Some computational aspects of HARMONIE

Ulf Andræ, SMHI
Trygve Aspelien, Ole Vignes, Dag Bjørge met.no
Torgny Faxén, NSC
Jacob Weissman Poulsen DMI

ALADIN 23nd WS - HIRLAM ASM
Reykjavik
15-19th April 2013

Content

- Benchmark activities

- Diagnosing the IO performance of cy38

- Portability

- Surface assimilation lesson learned

Benchmarking

- Several HIRLAM countries are in the process of upgrading their HPC.
- A benchmarking package have been created
  - harmonie-38h1.alpha.2, no netcdf dep, wrgp2fa.F90 update, OpenMP fixes,
  - Simple sample scripts
  - Required background data
  - 1h boundaries up to 6h for several domains with 2-5km and 65 vertical levels

  XS: 50x50, M:384x400, L:750x960, XL:1200x1200, XXL:1600x1600
- https://hirlam.org/trac/wiki/HarmonieSystemDocumentation/HarmonieBenchMark

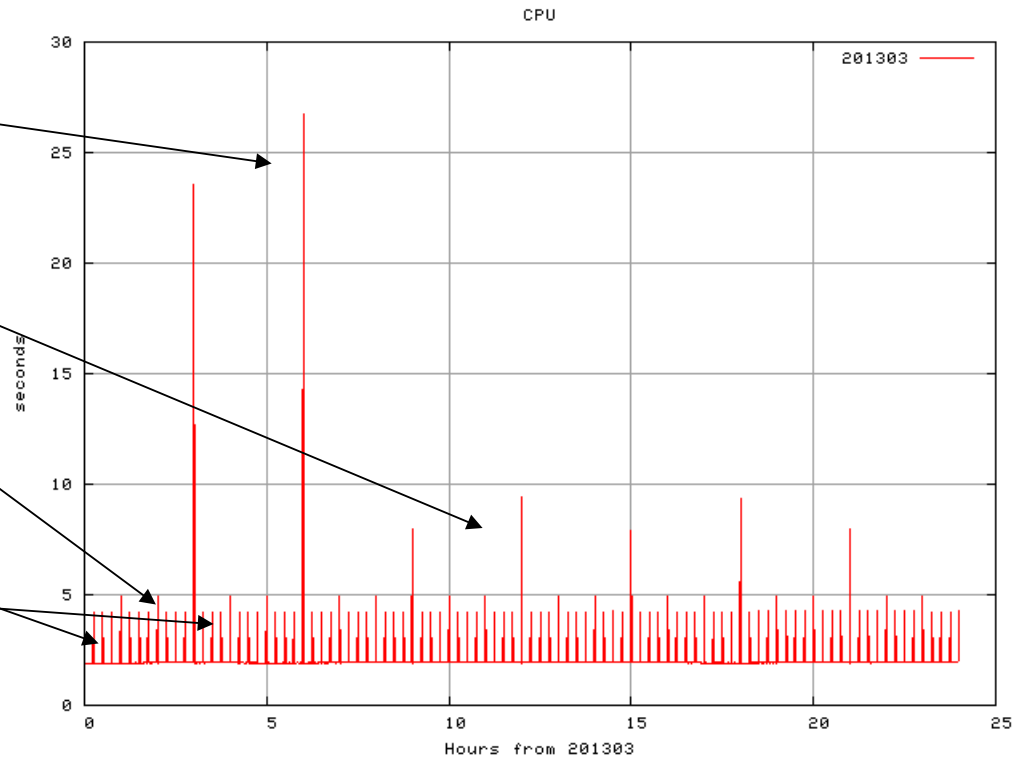
Benchmarking, some properties

- **Runs with EDKF (EDMFM had problems)**
- **Tested for IBM, gfortran, intel**
- **MPI reproducible, different decompositions**
- **OpenMP reproducbile for different number of threads**
- **Reproducibility issues with MKL libraries (intel) but good performance and reproducibility with other blas/lapack libraries.**
- **Forecast model only. Assimilation still considered second of importance (or to complicated to deal with)**

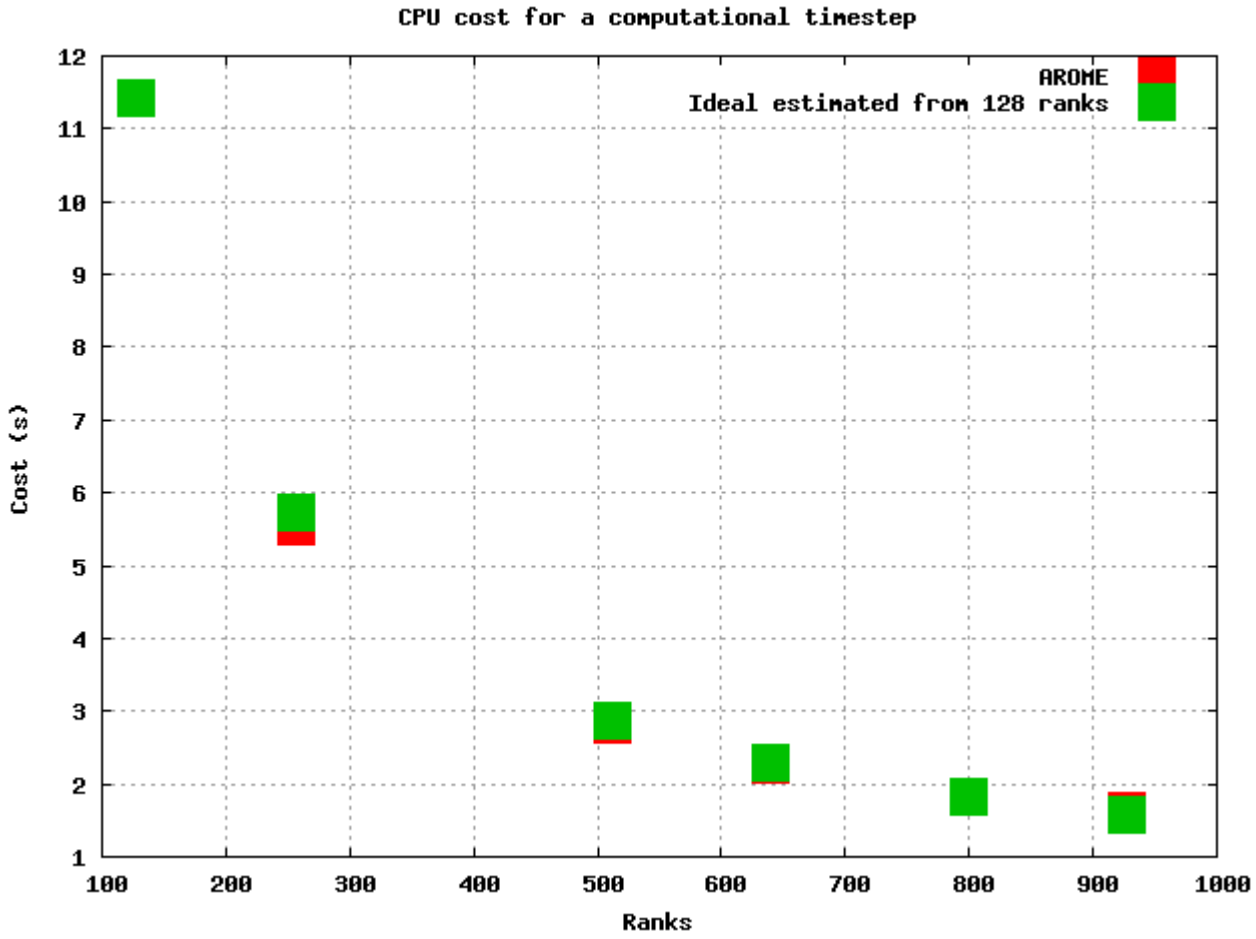
The typical cost of a forecast

- **SURFEX lfi output**
- **FA output**
- **BD input**

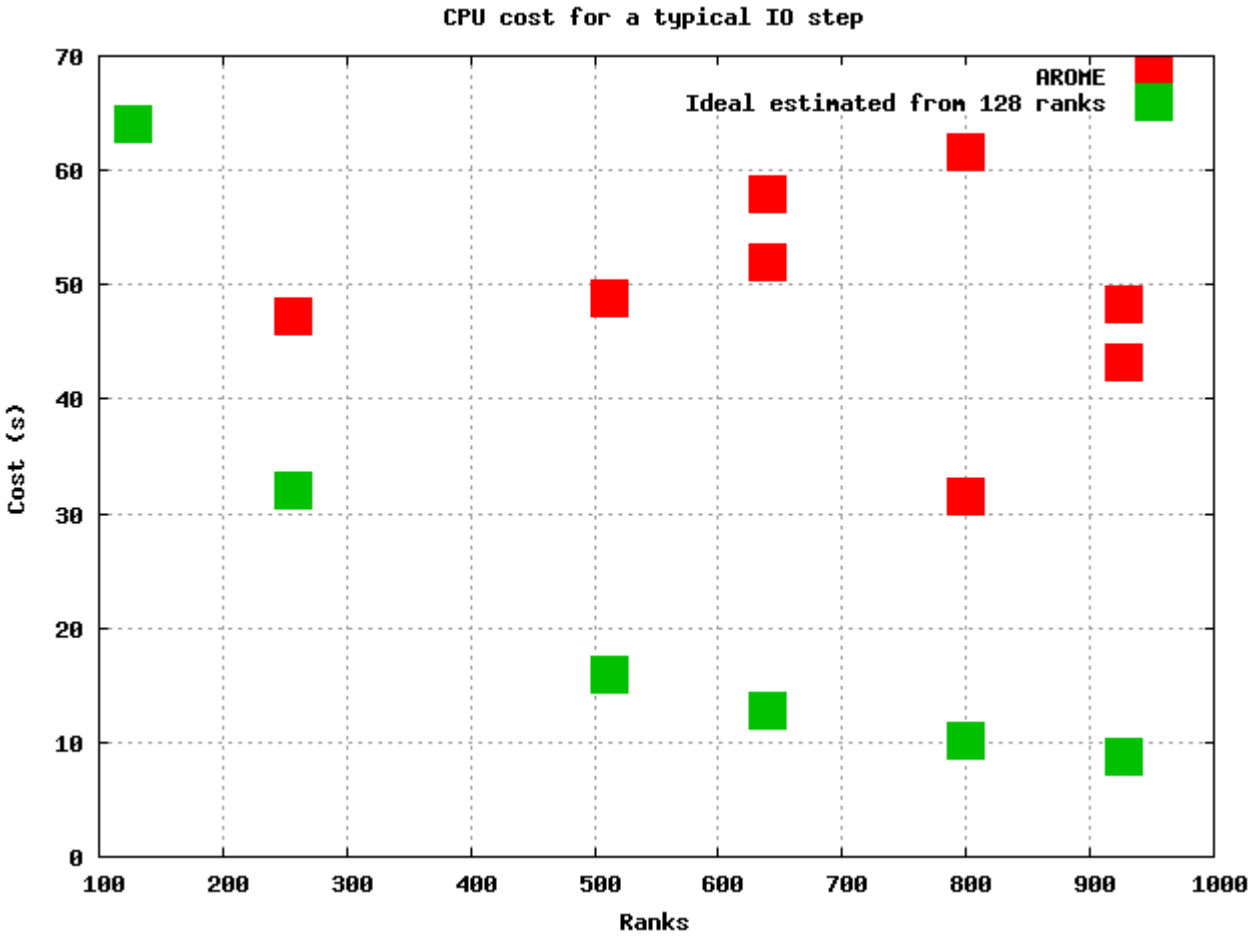
- **Computations**
- **Radiation timestep**



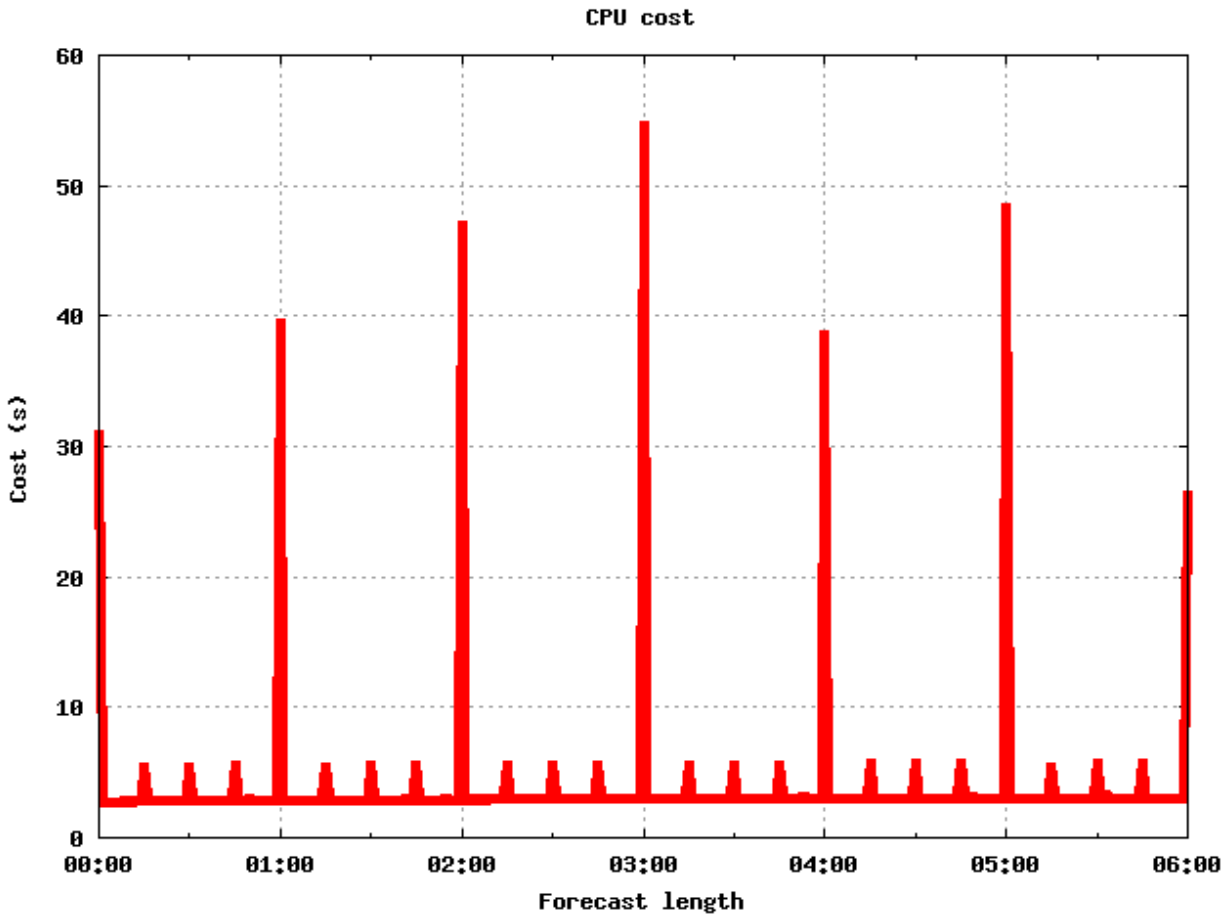
Scalability on a 1200x1200x65 domain
Sandybridge 2.2GHz 16 core nodes
Mellanox infinband



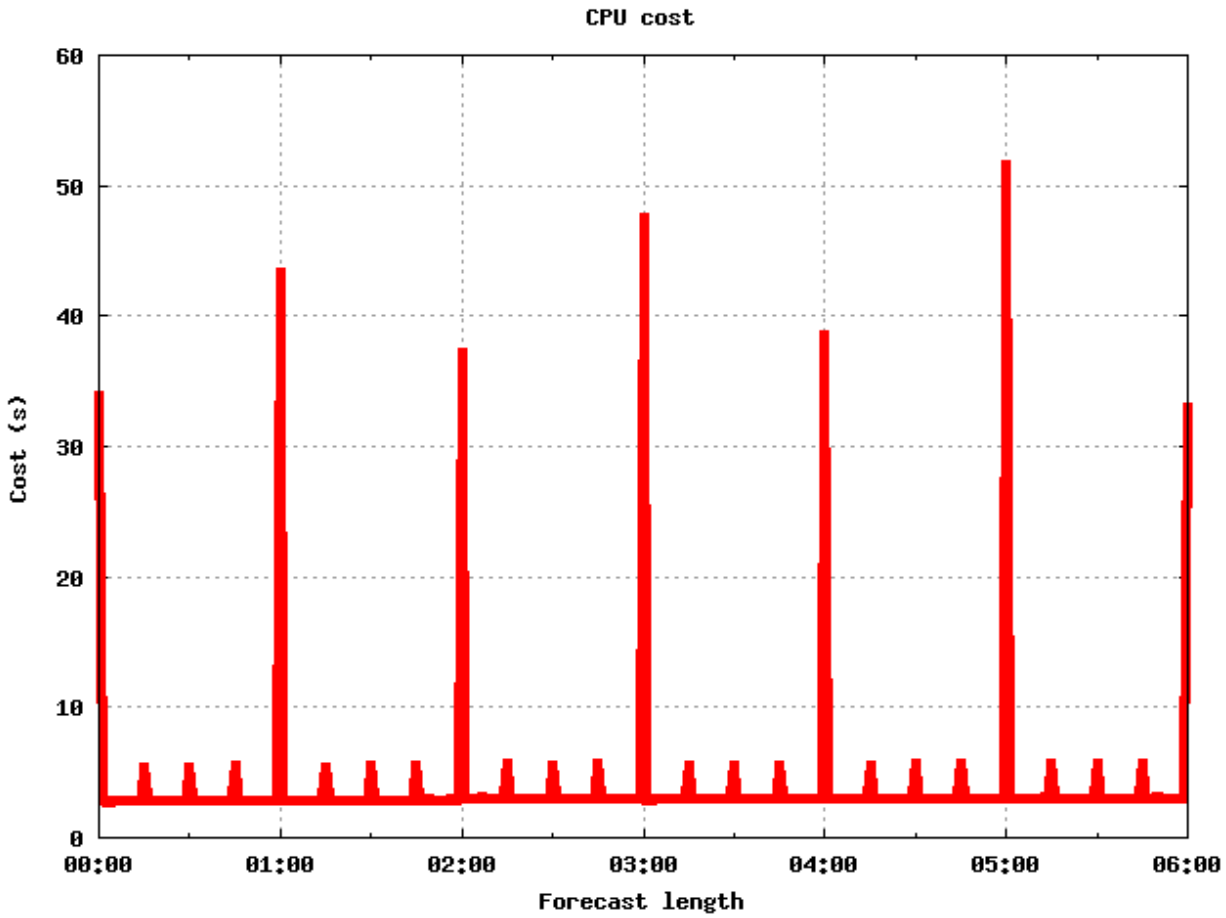
IO step only


CPU cost for a typical IO step
AROME
Ideal estimated from 128 ranks
Cost (s)
Ranks
0
10
20
30
40
50
60
70
100
200
300
400
500
600
700
800
900
1000

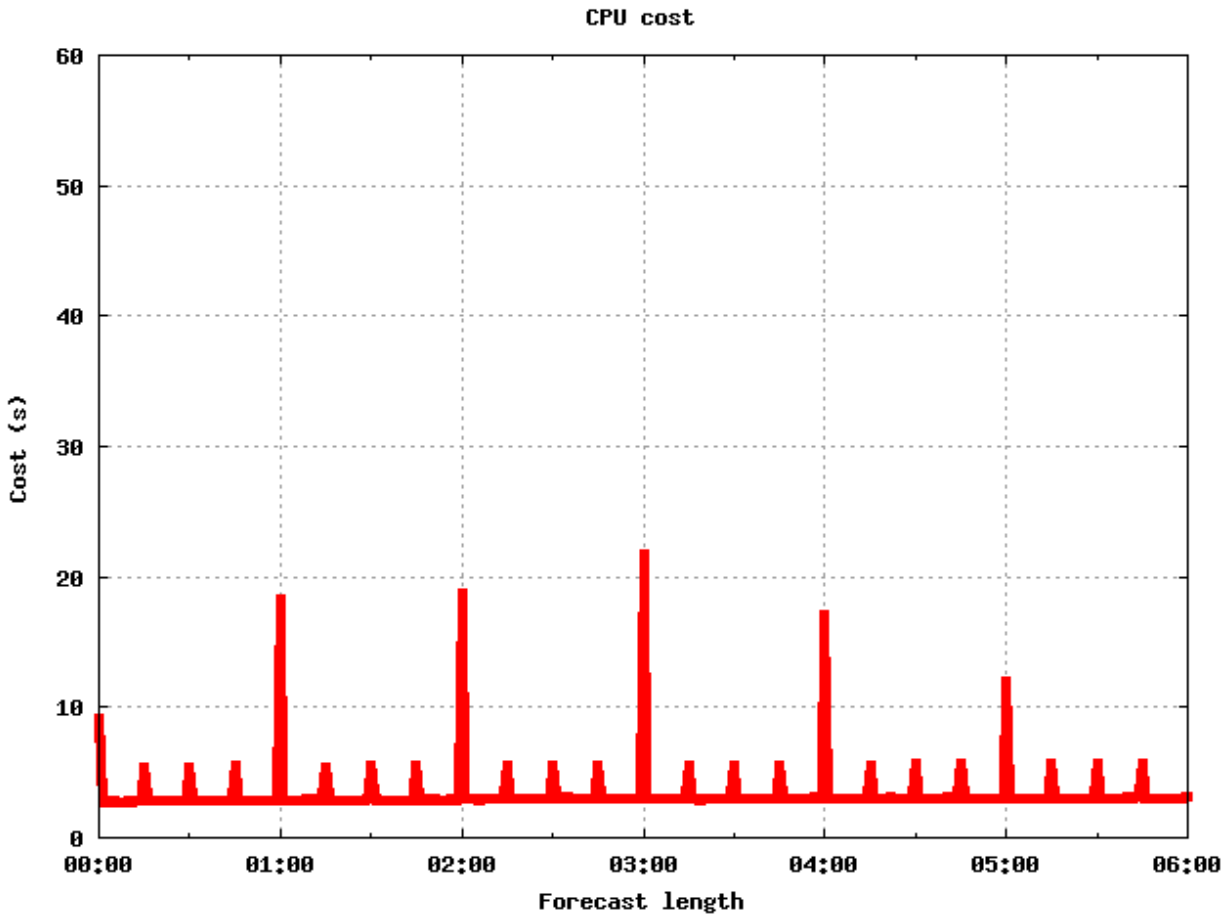
IO FA + LFI


CPU cost
Cost (s)
0
10
20
30
40
50
60
00:00
01:00
02:00
03:00
04:00
05:00
06:00
Forecast length

IO FA+SURFEX as FA



IO FA + SURFEX as FA + IO SERVER



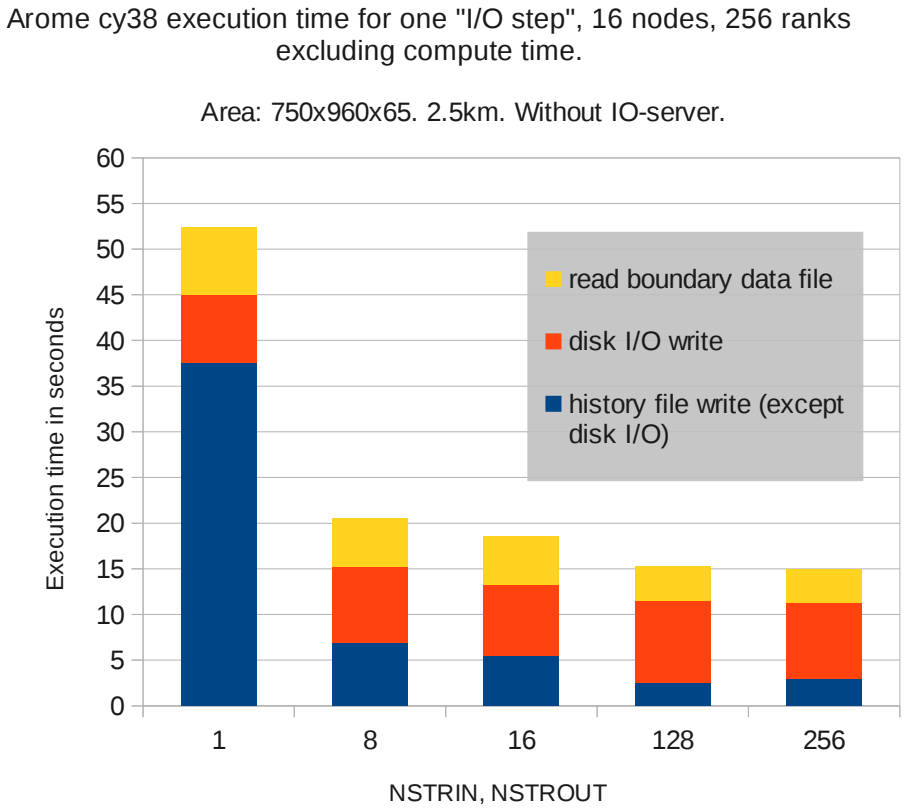
A more careful look on the IO steps (without IO server)

Runs done on Lustre file system, maximum BW for single file 120 MB/sec.

“Excluding compute” means subtracting the time for an ordinary time step or radiation time step.

NSTROUT important to maximize.

NSTRIN has rather small effect on execution time and can vary a lot depending on what else is going on and where file resides. Factor of three slower when NSTRIN=1 → NSTRIN=8 observed in one case!



Courtesy Torgny Faxen NSC

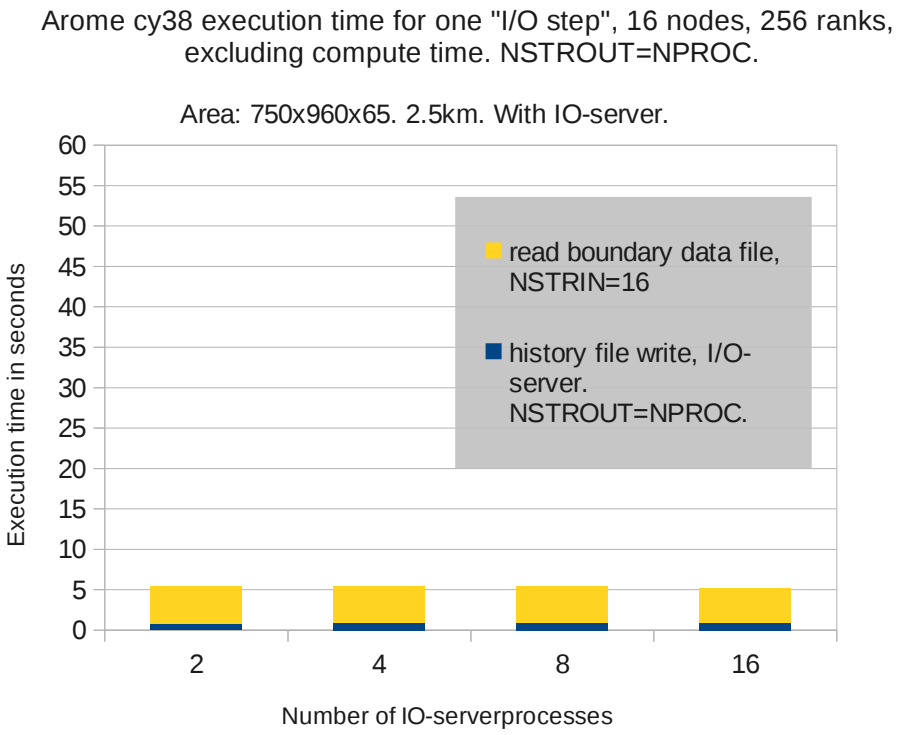
IO performance with IO server.

IO-server works fine. Time to concatenate output files not included though. Maybe reading of output data can be done in parallel instead?

With IO-server it seems like reading of boundary data is now the largest time consuming routine.

How to improve reading of boundary files?

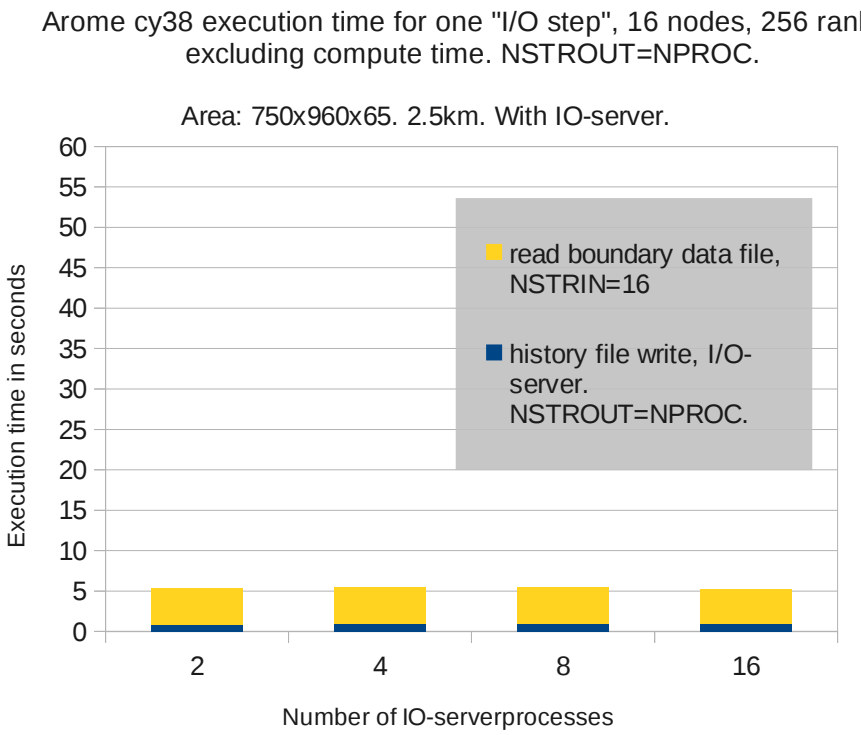
- Asynchronously through the IO-server should be possible?
- Modify the actual READ?



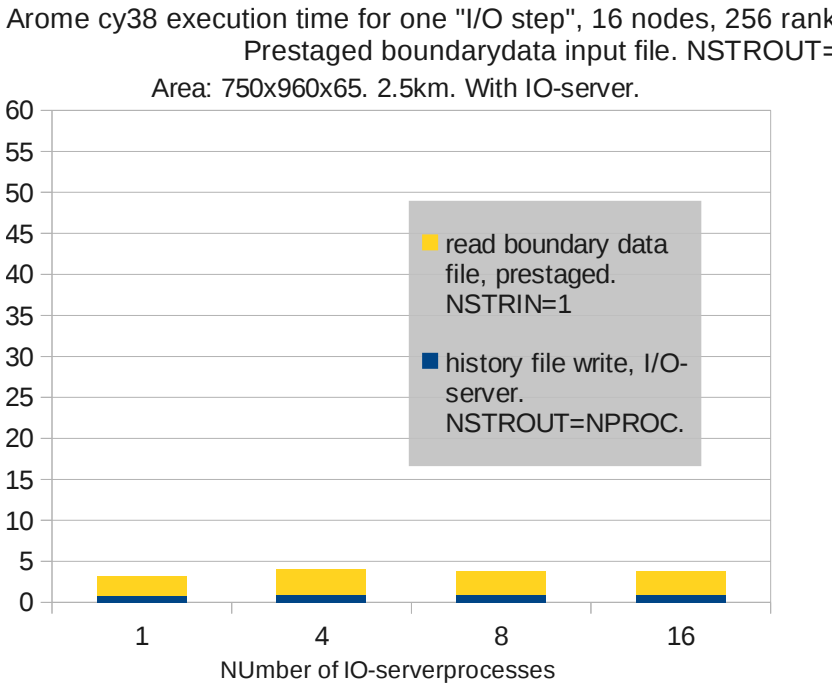
Courtesy Torgny Faxen NSC

Better performing by prestaging the input file (just an example)

Default



Prestage



So we believe we have a reasonably well working benchmark package!

Well....

Testing the portability (cy37h1)

Mixed MPI OpenMP

mpi	1 thread	2 threads	12 threads
pathscale	X	X	X
intel	X	X	X
cray	X	X	F
gfortran	X	X	F
pgi	F		

Pure MPI with different compiler options

MPI	Default	ieee	stack	bound
pathscale	X	X	F	F
gfortran	X	X	X	F
cray	X	X	X	F
intel	X	F	F	F
pgi	F			

Compilation warnings and interface problems some examples

- Fortran pointer variable ”FOO” is being used before being pointer assigned or allocated (2)
- Variable ”FOO” is used before it is defined (95)
- Dummy argument ”FOO” has the INTENT(OUT) attribute, but is never assigned a value or used as an actual argument (73)
- “FOO” is used but never set (31)
- Argument type differ from declaration (49)

(some of these were sent as corrections to cy39t1.)

Some warnings are more harmful than others

It's natural that “real” errors are dealt with first, but how can be do better here? (back to the cycling strategy)

How to speedup your code

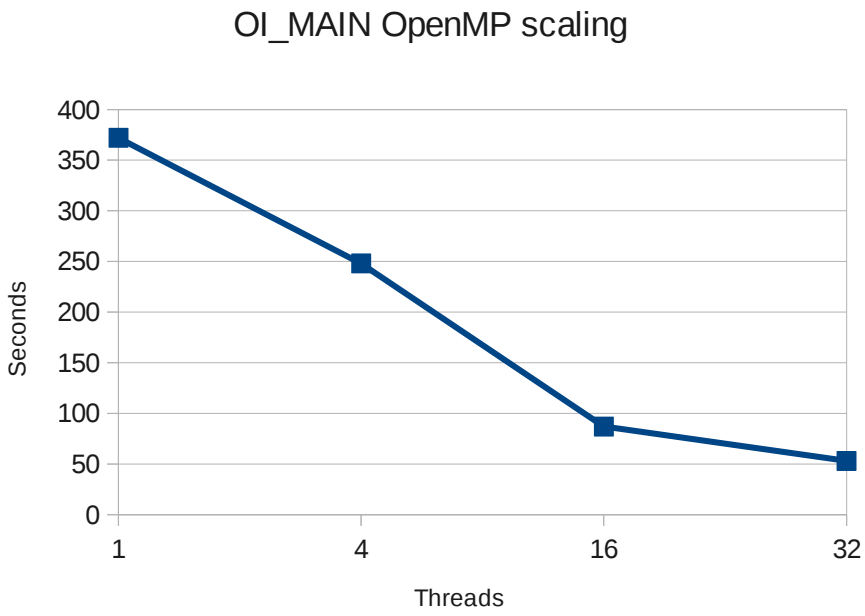
- **Nothing beats doing less**
- **If you have to do it, do it better**
- **Share your work**
  - **OpenMP, loop, single node parallelisation**
  - **MPI, distributed computations**

Fighting with the surface assimilation

- **On the way to 37h1.2 we modifided OI_MAIN and introduced SODA in parallel.**
  - **Increased the cost ~10 times compared to 37h1.1 OI main f**
  - **The bad setup) a**
  - **Could de extrapol**
- **In cy38h1 we have OI main comes for**
  - **HARMON inside CA**
  - **Extrapol[illegible] environment**

Back to OI_main

- **Share the work, add OpenMP directives to the painful part in the extrapolatio**
- **Scaling example f MetCoOp domain reasonable numbe**
- **All tricks tried**
  - **Doing less**
  - **Share with Open**
  - **Doing it right: Re reason for extrapolations!**



Courtesy Ole Vignes met.no

Conclusions

- We have a benchmark package for the forecast model in cy38h1

- The IO server works well for output. Will be optional in HARMONIE together with SURFEX FA/LFI output. More work needed for input

- Several versions of surface assimilation exists with different computational and meteorological properties. Convergence discussion started!

- Our system is sem-implicit, semi-lagrangian, semi-portable and semi-fortran standard compliant

Thanks again for your attention
Questions?