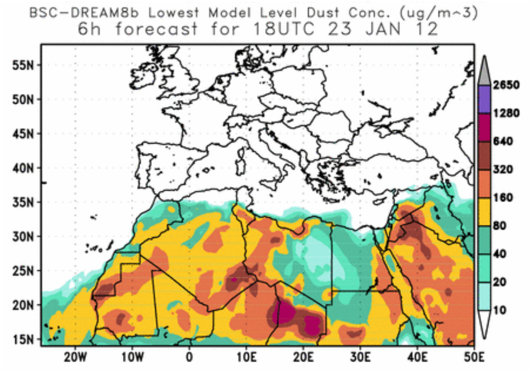




Models at BSC

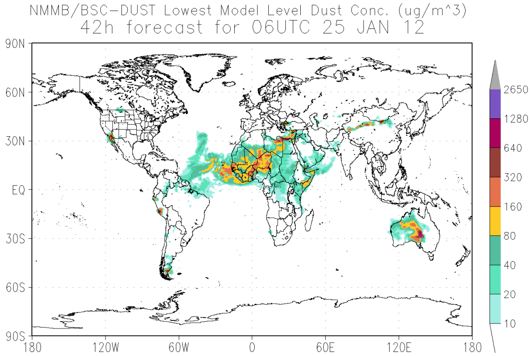
Mineral Dust Modeling

- BSC-DREAM8b V2: Dust REgional Atmospheric
  - Model
  - Fortran Code
  - Not parallel



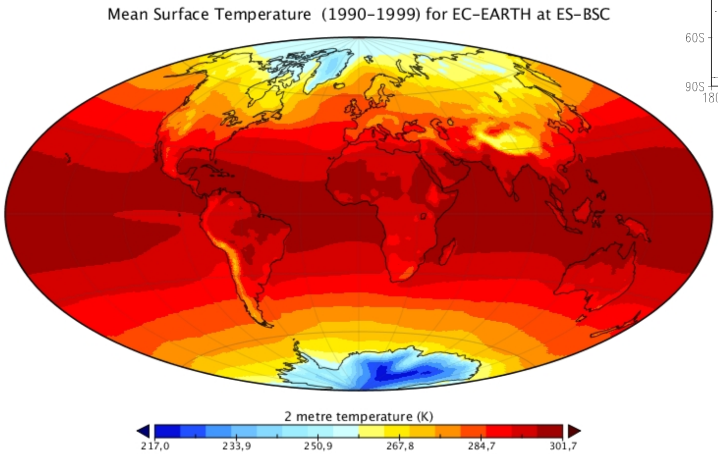
NMMB/BSC-CTM

- Meteorology-Chemistry coupled model
  - Meteo. Driver: Nonhydrostatic Multiscale
  - Model on the B grid (NMMB)
  - Fortran Code
  - MPI

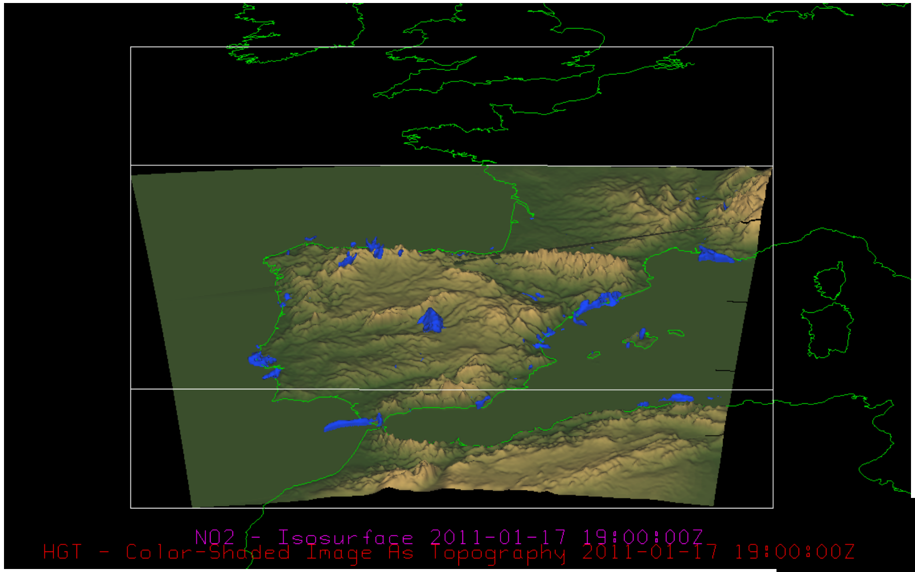


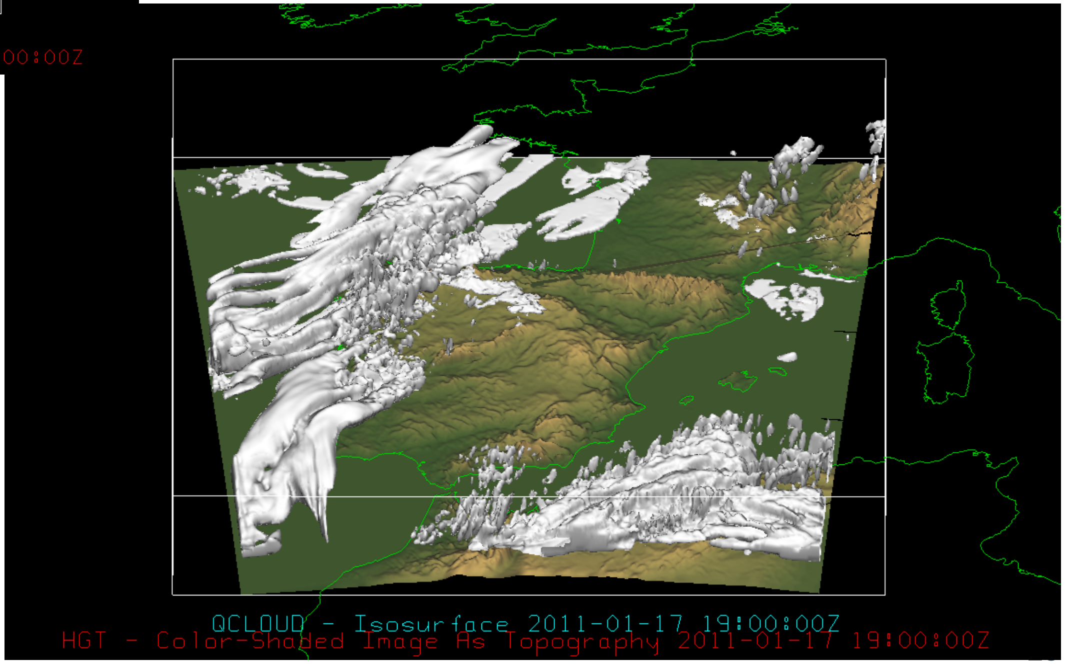
Climate Change

- EC-EARTH
  - Fortran, C
  - MPI, OpenMP



3D Outputs



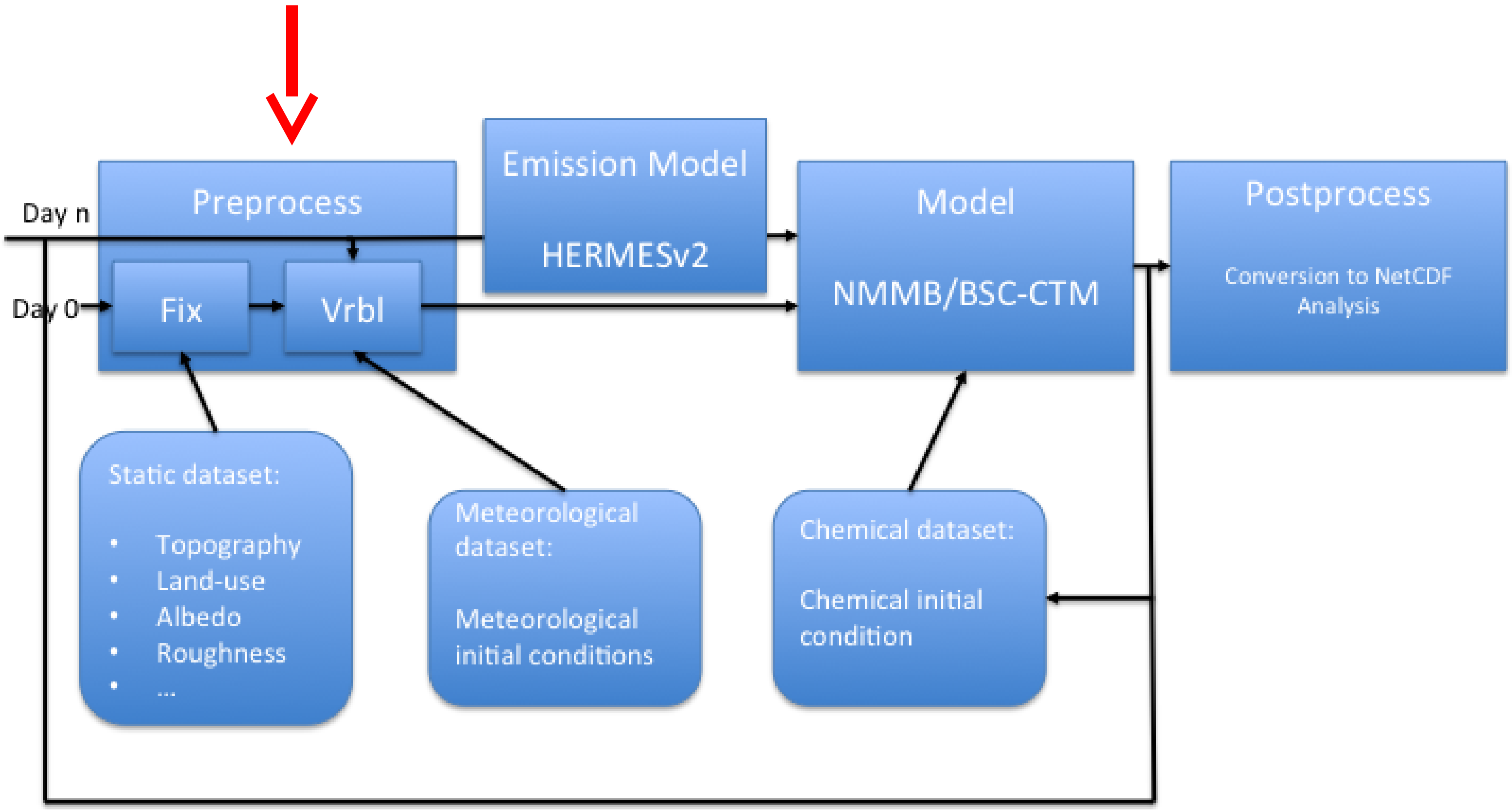


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Preprocess

Execution diagram: Focus on the Preprocessor



Tested in two cases: Global domain 1ºx1.4º resolution
Global domain 12km x 12km

CompSs

- COMPSs programming model intends to maximize the programmability of Java applications running on parallel and distributed infrastructures.
- COMPSs is fully developed at BSC.

Original Preprocess

- Preprocess is divided in two main tasks:
  - Fixed: which is only done once, when configuring the model
  - Variable: is done each run, as takes daily meteorological and surface sea temperature inputs.
  - Fixed and Variable are now run separately.
- Totally sequential, synchronous, ignore data dependencies between subprocesses.

```
#FIXED
./exe/smmount.x
./exe/landuse.x
./exe/landusenew.x
./exe/topo.x
./exe/stdh.x
./exe/envelope.x
./exe/topsoiltype.x
./exe/botsoiltype.x
./exe/toposeamask.x
./exe/stdhtopo.x
./exe/deeptemperature.x
./exe/snowalbedo.x
./exe/vcgenerator.x
./exe/roughness.x
./exe/gfdlco2.x
./exe/lookup_aerosol.x
```

```
#VARIABLE
ln -s ../meteo_data/wafs.00.0P5DEG.13042400.grib1
../output/gfs.t00z.pgrb2f00
ln -s ../meteo_data/sst2dvar_grb_0.5.13042400.grib1
../output/sst2dvar_grb_0.5

./degribgfs_generic_05.sh 00 00 03 pgrb2f ../output
./exe/gfs2model_rrtm.exe 00
./exe/inc_rrtm.x
./exe/cnv_rrtm.x
./exe/degribsst.x
./exe/albedo.x
./exe/albedorrtm1deg.x
./exe/vegfrac.x
./exe/z0vegustar.x
./exe/allprep_rrtm.x
./exe/read_paul_source.x
./exe/dust_start.x
```

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Original Performance

- The executions are done in MareNostrum3.
- Compiled with ifort compiler,
  - `FFLAGS=“-mcmodel=large -shared-intel -convert big_endian -traceback -assume byterecl -O3 -fp-model precise -fp-stack-check”`
- 9.3 Gb statical data required (geodata and GTOPO30 databases)
- Runtime for the global operational domain:
  - Fixed: 7m30s
  - Variable: 0m32s

Porting to COMPSs (I)

- Preprocess is a collection of Fortran codes.
- In order to port to COMPSs, we need to modify sources to manage files as arguments instead of being hardcoded.
- Example:
  - **smmount** creates two files, *seamaskDEM* and *heightDEM*.
  - With COMPSs, smmount is executed with files as arguments
    - ./smmount ../output/seamaskDEM ../output/heightDEM

- Fortran source code is modified to handle arguments.
- Each executable is wrapped in a Java method and selected as a task.
- This method is not hard to code, but **allprep** executable in variable, manages more <u>than 44 files </u>!!!

Porting to COMPSs (II)

- Then, three files are written in JAVA:
  - *Fixed.java*: main program of the application, contains task calls.
  - *FixedBinaries.java*: implementation of each task with the call to the executable.
  - *FixedItf.java*: selection of tasks, providing the necessary metadata about their parameters.

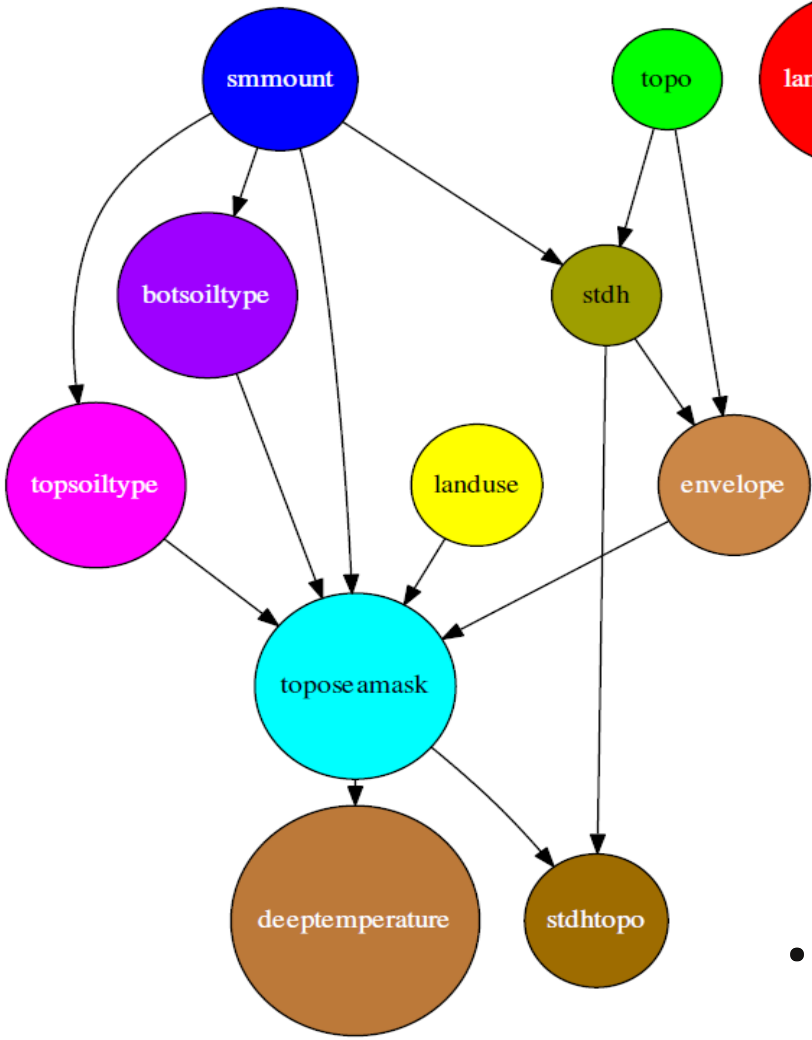
- The same files are written for Variable.

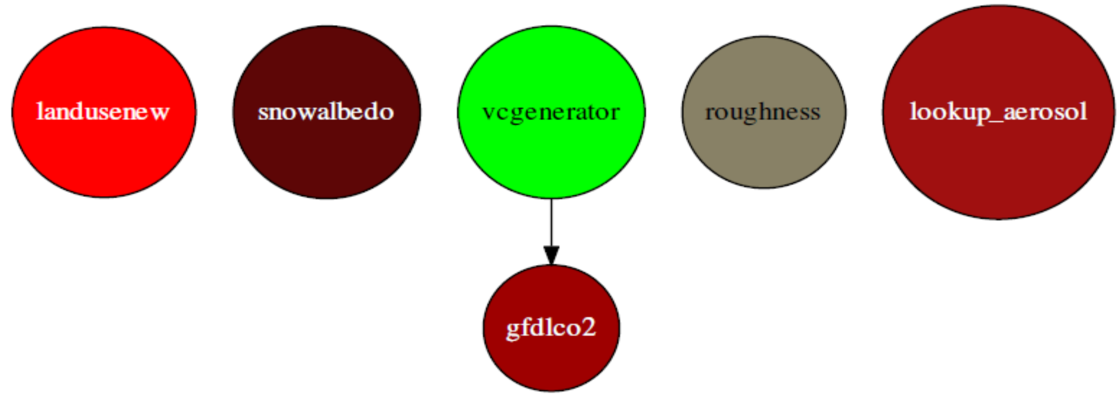
Execution

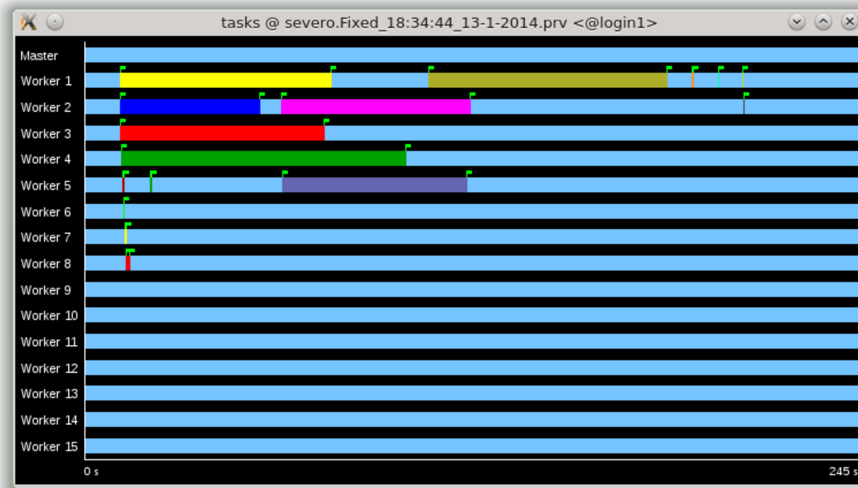
- We implemented a Fortran/MPI application only for the Fixed preprocess, using 5 cores of one node based on the dependency graph acquired from CompSs.

- Runtime for the global domain, 24 km:
  - Fixed: 2m30s.

Fixed – COMPSs

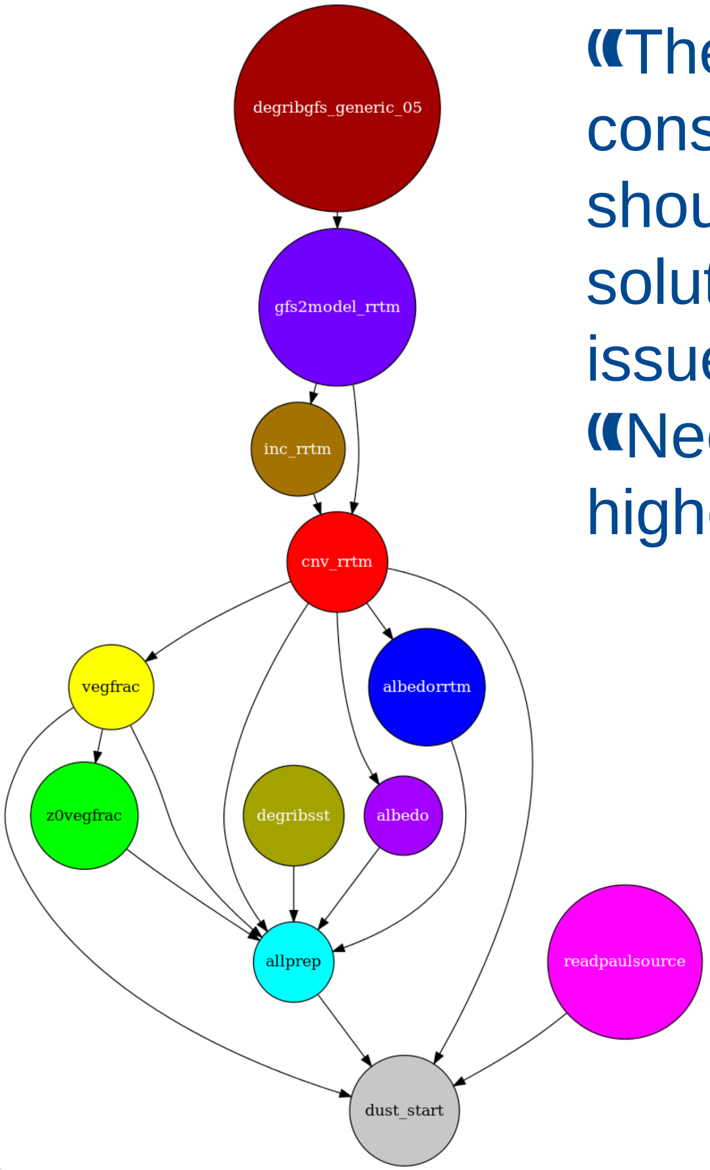






- We used 5 cores and achieved a speedup of 2.7 times!

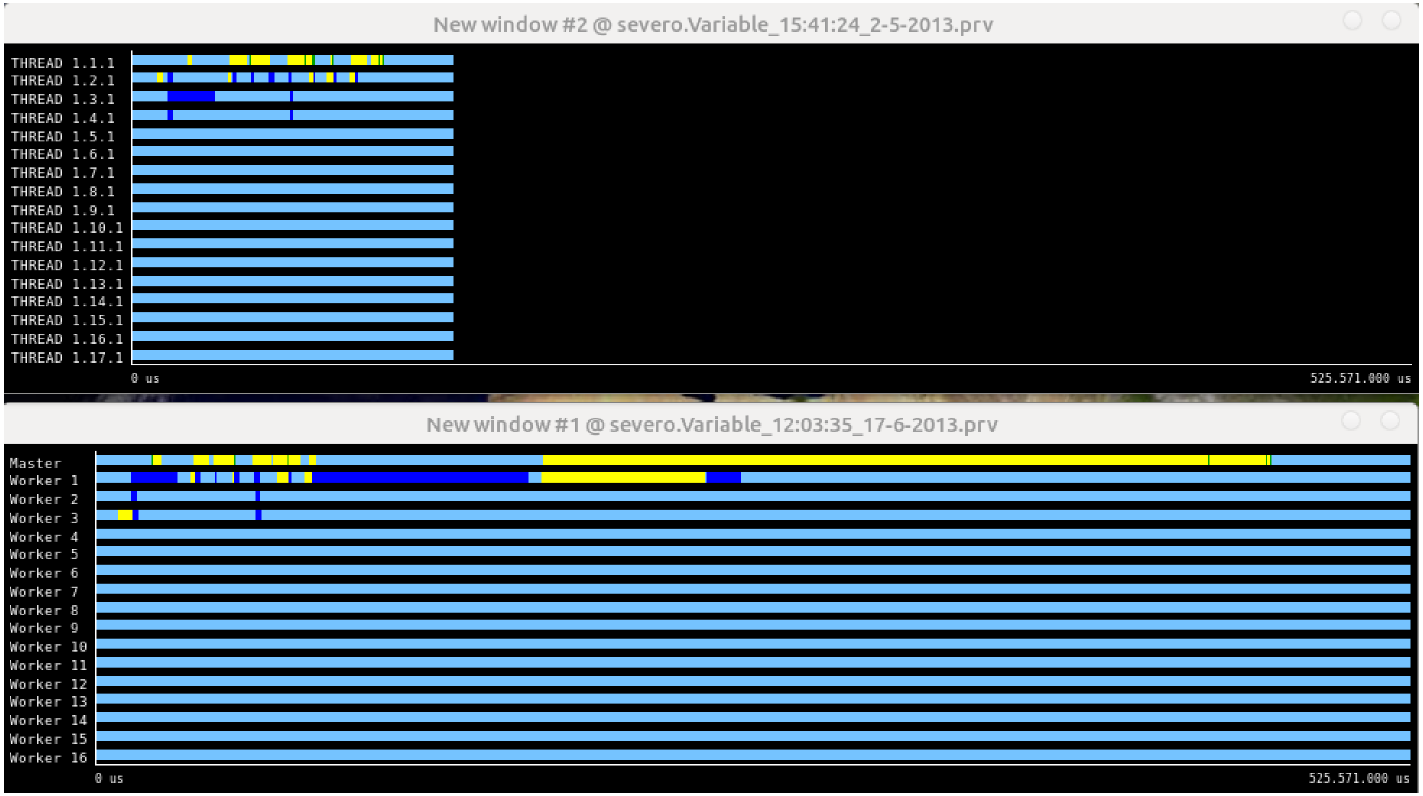
Variable – COMPSs



- The serial part *allprep* consumes a lot of time, we should investigate a hybrid solution because of memory issues
- Need to be improved for higher resolution forecasts

Test on a bigger case

- We applied this method to generate 12km global resolution input files (more than 6GB output files)



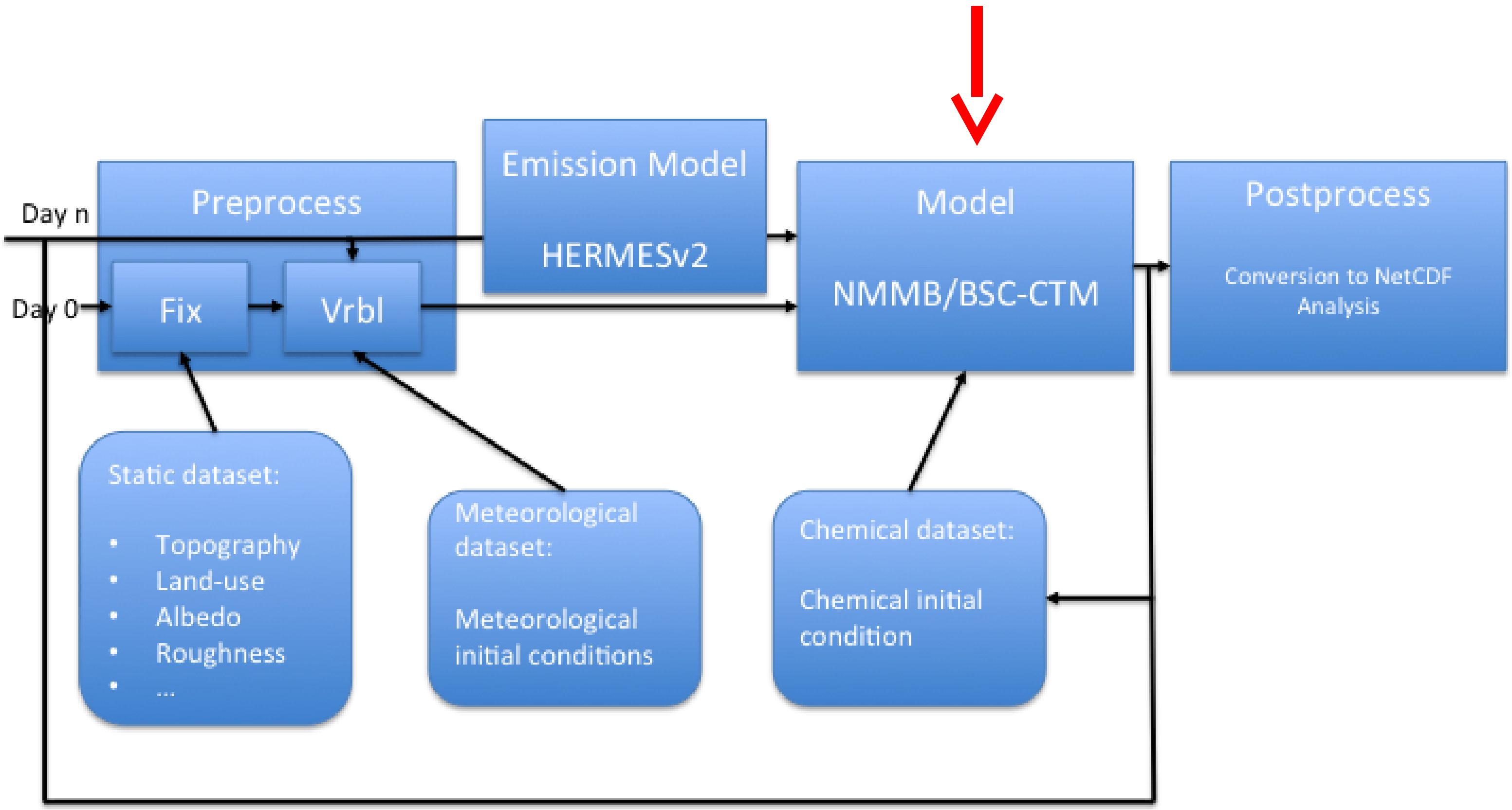
Execution Remarks

- Data dependencies between tasks are automatically detected, thus exploiting the inherent concurrency of the application when executing the tasks.
- In the Fixed application, 8 tasks are free of dependencies at the beginning, and therefore they can be sent for execution immediately.
- Performance
  - Fixed: the exploitation of task parallelism speeds up the process.
  - Variable: it has little computation and parallelism, which does not compensate the overhead of task processing and distribution (e.g. dependency analysis, file transfer, task submission), hence incrementing the execution time.

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Performance Analysis of NMMB/BSC-CTM Model

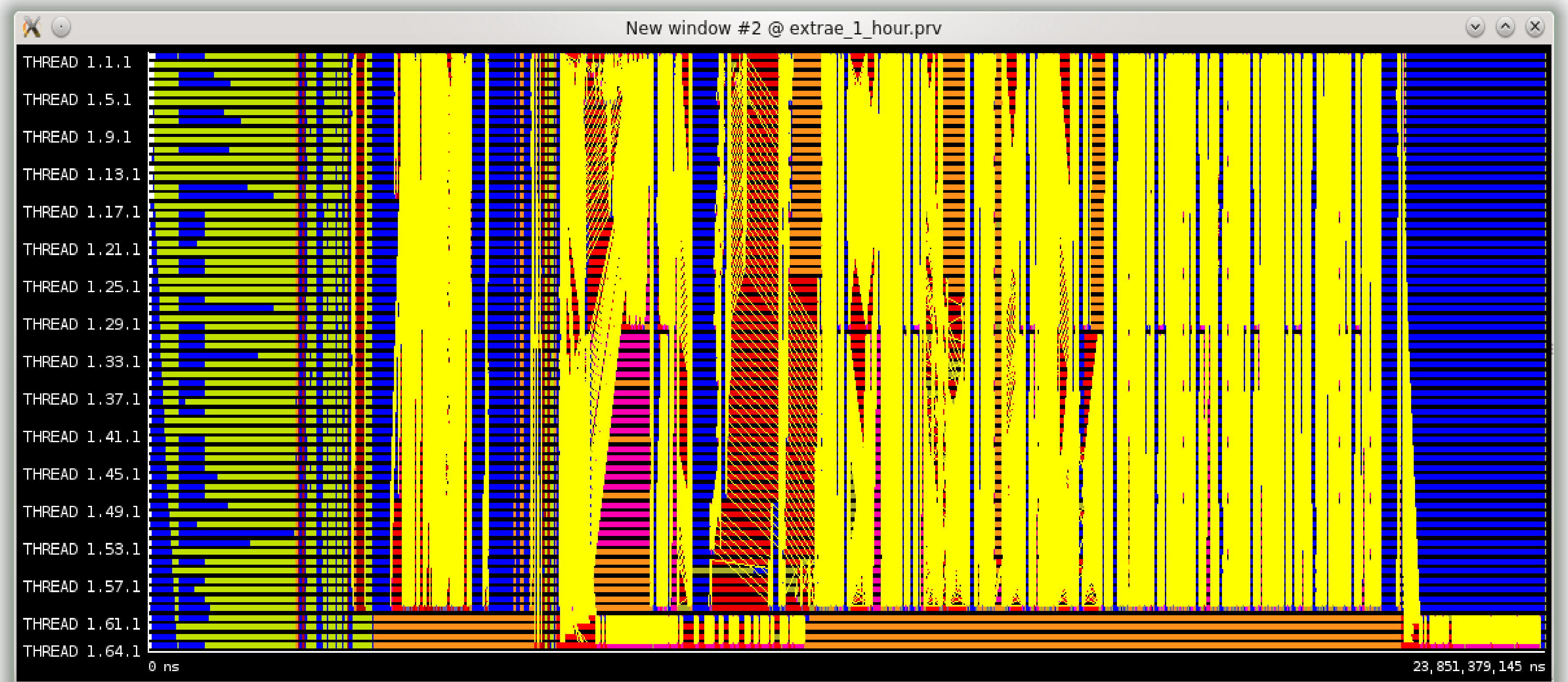
Execution diagram – Focus on the Model



Study domain: Global domain 24km x 24km resolution

Paraver

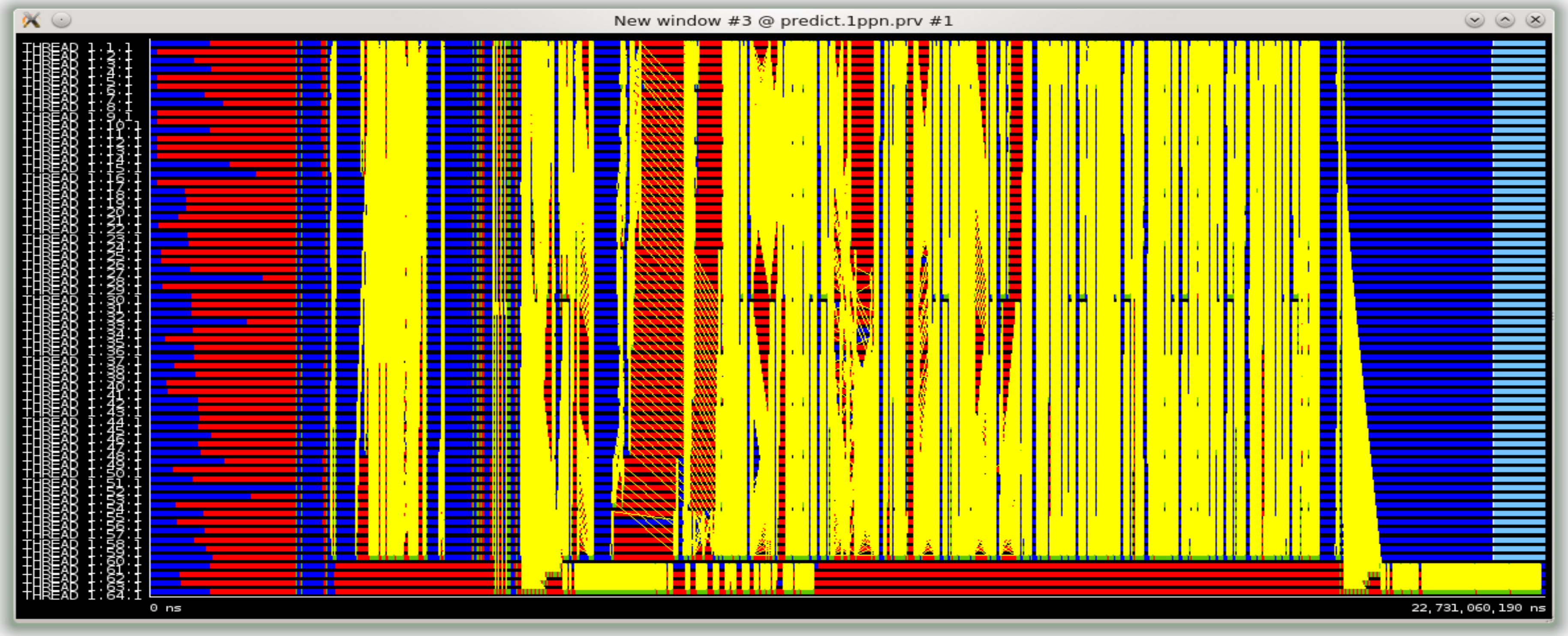
- One hour simulation of NMMB



- Last four processes are used for I/O

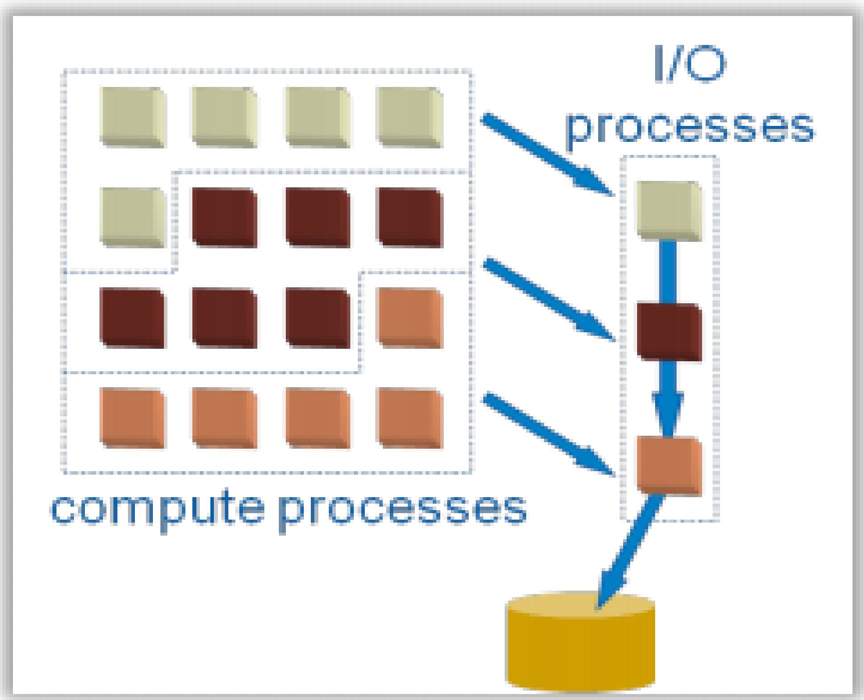
Paraver - Dimemas

- It seems that previously there was noise during the execution

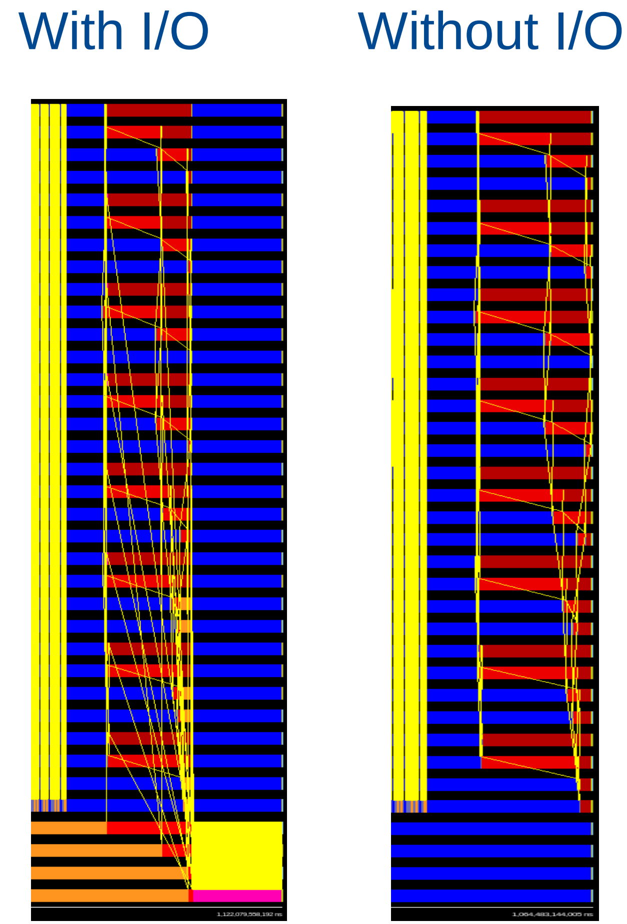


Issue with I/O

- There is no parallel I/O implemented!



Last hour



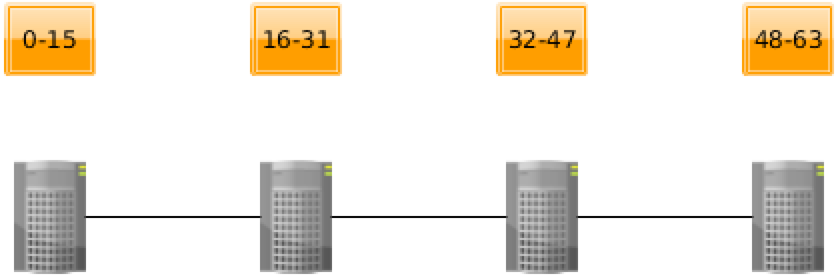
Issue with the last binary file

- Last binary is written with delay.
- Example regional 11km resolution

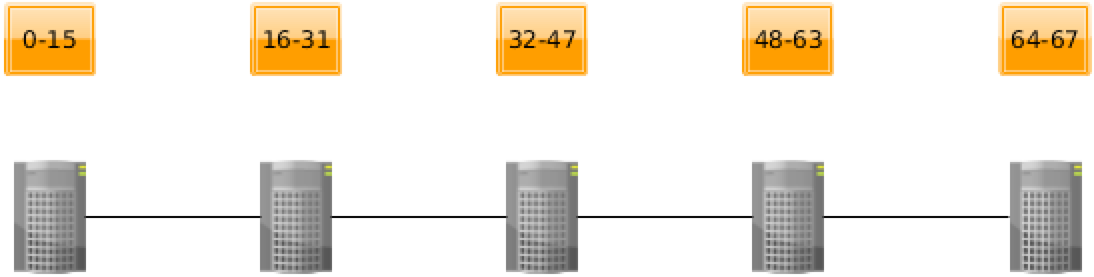
```
4778176548 Dec 15 09:25 nmmb_hst_01_bin_0000h_00m_00.00s
4778176548 Dec 15 09:28 nmmb_hst_01_bin_0001h_00m_00.00s
4778176548 Dec 15 09:31 nmmb_hst_01_bin_0002h_00m_00.00s
4778176548 Dec 15 09:34 nmmb_hst_01_bin_0003h_00m_00.00s
4778176548 Dec 15 09:38 nmmb_hst_01_bin_0004h_00m_00.00s
4778176548 Dec 15 09:41 nmmb_hst_01_bin_0005h_00m_00.00s
4778176548 Dec 15 10:42 nmmb_hst_01_bin_0006h_00m_00.00s
```

Issue with I/O – Mapping

- Initial mapping for an experiment with 64 cores where the last 4 ranks are the write tasks



- Final mapping



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

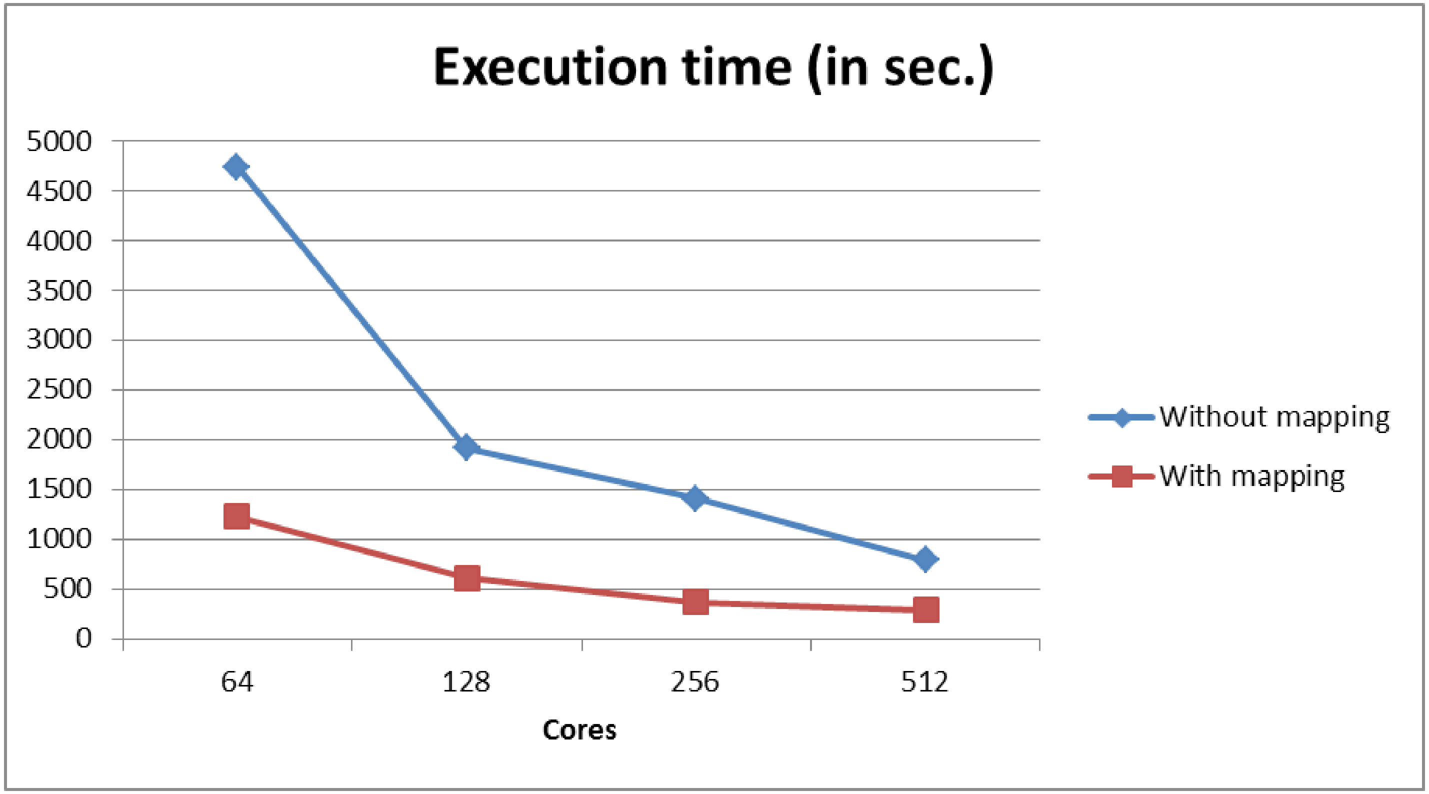
Issue with the last binary file solved

- The instrumented execution has no issue…

```
4778176548 Dec 15 11:14 nmmb_hst_01_bin_0000h_00m_00.00s
4778176548 Dec 15 11:17 nmmb_hst_01_bin_0001h_00m_00.00s
4778176548 Dec 15 11:21 nmmb_hst_01_bin_0002h_00m_00.00s
4778176548 Dec 15 11:24 nmmb_hst_01_bin_0003h_00m_00.00s
4778176548 Dec 15 11:27 nmmb_hst_01_bin_0004h_00m_00.00s
4778176548 Dec 15 11:30 nmmb_hst_01_bin_0005h_00m_00.00s
4778176548 Dec 15 11:33 nmmb_hst_01_bin_0006h_00m_00.00s
```

Performance of different mapping and more I/O servers

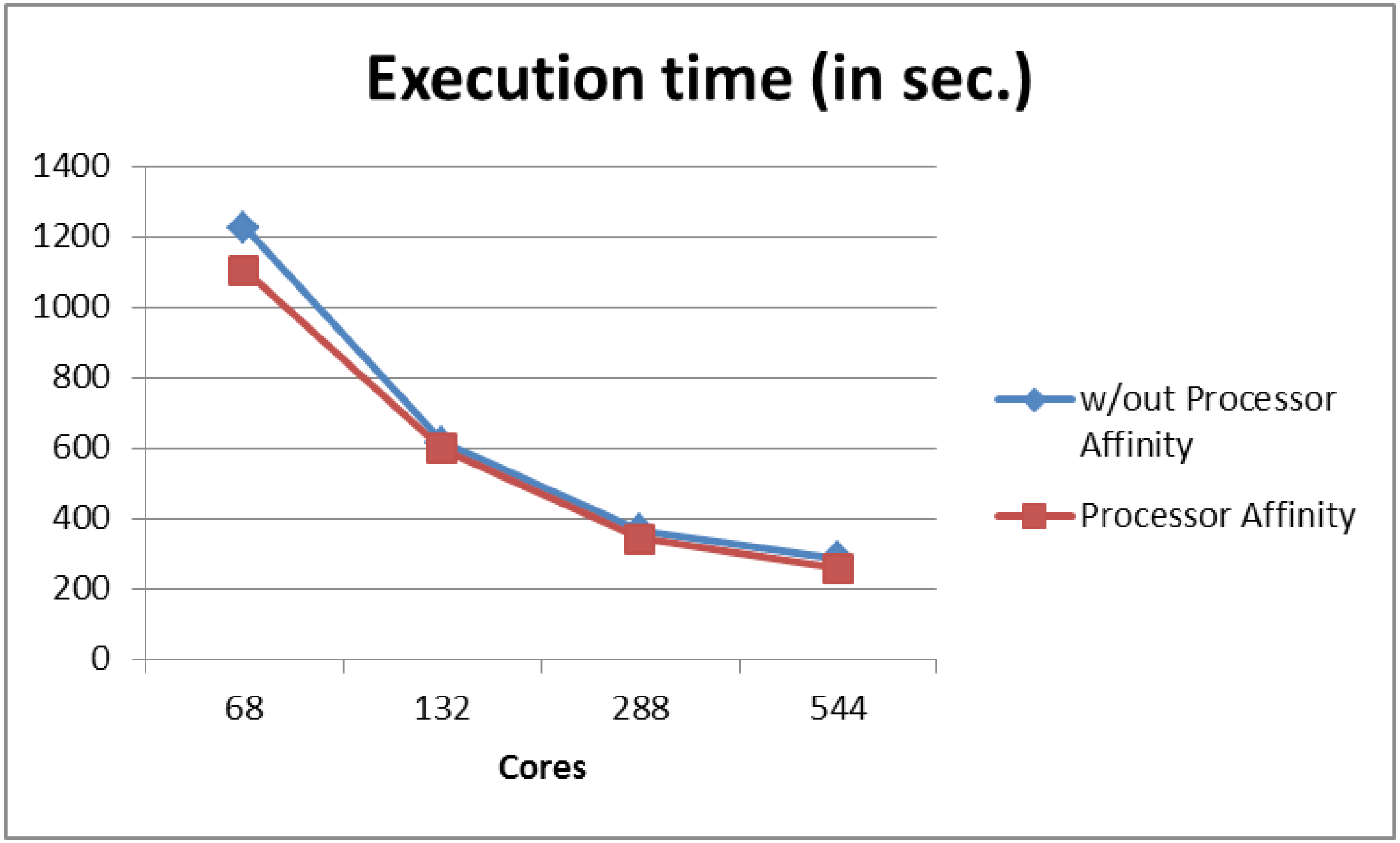
- The new mapping improved the execution time between 2.73 and 3.85 times



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

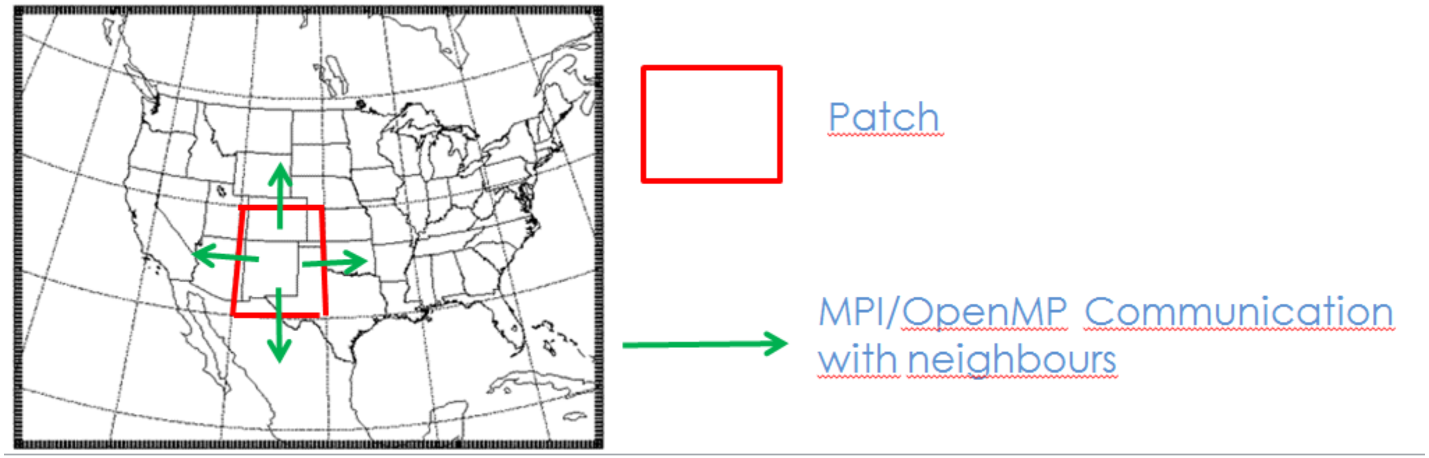
Processor Affinity

- Processor affinity improved the execution time between 2.8% and 10% (some colleagues reported 20% improvement)



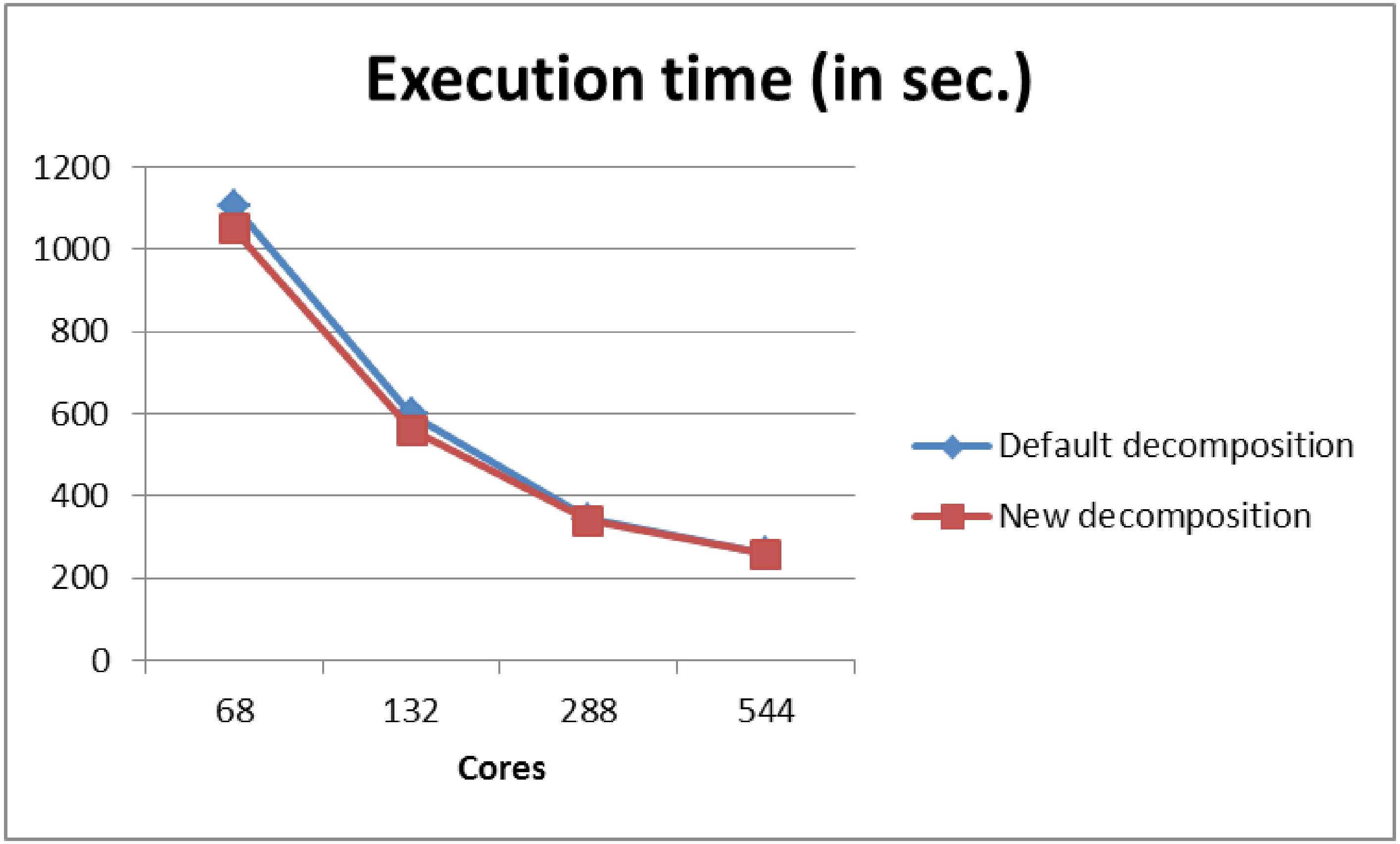
Decomposition (X,Y)

- Usually we use a square decomposition or something close to square.
- It is better to use values to a more rectangular decomposition (i.e. X<<Y). This leads to longer inner loops for better vector and register reuse, better cache blocking, and more efficient halo exchange communication pattern.



Decomposition

- New decomposition improved the execution time till 6.5%



Throttling mechanism

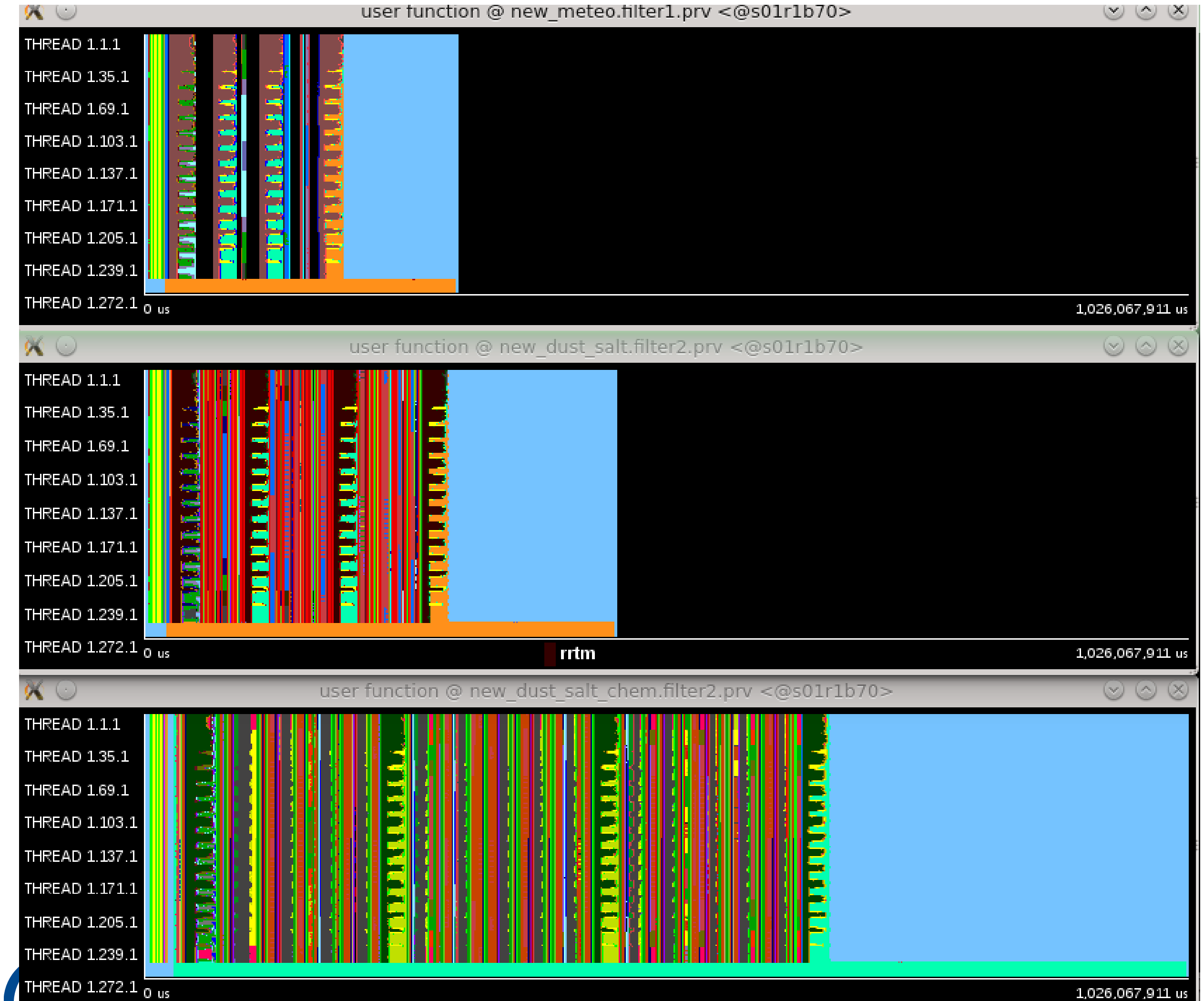
- An application is developed for many years and some times the scientists are not located anymore in the department

- Use gprof (-pg) to figure out number of calls and duration of functions

- Use Intel Fortran compiler with “-g -finstrument-functions” option and create a function list with the following rule, do not instrument the functions that are executed more than 10,000 times and the duration of each call is less than 1ms or 0%
  For example:
  00000000008c0230 # module_dynamics_routines_mp_hdiff_

Paraver

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

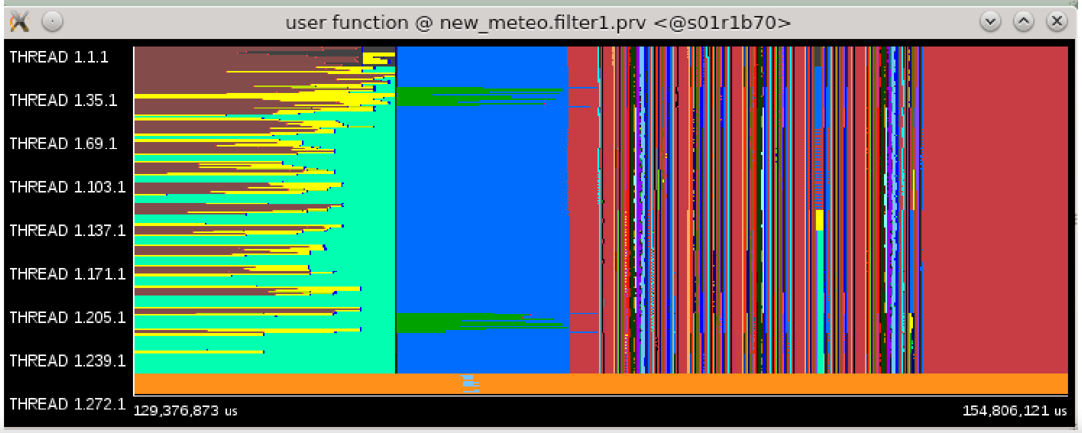


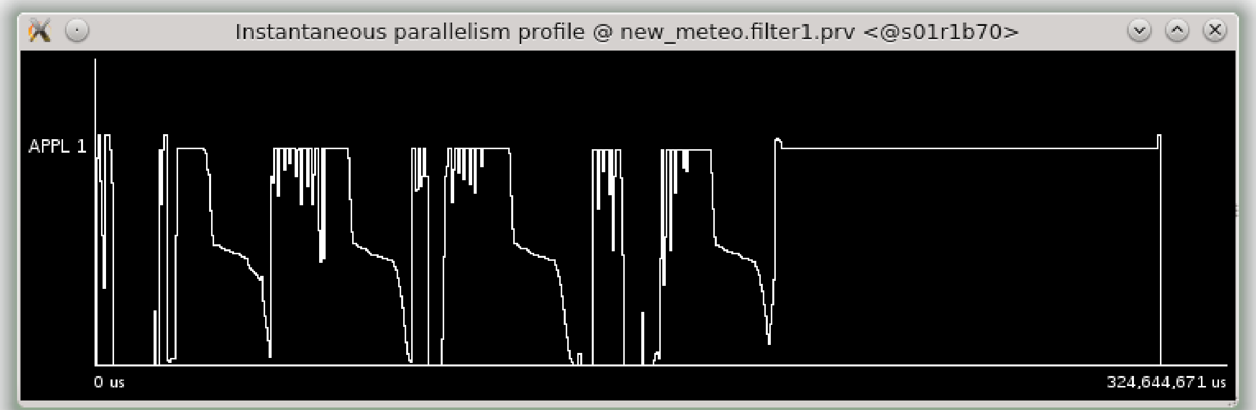
meteo: 9 tracers

meteo + aerosols: 9 + 16 tracers

meteo + aerosols + gases: 9 + 16 + 53

Paraver – Useful computation - Meteo





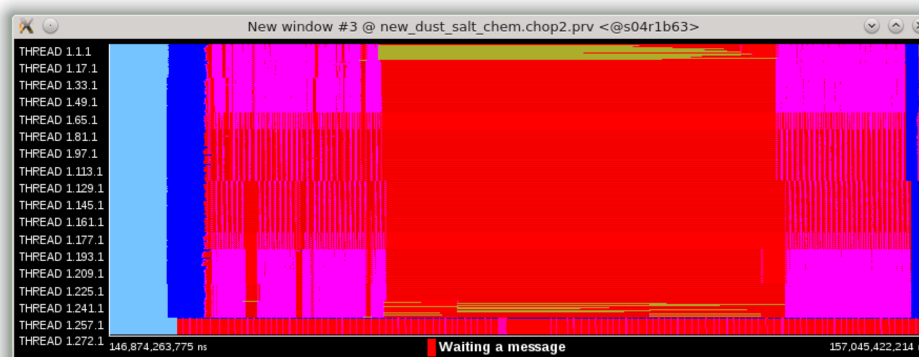
Paraver - Information about functions

- One hour simulation of NMMB, global 24km

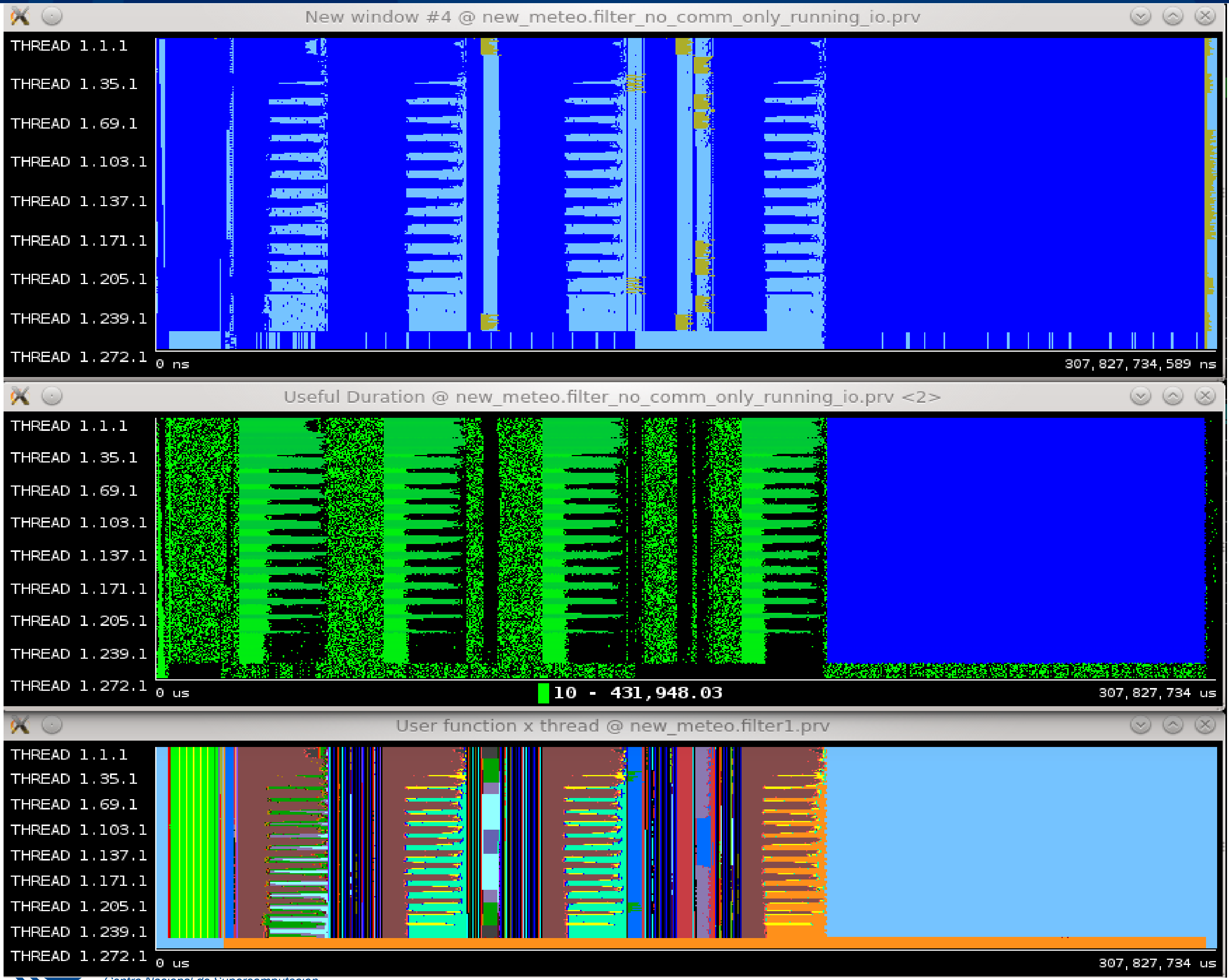
Meteo		
Functions	Percentage	IPC
rrtm	13.7% - 52% (31.3%)	2.18 - 2.38
gather_ layers	8.26% - 13.7% (11.1%)	X
scatter_ layers	10.6% - 14.1% (12.1%)	X

Meteo + aerosols		
Functions	Percentage	IPC
rrtm	8.8% - 33.4% (20.33%)	2.2 – 2.4
gather_layers	11.9% - 22% (17.4%)	x
scatter_layers	14.4% - 26.6% (19.5%)	X

Meteo + aerosols + chemistry		
Functions	Percentage	IPC
run_ebi	14% - 20.3% (16.55%)	0.71-1.11
rrtm	3.97% - 15.07% (9.05%)	2.17 – 2.37
gather_ layers	12.37% - 24.55% (16.93%)	X
scatter_ layers	14.65% - 26.58% (19%)	X

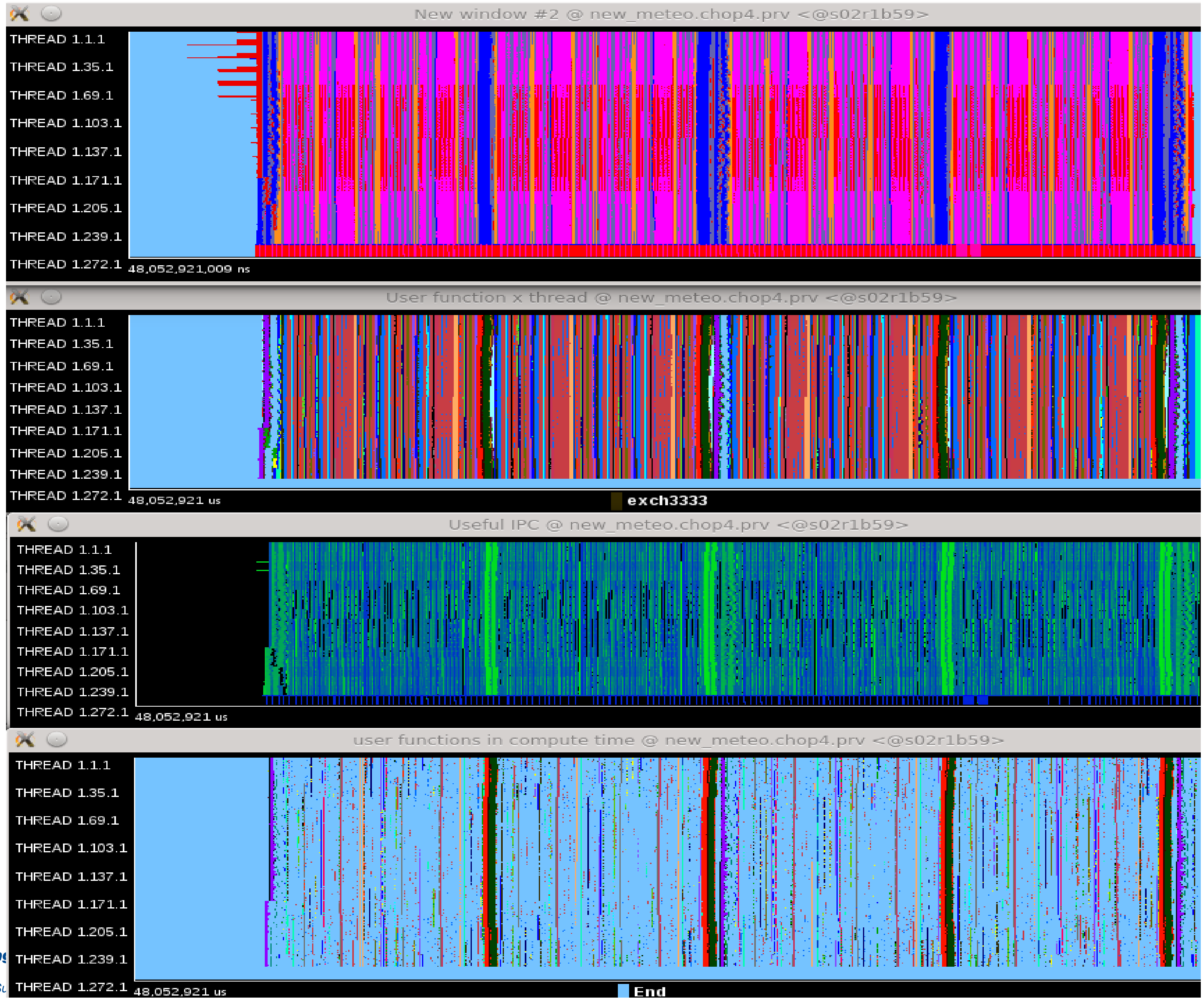


Paraver – Global – 24km - Meteo

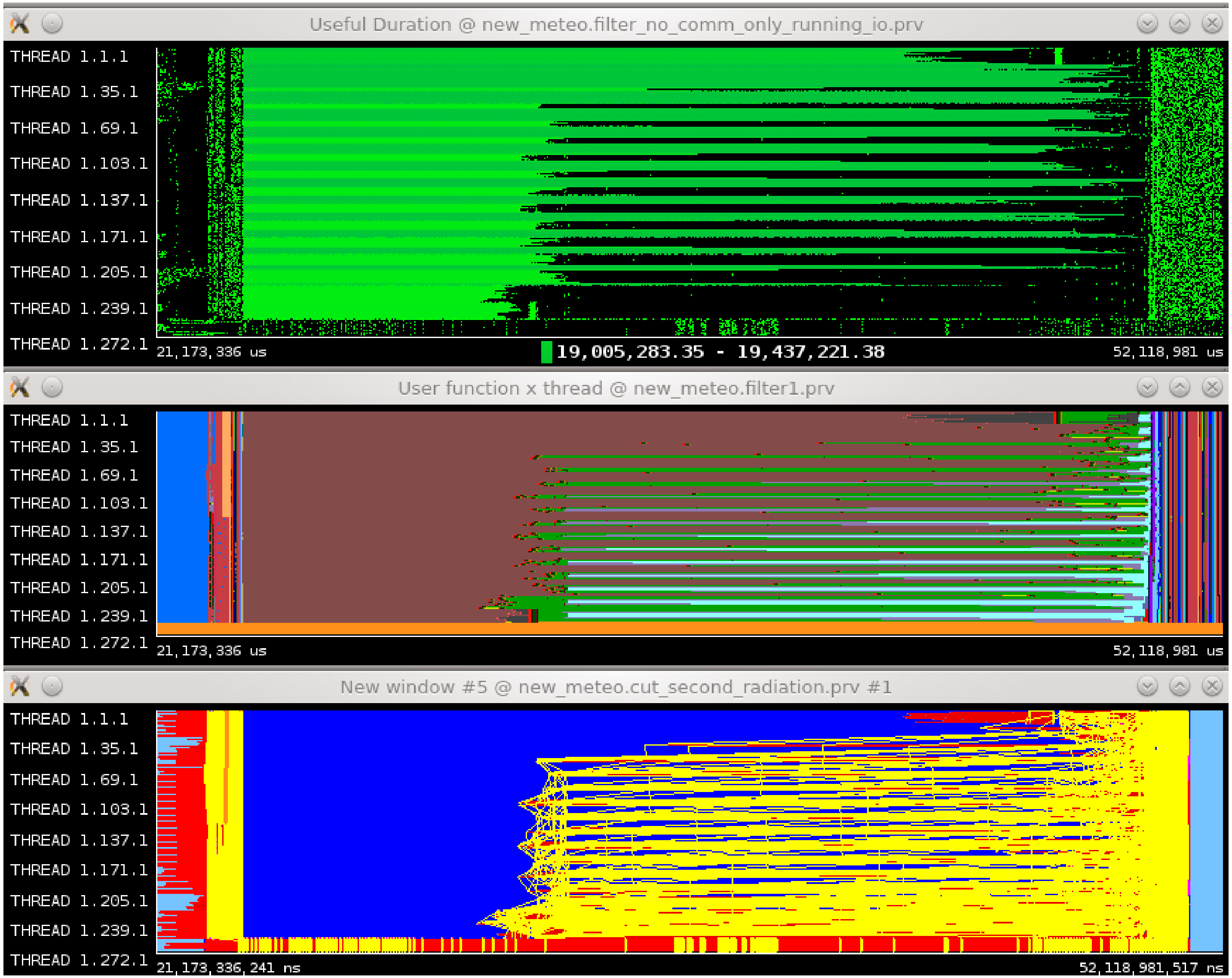


Simulation: 02/12/2005

Paraver – Global – 24km – Meteo – between radiations

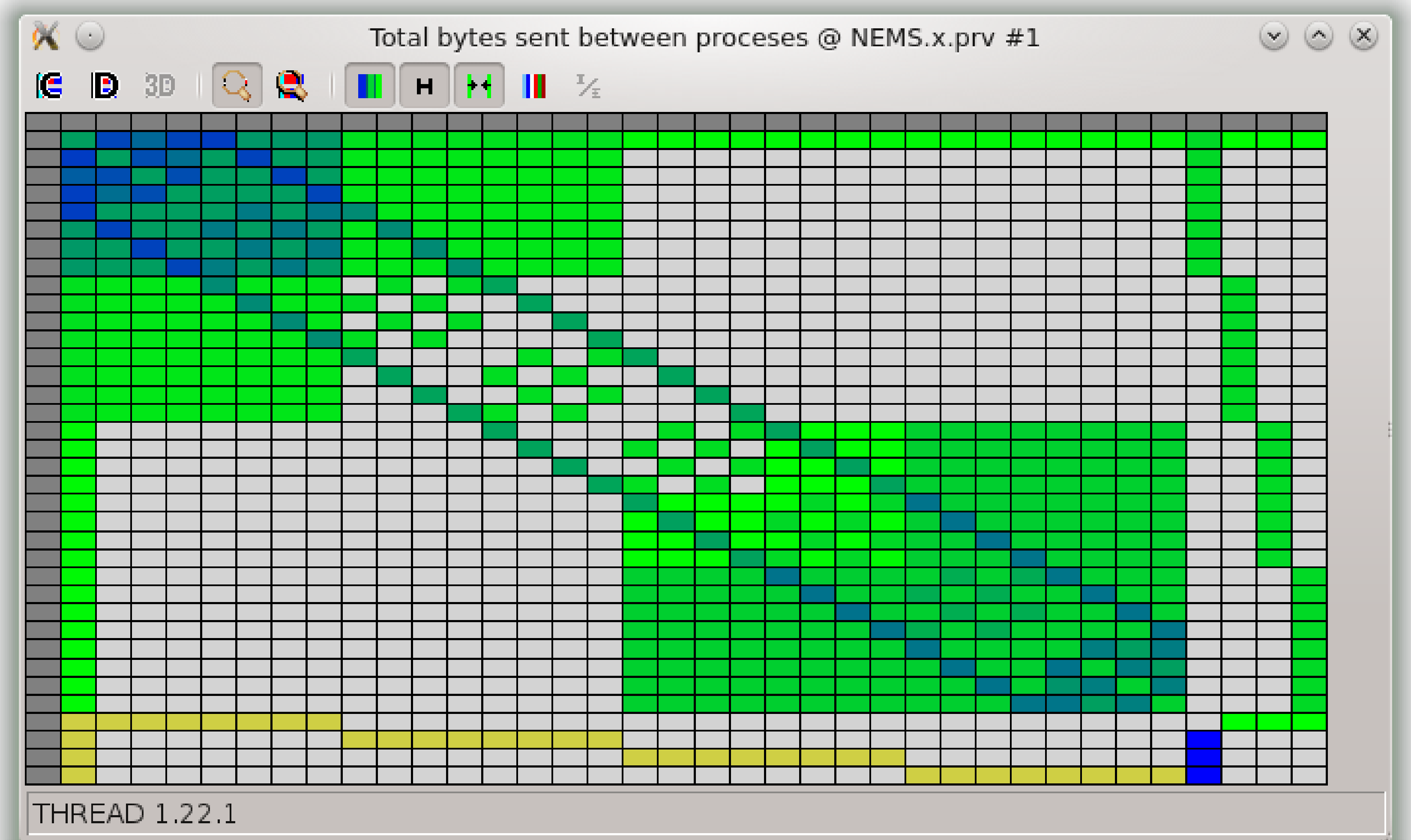


Paraver – Global – 24km – Meteo – radiation

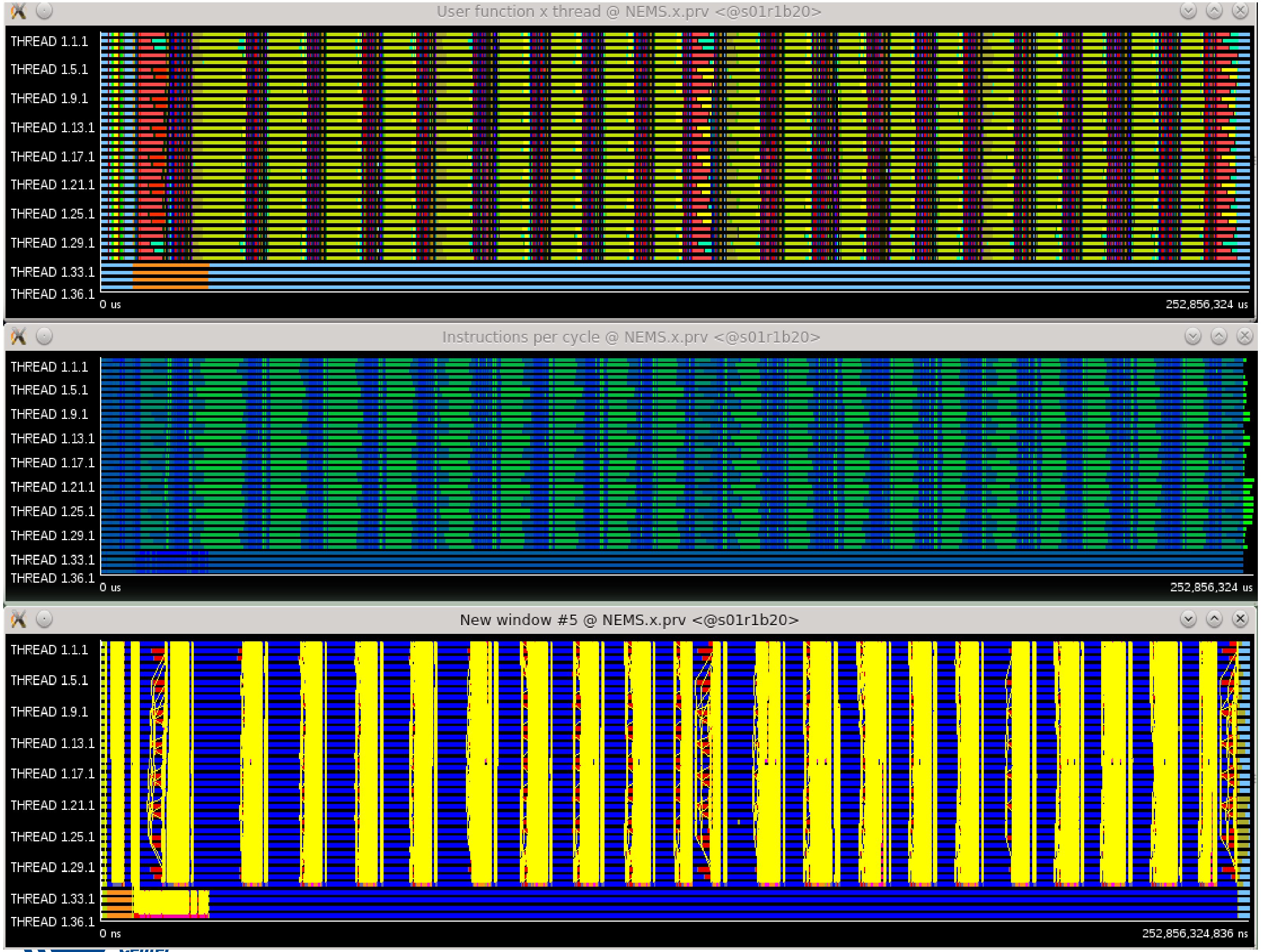


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Communication matrix

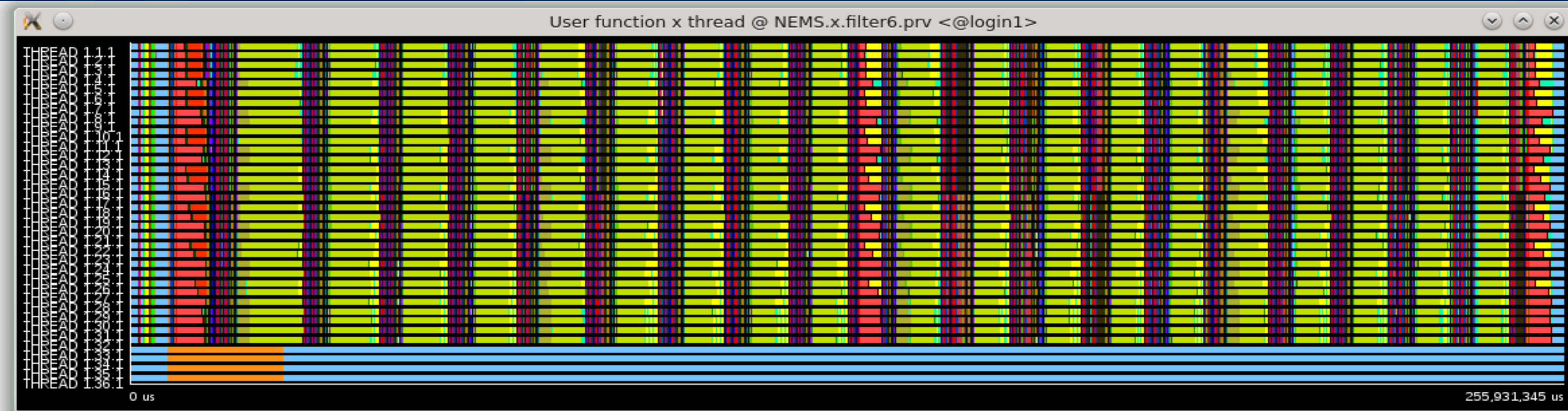


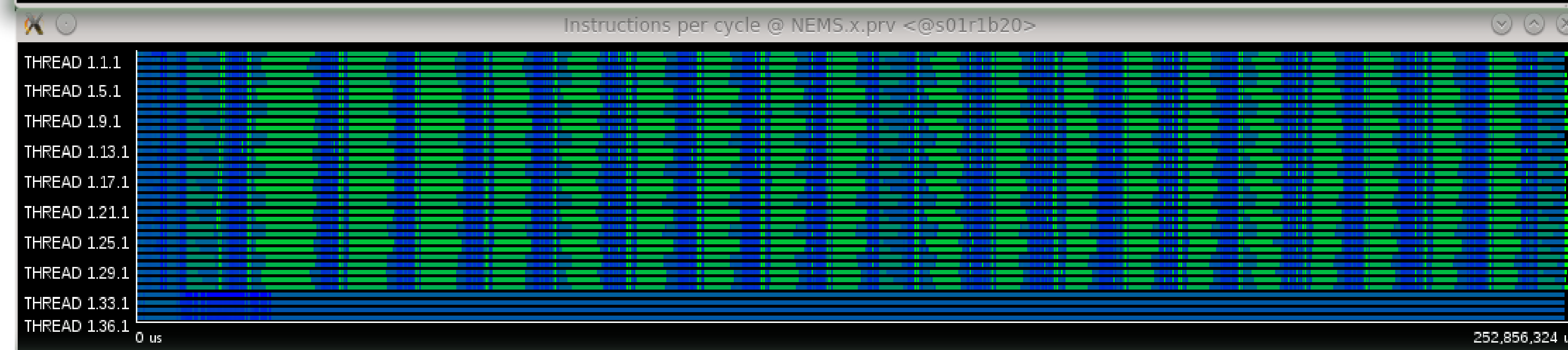
Paraver – Global – 24km – Meteo/Dust/Chem

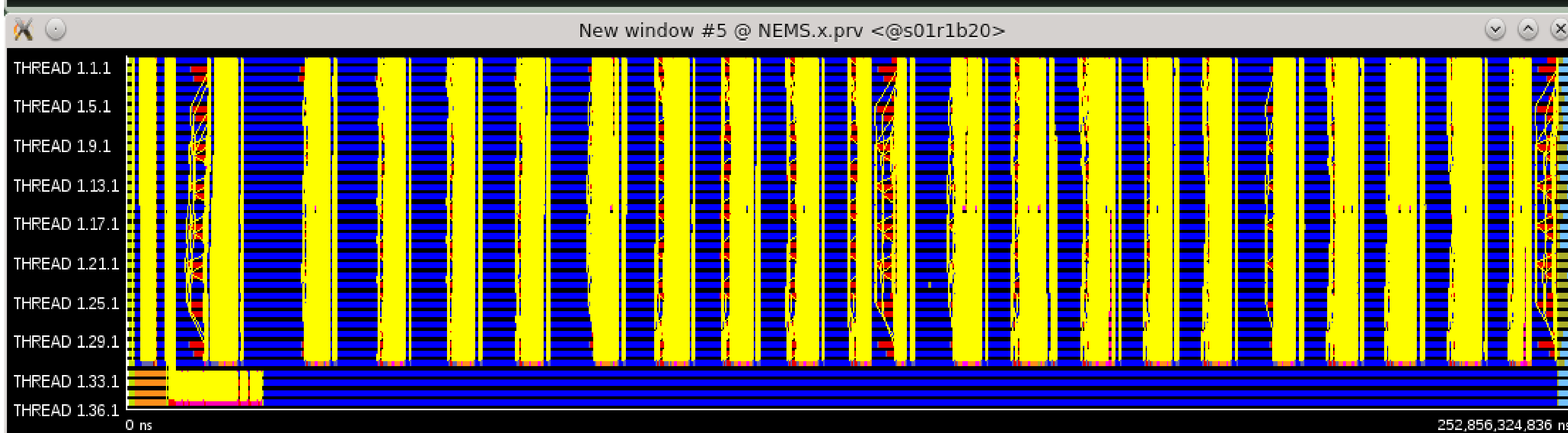


Simulation: 21/05/2010

Paraver – Global – 24km – Meteo/Dust/Chem







Simulation: 21/09/2010

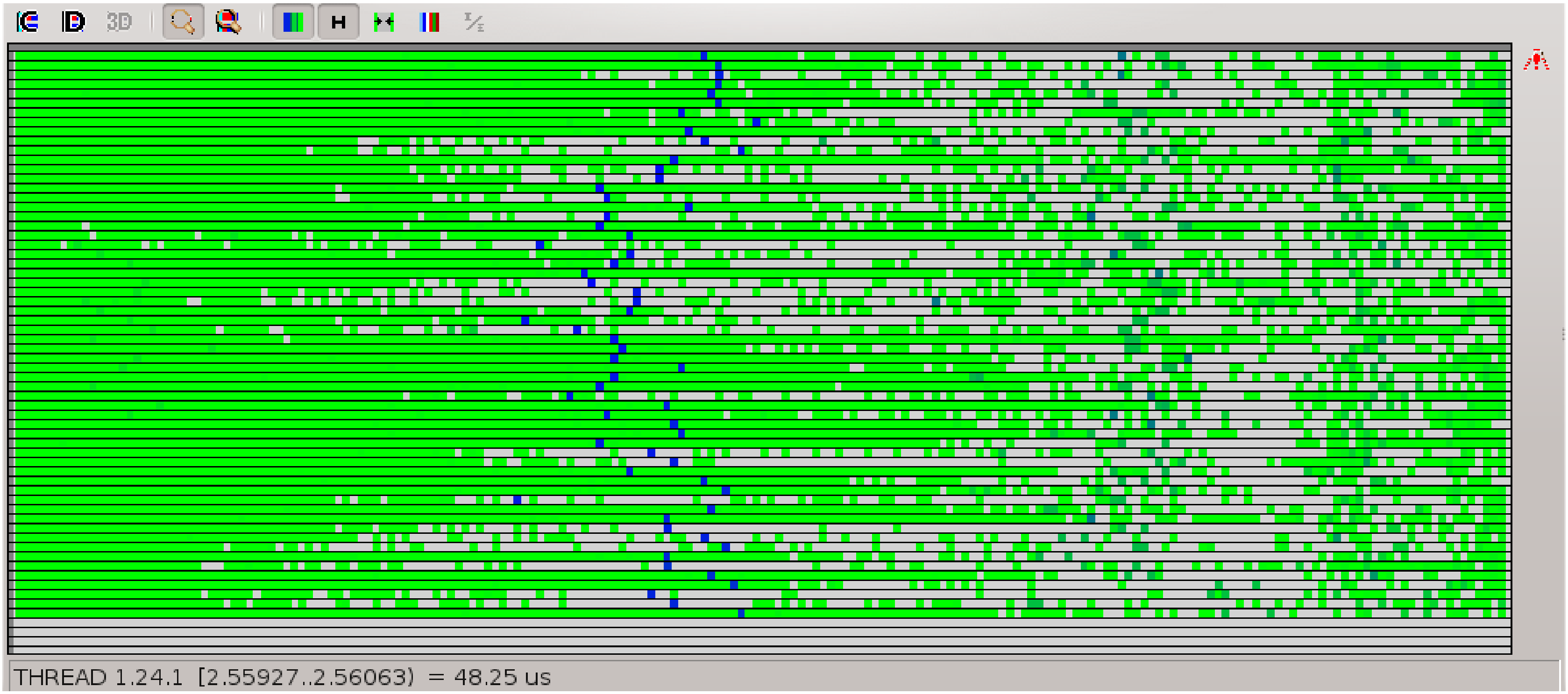
Paraver – (useful) user functions

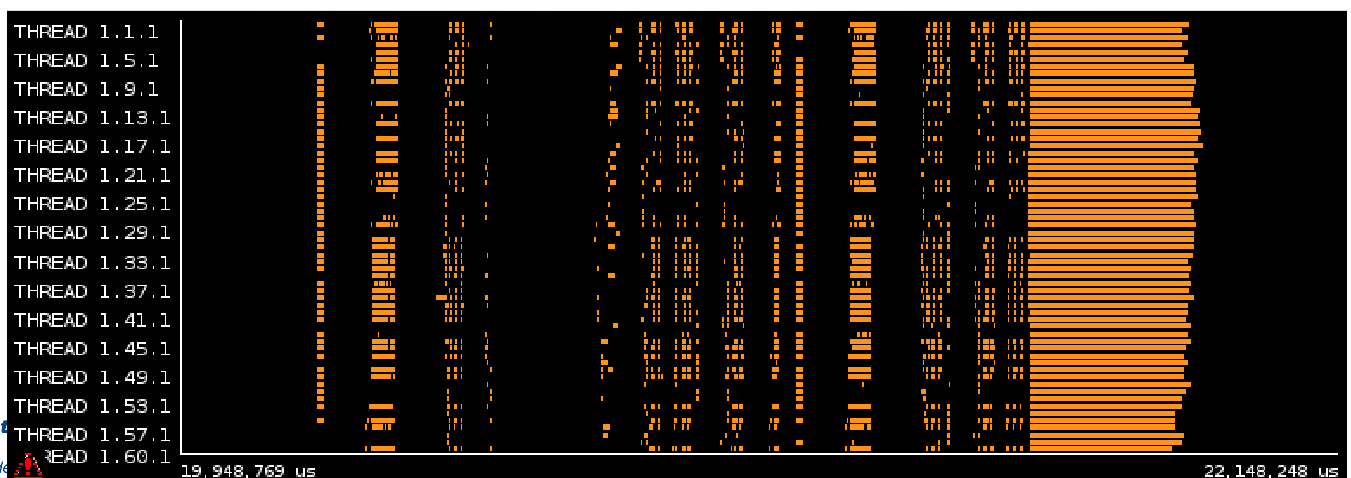
	rrtm	coszmn	rdtemp	sedimentation
THREAD 1.1.1	100,167,178.73 us	21,056.63 us	108,056.07 us	22,548,991.18 us
THREAD 1.2.1	88,414,567.01 us	21,016.15 us	116,538.77 us	23,014,879.62 us
THREAD 1.3.1	99,061,084.26 us	21,050.96 us	118,806.56 us	22,422,152.93 us
THREAD 1.4.1	99,296,557.47 us	21,144.29 us	113,197.49 us	22,703,664.54 us
THREAD 1.5.1	101,360,926.82 us	21,000.36 us	114,855.45 us	22,767,577.96 us
THREAD 1.6.1	101,527,185.54 us	20,899.77 us	112,238.29 us	23,099,842.97 us
THREAD 1.7.1	105,942,158.85 us	20,972.37 us	108,866.92 us	22,602,703.19 us
THREAD 1.8.1	100,998,572.75 us	21,024.79 us	120,283.78 us	22,297,857.18 us
THREAD 1.9.1	99,923,115.69 us	21,138.73 us	114,058.65 us	23,000,976.69 us
THREAD 1.10.1	69,672,261.36 us	20,526 us	112,094.26 us	22,811,064.18 us
THREAD 1.11.1	84,678,315.30 us	20,813.17 us	113,146.01 us	22,699,483.99 us
THREAD 1.12.1	103,738,349.58 us	21,091.16 us	117,928.27 us	22,577,006.32 us

Paraver – (useful) user functions

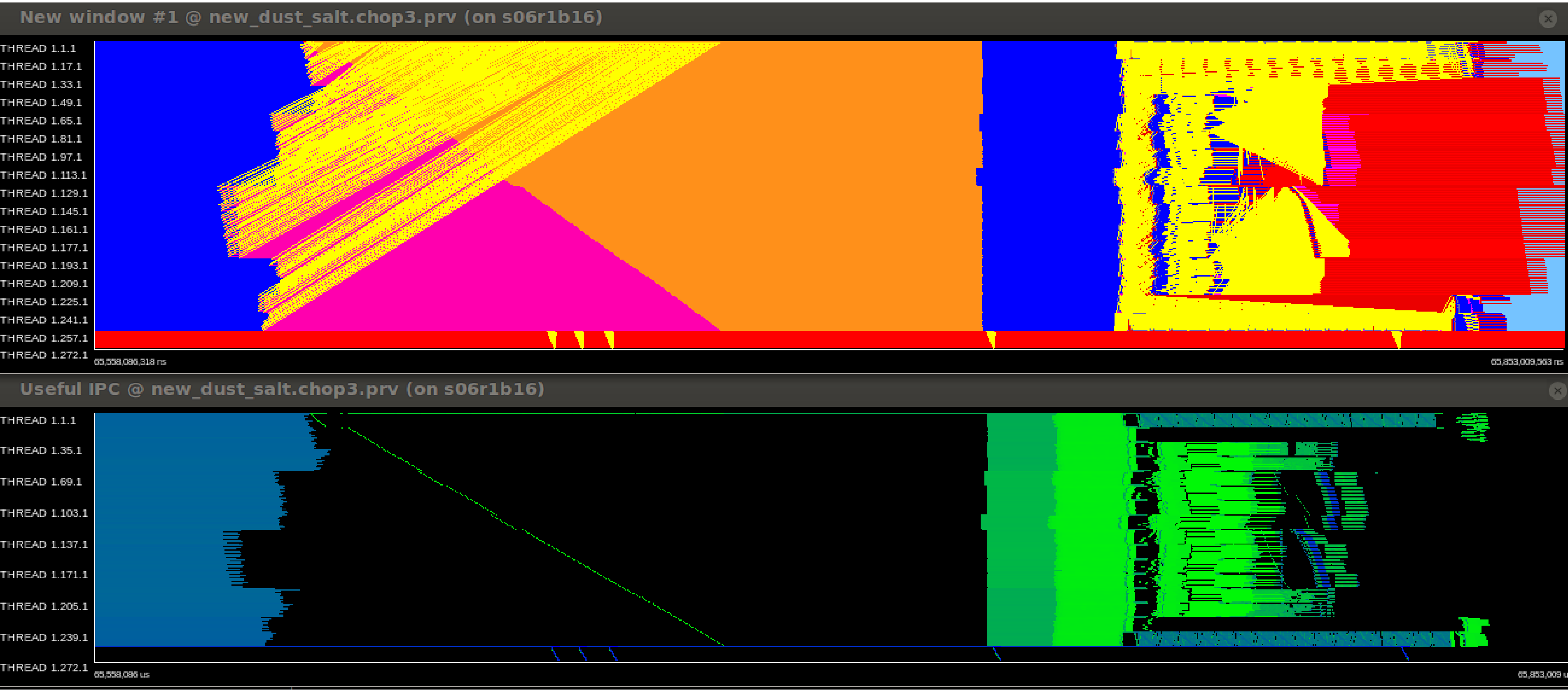
	rrtm	coszmn	rdtemp	sedimentation
THREAD 1.1.1	100,167,178.73 us	21,056.63 us	108,056.07 us	22,548,991.18 us
THREAD 1.2.1	88,414,567.01 us	21,016.15 us	116,538.77 us	23,014,879.62 us
THREAD 1.3.1	99,061,084.26 us	21,050.96 us	118,806.56 us	22,422,152.93 us
THREAD 1.4.1	99,296,557.47 us	21,144.29 us	113,197.49 us	22,703,664.54 us
THREAD 1.5.1	101,360,926.82 us	21,000.36 us	114,855.45 us	22,767,577.96 us
THREAD 1.6.1	101,527,185.54 us	20,899.77 us	112,238.29 us	23,099,842.97 us
THREAD 1.7.1	105,942,158.85 us	20,972.37 us	108,866.92 us	22,602,703.19 us
THREAD 1.8.1	100,998,572.75 us	21,024.79 us	120,283.78 us	22,297,857.18 us
THREAD 1.9.1	99,923,115.69 us	21,138.73 us	114,058.65 us	23,000,976.69 us
THREAD 1.10.1	69,672,261.36 us	20,526 us	112,094.26 us	22,811,064.18 us
THREAD 1.11.1	84,678,315.30 us	20,813.17 us	113,146.01 us	22,699,483.99 us
THREAD 1.12.1	103,738,349.58 us	21,091.16 us	117,928.27 us	22,577,006.32 us

Computation load impalance





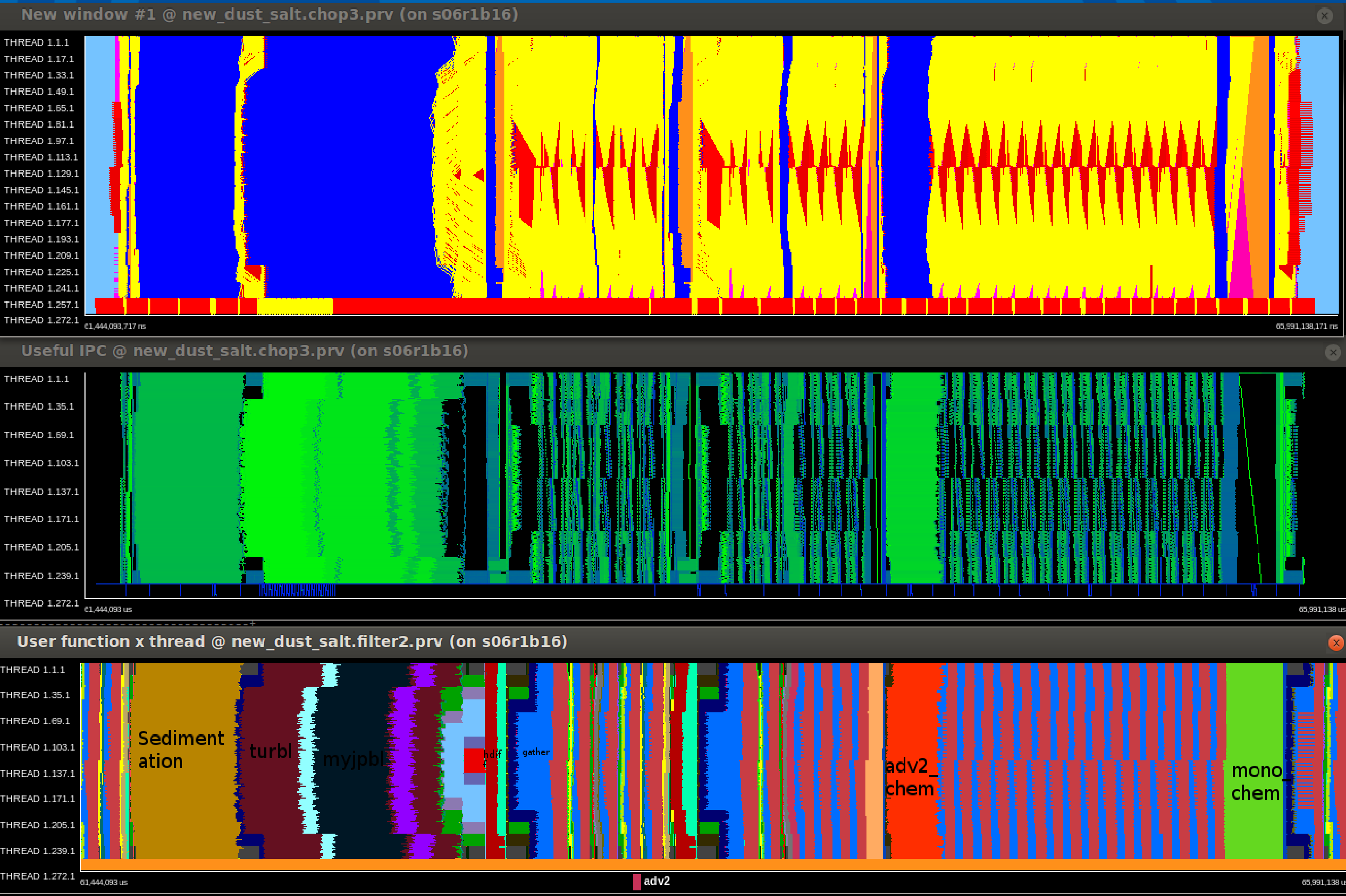
Tracer Monotonization



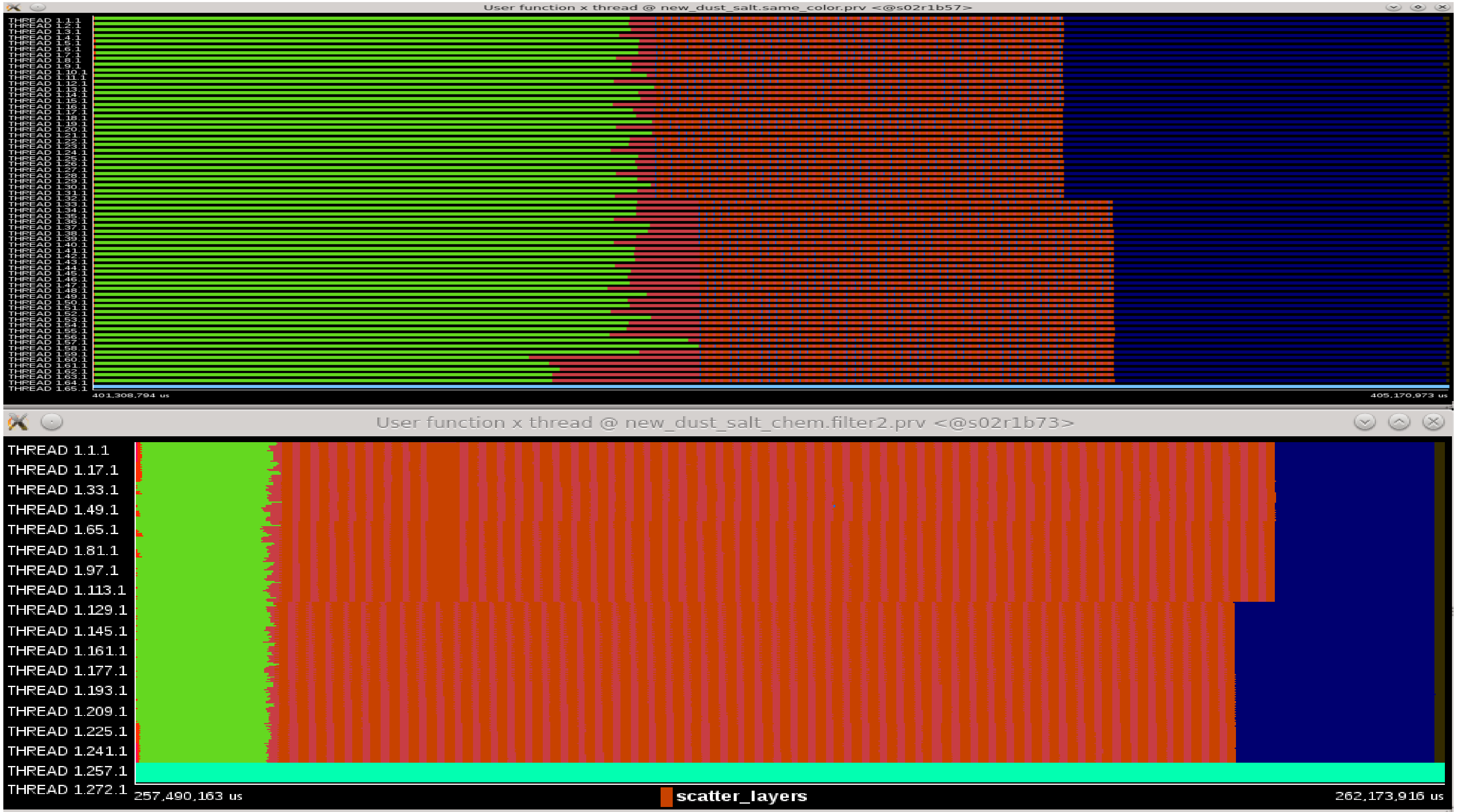
This routine is designed with a not efficient approach, the serialization can be observed

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Zoom between radiation calls for dust/sea-salt



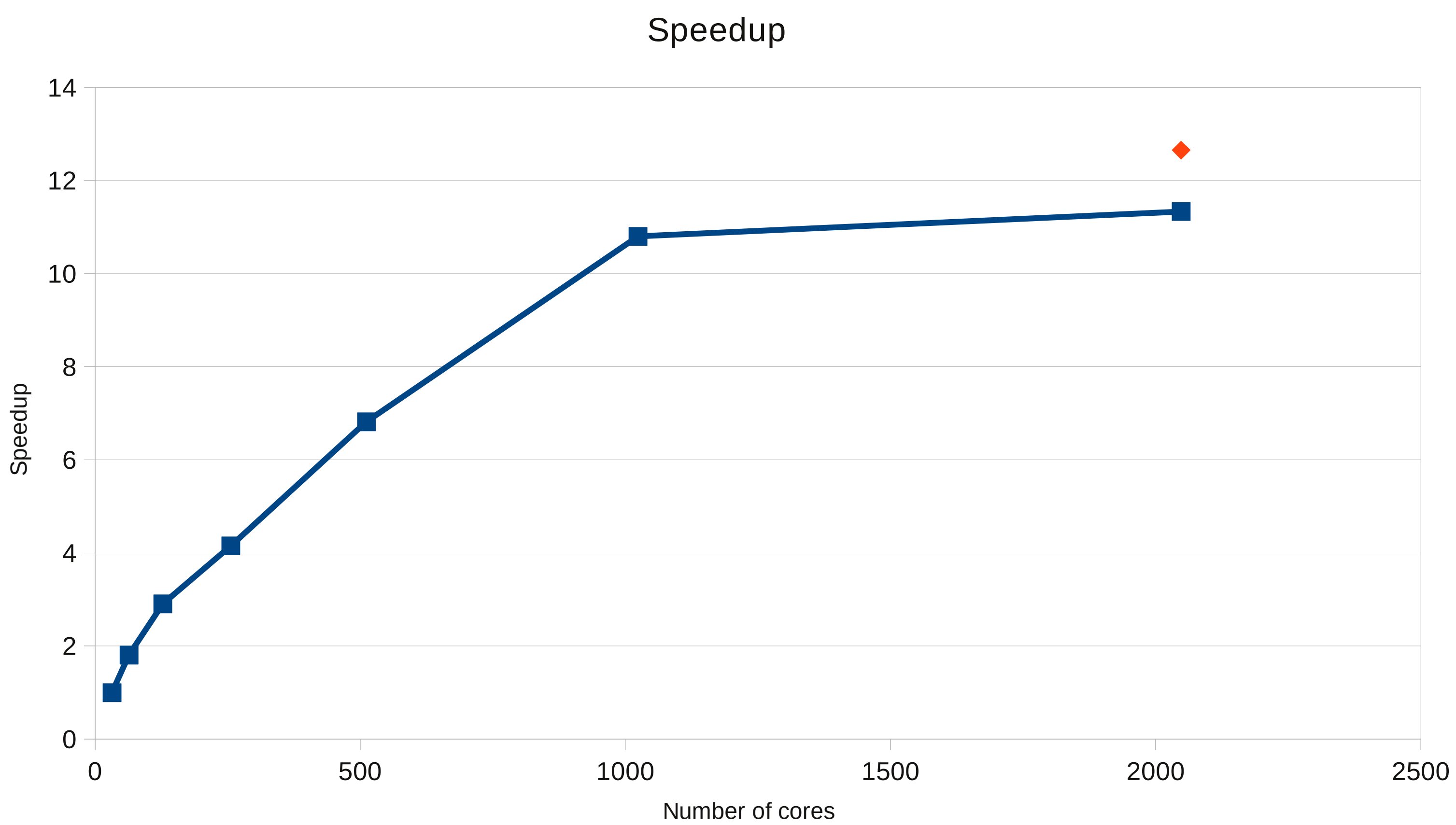
Polar filters



- The execution time with 65 cores is increased by 60% at least (without I/O) but the functions gather/scatter are improved by 5.2 - 5.8 times.

BSC
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Speedup – Global 24km – 64 layers



- For the extra datapoint we use a domain of 16 x 128 processors instead of 32 x 64

Code vectorization

```
% Vectorized code to
% add two vectors
  a= rand(1,4);
  b= rand(1,4);
  c= a + b;
```

```
% Non-vectorized version
  a= rand(1,4);
  b= rand(1,4);
  for k= 1:length(a)
      c(k)= a(k) + b(k);
  end
```

MUST - MPI run time error detection

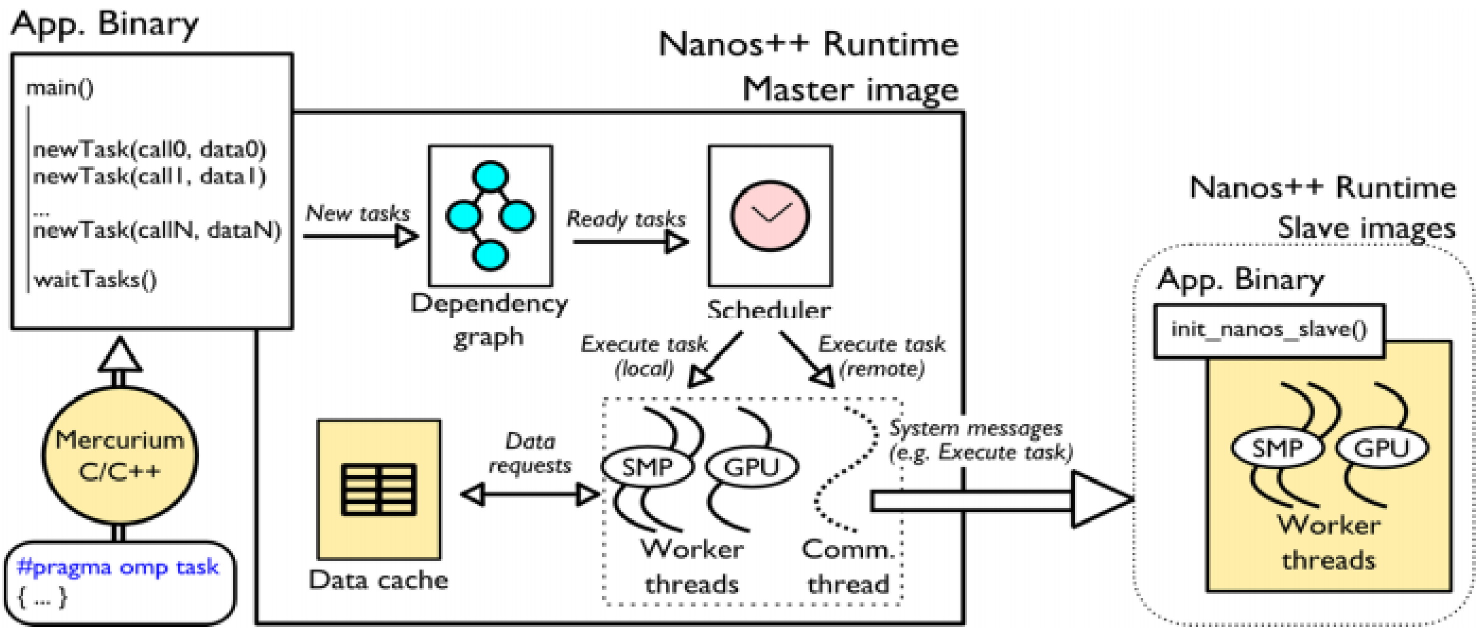
Rank(s)	Type	Message	From	References
0-35	**Warning**	Argument 2 (n) is zero, which is correct but unusual!	Representative location: call MPI_Group_excl (1st occurrence)	
35	**Error**	Argument 4 (source) specifies a rank that is greater then the size of the given communicator. (source=24, communicator size:4)!(Information on communicator: Communicator created at reference 1 size=4, is an intercommunicator remote group has size=32)	Representative location: call MPI_Recv (31th occurrence)	References of a representative process: reference 1 rank 35: call MPI_Intercomm_create (1st occurrence)
33	**Error**	Argument 4 (source) specifies a rank that is greater then the size of the given communicator. (source=8, communicator size:4)!(Information on communicator: Communicator created at reference 1 size=4, is an intercommunicator remote group has size=32)	Representative location: call MPI_Recv (31th occurrence)	References of a representative process: reference 1 rank 33: call MPI_Intercomm_create (1st occurrence)
34	**Error**	Argument 4 (source) specifies a rank that is greater then the size of the given communicator. (source=16, communicator size:4)!(Information on communicator: Communicator created at reference 1 size=4, is an intercommunicator remote group has size=32)	Representative location: call MPI_Recv (31th occurrence)	References of a representative process: reference 1 rank 34: call MPI_Intercomm_create (1st occurrence)

Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

OmpSs Programming Model

OmpSs Introduction

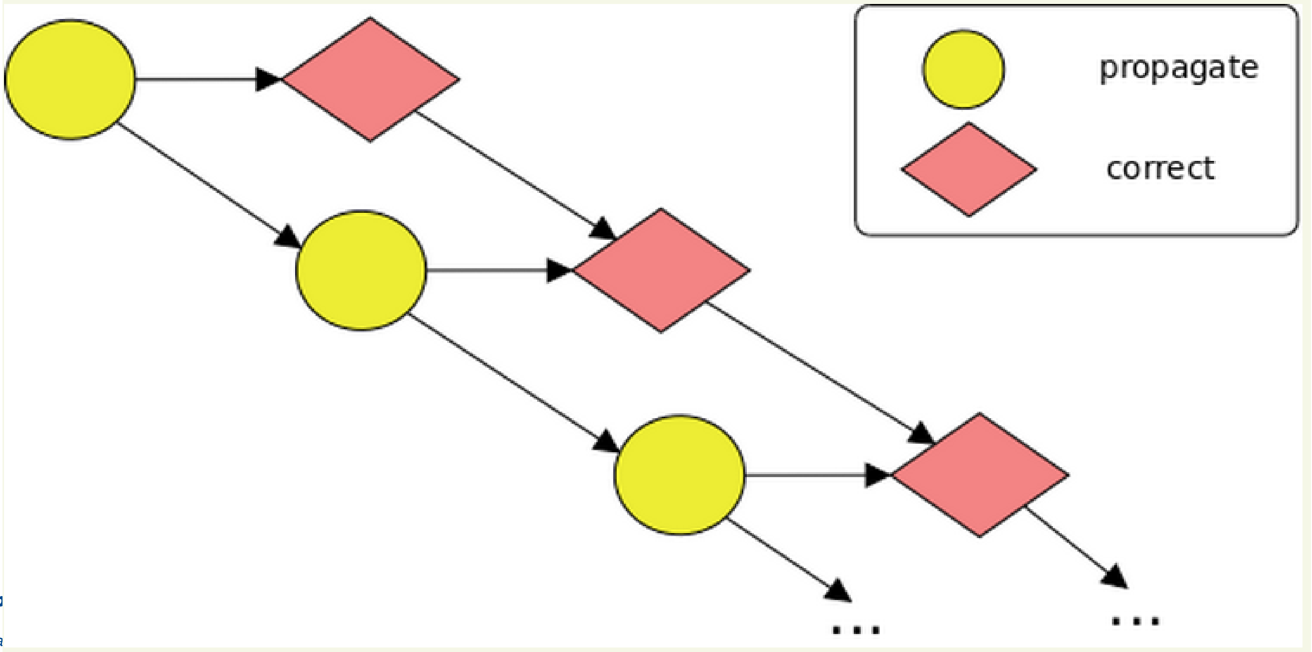
- Parallel Programming Model
  - Build on existing standard: OpenMP
  - Directive based to keep a serial version
  - Targeting: SMP, clusters, and accelerator devices
  - Developed in Barcelona Supercomputing Center (BSC)
    - Mercurium source-to-source compiler
    - Nanos++ runtime system



OmpSs Example

```
void foo ( int *a, int *b )
{
   for ( i  = 1; i < N; i++ ) {
      #pragma omp task in(a[i-1]) inout(a[i]) out(b[i])
         propagate(&a[i-1],&a[i],&b[i]);

      #pragma omp task in(b[i-1]) inout(b[i])
         correct(&b[i-1],&b[i]);
   }
}
```



Roadmap to OmpSs

- NMMB is based on the Earth System Modeling Framework (ESMF)

- The current ESMF release (v3.1) is not supporting threads. However, the development version of NMMB uses ESMF v6.3

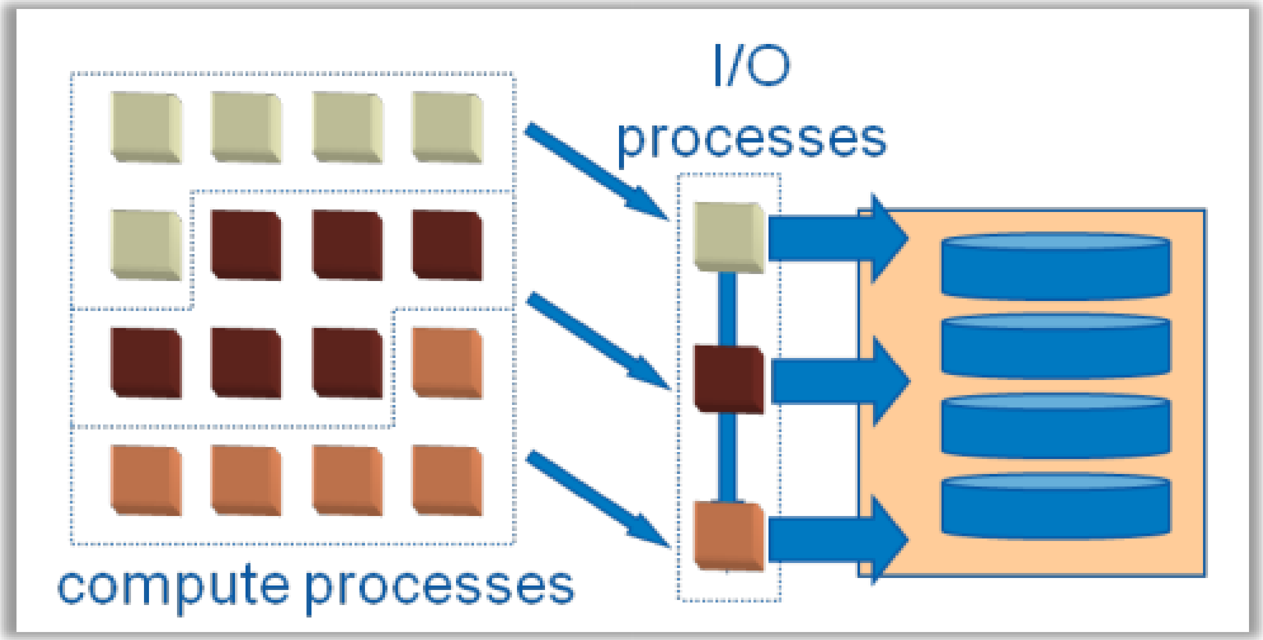
- Post-process broke because of some other issues but it was fixed

- The new version of NMMB with OmpSs support has been compiled and is ready to apply and test OmpSs

- Current work to be presented at PRACE Scientific and Industrial Conference 2014

Improved I/O (future work)

- Parallel NetCDF written to single files by all MPI tasks.



Future work

- Use OmpSs programming model
  - Study GPU case
  - Explore Xeon Phi

- Prepare NMMB model for higher resolutions, first milestone is the global model for 12km
  - Improve performance and scale NMMB for thousands of cores

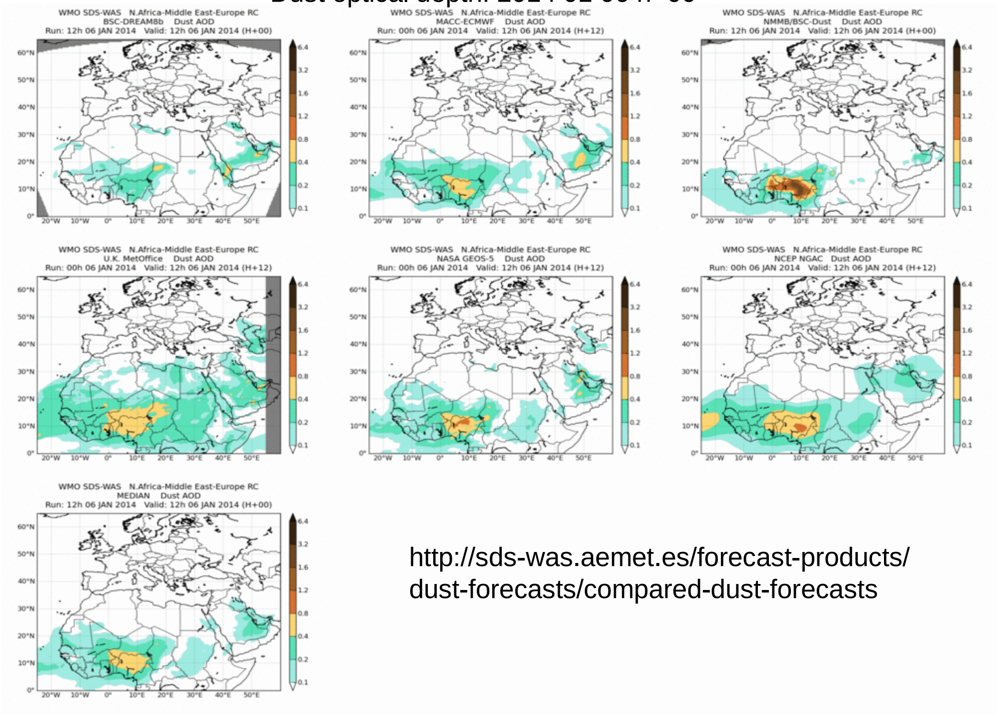
- Fix I/O issue
  - IS-ENES Exascale Technologies & Innovation in HPC for Climate Models workshop
  - Possible collaboration across the community to focus on a global solution

Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

Data Assimilation

Atmospheric models are far from being perfect

Dust optical depth: 2014 01 06 h+00



http://sds-was.aemet.es/forecast-products/
dust-forecasts/compared-dust-forecasts

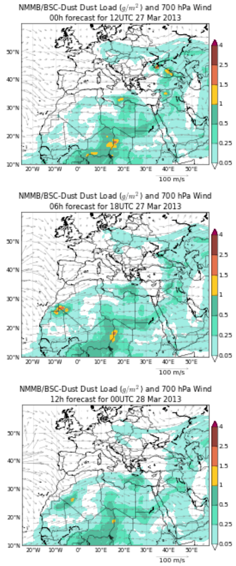
A considerable amount of accurate earth observations is available

http://www.wmo.int/pages/prog/gcos/

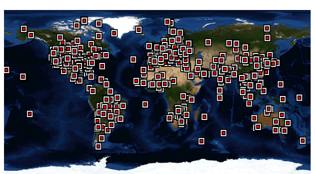
Data assimilation 'optimally' combines **models** and **observations**

Data Assimilation – Workflow

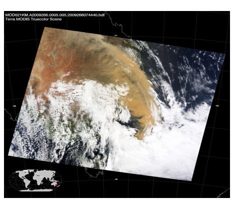
Ensemble background



Observations



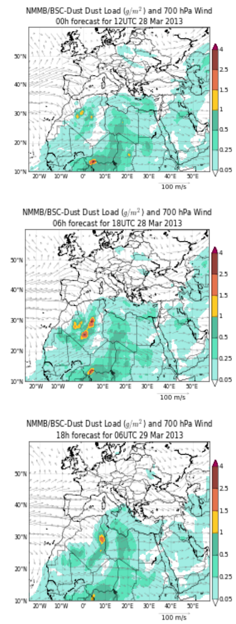
http://aeronet.gsfc.nasa.gov/



http://modis-atmos.gsfc.nasa.gov/

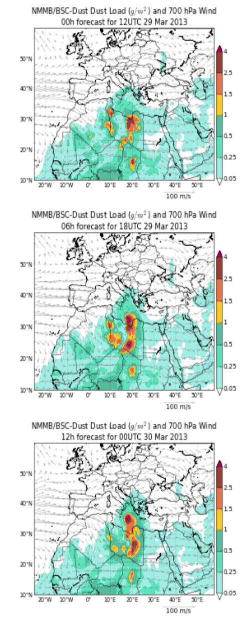
Kalman filter*

Ensemble analysis

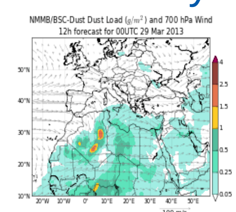


short-term forecast

Ensemble background



Mean analysis



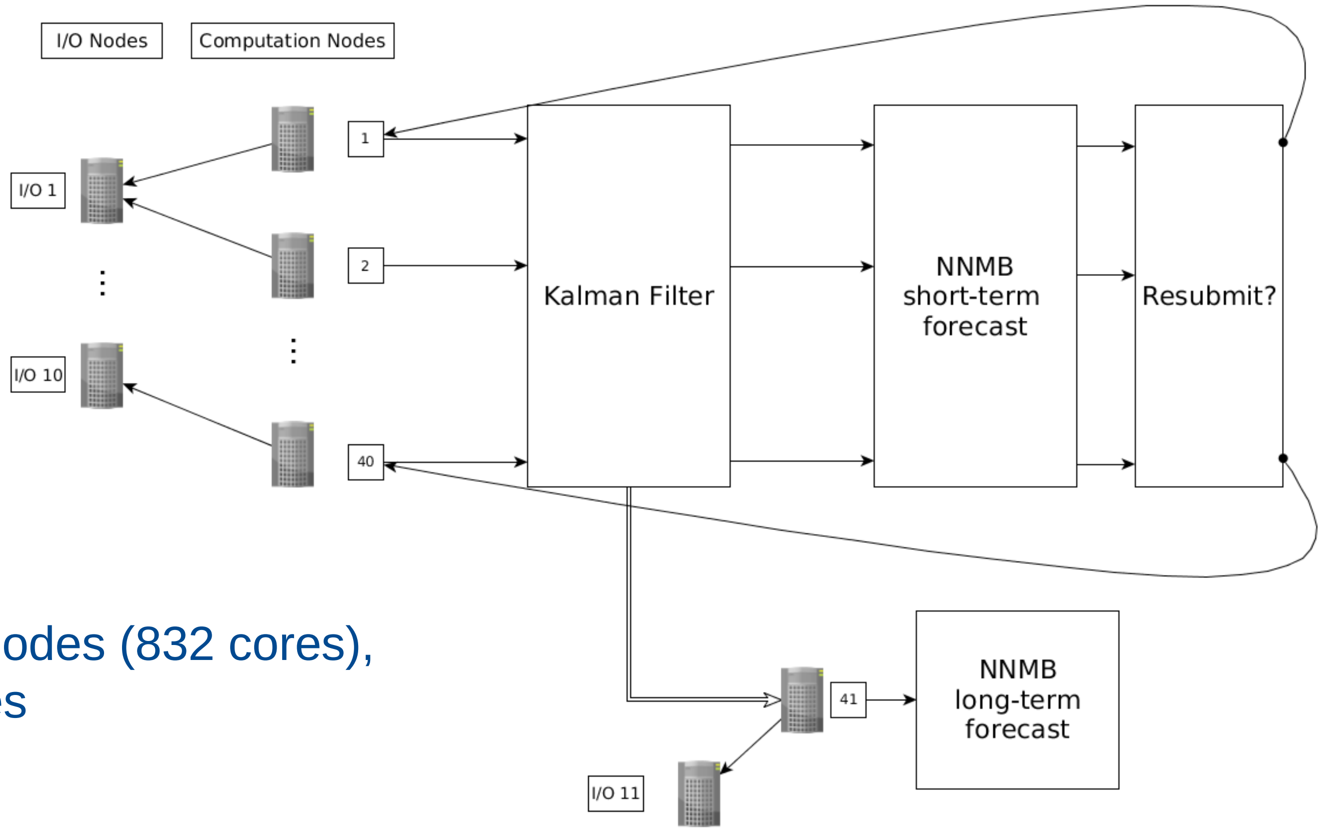
long-term forecast

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

* In collaboration with N. Schutgens (Uni. Oxford, UK)

Data Assimilation – Workflow

- BASH script starts the submission of the assimilation job
  - We want all the ensembles to be executed in parallel
  - We have 40 ensembles, we provide 20 cores for each execution and one ensemble for long-forecast. We should need totally 82 nodes (1,312 exclusive cores)



  - Now, we need 52 nodes (832 cores), ~36% less resources

http://modis-atmos.gsfc.nasa.gov/

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Thank you!

For further information please contact
georgios.markomanolis@bsc.es