FULLPOS-2

T. Dalkiliç (Meteoroloji Genel Müdürlüğü)

R. El Khatib (Météo-France)

and also

Karim Yessad (MF), Antonio Stanešić (DHZ)

Daan Degrauwe (IRM), Fabrice Voitus (MF)

23rd Aladin Workshop – Hirlam All Staff Meeting 2013

Reykjavik 15-19/04/2013

METEO FRANCE
Toujours un temps d'avance

Plan

- Introduction
- Technical aspects
- Validation issues
- Performance
- Next developments

Introduction : What is Fullpos

- Full-Pos is a post-processing software used for :
  - **Gridpoint interpolations + spectral filtering for backend post-processing**
  - **Spectral models geometry changes** (« configurations 927 »)
    - Global to Global (4DVar, Ensemble forecast)
    - Global to LAM (Coupling)
    - LAM to LAM (Nesting)
- **High performance is crucial** for models geometry changes in operations
- Full-Pos has been designed to **re-use existing pieces of software in the model**, especially :
  - Spectral transforms
  - Horizontal communications
  - => Overall dataflow

Mechanism of Fullpos « configuration 927 »

Because the spectral transforms software was not external, a complex mechanism has been developped in 1993 :

First part :

- Setup in the input model geometry
- Horizontal interpolations of the input model fields
- **Save to disk intermediate results**
- **Deallocate all arrays ; restart from top :**

  Second part :

  - Setup in the output model geometry
  - (Spectral fit of formerly interpolated fields)
  - Vertical interpolations
  - Spectral fit of interpolated fields
  - Save to disk final results
- **... Leave !**

METEO FRANCE
Toujours un temps d'avance

Concept of Fullpos-2

Since 2003

the spectral transforms package has been externalized, making possible a simpler 1-part mechanism :

- Setup in the input model geometry + Setup of the output geometry
- Horizontal interpolations of the input model fields
- **Data transposition toward the output model geometry**
- (Spectral fit of output orography if needed)
- Vertical interpolations
- Spectral fit of interpolated fields
- Save to disk final results

=> **Easier** : a change of geometry becomes straightforward,
like a back-end post-processing

=> **Expected faster and more scalable** : less setup + less I/Os

Milestones toward Fullpos-2

A long way to run ...

- **2007** : preliminar gridpoint transposition for Fullpos – not considering spectral transforms aspects (K. Yessad)
- **2008** : check multi-spectral capability of Aladin spectral transforms + externalize the biperiodicization (A. Stanešić)
- **2010** : OOPS project offers the opportunity to officially start this « new Fullpos 927 » framework (T. Dalkiliç)
- **2012** : Boyd biperiodicization brings up (unexpectedly !) the solution to a common code architecture for Global/LAM gridpoint transposition handling (D. Degrauwe, Fabrice Voitus)

- **Cycle 39** : First release of Fullpos-2

Technical aspects : user control

Fullpos is
in cycle 38T1 and before :

- NFPOS=0 :
  Fullpos is switched off
- NFPOS=1 : Post-processing
  for backend usage
- NFPOS=927 : Changes
  of models geometry

Fullpos-2 :
a slight change of paradigm
where NFPOS=2
should replace NFPOS=927

Fullpos-2 is
in cycle 39 and after :

- NFPOS=0 :
  Fullpos is switched off
- NFPOS=1 : Post-processing
  to make gridpoint fields
- NFPOS=927 : Changes
  of models geometry
- NFPOS=2 : Post-processing
  to make spectral field
- NFPOS=928 :
  An optimization of NFPOS=927
  which can validate NFPOS=2

Technical aspects : STEPO for Fullpos

- **Fullpos was using STEPO (= a model time step structure)**
  - In STEPO,
    - Input data = model data
    - Output data = model data
  - => using STEPO in Fullpos had needed specific developments
  - => complex : 1 fullpos/NFPOS=927 step => 5 calls to STEPO
- **Fullpos-2 uses its own control subroutine STEPO_FPOS :**
  - In STEPO_FPOS,
    - Input data = model data
    - Output data = *target* data
  - 1 fullpos-2/NFPOS=927 step => 2 calls to STEPO_FPOS
  - 1 fullpos-2/NFPOS=2 step => 1 call   to STEPO_FPOS

Technical aspects : Extension zone (1)

- **In Fullpos, «C+I» and «E» are stored in 2 distinct arrays**
  - Interpolate over « C+I »
  - Fill « C+I+E » with « C+I »
  - Transpose then biperiodicize over « C+I+E »
  - Extract « E » from « C+I+E »
  - Re-transpose
  - Merge « C+I » and « E »
  - Write out
- **In Fullpos-2, «C+I» and «E» are fused in a single array**
  - Interpolate over « C+I+E »

    => extra interpolations over a virtual E-zone
  - Transpose then biperiodicize over « C+I+E »
  - Re-transpose
  - Write out

Technical aspects : Extension zone (2)

Consequences of a single array for C+I+E en Fullpos-2 :

- **Less code to maintain**
  - No more E-zone specific buffers
- **Surfex could get an extension zone from Fullpos-2**
  - Not the same behavior in cycles 38t1 and 39 !
- **The biperiodicization algorithm *must* be idempotent over C+I, too**
  - Validation issues on physical fields
- **Unlike in the Boyd biperiodicization, the E-zone location is virtual and not necessarily geographic**
  - Located at the antipodes of the target domain
  - => Extra cost ≈ 0 with at least 2 MPI tasks

Technical aspects : other modifications

- Setup reorganization
- **Extensive use of transforms inquiries** for dimensionning variables
- Spectral transforms for Fullpos re-written
- Spectral filters computation re-organized
- **Use of spectral transforms communication routines** to gather the output spectral arrays
- **Substantial re-write of gridpoint management**, including **memory savings in cycle 39T1**
- Enhancements in optimizations and open-MP parallelization
- Cleaning, preparing the removal of NFPOS=927/928

Validation issue : spectral fit of orography

- In Fullpos with NFPOS=927,
  - Beside the output orography, the algorithm is such that the interpolated orography (needed for vertical corrections) is fitted in spectral space
  - Consequently the interpolated surface temperature (needed for vertical corrections, too) is fitted in spectral space in order to fit the interpolated orography
- In Fullpos-2 with NFPOS=2,
  - There is no need to fit in spectral space anything but the output orography
- => How to validate NFPOS=2 with respect to NFPOS=927 ?
  - **=> NFPOS=928 : like NFPOS=927 but only the output orography is fitted in spectral space :**

    possible since the existence of GMV/GFL structure

Validation issue for LAM : extension zone (1)

- In Fullpos with NFPOS=927,
  - The biperiodicization is performed once :
    - After the horizontal interpolations,
    - Before the intermediate spectral fit

    => The extension zone is interpolated on the vertical
- In Fullpos-2 with NFPOS=2,
  - The biperiodicization is performed once :
    - After the horizontal *and* the vertical interpolations,
    - Before the final spectral fit

    => The extension zone is computed
    from the vertically interpolated core area (C+I)

Validation issue for LAM : extension zone (2)

- Considering :
  - the new biperiodicization framework does not separate C+I and E,
  - the new biperiodicization algorithm is idempotent over C+I+E,
  - the intermediate spectral fit is bypassed in NFPOS=928,

If NFPOS=928, the biperiodicization is performed twice :

1) After the horizontal interpolations (and before the intermediate spectral fit of the orography only)

2) Again after the vertical interpolations and before the final spectral fit

=> The extension zone is computed

from the vertically interpolated core area (C+I)

... like in NFPOS=2

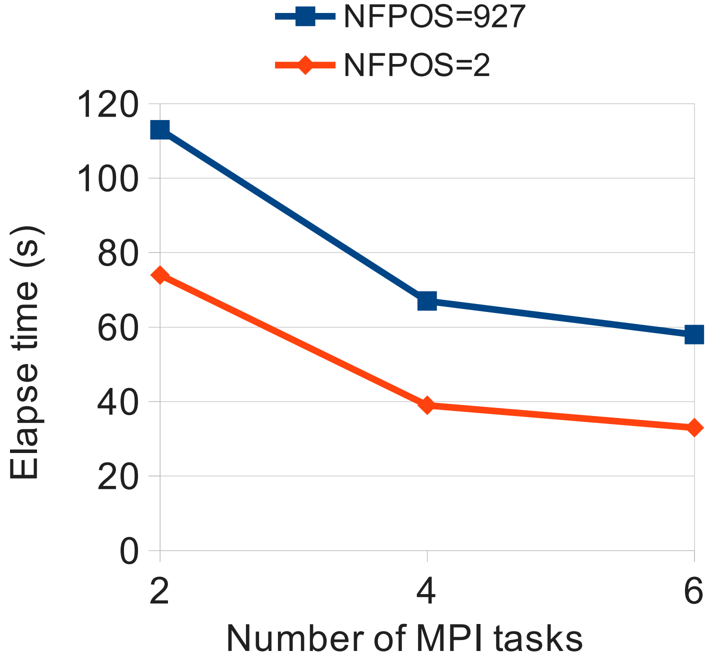
Validation issue for LAM : Map factor

- Spectral smooting of the map factor in NCONF=001
- No possible spectral smooting of the map factor in Fullpos *(since the target spectral geometry was not accessible at the right moment)*
- Fullpos-2 : Smooting of the map factor is possible :
  - Needs the spectral smooting code to be made modular
  - => Map factor truncation needs now to be controlled by the linearity of the grid instead of the model advection scheme
  - **Validation of NFPOS=2 vs NFPOS=928 made by forcing the smooting of the map factor in NFPOS=928**
  - Still no smooting of the map factor in NFPOS=927 to simplify the validation

Performance on vector machine (1)

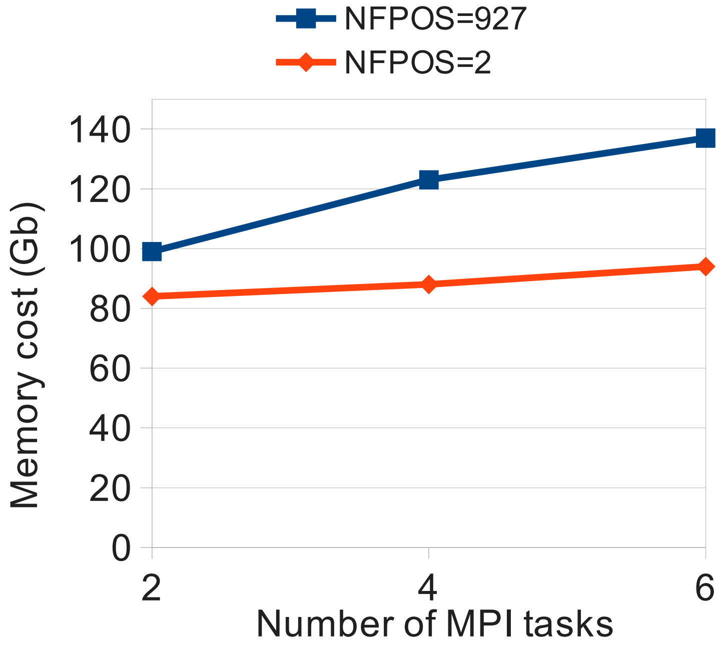
Fullpos-2 performance in cycle 39

T798 => T1198 / NEC SX-9



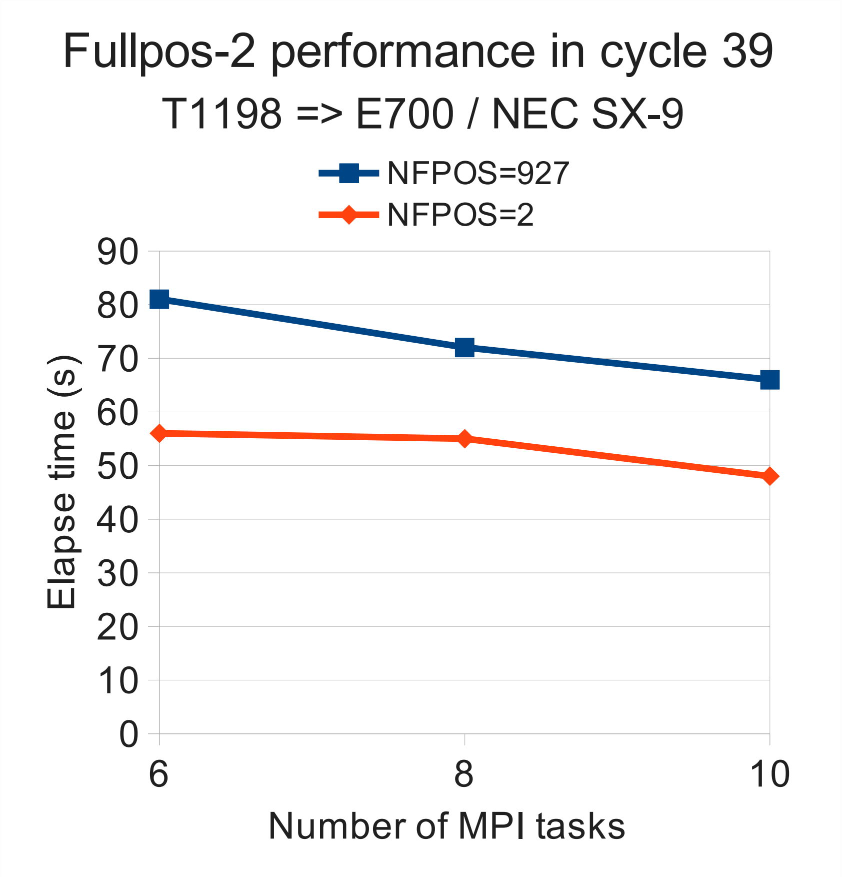
Fullpos-2 memory cost in cycle 39

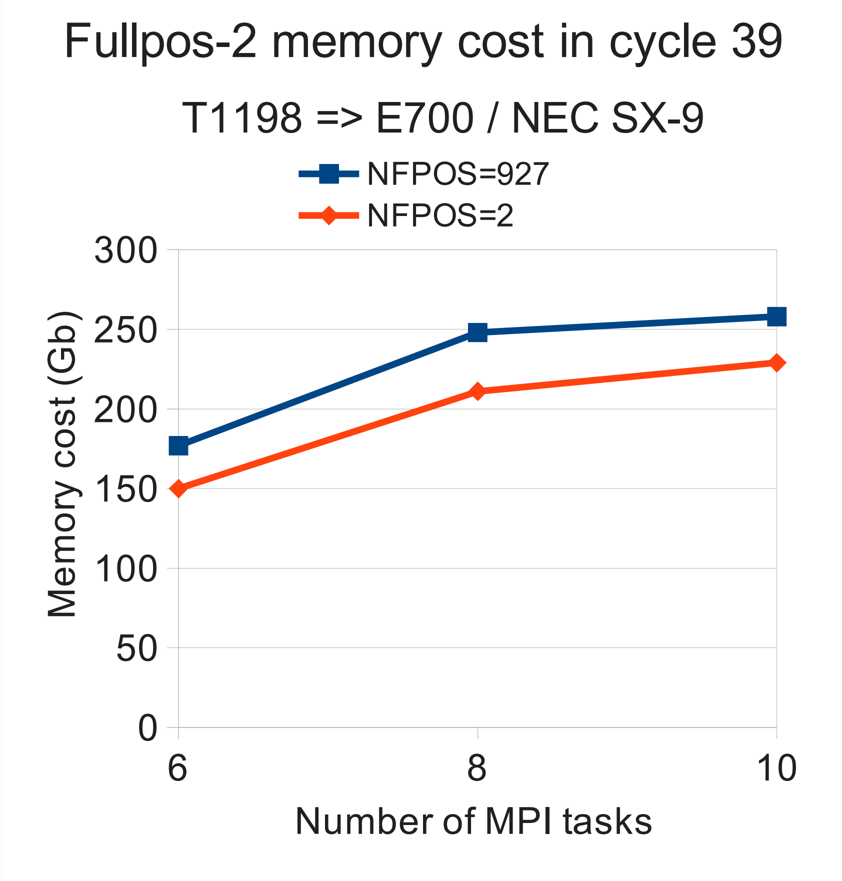
T798 => T1198 / NEC SX-9



- Well vectorized
- ≈ 40 % faster
- Better control of the memory cost

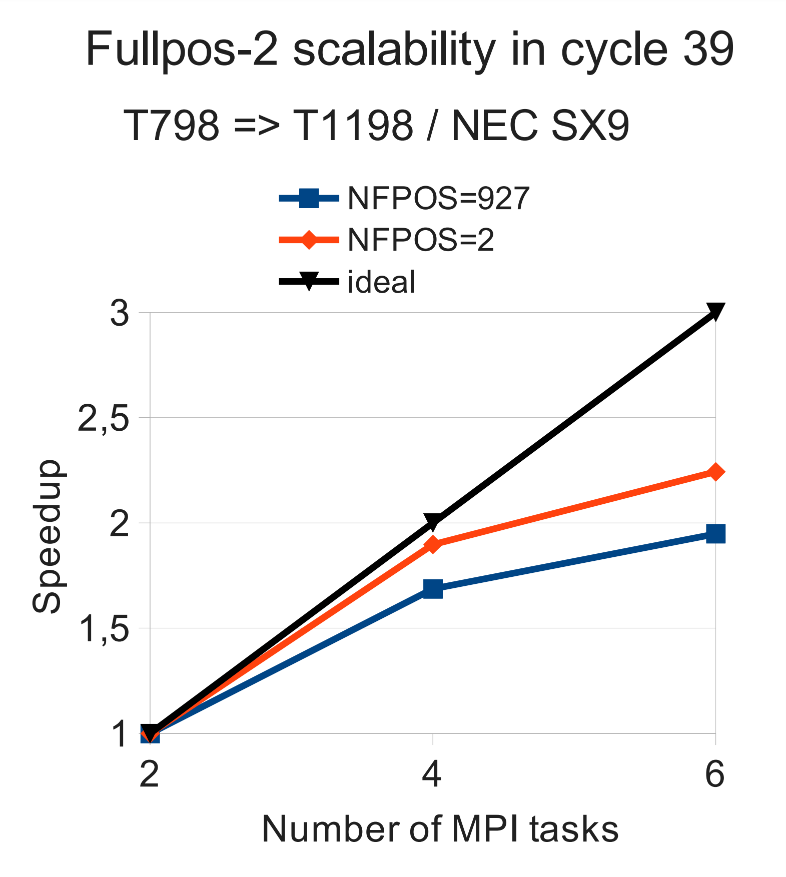
Performance on vector machine (2)

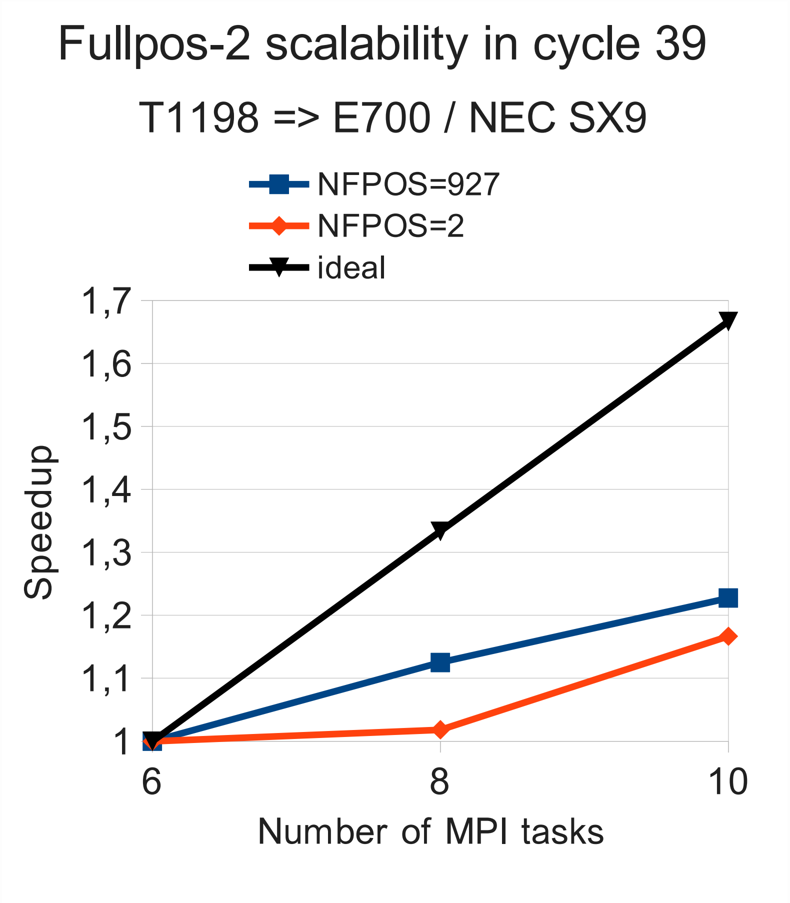




- Well vectorized
- ≈ 25 to 30 % faster
- Still very memory-consuming

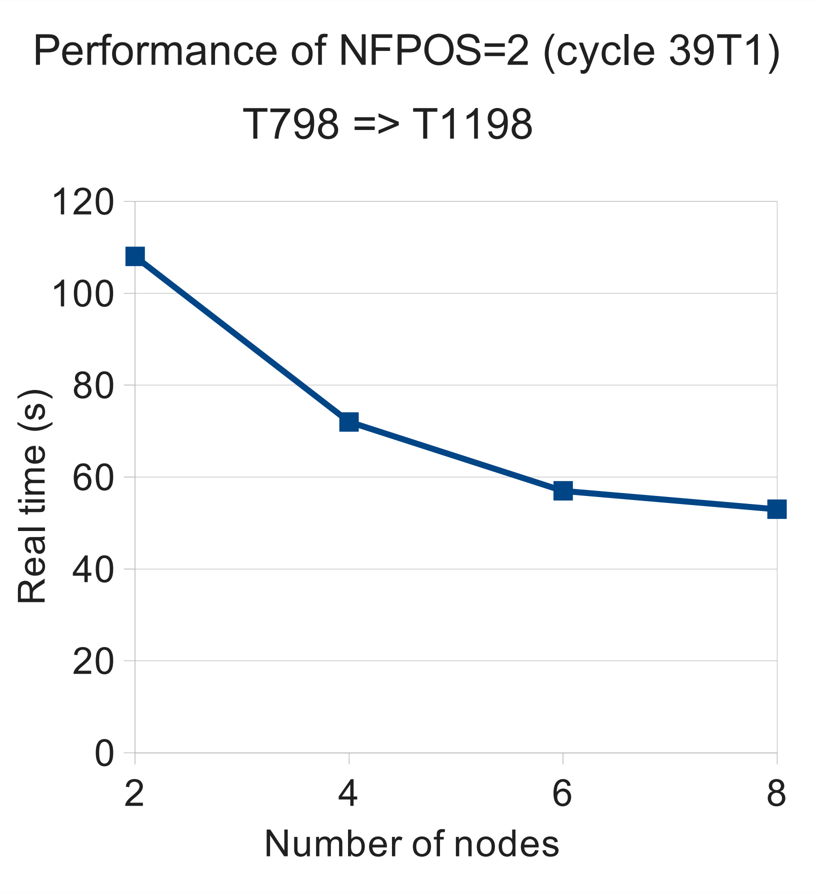
Scalability on vector machine

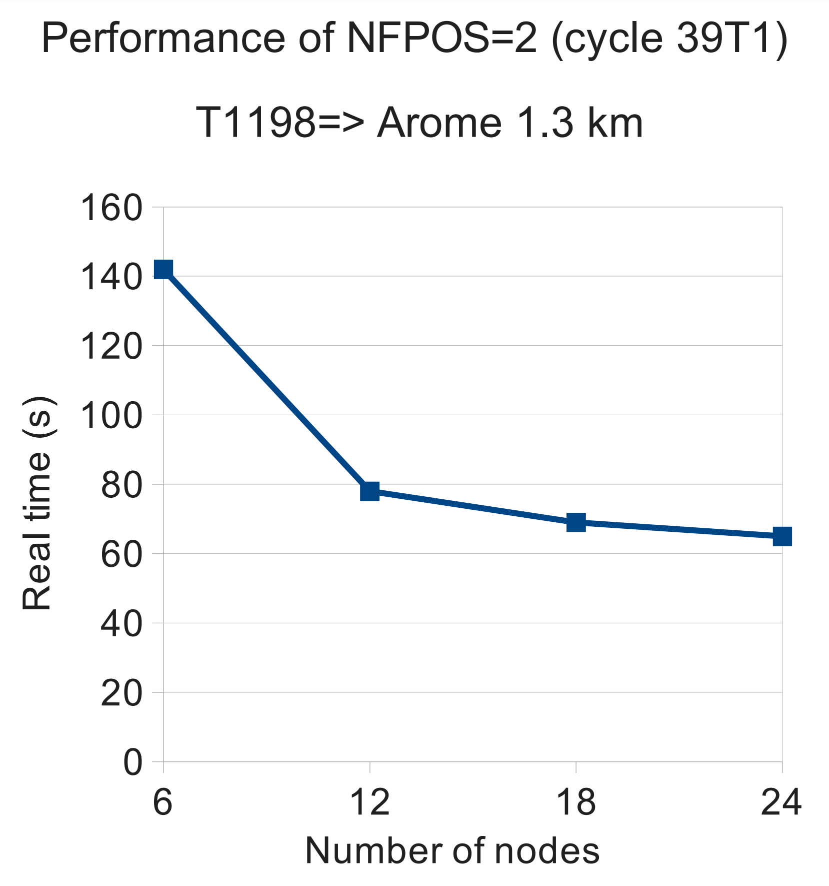




- Correct for global transformation
- Problematic for Global->LAM transformations

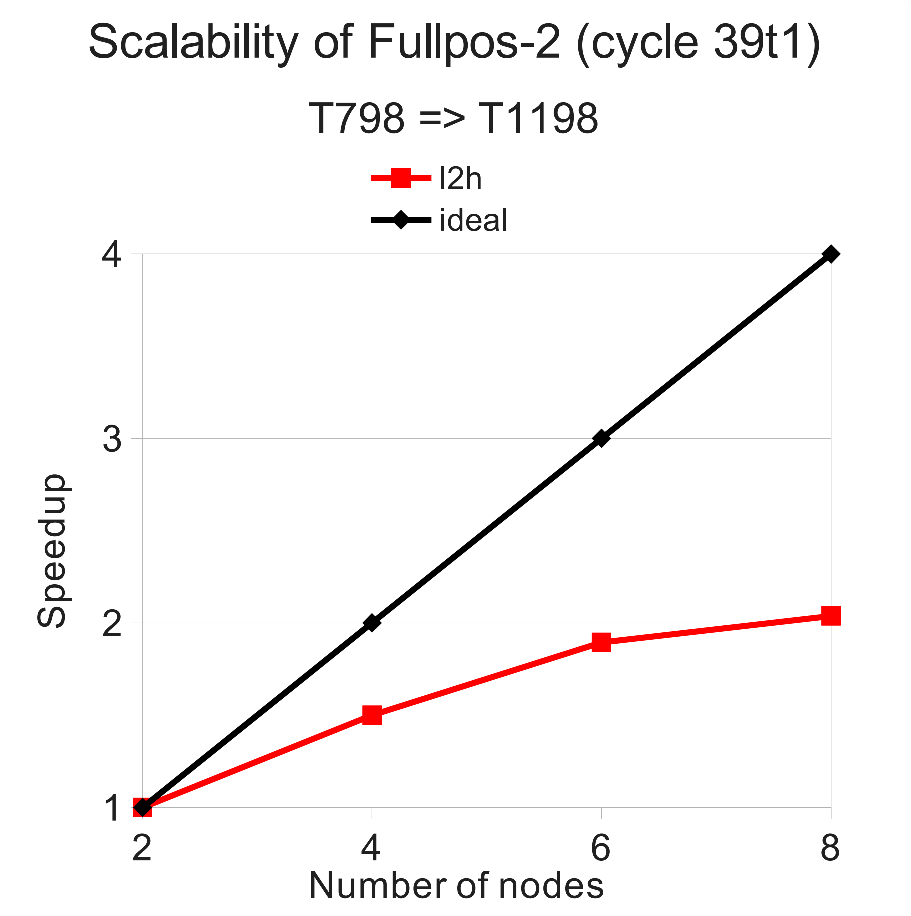
Performance on scalar machine

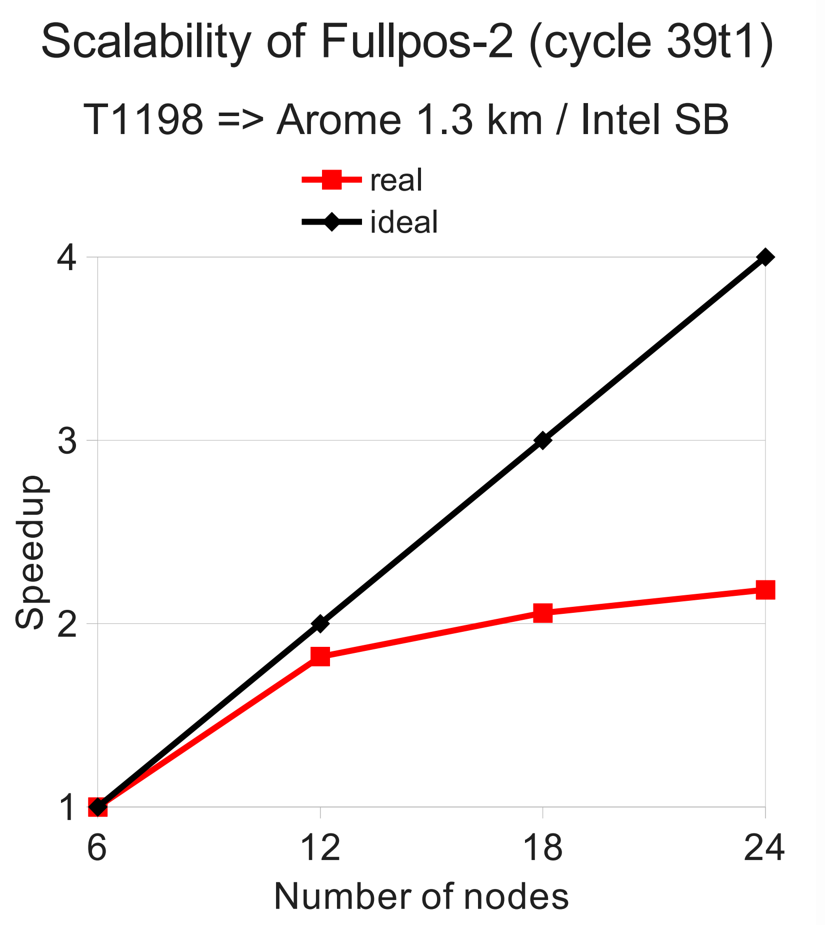




Same order of magnitude than for the vector machine

Scalability on scalar machine

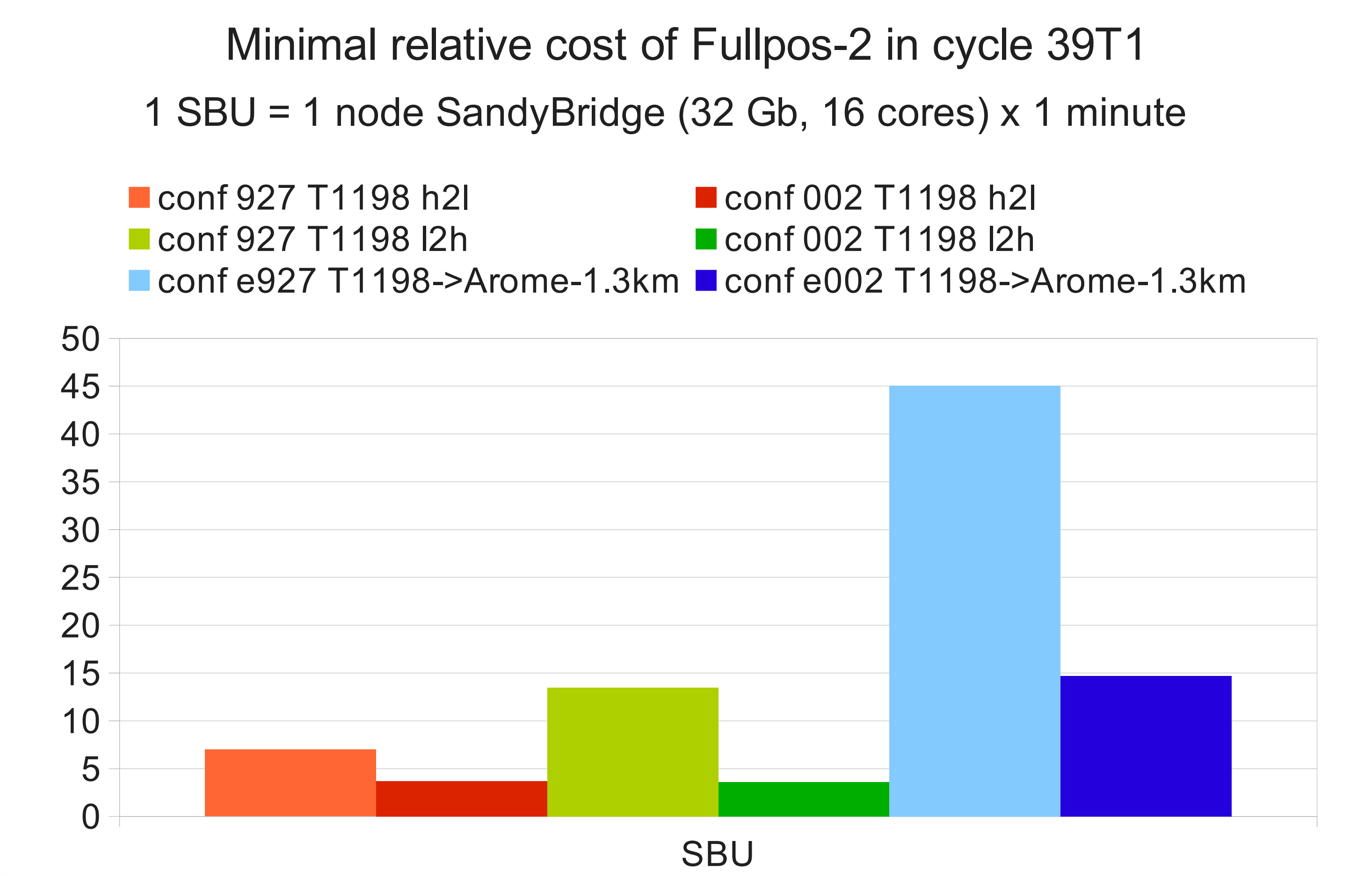




- Similar for Global or LAM
- ... could be better in both case !

METEO FRANCE
Toujours un temps d'avance

Billing units on scalar machine



- NFPOS=2 is ≈ twice cheaper for Global->Global transformations
- NFPOS=2 is even cheaper for Global->LAM transformations

Suspected performance weaknesses

- **Output data handling :**
  - Lack of Open-MP parallelization in data packing
  - Blocking communications in gridpoint data gathering
- **Setup of spectral transforms :**
  - Lack of Open-MP parallelization ?
  - Blocking communications ?
- **Climatological data handling :**
  - Sequential scattering of the fields
  - Blocking communications in data scattering
- **Gridpoint distribution :**
  - LEQ_REGIONS not transparent for LAM
  - Based on the source grid instead of the target grid

The Boyd biperiodicization issue

- **Boyd biperiodicization needs 2 extension zones :**
  1) A geographical over-dimensionned one for the interpolations
  2) A final one which is an extraction & relocation of the former one
- In NFPOS=927,
  - The first E-zone is defined/computed in the first part
  - The second (final) E-zone is defined/computed in the second part
- In NFPOS=2,
  - There is only 1 part => only 1 E-zone
  - **=> A specific additional E-zone has to be defined/computed**
    - With extra points for the interpolations
    - Without extra points for the map factor smooting

Other topics raised by Fullpos-2

- Access to derivatives on terrain-following levels on output grids ?
- Spectral filtering on the target grids rather than the source grid ?

- Simultaneous multiple changes of geometry ?
- On-line changes of geometry can work
- Multi-files sequential change of geometry (« post-processing server »)

- Fullpos-TL ?
- Full externalization of Fullpos

- Polymorphic spectral transforms (the napkin at George & Dragon)
- Simultaneous multi-spectral transforms (OOPS) ?

Conclusion

- Fullpos-2 is a substantial re-write of Fullpos, which takes advantage of the spectral transforms modularity to optimize the model changes of geometry
- The new code design should make the maintenance easier
- Performance on vector or scalar platforms shows that the computing resources are devided by roughly a factor of 2
- There is still room for scalability improvement
- The Boyd biperiodicization is not yet coded in the new framework (but the old framework is still working)
- There are various other topics raised by Fullpos-2, concerning the spectral transforms package or the post-processing itself

Arpege-IFS ski-shop

Pyrenees, January 2013