EC-Earth meeting
2-4 February 2016

Reproducibility of EC-Earth

F. Massonnet, M. Asif, **M. Acosta**, O. Bellprat, E. Exarchou, **J. García-Serrano**, V. Guemas, M. Ménégoz, O. Mula-Valls, K. Serradell, F. J. Doblas-Reyes, X. Yepes

Where were we?

At the last EC-Earth meeting (May 2015):

- Machine error (e.g., roundoff) also contributes to model uncertainty

  This error adds to well-known model physics/coupling, initial condition and forcing errors

Where were we?

At the last EC-Earth meeting (May 2015):

- Machine error (e.g., roundoff) also contributes to model uncertainty

  This error adds to well-known model physics/coupling, initial condition and forcing errors
- EC-Earth3.1 was found to be bit-reproducible

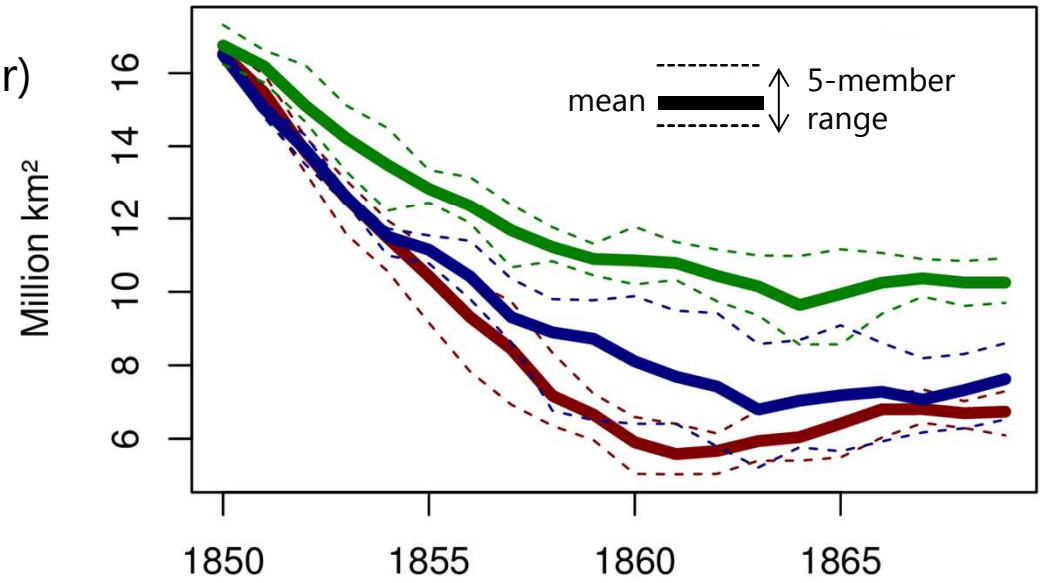
  Two runs performed under exactly the same conditions produced exactly the same output

Where were we?

At the last EC-Earth meeting (May 2015):

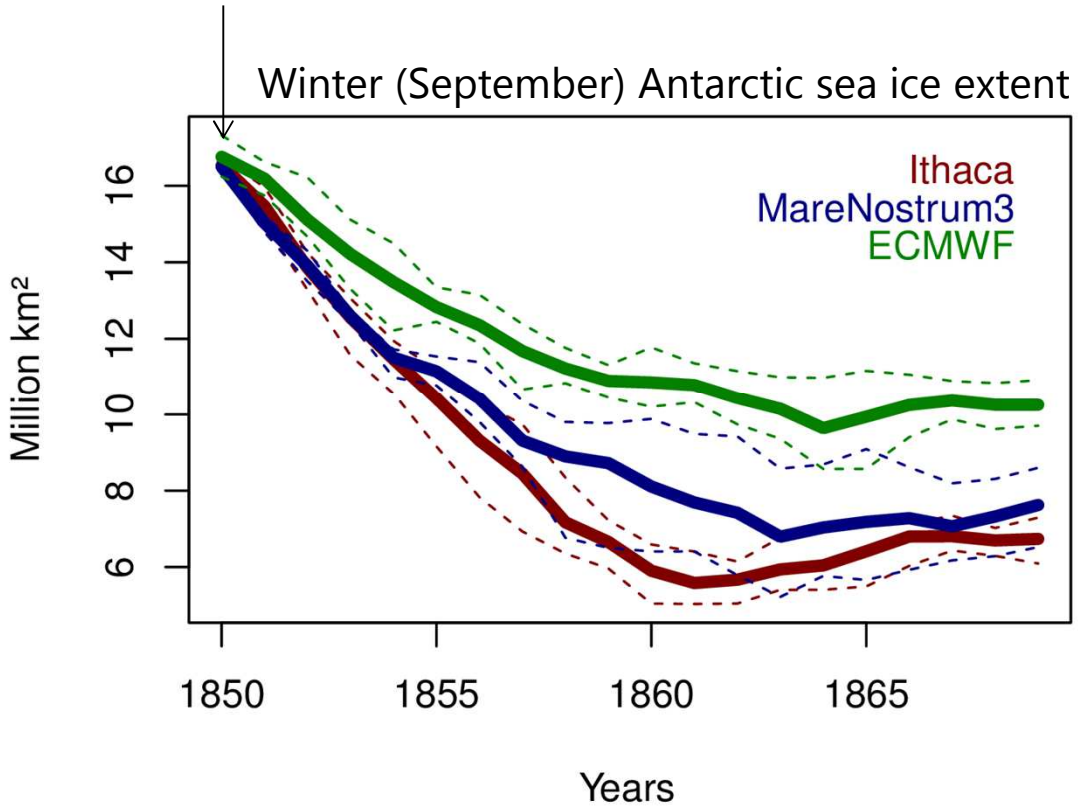
- Machine error (e.g., roundoff) also contributes to model uncertainty
  - This error adds to well-known model physics/coupling, initial condition and forcing errors
- EC-Earth3.1 was found to be bit-reproducible
  - Two runs performed under exactly the same conditions produced exactly the same output
- EC-Earth3.1 was found to be **not** climate-reproducible
  - Moving from one machine to another caused different climates

Winter (September) Antarctic sea ice extent



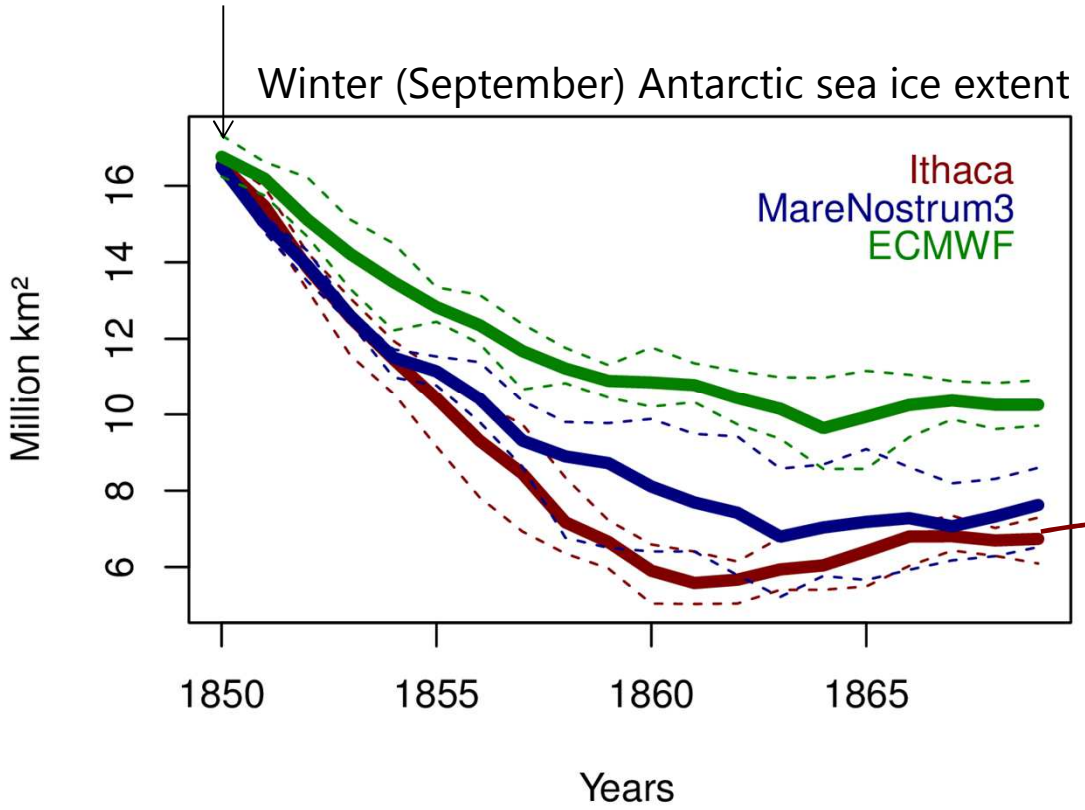
Issue 1: the massive drift: can we reliably exchange restarts among centers?

Initial conditions were from an equilibrated run carried out at CNR



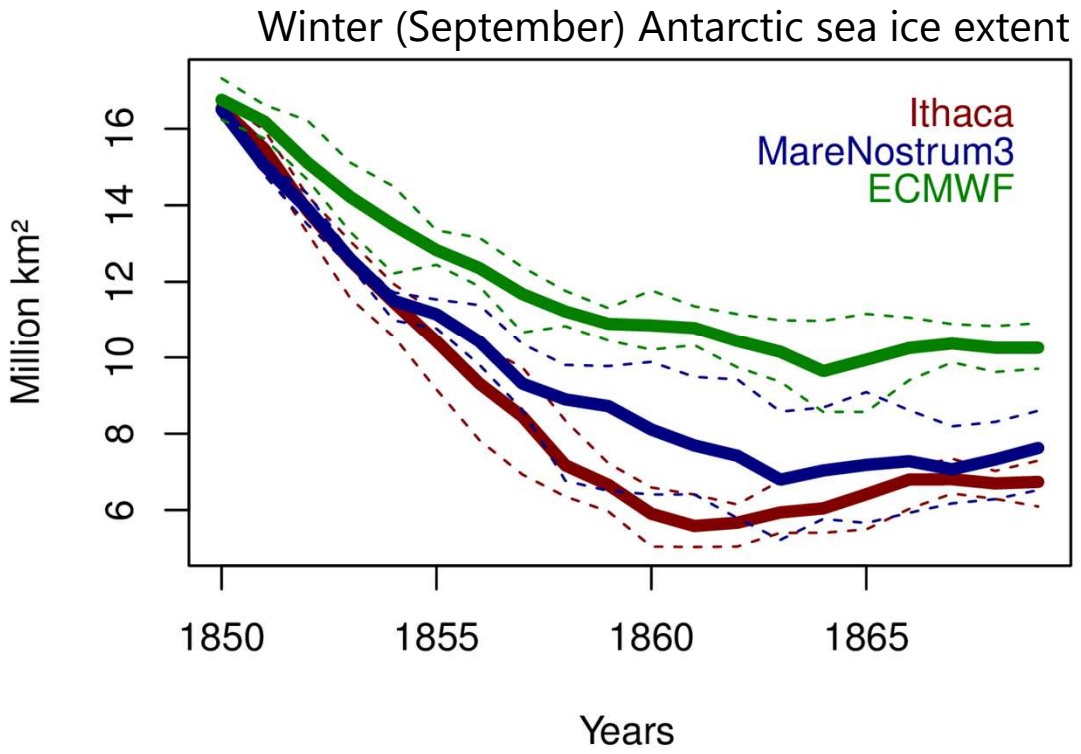
Issue 1: the massive drift: can we reliably exchange restarts among centers?

Initial conditions were from an equilibrated run carried out at CNR



We decided to let EC-Earth equilibrate for 40 more years before starting a new stream of reproducibility experiments

Issue 2: we didn't maximize the odds of success



598 CPUs

512 CPUs

72 CPUs

-Floating Point (FP) options were not set to ensure reproducibility

-Number and distribution of CPUs were different

User-approach: climate-reproducibility

Results can be different, but statistics must be the same

Developer-approach: bit-reproducibility

Results must be the same, strictly

User-approach: climate-reproducibility

Results can be different, but statistics must be the same

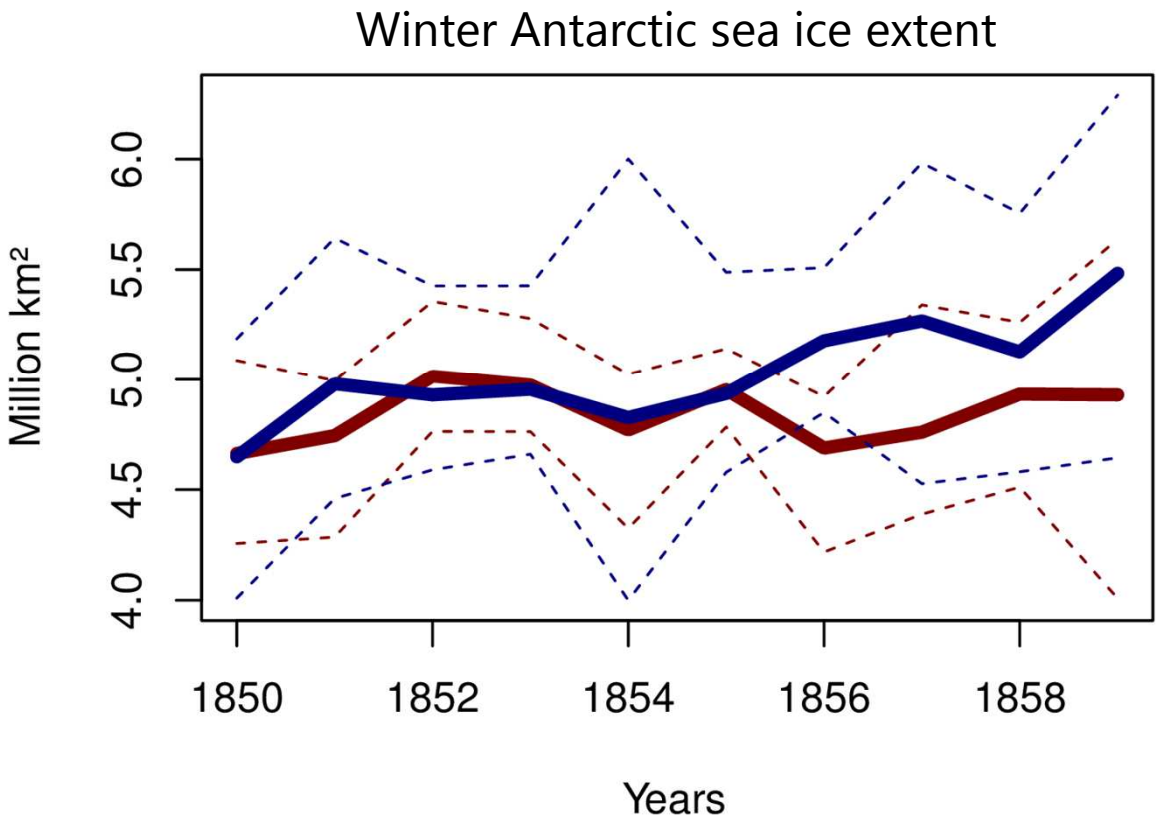
EC-Earth3.1
Initialized from **our own** equilibrated run
5 members
Pre-industrial forcing

2 platforms, **same # CPUs and domain decomposition**
Only 10 years for now
Same compilation options as before

Developer-approach: bit-reproducibility

Results must be the same, strictly

This time, the two ensembles cannot be statistically distinguished from each other



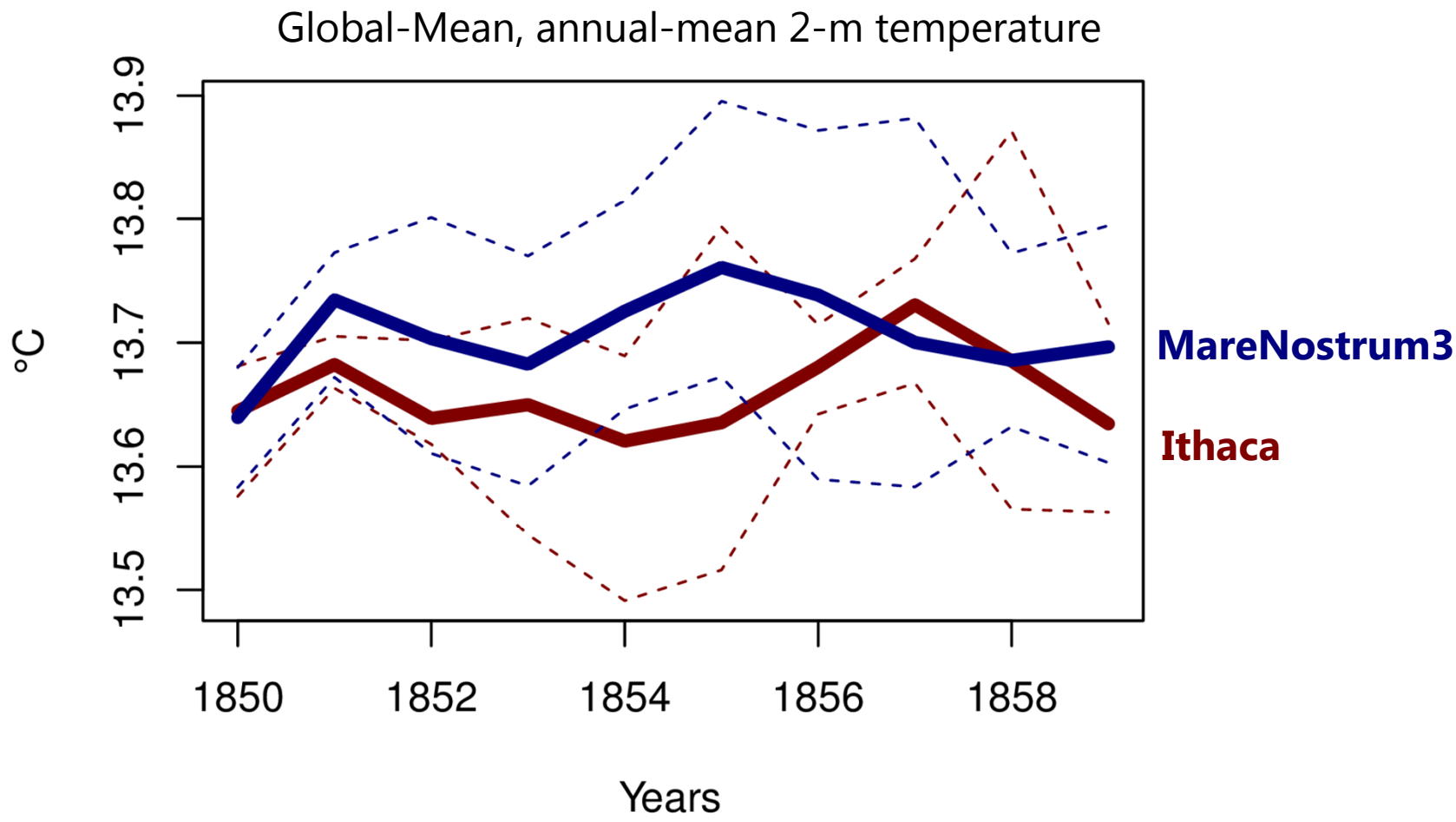
MareNostrum3

Ithaca

(same domain decomposition)

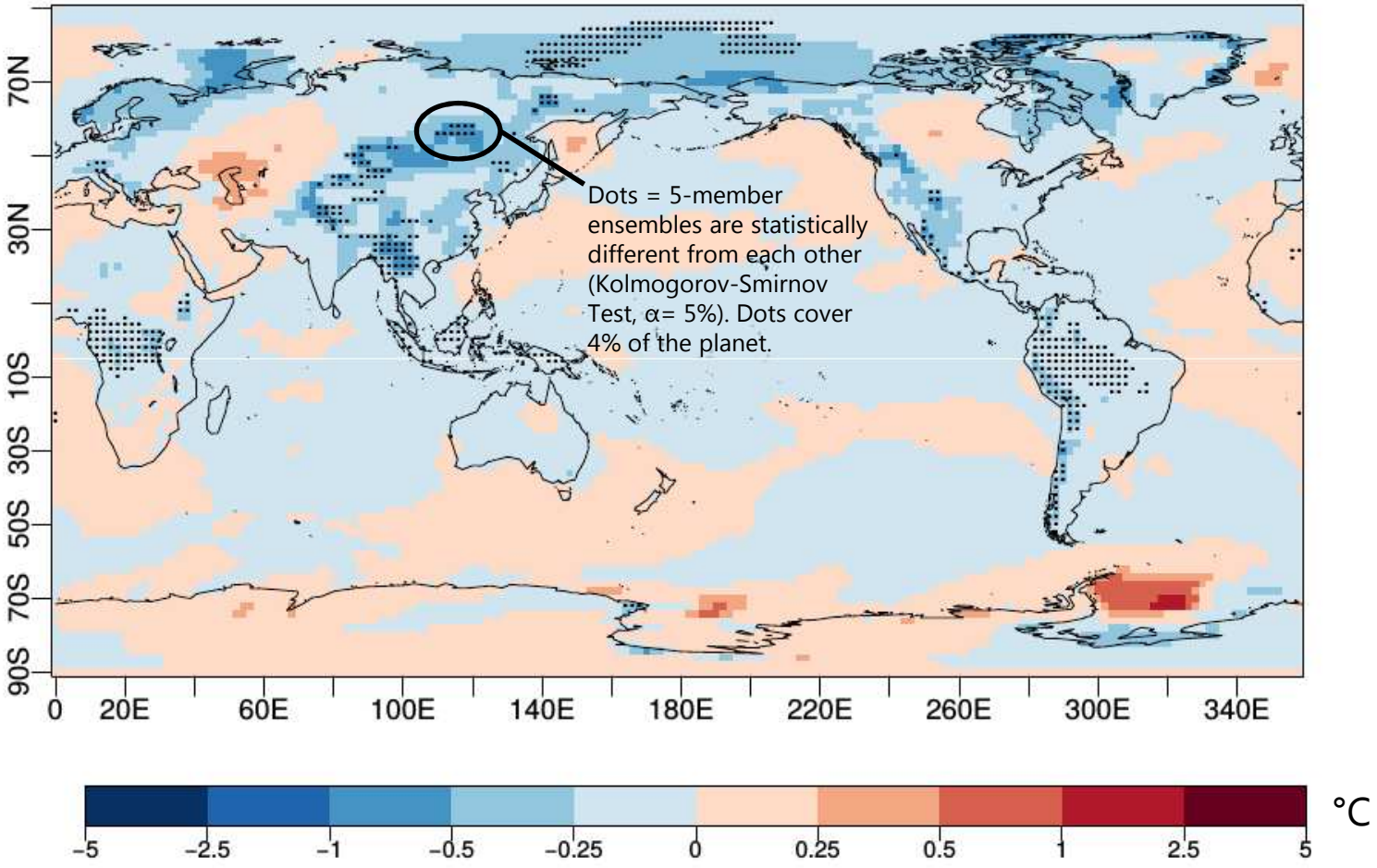
- **Only ten years**
- **Only two platforms**
- **Only five members**
- **ECMWF (missing here) was the one standing out**

This time, the two ensembles cannot be statistically distinguished from each other

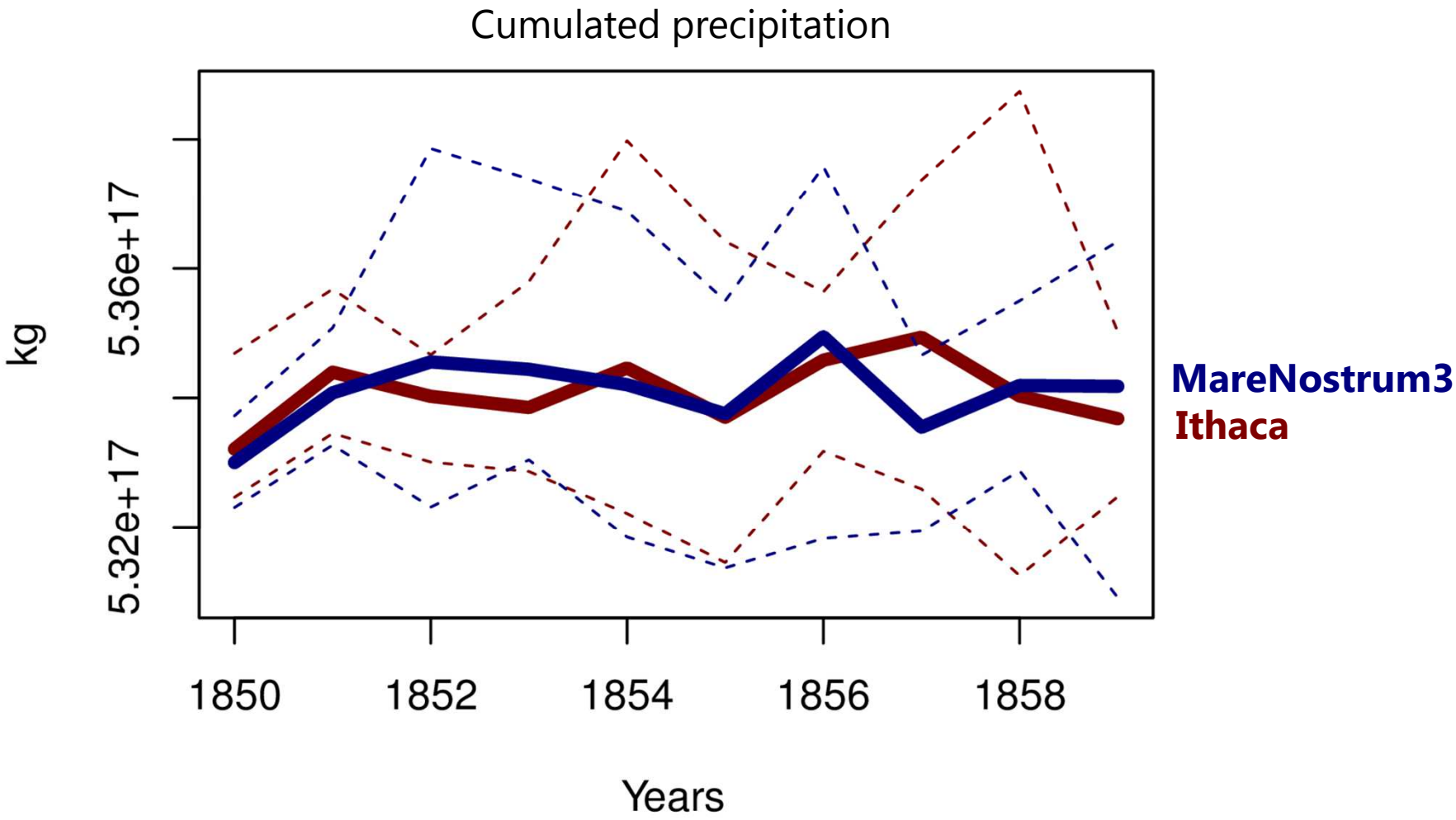


This time, the two ensembles cannot be statistically distinguished from each other

2-m air temperature difference between Ithaca and MareNostrum3 (ensemble means)



This time, the two ensembles cannot be statistically distinguished from each other



Next experiments to be conducted

- At MareNostrum3, with different #CPUs from Ithaca's
- At ECMWF, with same #CPUs as Ithaca's
- At ECMWF, with different #CPUs from Ithaca's

And the experiments should, ideally, be repeated with EC-Earth3.2!

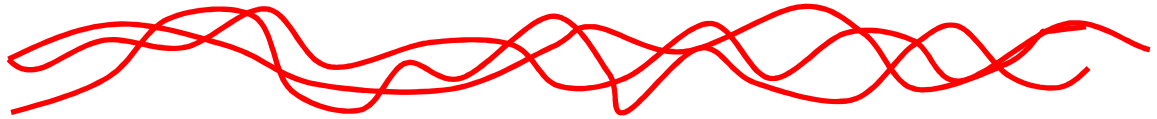
Conclusions

- Don’t exchange restarts across centers unless you are sure of what you are doing
- EC-Earth3.1, after equilibration, and for the same domain decomposition, now looks climate-reproducible (! Cautions)
- A systematic reproducibility procedure has to be defined and applied with EC-Earth3.2, ideally before CMIP6 runs are started

Strategic question to discuss

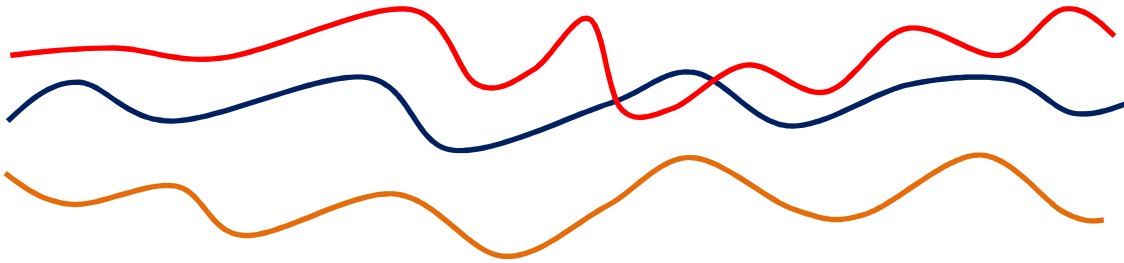
How do we handle CMIP6 simulations?

- Option 1 (clean, but unrealistic): One partner does all the simulations on the same machine.



3 members run at BSC

- Option 2 (fast, but questionable): Split the load of simulations



1 member run at BSC

1 member run at KNMI

1 member run at SMHI

In that case, we have to recognize that this « 1 + 1 + 1 » ensemble will have likely a **larger spread**: **it also comprises hardware/software uncertainty**

•Option 3 (tradeoff): One partner takes one type of experiment

(examples)
piControl → BSC
Historical: 5 members → KNMI
RCP4.5: 6 members → SMHI

In that case, ensembles would be more consistent but we may have trouble in experiments that are connected, e.g. historical and RCP.

User-approach:
climate-reproducibility

Results can be different, but statistics must be the same

Developer-approach:
bit-reproducibility

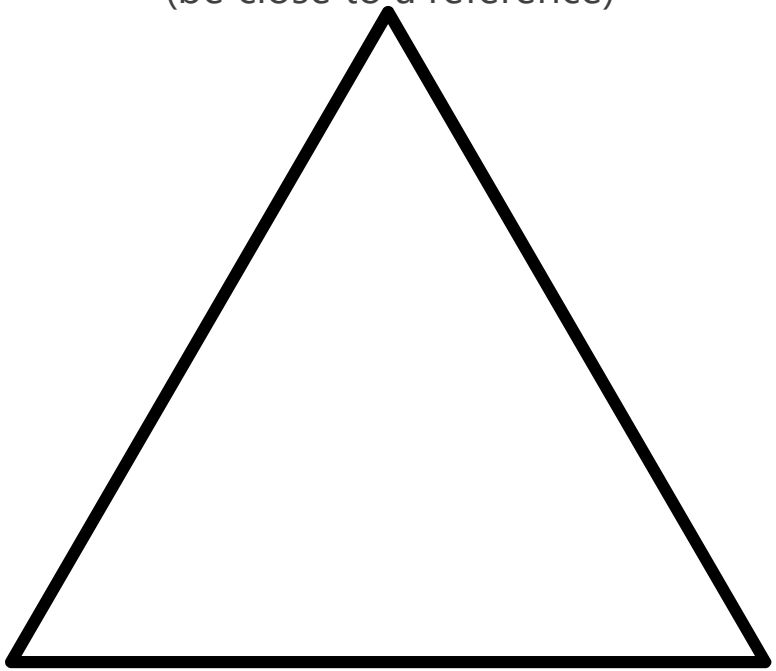
Results must be the same, strictly

EC-Earth 3.2beta
Initialized from default initial conditions

1 platform (MareNostrum 3)
Two domain decompositions:
- IFS: 320 & NEMO: 288
- IFS: 128 & NEMO: 64
Only 1 month
Different compilation options

Model development has the following objectives

Accuracy
(be close to a reference)



Reproducibility
(be similar across configurations)

Performance
(use resources efficiently)

Compiler options let you control the tradeoffs between accuracy, reproducibility and performance

- Different compilation flags can be used to control the tradeoffs between accuracy, reproducibility and performance
  - To control Floating-Point (FP) operations
    - fp-model precise, fimf-arch-consistency, fpe0, fma, ftz ...
  - To control optimization options
    - O1, O2, O3, xHost, ipo ...

- Why is it necessary to control FP operations?
  - FP numbers have finite resolution 1.77777777 → 1.77778
  - Rounding can change intermediate results
    - A+B+C =/ A+C+B
- FP errors are caused by:
  - Algorithm
    - Conditional numerical computation for different systems and/or input data can have unexpected results.
  - Non-deterministic task/process scheduler
    - Asynchronous task/process scheduling can change the order of some operations between reruns.
  - Memory alignment
    - If memory alignment is not guaranteed, the results could be computed differently between reruns.

- Compilation flags to control optimizations

Flag	Description
-O0	No optimizations
-O1	Enables optimizations if they do not increase code size
-O2	Enables optimizations for maximized speed, such as code inlining, loop unrolling, variable renaming...
-O3	Performs O2 optimizations and enables aggressive loop transformations such as Fusion, collapsing “if” statements...
-xHost	Generates instruction sets up to the highest that is supported by the compilation host.

- Compilation flags to control FP operations

Flag	Description
-fp-model precise	Allow value-safe optimizations only
-fp-model source/double/extended	Intermediate precision for FP calculations (For Fortran only source)
-fimf-arch-consistency=true	Math library functions produce consistent results
-no-fma	FP contractions are disallowed
-fp-model except	Determine whether floating point exceptions semantics are used
-fp-model strict	Enables precise and except, disables contractions, and enables the property that allows modification of the floating-point environment (include -no-fma, fpe0)
-ftz	Flush denormal results to zero
-fpe0	Unmask floating point exceptions and disable generation of denormalized numbers (include ftz)

- These flags enable or disable:
  - Value safety
    - Make safe some operations such as Reassociation → **(a+b)+c or a+(b+c)**
  - Floating-point expression evaluation
    - Avoid operations using a different precision between variables
  - Precise floating-point exceptions
    - FP exceptions (overflow, underflow, divide by zero…) are synchronized with the operation causing it and optionally unmasked.
  - Floating-point contractions → **a=b*c+d**
  - Floating-point unit environment access
    - Control some options such as the rounding mode

- Classification of flags
  - –Value safety
    (fp-model precise,ftz)
  - –Floating-point expression evaluation
    (fp-model source/double, fimf-arch-consistency=true)
  - –Precise floating-point exceptions
    (fp-model strict,fpe0)
  - –Floating-point contractions
    (fp-model strict, no-fma)
  - –Floating-point unit environment access
    (fp-model strict, ftz)
  - –Optimization options
    (O2,O3,xHost)

List of compilation flags used in the experiment

-O2 -fp-model precise -xHost -r8
-O2 -fp-model strict -xHost -r8
-O3 -fp-model precise -xHost -r8
-O3 -fp-model strict -xHost -r8
-O2 -fp-model precise -fp-model double -fimf-arch-consistency=true -no-fma -r8
-O2 -fp-model double -fimf-arch-consistency=true -no-fma -ftz -r8
-O2 -fp-model precise -fp-model double -fimf-arch-consistency=true -no-fma -fpe0 -r8
-O2 -fp-model strict -fp-model double -fimf-arch-consistency=true -no-fma -fpe0 -r8

Comparing outputs, they are different

-Using same domain decomposition and different flags among runs

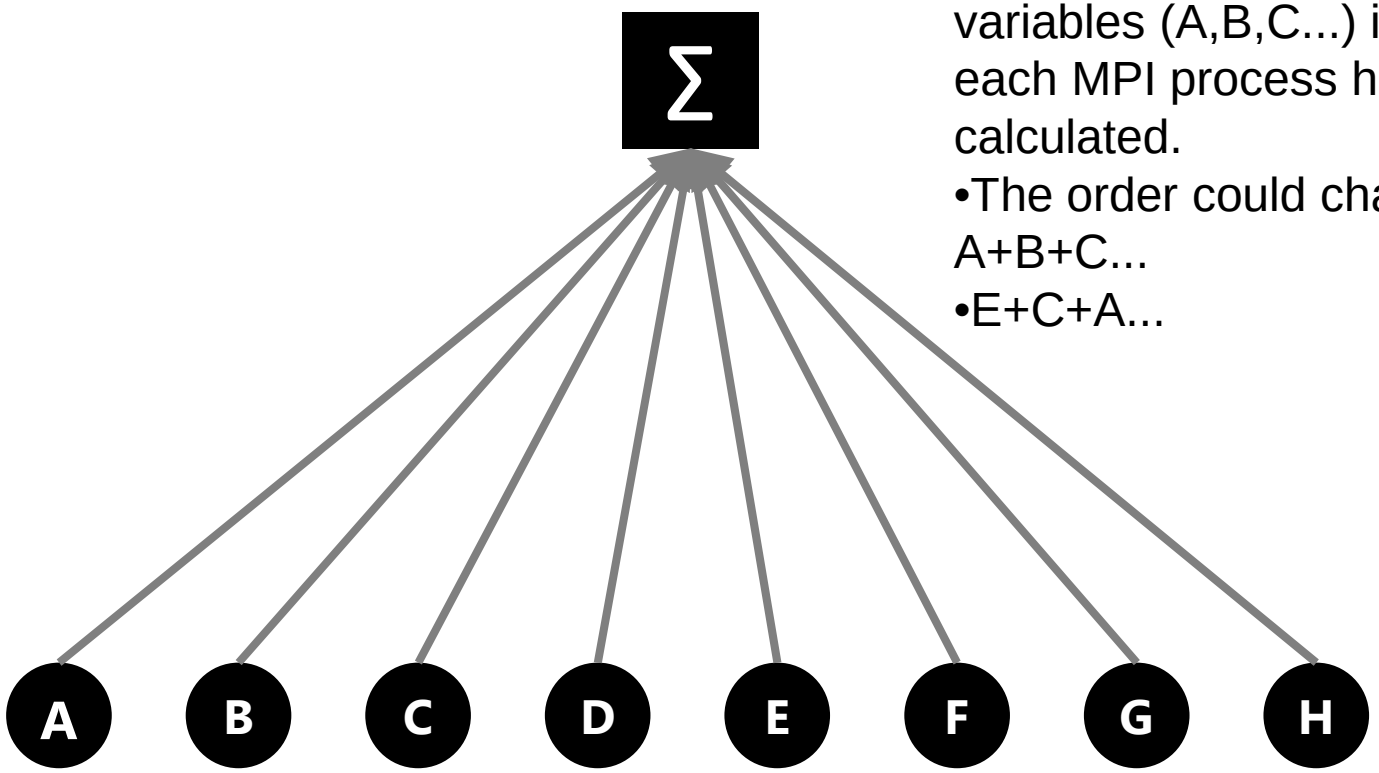
-Using same flags and different domain decomposition

-Using same flags and domain decomposition

(in 1-month simulations with EC-Earth 3.2beta)

Non-deterministic task in parallel applications

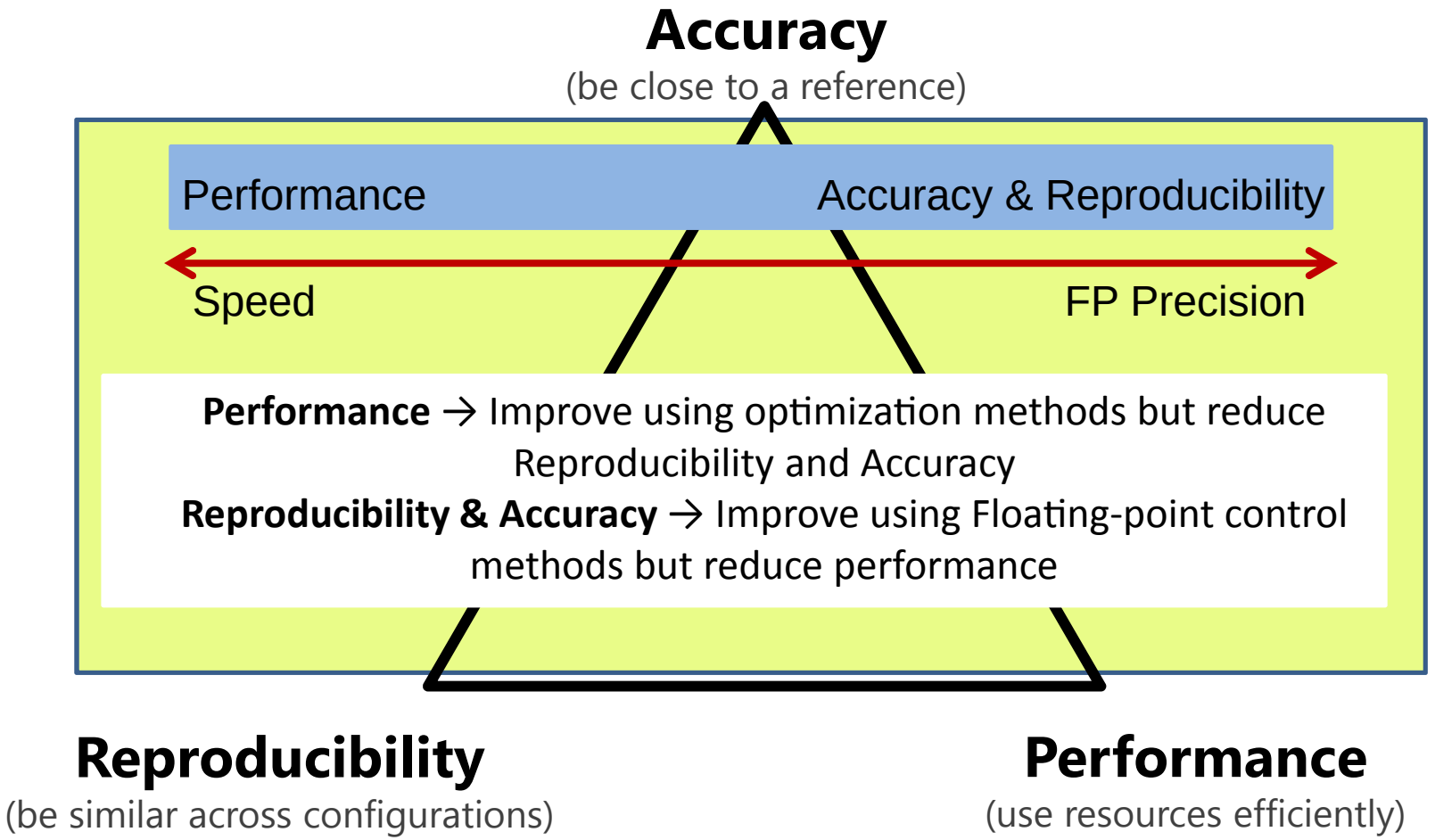
MPI reduction(+): order of summation depends on several external factors



8-cores MPI job

- In asynchronous tasks done in a efficient way, the order to add all variables (A,B,C...) is according to when each MPI process has the value calculated.
- The order could change the result A+B+C...
- E+C+A...

Relation between Performance and Accuracy & Reproducibility



Conclusions about bitwise reproducibility

- Is the modelled climate able to obtain a bitwise precision using some combination of compilation flags? → NO
- Is bitwise precision possible without losing parallel and sequential performance? → NO
- How do we deal with the associated uncertainties?
  - Use a statistical method to quantify the differences and propose a minimum to achieve instead of bit-for-bit precision in order to avoid critical restrictions in performance.
  - Determine a combination of flags (Floating-point control and optimization) and optimization methods which achieve a balance between performance and accuracy & reproducibility in both, runs in a particular platform and runs in two different platforms with a similar architecture.