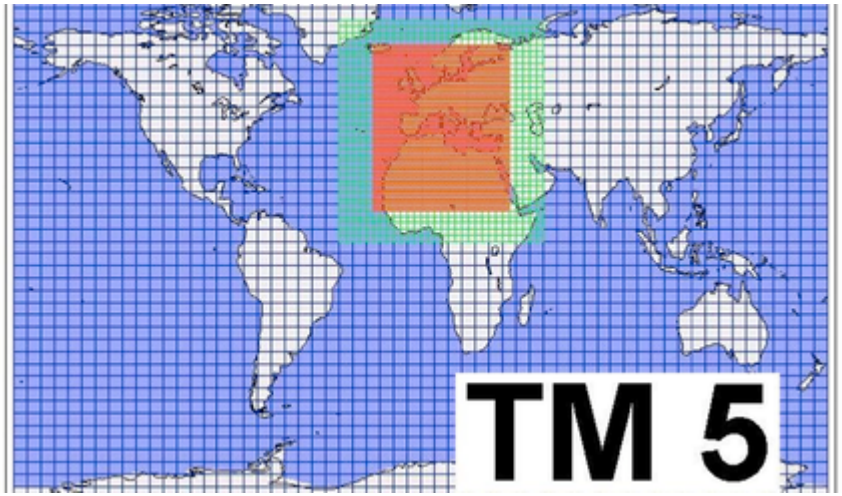
Proposal for a faster TM5

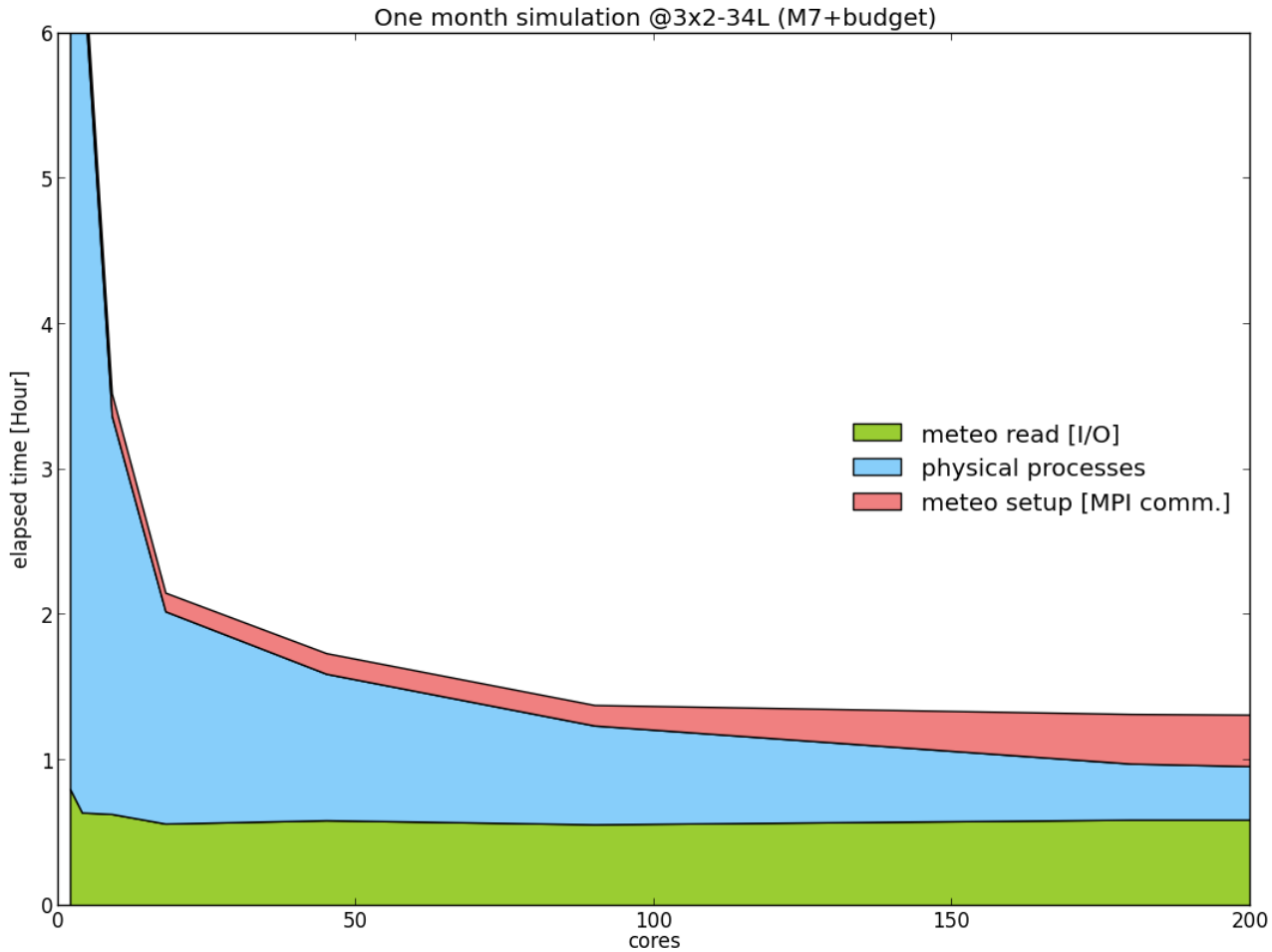
(what's up with netCDF?)



P. Le Sager
R&D Weather and Climate Modeling
KNMI

S. Basu, A. Jacobson
NOAA

The bottleneck



Meteo format & libraries

File format

- HDF4 → uncompressed netCDF4

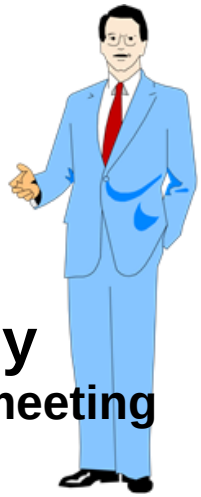
Library

- HDF4 → netCDF4 → netCDF4_parallel_IO

Required for:
Restart files
Timeseries output Chemistry

Meteo format & libraries

"Uncompressed netCDF much **SLOWER** than either of the two other options [compressed NC and HDF] !!"



Andy
last meeting

We have a problem…
buggy library?
file system issue?

How did we get there?

Speedup
=
runtime(HDF)
/
runtime(NC)

Arjo's presentations 2009-2010

- 2009-12 "one file, outside TM5"
  - nc4 : speedup ~**5** !!
  - nc4-L6 : speedup ~**3.7** !
- 2010-06
  - nc3 : speedup of **1.33** for "step init"
  - netcdf3 lib (b/c bug HDF5)
- 2010-11
  - nc4 : "read meteo" **1.25** faster

How did we get there?

Arjo's presentations 2009-2010

- 2009-12 "one file, outside TM5"
  - nc4 : speedup ~**5** !!
  - nc4-L6 : speedup ~**3.7** !
- 2010-06
  - nc3 : speedup of **1.33** for "step init"
  - netcdf3 lib (b/c bug HDF5)
- 2010-11
  - nc4 : "read meteo" **1.25** faster

How did we get there?

Arjo's presentations 2009-2010

- 2009-12 "one file, outside TM5"
  - nc4 : speedup ~**5** !!
  - nc4-L6 : speedup ~**3.7** !
- 2010-06
  - nc3 : speedup of **1.33** for "step init"
  - netcdf3 lib (b/c bug HDF5)
- 2010-11
  - nc4 : "read meteo" **1.25** faster

How did we get there?

Arj

–

–

–

ISSUES

- The smallest number was picked up
- netCDF4 lib **can** read netCDF3 (“nc4_classic”)
- Short runs (12h)
- TM5 L60 with 5 tracers, 2 zoom regions
- Concomitant runs?
- Compression was too quickly thrown away
  - L1 → L9
  - Balance b/w IO and CPU to decompress

  – nc4 : “read meteo” **1.25** faster

More comprehensive tests

	HDF4	NC4	NC4 L1	NC4 L9	NC3	NC3 L1	NC3 L9
Serial library							
Parallel library with serial IO							
Parallel library with parallel IO (TM5-MP only)							

- Runs on each row should be concomitant, and long enough
- On different machines

A word about compression

	NC4	NC3
unlimited	596	596
fixed	596	596
unlimited-L1	117	117
fixed-L1	114	114
unlimited-L9	-	99
fixed-L9	100	100

- meteo ouput is written with **unlimited** dimension
- can reduce the compression
- but not by much here

Compression
- L1 is fastest to decompress
- L9 is very slow
  for **little** more compression

Size in MB of
ec-ei-fc012up2tr3-ml60-glb100x100-2006-cld_20060124_00p03.nc
for different compression levels, time dimension types, and nc format.

A word about compression

	NC4	NC3
unlimited	596	596
fixed	596	596
unlimited-L1	117	117
fixed-L1	114	114
unlimited-L9	-	99
fixed-L9	100	100

- meteo ouput is written with **unlimited** dimension
- can reduce the compression
- but not by much here

NOT TESTED

Compression
- L1 is faster to decompress
- L9 is very slow for **little** more compression

TM5 run times with L9 are similar or larger than with L1. Not shown.

Size in MB of
ec-ei-fc012up2tr3-ml60-glb100x100-2006-cld_20060124_00p03.nc
for different compression levels, time dimension types, and nc format.

CCA (ECMWF, UK) – CRAY XC40

Environment

- CRAY and IFORT compilers (switch PrgEnv)
- netcdf 4.4.0
- Lustre file system

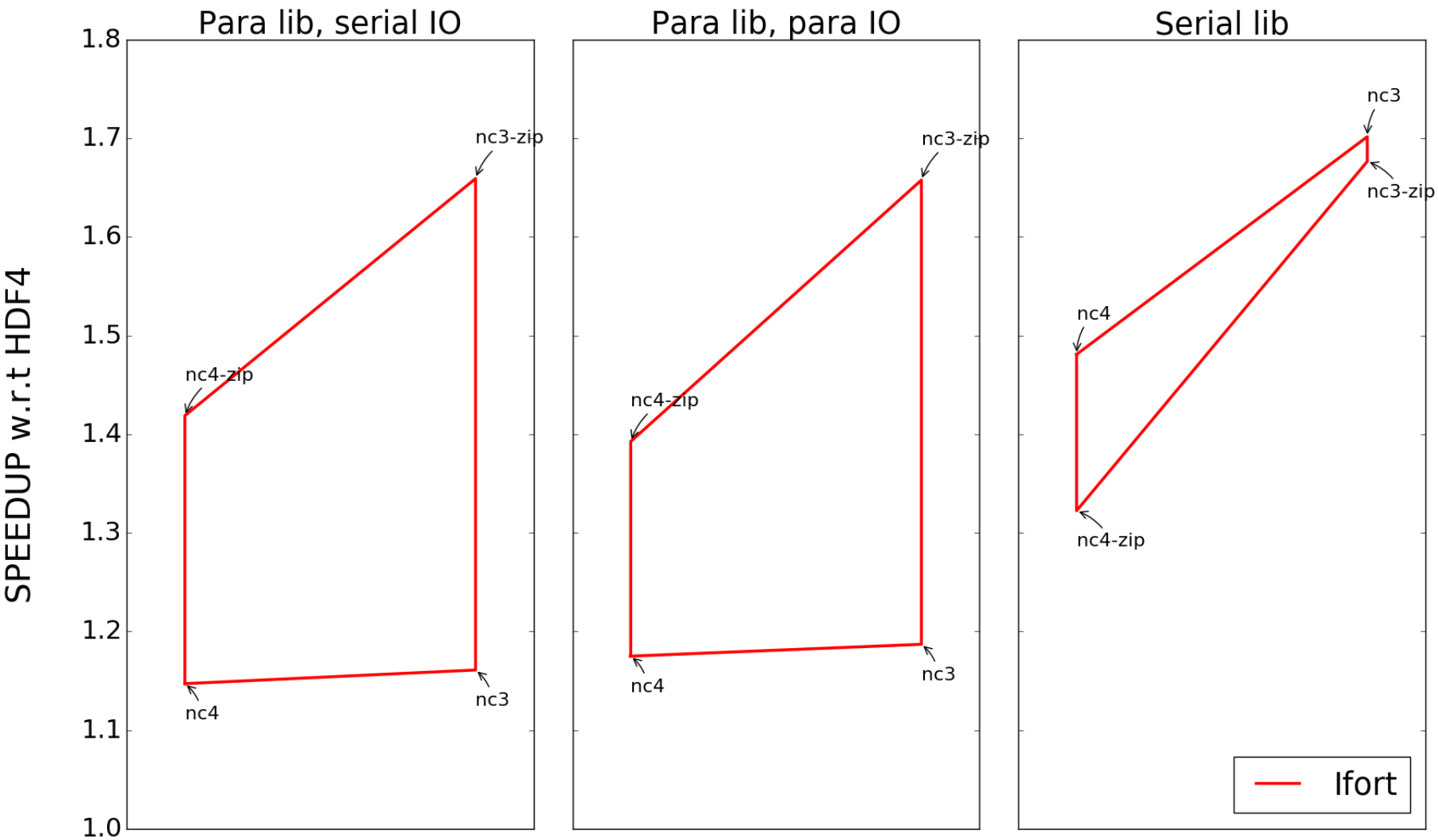
Model setup

- TM5-MP r464, full-chemistry, 1 month run, 3x45-34L
- reading 1x1-60L met fields
- without restart or timeseries output to test serial lib
- 3x45 (135) cores

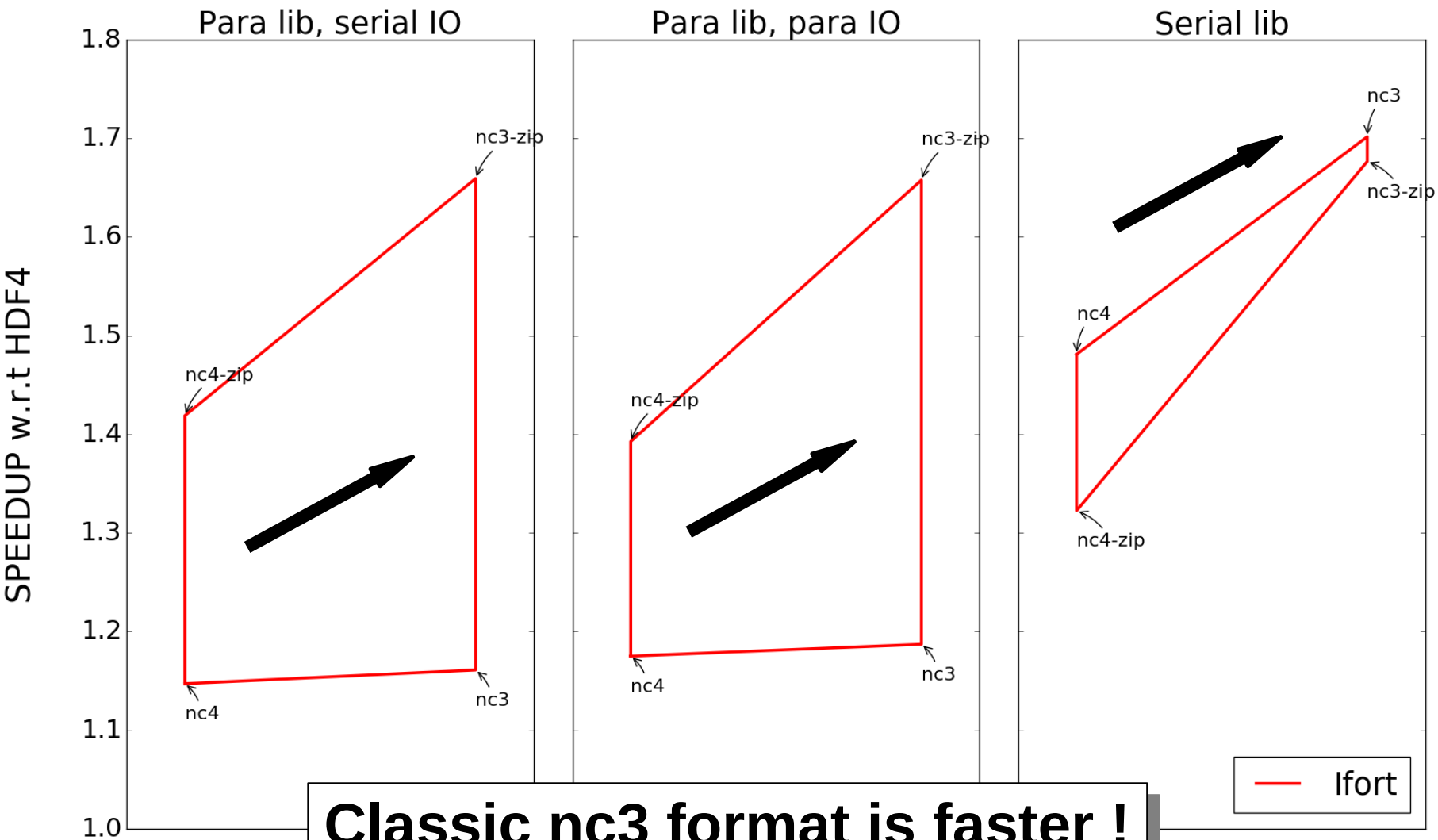
Result

- Speedup (w/r/t/ HDF4) of the “readrecord” timer (from “timing.output: T” setting)
- repeat a couple of times, and average

CCA - Ifort


Para lib, serial IO
Para lib, para IO
Serial lib
SPEEDUP w.r.t HDF4
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1.0
nc3-zip
nc4-zip
nc4
nc3
nc3-zip
nc4-zip
nc4
nc3
nc3
nc3-zip
nc4
nc4-zip
Ifort

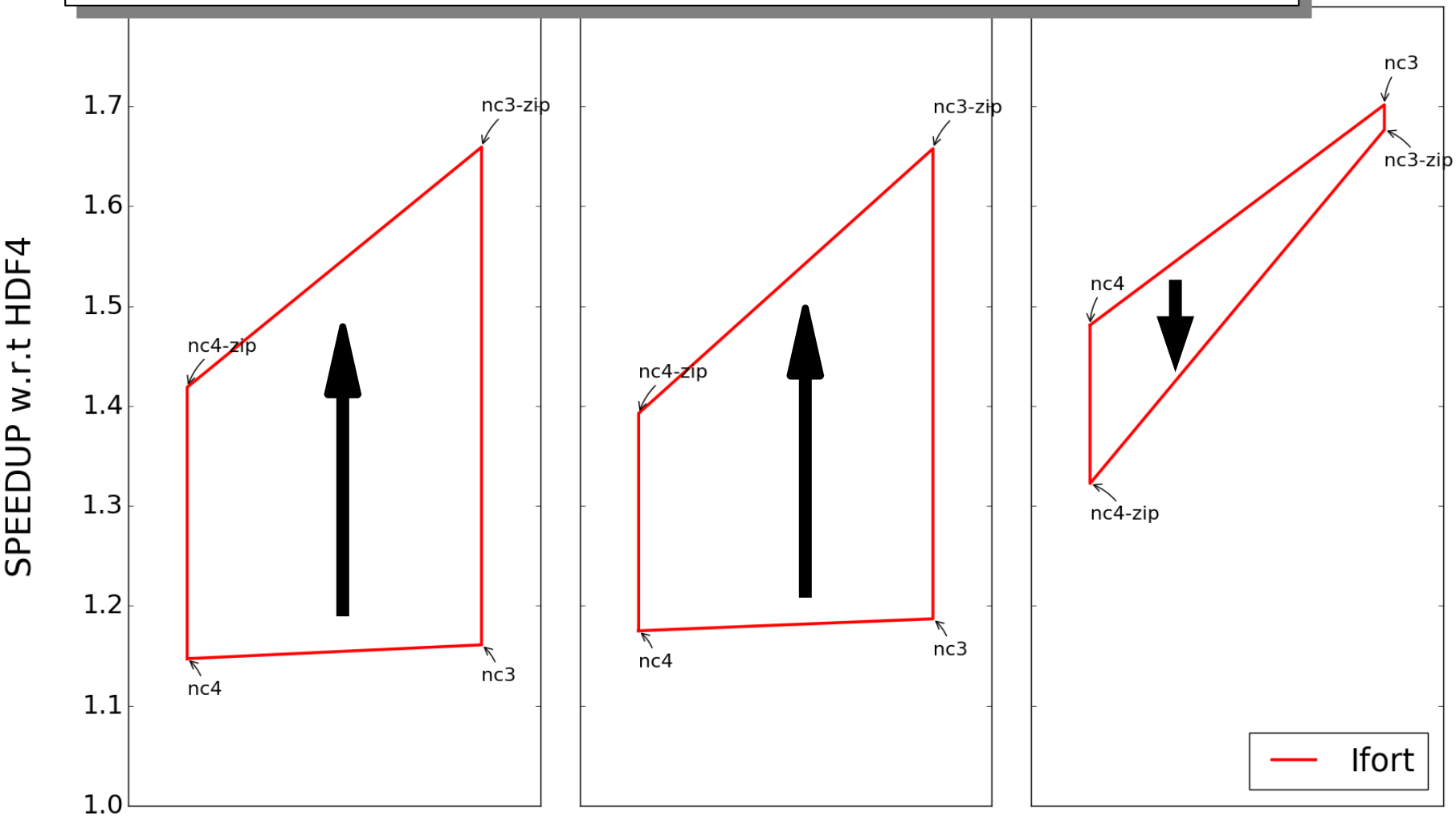
CCA - Ifort



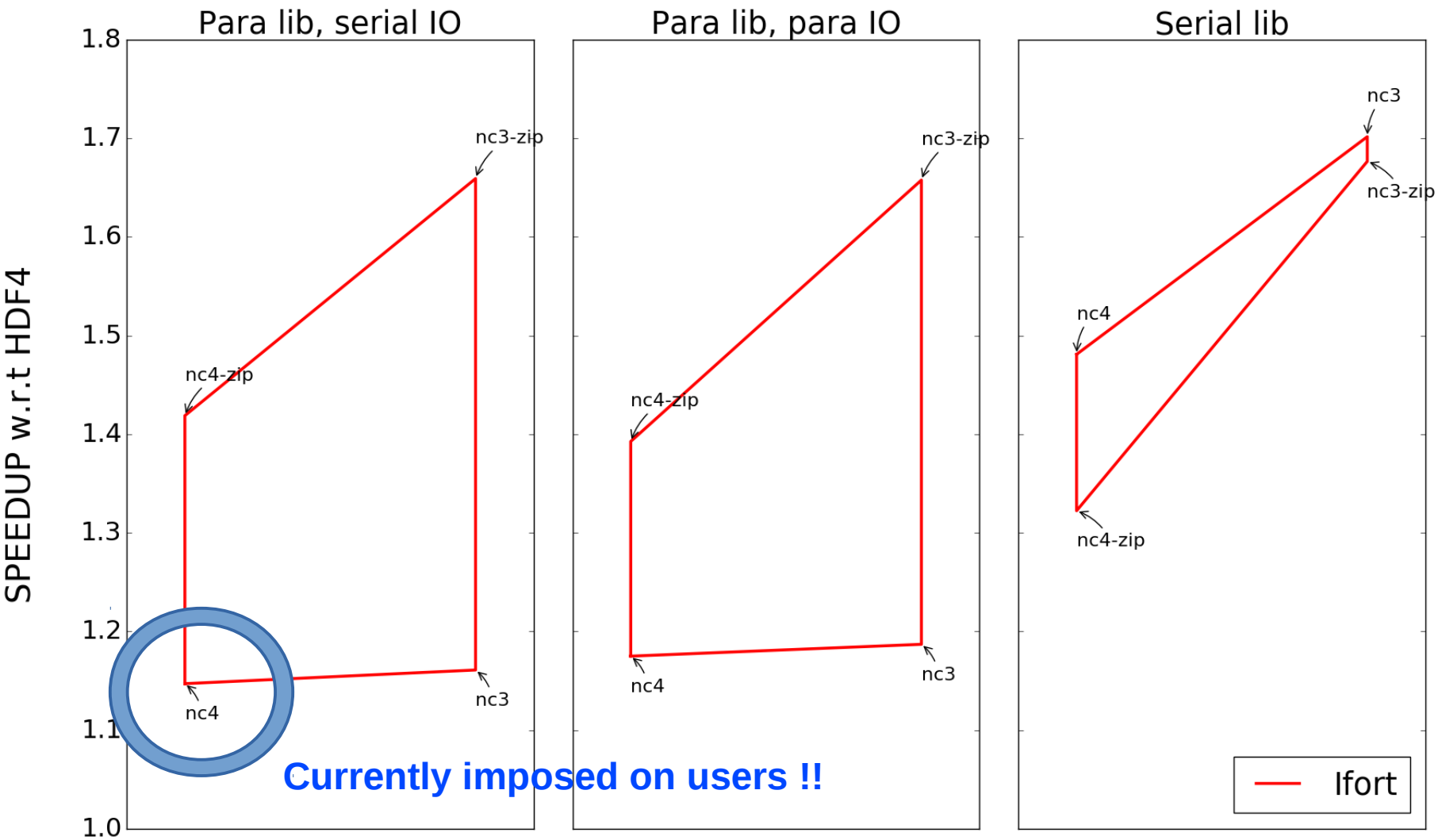
Classic nc3 format is faster !

Reading files with internal compression

- large speed up for parallel lib (larger with nc3)
- small penalty for serial lib (smaller with nc3)



CCA - Ifort



Currently imposed on users !!

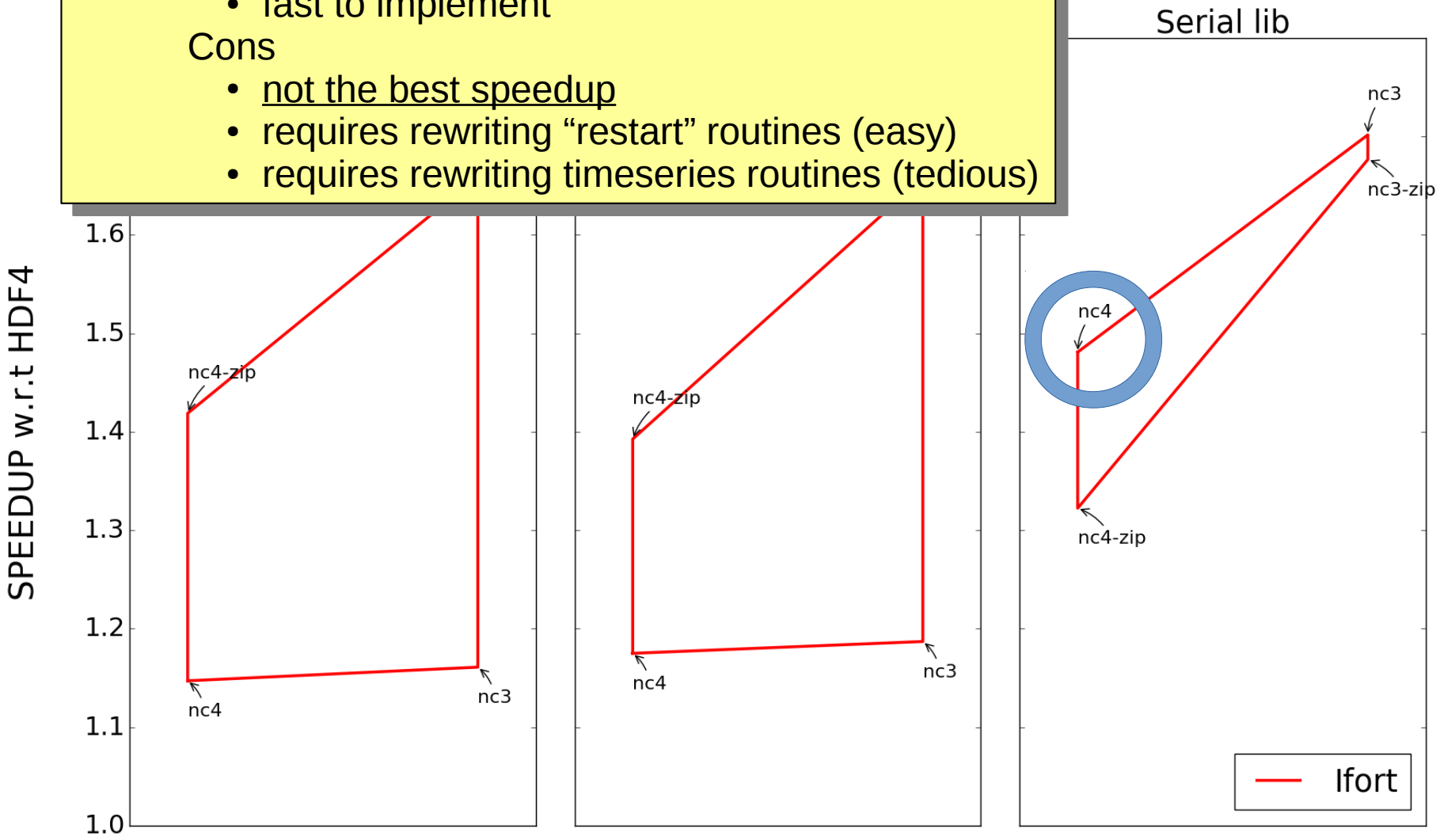
Idea #1 : switch to serial lib

Pros

- no change to the archive
- fast to implement

Cons

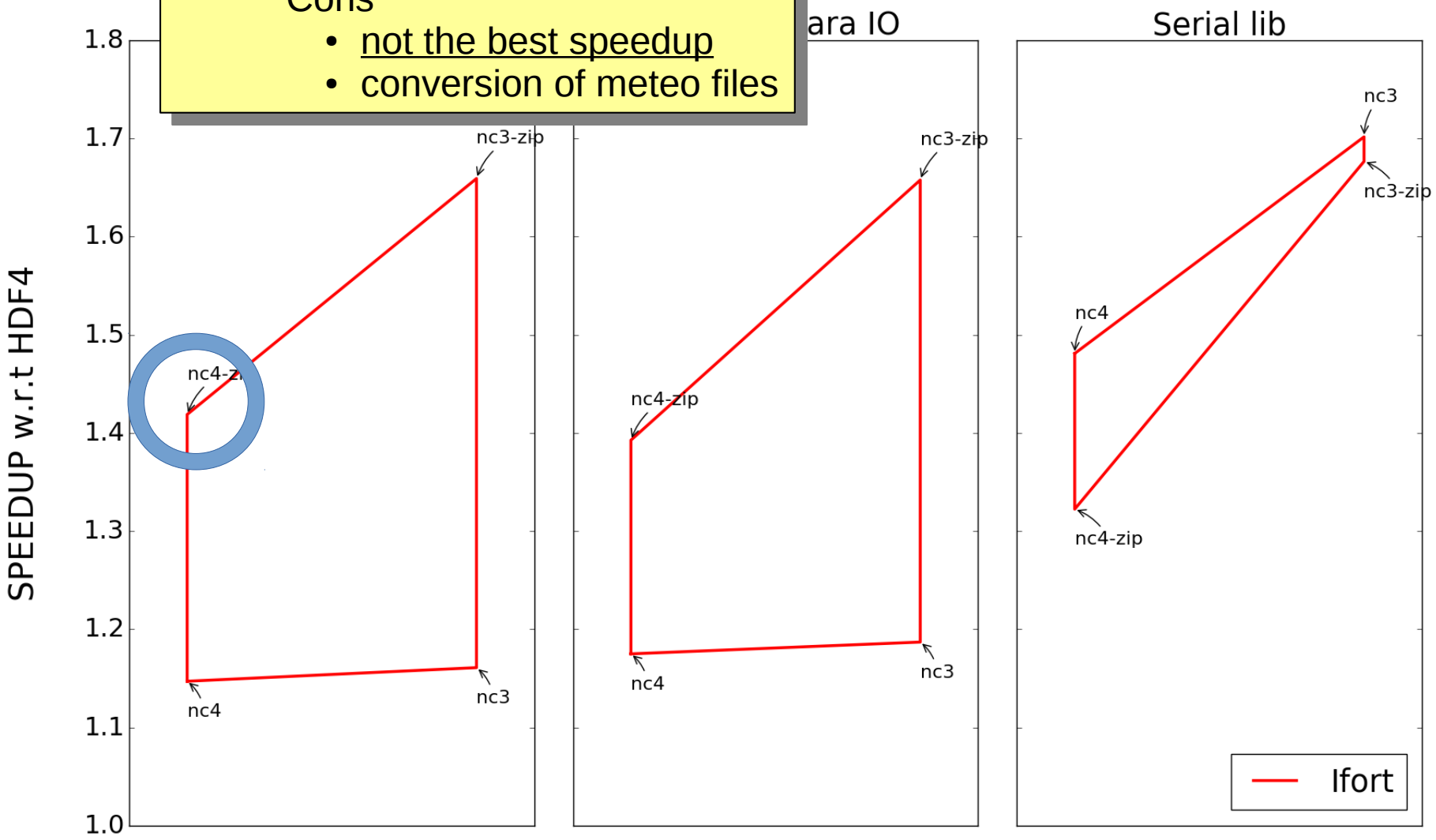
- <u>not the best speedup</u>
- requires rewriting “restart” routines (easy)
- requires rewriting timeseries routines (tedious)



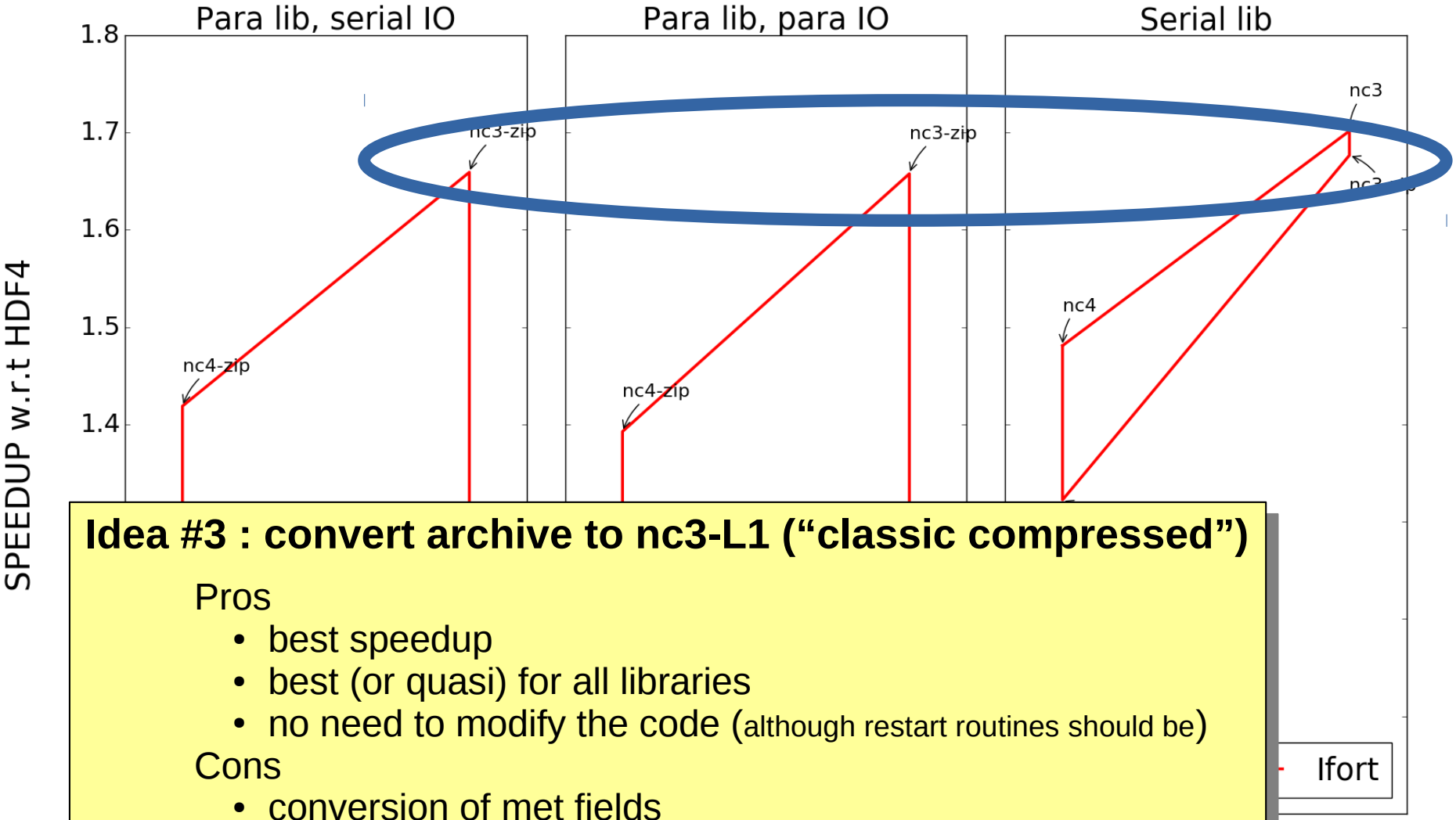
Idea #2 : Compress nc4

Pros
- no code rewrite

Cons
- <u>not the best speedup</u>
- conversion of meteo files



CCA - Ifort



Idea #3 : convert archive to nc3-L1 ("classic compressed")

Pros

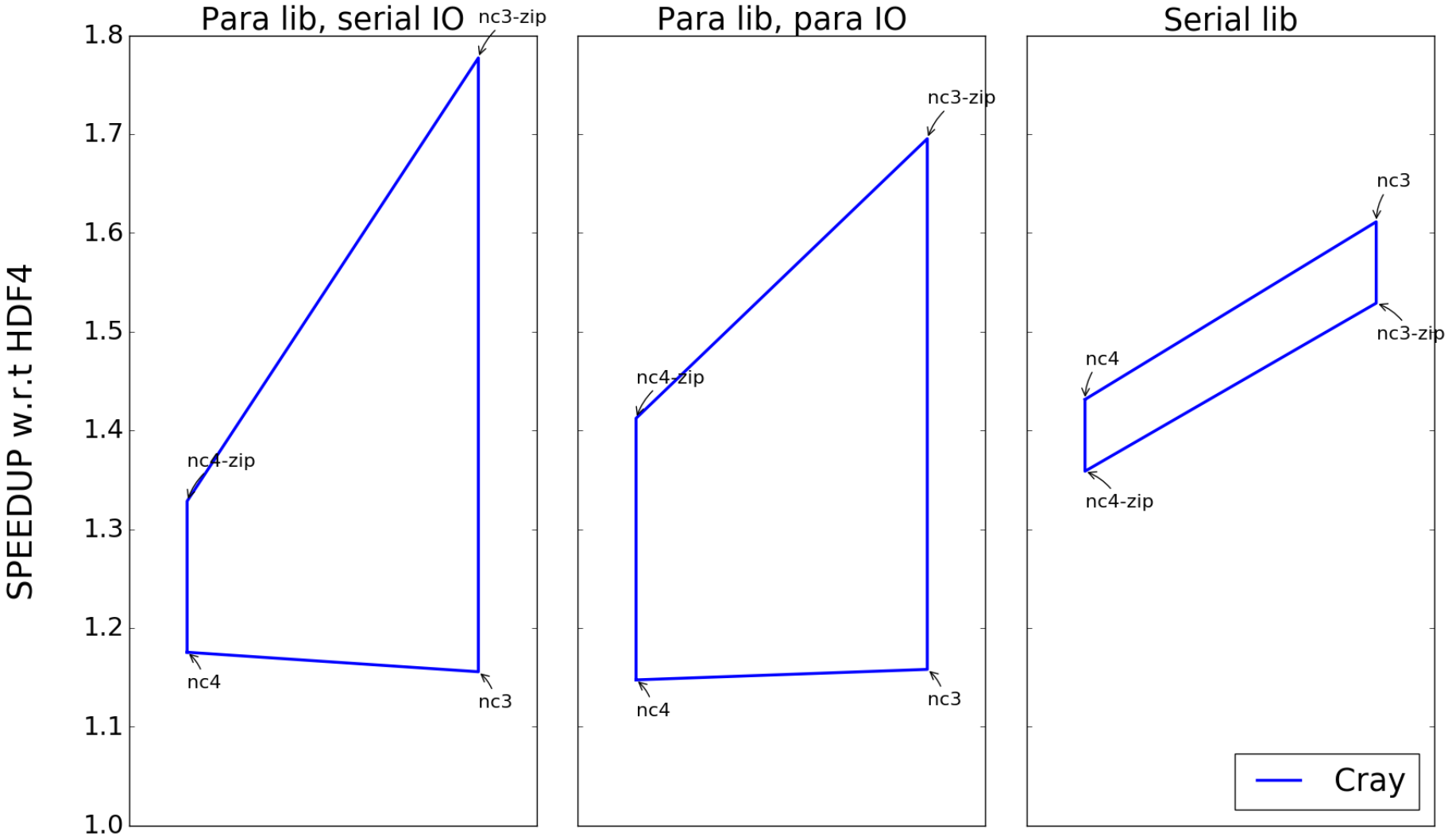
- best speedup
- best (or quasi) for all libraries
- no need to modify the code (although restart routines should be)

Cons

- <u>conversion of met fields</u>

But wait!
What about the other platforms?

CCA – Cray compiler



Cartesius (SurfSARA, NL)
Theia (NOAA, USA)

With nc4 files

Paralle lib:
compressed files is faster than uncompressed
Serial lib:
uncompressed files is faster than compressed

Same time to read compressed files
with serial and parallel libraries

Uncompressed much longer in the parallel
than in the serial lib

Sourish

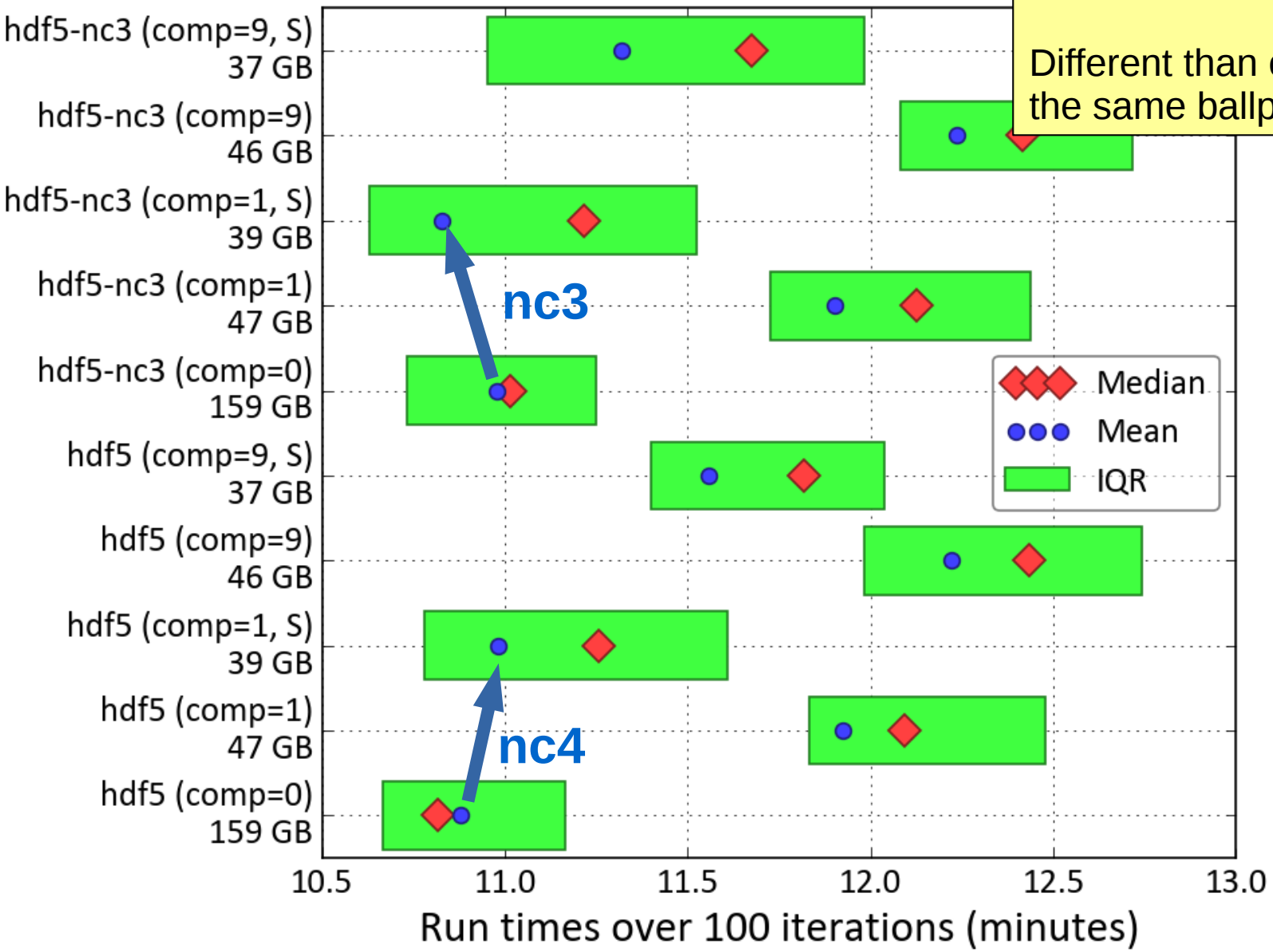
SAME RELATIVE RESULTS !

Cartesius (SurfSARA, NL)

SERIAL LIB

Total TM5 runtime for 2 months, reading glb1x1: nc3-L1 & nc4 are the best

Different than cca, but in the same ballpark



Theia (NOAA, USA)

Huge difference:
the speed up relative to HDF with the parallel lib:

compressed nc4 is 20% faster than HDF4,
but
uncompressed nc4 is something like 2x slower

Andy

is HDF4 super fast
or
netCDF4_para lib performance not optimum ??

Conclusion

1. Stop using HDF4
2. With current nc4, zip it OR use serial lib if possible

   (Sourish already has a restart module for some versions)

3. Rewrite the restart module to do IO in serial

   (needed regardless of the meteo strategy, b/c //-write does not scale)

Conclusion

1. Stop using HDF4
2. With current nc4, zip it OR use serial lib if possible

   (Sourish already has a restart module for some versions)
3. Rewrite the restart module to do IO in serial

   (needed regardless of the meteo strategy, b/c //-write does not scale)

4. Convert meteo archive to classic-L1 ??

   could be done locally, but WAIT, because:

5. **Read all time steps in a file <u>at once</u> (at least in TM5-MP)**

   larger (than from switching format) speedup expected

   maybe 5x faster like Arjo's first test ?

   redo all these lib/fmt tests

Explanation

How can “classic” be faster than netcdf4 format?

“the IO is **not handled by hdf5** if you read classic format,
but directly through an I/O layer implemented
much like the C standard I/O library”

Why can compression lead to faster IO?

time saved reading less data, larger than:
extra computing time to decompress data

(but not necessary always the case)

Technicalities

F90 code enforces nc4 format when writing met field.
To produce netcdf4_classic format, you have to either

- modify the code, or
- do it in a post-processing step

Meteo output is written with unlimited dimension:

- postprocessing needed outside the fortran code to remove the unlimited dimension (a simple ncks cmd)
- compression for netcdf_classic (nc3) cannot be done by simple ncks. Need to use nccopy with the shuffle option. But that may depends on your NCO version.