Optimizations on ARPEGE and AROME

philippe.marguinaud@meteo.fr

- Porting to single precision our forecast models

- Using a post-processing server coupled to our forecast models

Météo-France models

- ARPEGE, global model, T1198L105

- AROME, regional model over France, 1440x1536, 1.3kmL90, NH

Both share the same code, spectral, semi-implicit, semi-Lagrangian

Port to single precision

Follows the work of F. Vana (ECMWF) on IFS in single precision

- Port the I/O
- Make the physics work
- Run our test suite
- Run real forecasts & compute scores

Make the physics work

- Avoid huge (ie > 3E+38) numbers
- Avoid divisions by zero :
  - Either too small numbers (<1E-38)
  - Or zero divided by zero
- Find more accurate formulations

  (eg (X - Y) x (X + Y) vs (X x X - Y x Y))

→ Most of the time, the code is improved and more robust

Validation

- A single test case of our test suite does not work
- Impact of using single precision small; comparable to :
  - A change in compilation options
  - Reformulating physics
- One month run using ARPEGE and AROME (current operational resolutions); no visible impact

Performance

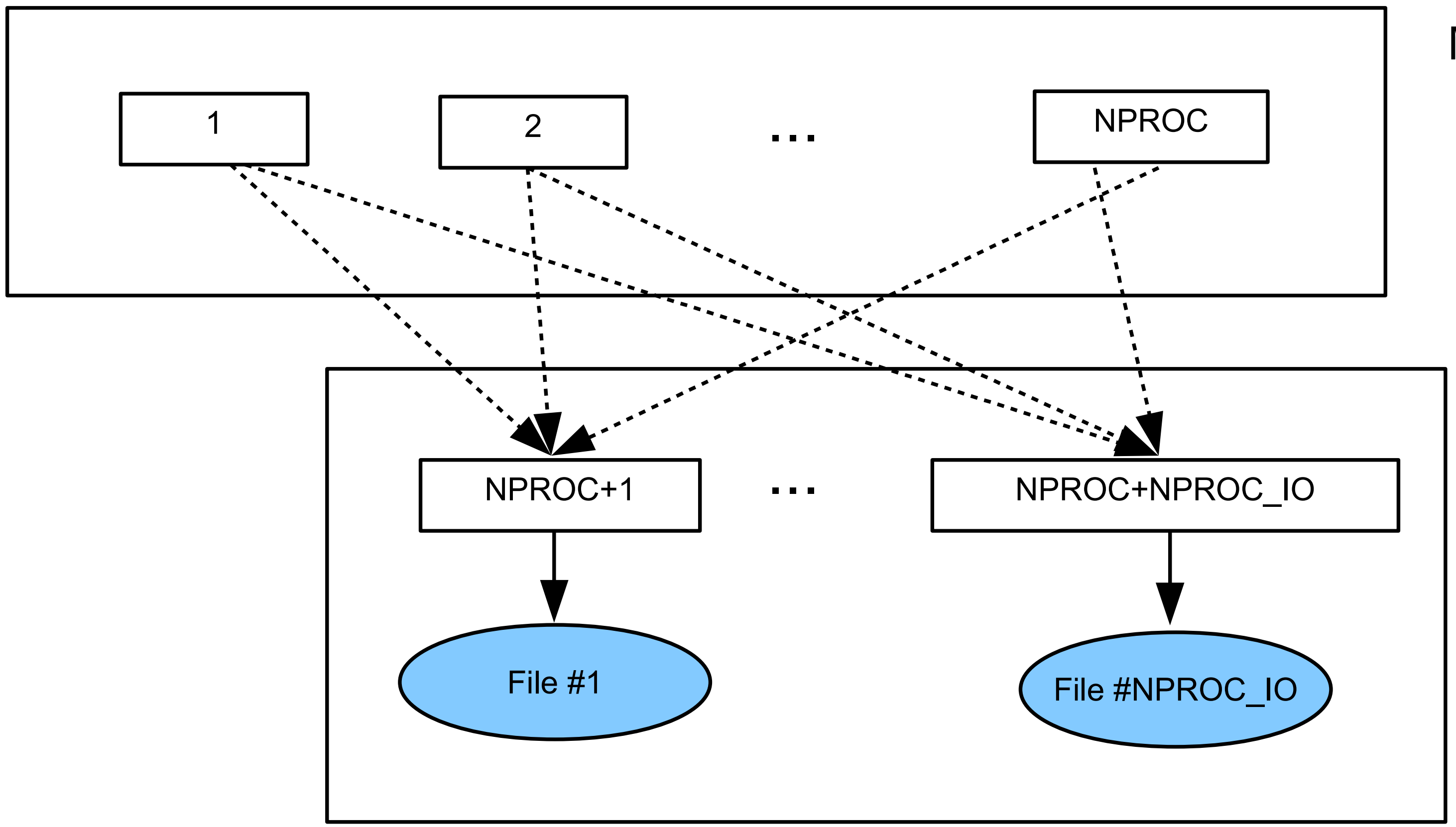
- 40 % reduction in elapsed time (+5% with NPROMA tuning)
- Scalability does not appear to be affected

Using a post-processing server

Current situation :

- ARPEGE and AROME use an IO server : write model state, post-processed fields, read coupling data
- IO server nodes used for their memory (very little processing)
- The post-processing (horizontal + vertical interpolations, derived fields, etc...) is integrated to the model code (“Fullpos”)

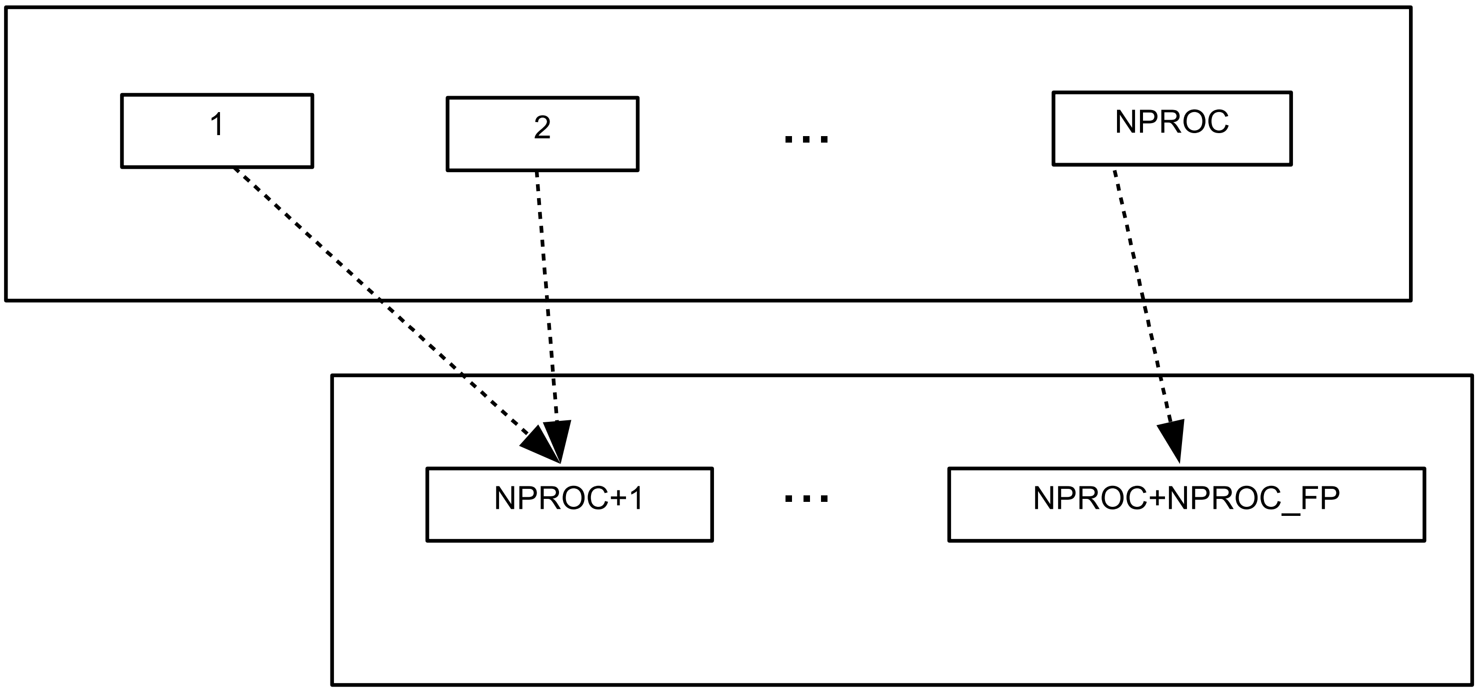
IO server



Model: send model state + post-processed data

IO server : receive model state + post-processed data; write to disk

Post-processing server



Model: send model state

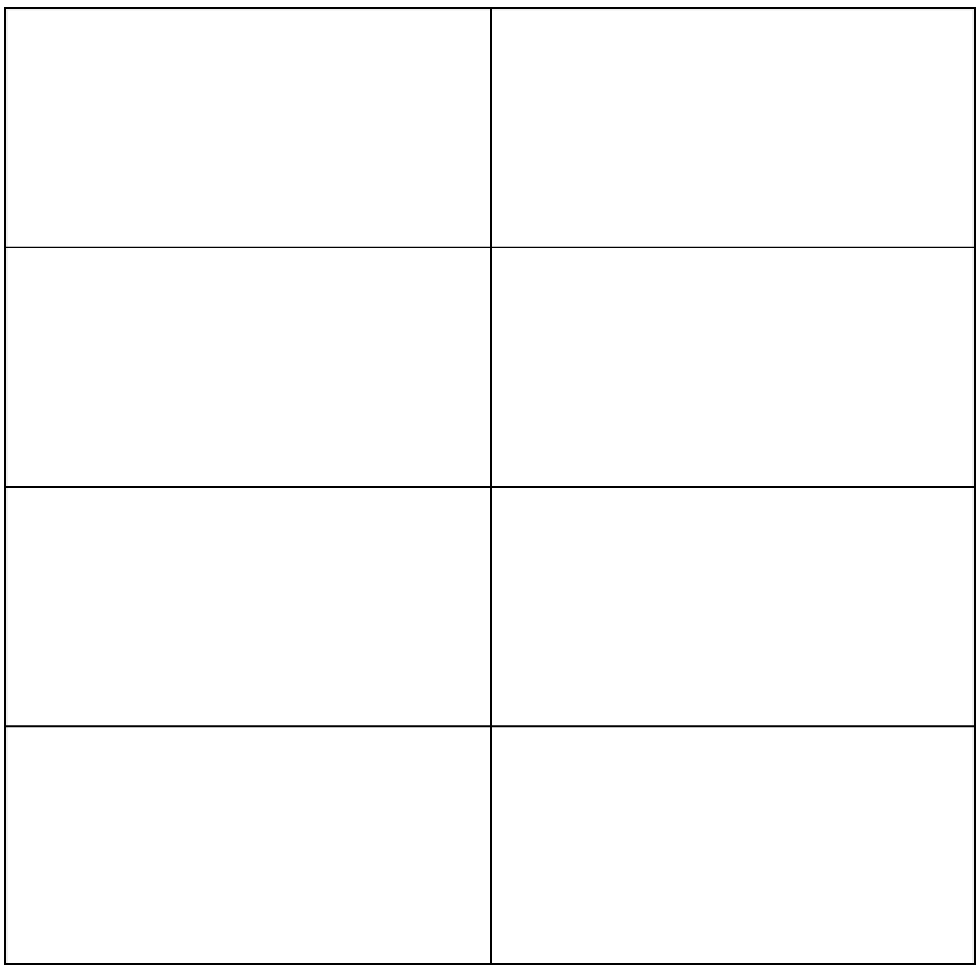
FP server : receive model state, post-process, write to disk

Principles

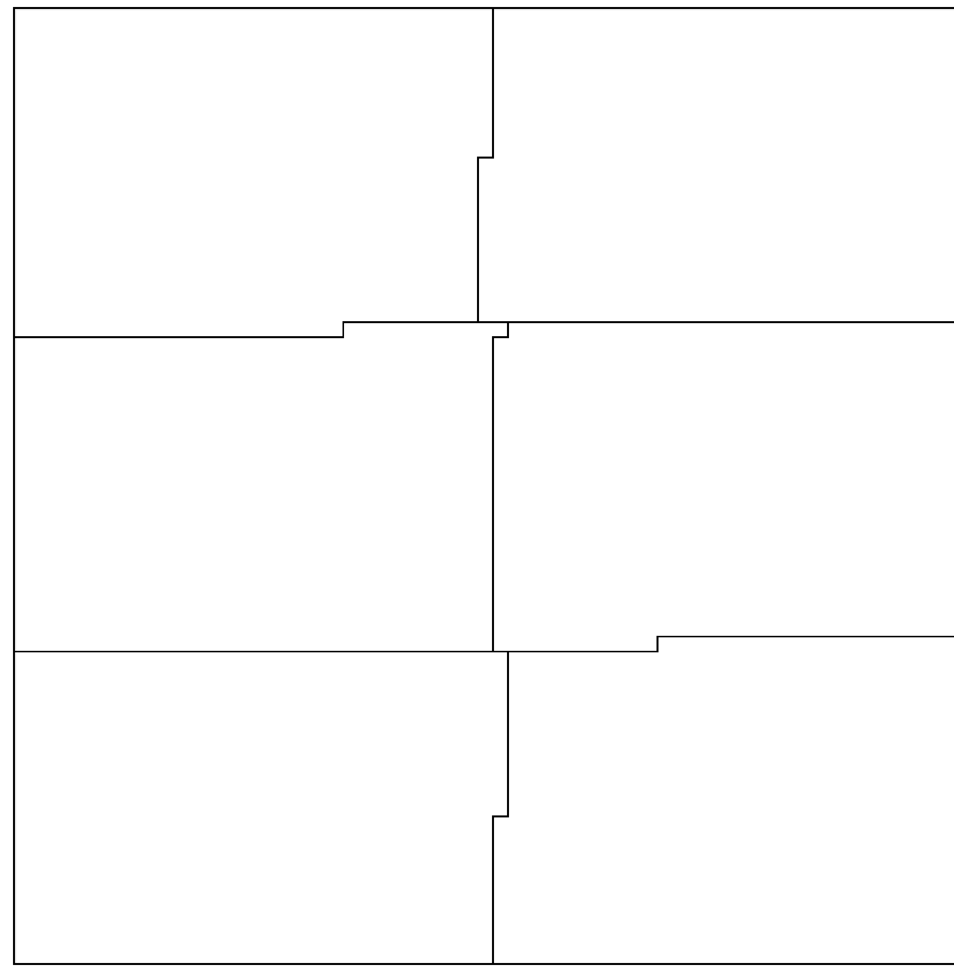
- Introduce a new degree of parallelism
- “Weaker” transposition than for the IO server
- Useful when :
  - Model scales poorly
  - Post-processing becomes expensive
- Direct transposition, but relies only on scatter & gather functions of the distribution (see next slides)
- Asynchronous sends
- Send data + meta-data

Computing redistribution parameters

AROME with 8 tasks

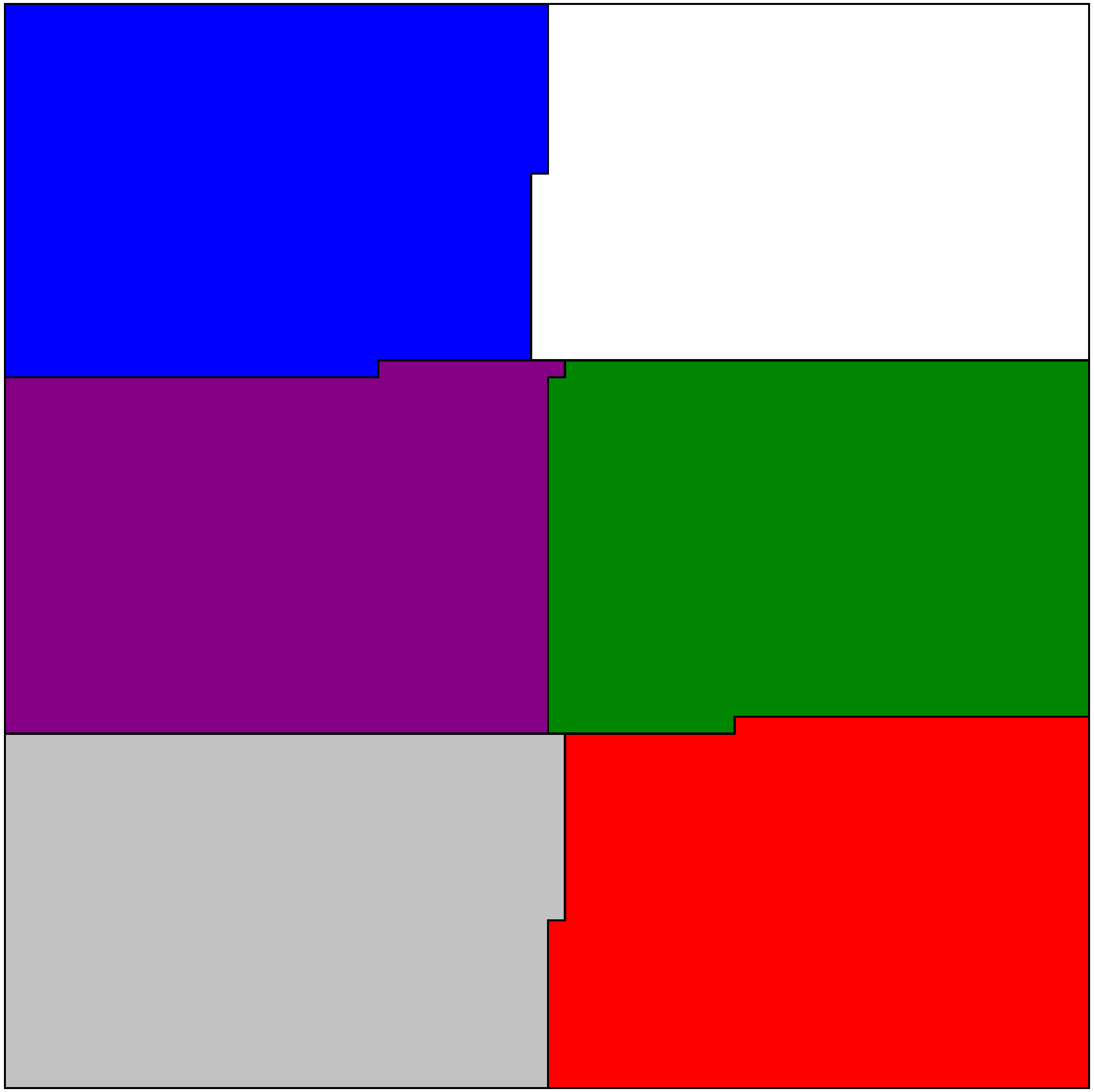


AROME post-processing
server, with 6 tasks



AROME server with 6 tasks

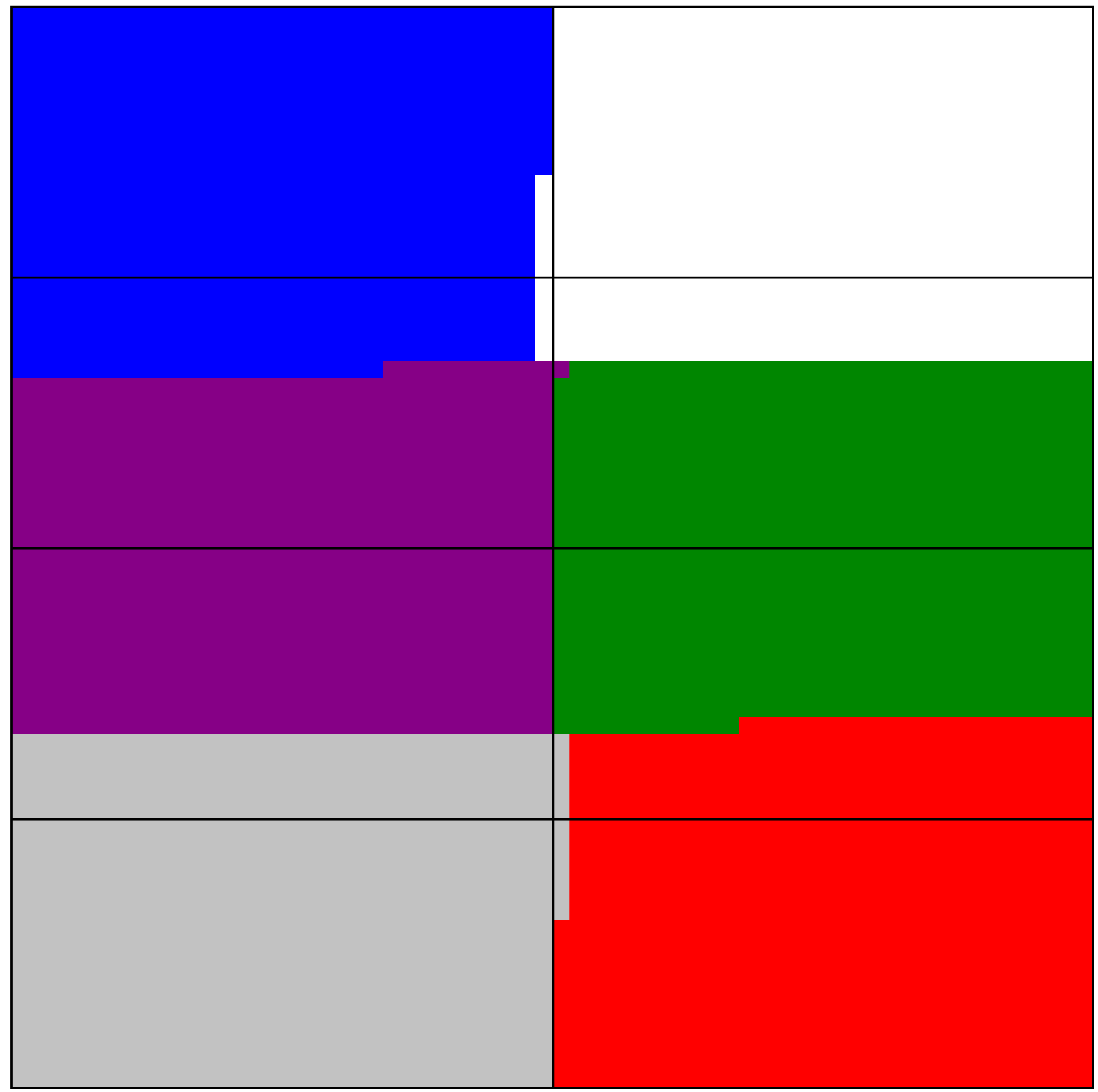
- Create a distributed global field, whose value is MPI rank
- Gather the field
- Send the global field to the model (MPI #1)



AROME model, with 8 tasks

- The global field is received by the MPI #1 of the model
- This field is then scattered on all model tasks

→ Each model task knows how many points it will receive from each post-processing server task



Example configuration

All test cases on Broadwell nodes, operational post-processing x 2

- AROME, 171 nodes + 12 nodes (FP server), 2068s
- AROME, 171 nodes + 4 nodes (IO server), 2192s
- AROME, 180 nodes + 4 nodes (IO server), 2074s

→ The post-processing server brings some little improvement

Conclusion

- Single precision forecast & post-processing server available in 43t2
- Post-processing server still experimental

- Single precision port = 3 man-months (long and boring)
- Post-processing server = 1 man-month (short and exciting)