Status of optimisation work on the NEC SX8R at Météo-France

R. El Khatib
with the help of many :
Damien Lambert, NEC, Dominique Puech, Sami Saarinen (ECMWF), Yann Seity, Eric Sevault, Filip Vana (CHMI)

Aladin workshop / Hirlam all staff meeting
Bruxelles 07-10/04/2008

METEO FRANCE
Toujours un temps d'avance

Plan

- **INTRODUCTION**
  - Purposes of this work
  - Overview of a node SX8R
  - Tools at disposal for optimisation
  - -ftrace : example
  - Methodology
- **OPTIMISATION WORK**
  - Main features
    - I/Os
    - Vectorisation (« NPROMA »)
    - Message passing
    - Multitasking (Open-MP)
  - Specific issues
    - ODB & observations preprocessing (« Bator »)
    - Screening
    - Minimisations
    - Arpege/Aladin - Arome
    - Fullpos
- **CONCLUSIONS & OUTLOOK**

INTRODUCTION : Purposes of this work

- **After porting :**
  time to have a look on the profiles with this new machine
  (even if still vector ...)
- **Make sure the code is optimised for higher truncations :**
  - especially for the now operational truncation T538
  - And then for the next truncation increasement (T799 on SX9)
- **Optimise AROME**
  - investigations needed
- **New technologies at disposal for MF (on site) :**
  - Local vs global file system : how to get the best profit from them ?
  - Open-MP : should we use it to improve the performances ?

INTRODUCTION : Overview of a node SX8R

- **Processors :**
  - 8 processors/node (VPP5000 : 1 processor/node)
  - Vector processors with 256 vector registers
- **Memory :**
  - 128 Gb/node (compared to VPP5000 : 4 or 8 Gb/node)
  - 4096 memory banks (VPP5000 : 512)
- **I/Os :**
  - Local disk : fast, but private to a node
  - Global file system (GFS) : slower but visible by all nodes
  - Expected transfer rate (GFS) : > 100 Mb/s
- **CPU performance :**

  Peak performance : 35 Gflops/processor. In practice we can expect ≈ 20% of the peak performance from a vector machine
  => 7 Gflops/processor

INTRODUCTION :
Tools at disposal for optimisation

- Compiler option -ftrace providing detailed informations per subroutines, such as :
  - frequency of calls, real time, Mflpos, vector length, cache-misses, banks conflicts ...
  - Elapse time, communications time, messages length
- Environment variables providing :
  - general informations per processors or for all processors, such as timings, Mflpos, mean vector length, ...
  - Detailed informations for each accessed file

- + Environment variables able to set miscellaneous parameters, especially buffer sizes for reading files

INTRODUCTION : -ftrace : example (1)

5.4 Prévision à 6 heures pour l'assimilation

5.4.1 "Top 10" en CPU

PROG.UNIT	FREQUENCY	EXCLUSIVE TIME[sec](%)	AVER.TIME [msec]	MOPS	MFLOPS	V.OP RATIO	AVER. V.LEN	VECTOR TIME	I-CACHE MISS	O-CACHE MISS	BANK CONF
rrtm_rtrn1a_140gp	2794260	253.246(13.0)	0.091	7609.2	3140.0	98.42	107.5	224.727	8.9242	4.8657	0.2891
rrtm_gasabs1a_140gp	2794260	96.694(5.0)	0.035	4543.9	1952.3	96.57	253.9	44.591	16.9116	18.1661	1.6708
rrtm_taumol3	2794260	75.777(3.9)	0.027	3577.5	1148.8	97.27	33.0	69.443	4.3025	1.3062	0.9463
rrtm_taumol9	2794260	74.813(3.8)	0.027	2713.4	688.8	94.64	36.2	55.589	6.9807	5.5858	0.7057
sgemmx	75460	73.815(3.8)	0.978	41458.8	30239.9	99.29	239.1	71.138	0.2866	1.9245	0.3522
laitqmh	23040	63.045(3.2)	2.736	23125.5	5972.0	99.97	232.7	63.028	0.0065	0.0054	2.1738
advprc	4608	59.901(3.1)	12.999	22588.2	8783.1	99.46	235.1	59.743	0.0815	0.0366	1.8773
rrtm_taumol2	2794260	58.468(3.0)	0.021	2802.0	719.4	96.09	33.6	51.693	4.4016	1.3813	0.6617
rrtm_taumol7	2794260	56.993(2.9)	0.020	3286.5	885.8	97.06	35.5	53.960	1.7289	0.3985	0.6733
rrtm_ecrt_140gp	2794260	56.751(2.9)	0.020	1009.6	263.9	85.51	40.6	15.326	12.3119	21.5948	0.1617
total	64184106	1950.315(100.0)	0.030	10693.5	4684.2	98.77	141.0	1624.563	101.5138	89.8596	37.1402

INTRODUCTION : -ftrace : example (2)

5.4.2 "Top 10" en ELAPSE

PROG.UNIT	ELAPSE [sec]	COMM.TIME [sec]	COMM.TIME / ELAPSE	IDLE TIME [sec]	IDLE TIME / ELAPSE	AVER.LEN [byte]	COUNT	TOTAL LEN [byte]
rrtm_rtrn1a_140gp								
	44.019	0.000		0.000		0.0	0	0.0
mpl_send_real8	32.575	32.536		31.224		696.8K	54098	35.9G
lfiedo	18.197	0.000		0.000		0.0	0	0.0
rrtm_gasabs1a_140gp								
	17.706	0.000		0.000		0.0	0	0.0
rrtm_taumol3	12.777	0.000		0.000		0.0	0	0.0
rrtm_taumol9	12.627	0.000		0.000		0.0	0	0.0
sgemmx	12.394	0.000		0.000		0.0	0	0.0
laitqmh	10.736	0.000		0.000		0.0	0	0.0
mpl_alltoallv_real8								
	10.694	10.693		0.122		356.6M	288	100.3G
advprc	10.036	0.000		0.000		0.0	0	0.0

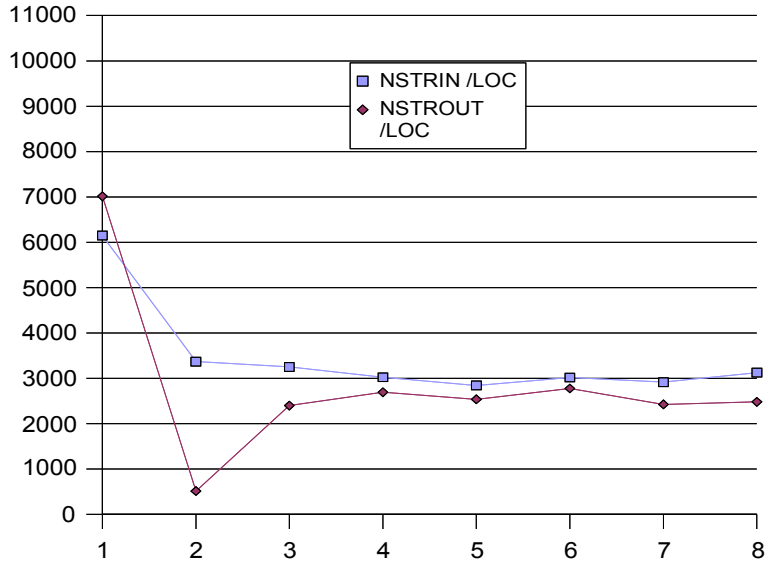
INTRODUCTION : Methodology

- Scalability comparisons for various software configurations :
  - Over 1, 2, 4, 8, 16 processors
  - MPI vs. OPEN-MP whenever possible
  - North-South vs. Square MPI distribution
  - Local disk vs. Global file system whenever possible
- Profilings from the best performing configuration,
  usually over 16 processors (ie : 2 full nodes)
- Tunings from the best performing configuration
  (mainly : namelists)

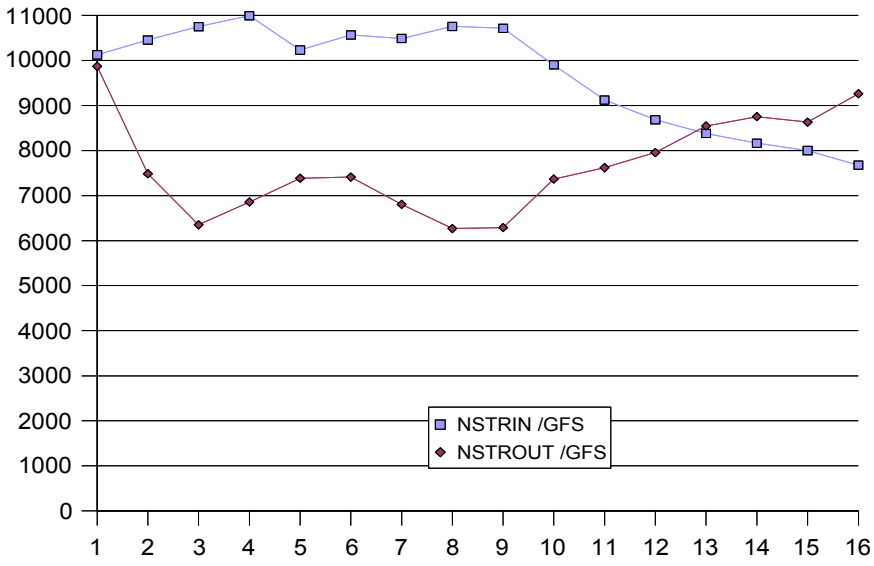
Unfortunately : not a full node used for tests over 1, 2, 4 processors
... but jobs running during summer !

Main features : I/Os distribution

Real time I/Os on local disk (millisec.)



Real time I/Os on GFS (millisec.)

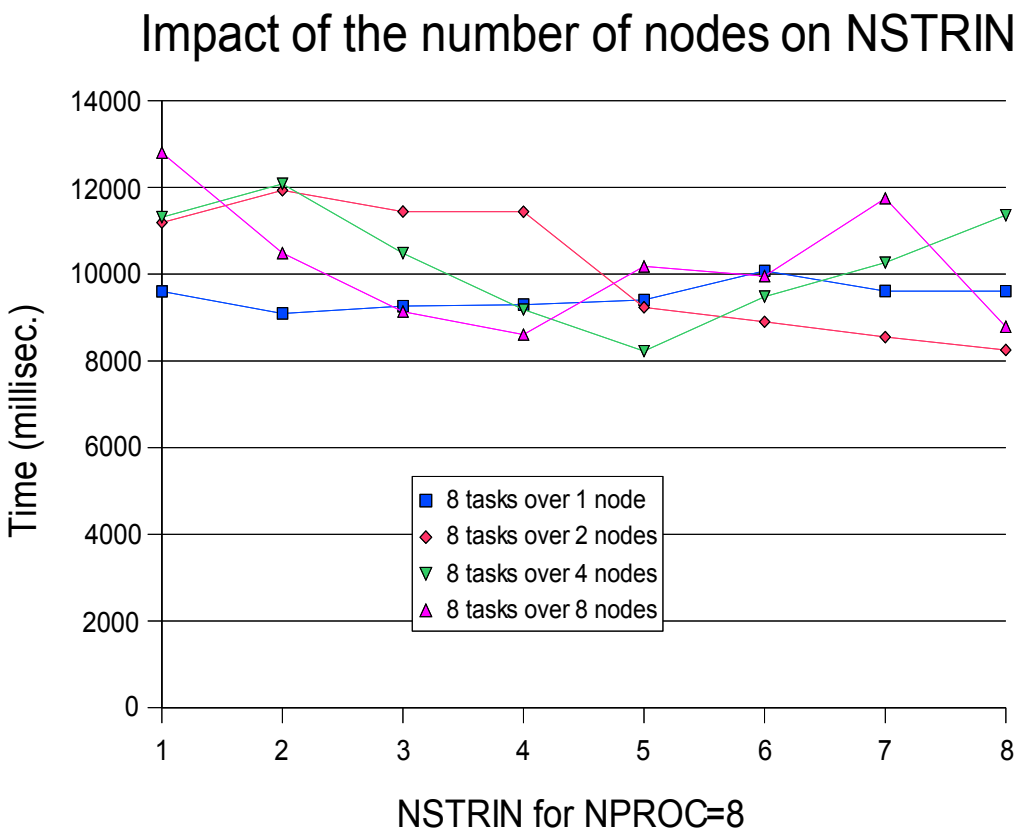


NSTRIN : number of processors for reading/decoding

NSTROUT : number of processors for encoding

- Local disks usage seems always better than global file system
- It is useless/detrimental to distribute encoding work on much processors
- It seems beneficial to distribute decoding work on much processors : Influence of multi-nodes ??

Influence of nodes on I/Os



- Better spread over nodes
- But not too much :
  2 nodes or 4/6 MPI tasks for I/Os
- => I/Os distribution should be spread over nodes, not over MPI tasks

- **BEWARE OF CONCURRENT DISK ACCESSES !**

I/Os : 2 examples of disk accesses issues

- Fullpos-Arpege (stretched):

  For a single file to post-process :

  short job but large matrixes files to read (≈ 133 Mb/s)

  => scalability mostly limited by the I/O part

- SURFEX (in the framework of AROME) :

  An input file (≈ 320 Mb) was treated by each processor :

  - 16 tasks => ≈ 480 s. (0.7 Kb/s) !
  - 32 tasks => ≈ 600 s. (0.5 Kb/s) !!
  - 64 tasks => ≈ 800 s. (0.4 Kb/s) !!!

  => Once read by a unique processor + MPI distribution (*) :

  - 19 s. (17 Mb/s) + ≈ 40 s. communications = 60 s.

(*) unfortunately still to be debugged ...

I/Os : files formats

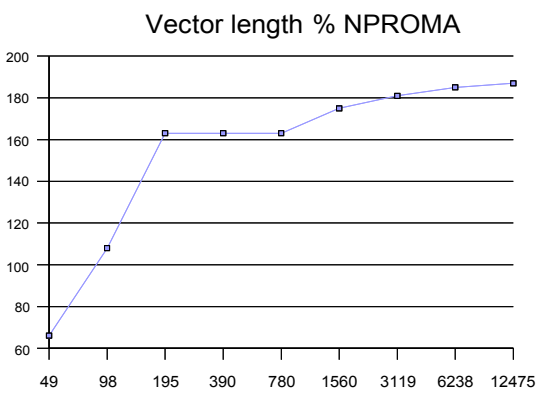
- *Better read binary files than ASCII files :*

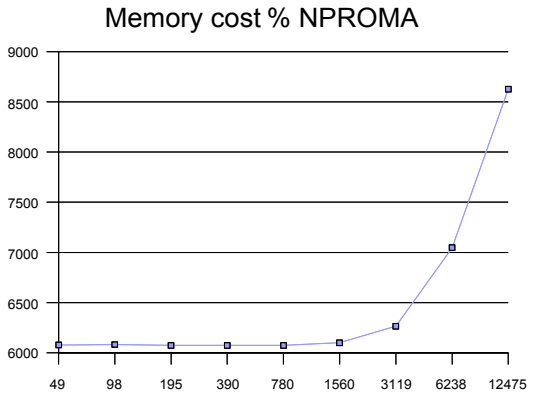
  see the case of RTTOV coefficients file `rtcoef_eos_2_airs.dat` :
  - ASCII version :
    - ≈ 60 Mb read in more than 18 s. (3 Mb/s)
    - Occurrence of 'BACKSPACE' breaking the use of a memory buffer
  - BINARY version :
    - ≈ 30 Mb read in ≈ 2 s. (14 Mb/s.)
    - Now acknowledged by ECMWF : faster on IBM as well.

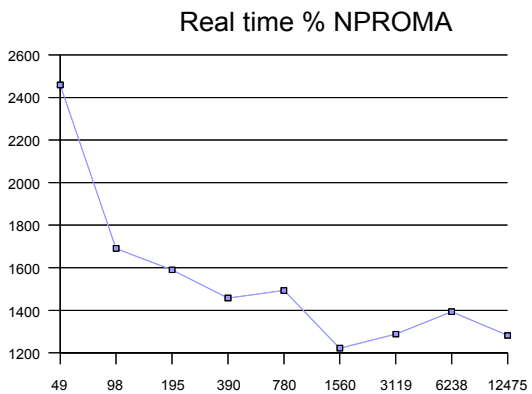
- *Is the « FA/LFI » file format (ie : Fortran indexed sequential files) performant enough ?*
  - To be investigated. Apparently not so bad.

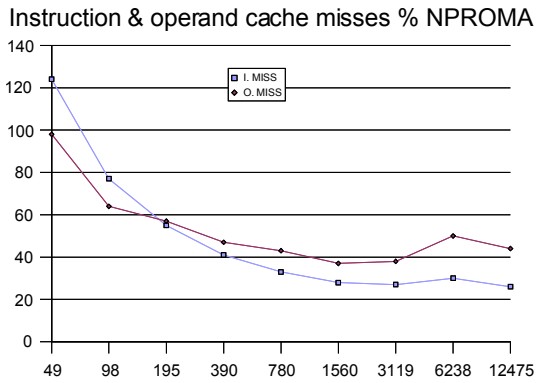
  Actual transfer rate for a historical file : less than 120 Mb/s

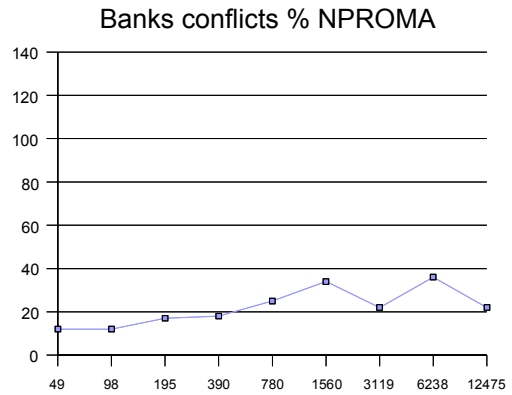
Main features : vectorisation (NPROMA)

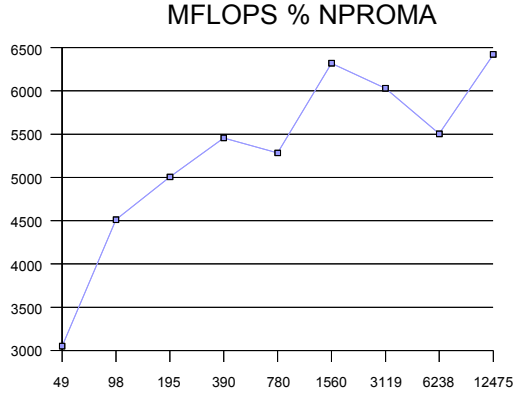










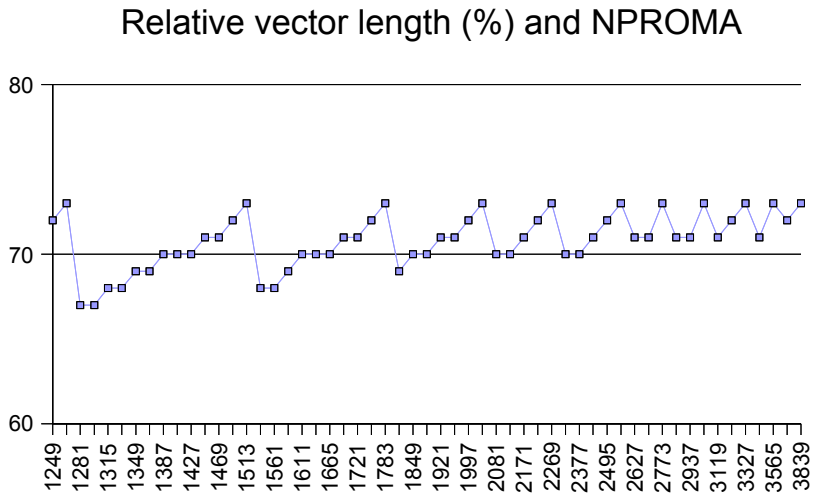


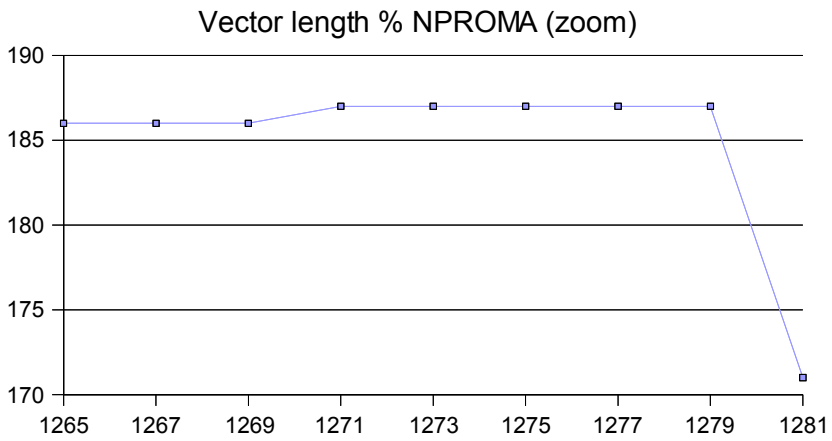
Forecast H24 T538 over 4 processors

- Best values : between 1500 and 3000
- Chaotic performances for such values or bigger values
  => further study

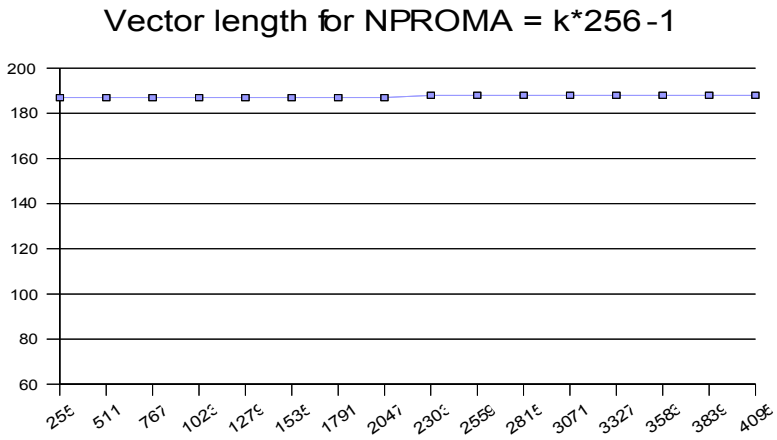
METEO FRANCE
Toujours un temps d'avance

NPROMA : vector length aspect



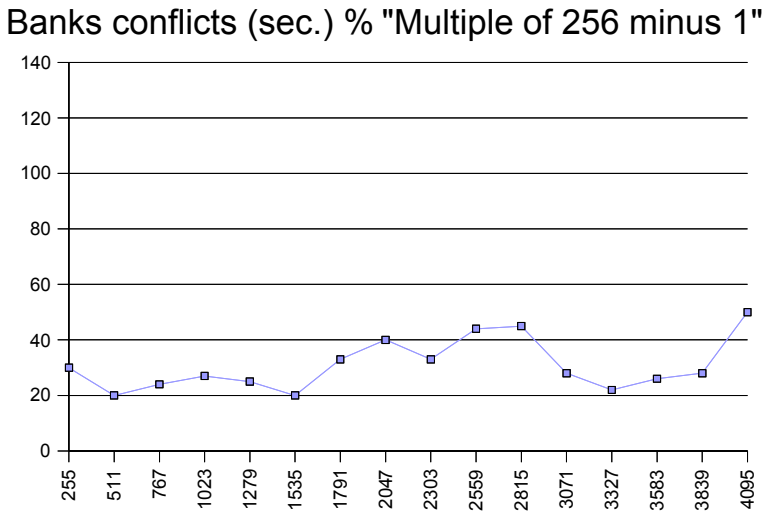


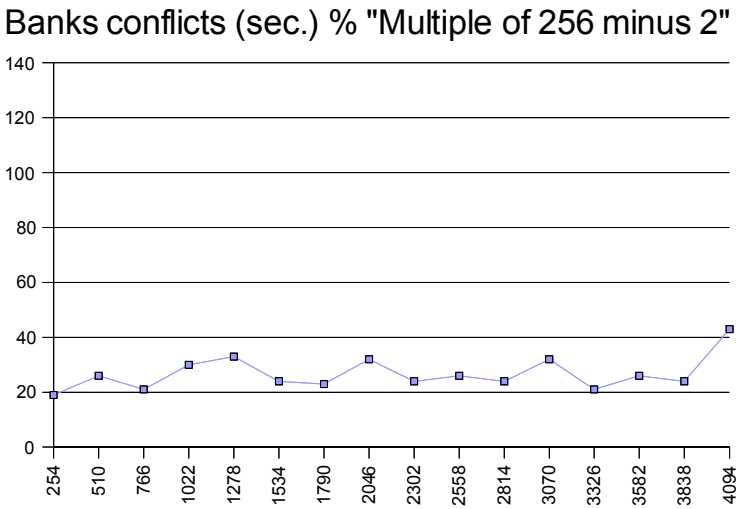
- « Kings » show that there must exist optimal values
- *Rmq* : There are 256 vector registers on a vector unit of a SX8R processor
- => Best NPROMA should be less than a multiple of 256



NPROMA : Banks conflicts aspect

- Inspired by an old recommandation by Fujitsu :
  « a power of 2 minus 2 », test on the multiples of 256 minus 2 :

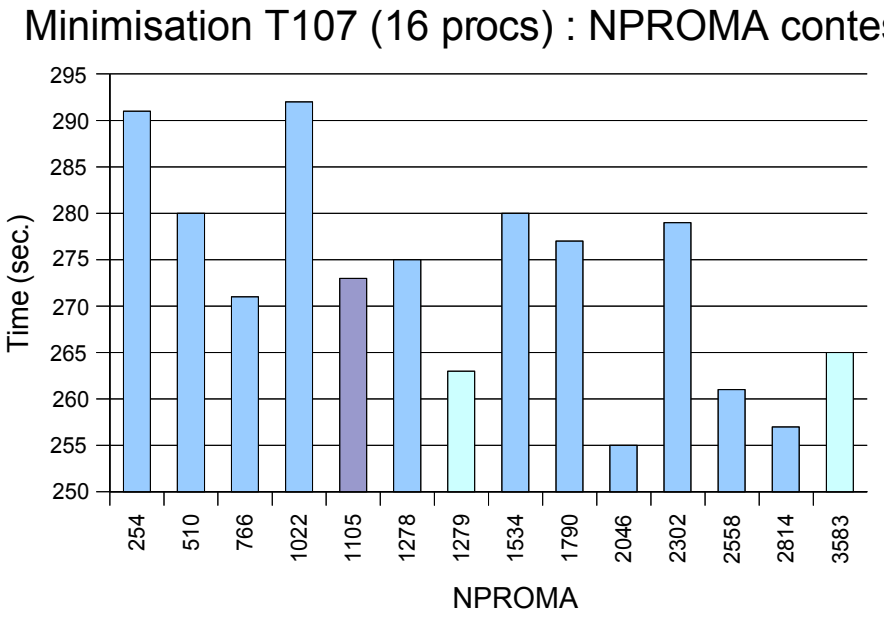




- => Looks slightly better ! Justification ??

NPROMA final recommendations & example

- A multiple of 256 minus 2
- Between 2000 and 3000
- Advice :
  - Either « NPROMA=-2046 » *(REK after smoothing graphics)*
  - Or « NPROMA=-2814 » *(ES after global 4DVar on SX8R « SUMO »)*
- It can be worth (regularly re-)trying a set of values between 2000 and 3000 :

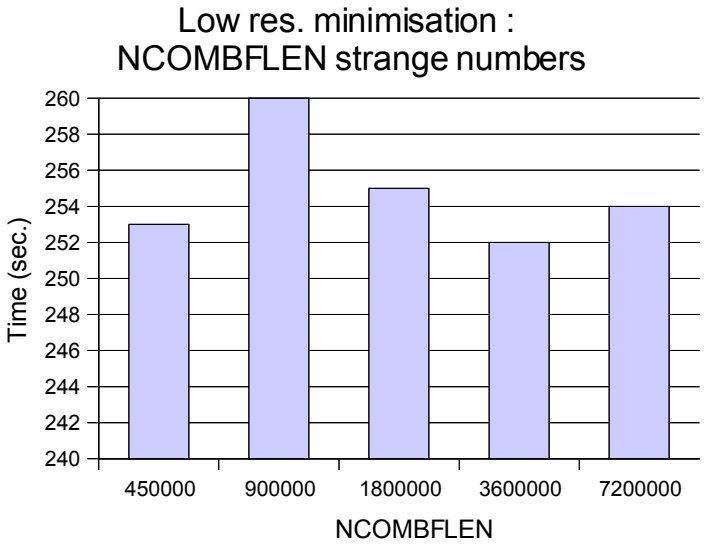


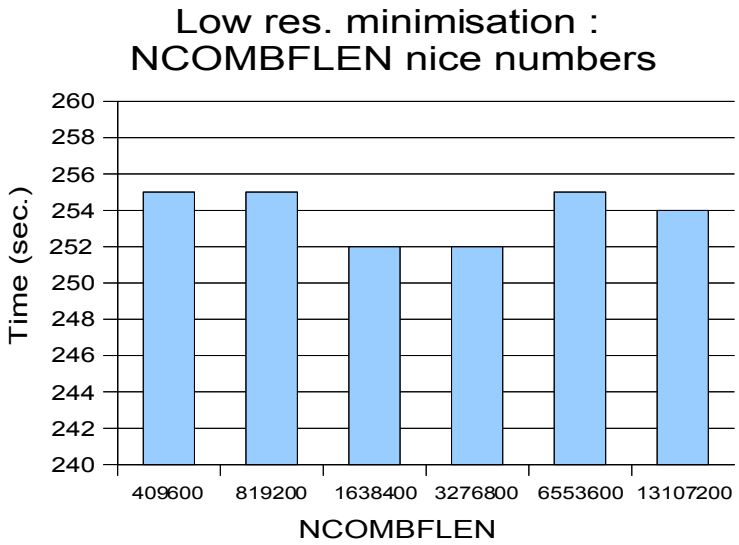
- Performance +/- 1% to 5 %
- Here the best values are exactly : 2046 and 2814
- Prime numbers : not the best
- Best NPROMA bigger than NGPTOT (1105) !?

Main features : Message passing (1/3)

- *For all models :*
  - Switch to LIMP_NOOLAP & LSLONDEM
    (LIMP_NOOLAP was not supported on VPP5000)
  - Communication buffer length tuned to a more « binary » value (3276800 instead of 1800000)

=> saves ≈ 0.5%





Message passing (2/3)

- *For global model only* :
  - **NORTH-SOUTH distribution** always slightly faster than, or equivalent to SQUARE distribution :
    - tested on forecast with 4, 8, 16 & 32 processors.
    - Tested on screening & minimisation with 4, 8 & 16 processors.
- *For LAM only* :
  - **SQUARE distribution** is faster than NORTH-SOUTH distribution (by ≈ 7 % according to a test with AROME) because coupling is then balanced betwen Western and Eastern processsors
  - For a better load balance, **Y-axis distribution** should consider :
    - (C+I) if physics is more cpu-consumming than dynamics
    - But (C+I+E) if dynamics (predictor-corrector) is more cpu-consumming than physics

      => saves ≈ 6 % according to test with AROME

Message passing (3/3)

The *GSTATS* mystery :

- More MPI barriers makes AROME *faster !?*
- *Explanation : ... not very clear. Is it because « sends » are « non-blocking standard » and « receives » are « blocking standard » ? Rmq : Barriers are already coded in mpobseq/mpobseqad.*
- => in cycle 33, two new namelists variables :
  - **LSYNC_TRANS** : to synchronise MPI tasks in spectral transforms *before* communications (MPL_ALLTOALLV)
  - **LSYNC_SLCOM** : to synchronise MPI tasks *before* semi-lagrangian communications

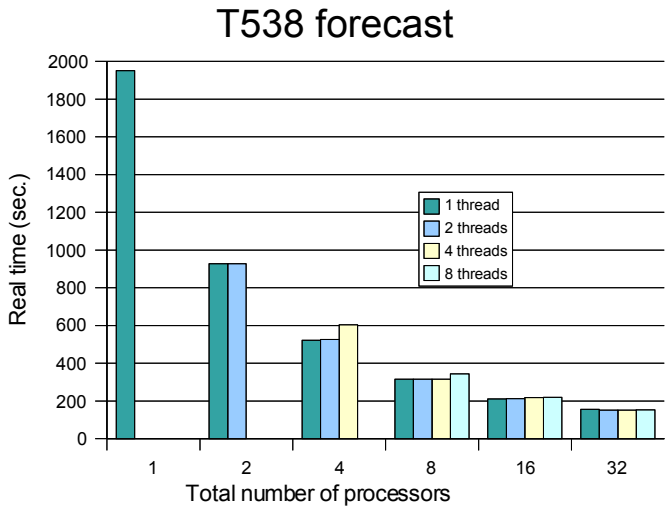
=> Saves ≈ 10 % cpu in AROME

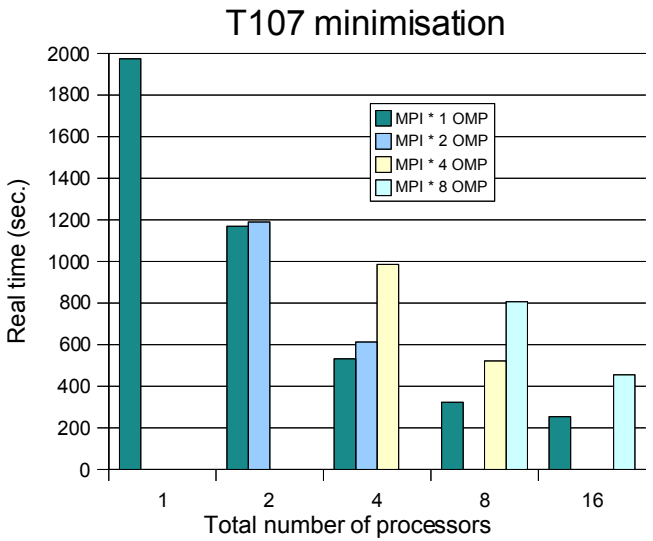
Main features : Multitasking (Open-MP)

- Works on ARPEGE conf 001 and Fullpos, but slightly **less efficient** than MPI

- **Partly working** on the minimisations :
  - Problems of interpretation of the norm OPEN-MP
  - Remaining bugs in compiler ?
    - Workarounds : a few OMP directives modified or switched off
    - Still not working when mixing MPI and OMP if more than 2 MPI tasks
  - SL adjoint : a synchronisation lock on threads can do worse than a single well-vectorising thread

  => still **less efficient** than pure MPI distribution





Multitasking (Open-MP)

- **Not yet working** on Screening, even partly (crash)
- **Not yet working** on ALADIN (a problem probably located inside spectral transforms)
- AROME **not ready** for Open-MP : needs SURFEX and Meso-NH physics to support Open-MP

Open-MP : a disappointing technique on NEC ?

Specific issues : ODB & observations preprocessing (« BATOR »)

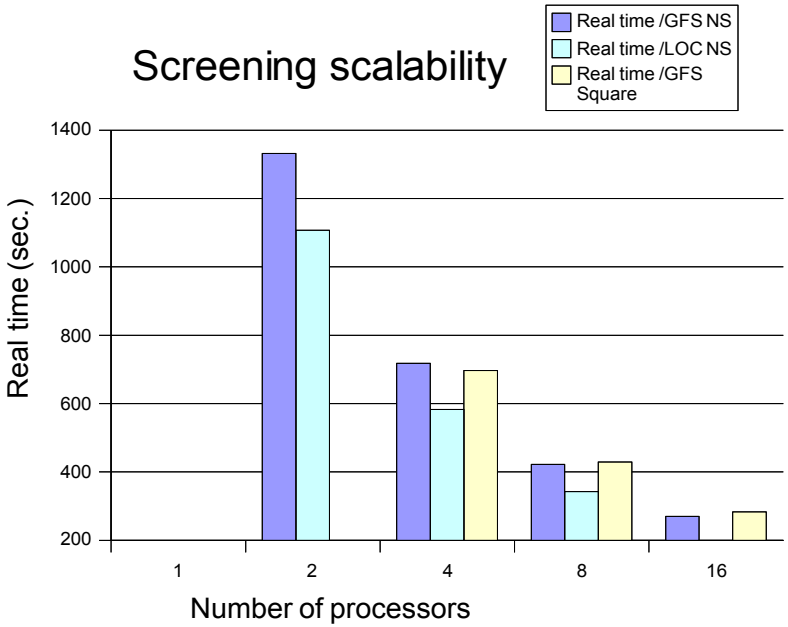
ODB :

- Porting has been partly missed :
  - An #ifdef VPP not translated => computation via a scalar subroutine (in ifsaux/xrd). Now fixed.
  - Vectorisation directives inside C codes not interpreted by the compiler, unless on top of the subroutine => need to use Sami Saarinen's wrapper for the sxcc compiler. Now active.
- Tuning of ODB environment variables under progress.
- Local disks synchronisation tool : to be re-written (failures occured)

BATOR :

- Profilings have been done early on top of the mis-porting of ODB
  => to be traced again
- However some optimisations have been realised (D. Puech)
- Further optimisations under progress to enable distribution

Screening



- Scalability looks good
- Better use local disks

  => needs inter-nodes synchronisations when more than 8 processors
- Bug if 1 processor only ? (time limit systematically exceeded)
- Performance *was* limited by RRTM – better vectorized.

TO DO :

- Re-examine profiles since RRTM optimisation and when ODB environment variables are properly tuned

Minimisation – High resolution

- Performance limited by larcinbad
  (500 Mflops ≈ 7% of what could be expected on a SX8R !)
- NEC reported poor performance because of « gather » operations. Now optimised via compiler directives (≈ *1 Gflops)*

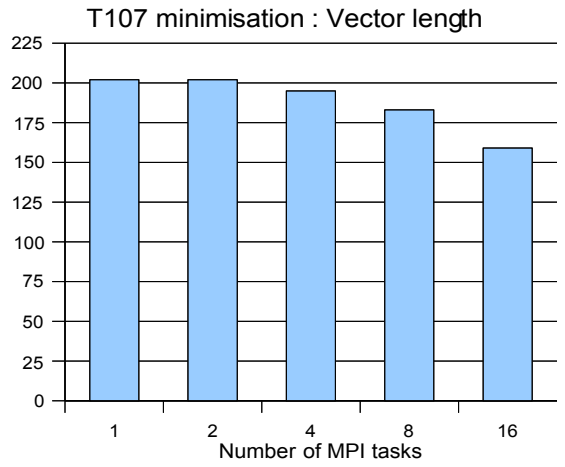
« Top 10 » in CPU for minimisation T224 (16 processors)

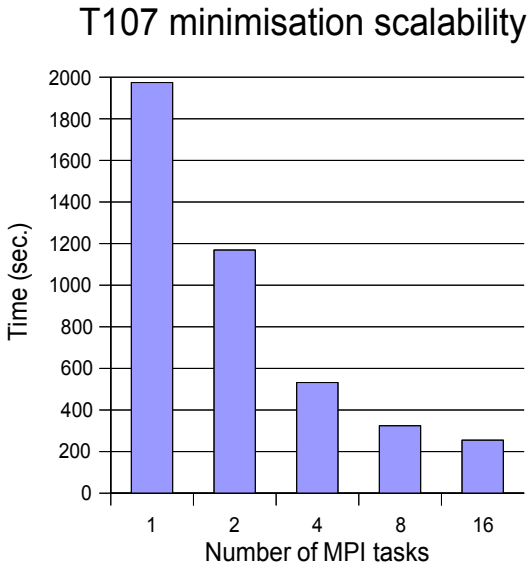
PROG.UNIT	FREQUENCY	EXCLUSIVE TIME[sec](%)	AVER.TIME [msec]	MOPS	MFLOPS	V.OP RATIO	AVER. V.LEN	VECTOR TIME	I-CACHE MISS	O-CACHE MISS	BANK CONF
larcinbad	18432	729.675(8.3)	39.587	2627.4	503.3	98.88	246.3	726.751	0.8070	0.7396	28.4321
mpl_alltoallv_real8	34464	411.726(4.7)	11.947	1289.6	0.0	95.84	86.7	330.736	3.4092	3.2192	2.5835
sgemmx	7604760	323.400(3.7)	0.043	32955.1	23321.5	98.95	230.1	294.420	5.2454	15.6378	2.9159
laitritlad	73728	287.668(3.3)	3.902	18410.4	5165.4	99.80	253.4	283.547	0.4542	0.1474	39.8357
mpl_barrier	7936	274.377(3.1)	34.574	158.1	0.0	32.15	64.1	9.727	0.0785	0.0572	0.0077
jgvcor	1040	255.555(2.9)	245.726	5550.6	1552.3	96.50	131.1	249.505	0.0145	0.1027	0.0351
laitritl	73728	240.155(2.7)	3.257	22302.1	7111.3	99.94	253.4	240.052	0.0465	0.0309	13.2983
mpl_recv_real8	527999	226.704(2.6)	0.429	1885.9	0.2	96.79	118.1	182.394	4.3155	3.8550	2.3810
acdraglad	18432	186.728(2.1)	10.131	15104.0	5875.7	99.71	253.4	185.332	0.8763	0.2748	22.4471
verint	690608	185.503(2.1)	0.269	35065.1	26829.5	99.20	244.4	182.296	1.4914	1.1125	7.6109
total	698388450	8823.367(100.0)	0.013	9381.0	4004.1	98.22	216.1	6275.324	324.2783	561.7844	457.7904

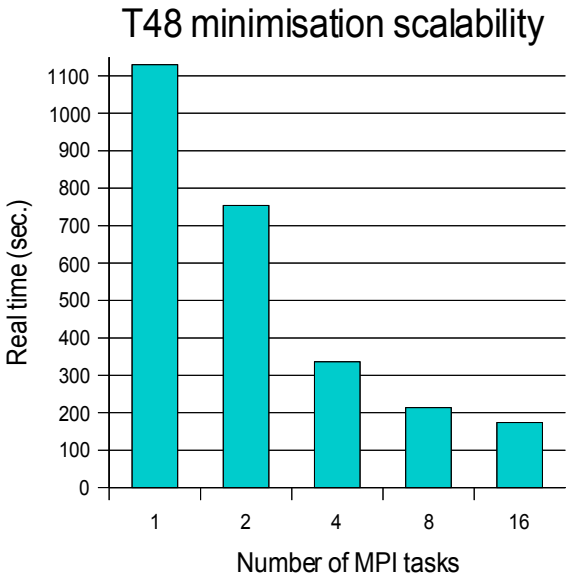
Minimisation – Low resolution

CPU :

- Max. number of gridpoint per processor can become too small for optimal vectorisation







ELAPSE :

- However communications are driving the performance. Scalability remains unchanged with an even lower truncation (<=> relatevely with more obs.)

ARPEGE/ALADIN - AROME

- ARPEGE/ALADIN :
  - Performances were driven by RRTM
    => profiles to be re-examined since its optimisation
- AROME :
  - Performance still driven by communications on SX8R.
  - *(ref : AROME-France, NPROC=64 to run 24 hours forecast in 30 min. elapse)*
  - Miscellaneous tiny improvement must be still possible
  - Could Open-MP reduce the load imbalance caused by the E-zone ? *(tests with ALADIN in CHMI would not advocate for it ...)*

AROME profile (1)

« Top 10 » CPU

PROG.UNIT	FREQUENCY	EXCLUSIVE TIME[sec](%)	AVER.TIME [msec]	MOPS	MFLOPS	V.OP RATIO	AVER. V.LEN	VECTOR TIME	I-CACHE MISS	O-CACHE MISS	BANK CONF
mpl_barrier_mod.mpl_barrier											
	1536640	17233.845(18.7)	11.215	328.7	0.0	66.28	255.3	966.270	10.6682	6.2616	0.6095
mpl_alltoallv_mod.mpl_alltoallv_real8											
	399360	8574.898(9.3)	21.472	2300.1	0.0	97.96	211.3	6897.549	58.1832	84.6792	25.0620
laitri	1659456	5176.077(5.6)	3.119	22911.2	7129.7	99.97	252.6	5175.162	0.2685	0.2395	336.5046
mpl_recv_mod.mpl_recv_real8											
	5875121	3655.534(4.0)	0.457	1846.4	0.0	95.85	241.7	1607.094	48.6462	134.9250	53.9369
laitqmh	921920	3555.263(3.9)	3.856	25881.2	6685.3	99.97	252.6	3554.423	0.3736	0.2597	185.5754
apl_arome	92224	2619.601(2.8)	28.405	8303.9	1928.0	99.11	252.7	2583.540	19.2293	9.9450	568.3468
rpassf	10296160	2133.639(2.3)	0.207	19604.4	9380.3	99.75	248.2	2109.949	8.3502	7.4220	204.1273
mpl_send_mod.mpl_send_real8											
	5875121	2108.973(2.3)	0.264	941.4	0.0	71.40	200.1	880.261	21.9603	18.2828	1.4300
laitli	2581376	1885.801(2.0)	0.731	16941.8	4868.7	99.90	252.6	1884.065	0.4754	0.5186	115.8275
slcomm2a	184384	1802.231(2.0)	9.774	509.0	0.0	68.88	64.8	738.970	18.3547	77.6585	855.7301
elascaw	1106304	1300.606(1.4)	1.176	15613.6	1478.7	99.83	239.1	1294.891	2.3991	1.8066	103.9366
total	1111073850	91584.656(100.0)	0.082	8840.5	2703.3	99.08	200.7	79404.147	1724.5077	1687.0571	5167.0221

AROME profile (2)

« Top 10 » ELAPSE

PROG.UNIT	ELAPSE [sec]	COMM.TIME [sec]	COMM.TIME / ELAPSE	IDLE TIME [sec]	IDLE TIME / ELAPSE	AVER.LEN [byte]	COUNT	TOTAL LEN [byte]
mpl_barrier_mod.mpl_barrier								
	409.967	408.869		41.312		0.0	0	0.0
mpl_alltoallv_mod.mpl_alltoallv_real8								
	165.311	165.247		114.999		25.4M	399360	9.7T
mpl_recv_mod.mpl_recv_real8								
	141.338	140.911		109.412		1.5M	5875121	8.2T
mpl_send_mod.mpl_send_real8								
	114.534	114.295		37.115		1.5M	5875121	8.2T
laitri	85.088	0.000		0.000		0.0	0	0.0
laitqmh	58.738	0.000		0.000		0.0	0	0.0
apl_arome	40.747	0.000		0.000		0.0	0	0.0
lfildo	39.572	0.000		0.000		0.0	0	0.0
opdis	39.477	0.000		0.000		0.0	0	0.0
rpassf	35.319	0.000		0.000		0.0	0	0.0
total	1745.378							

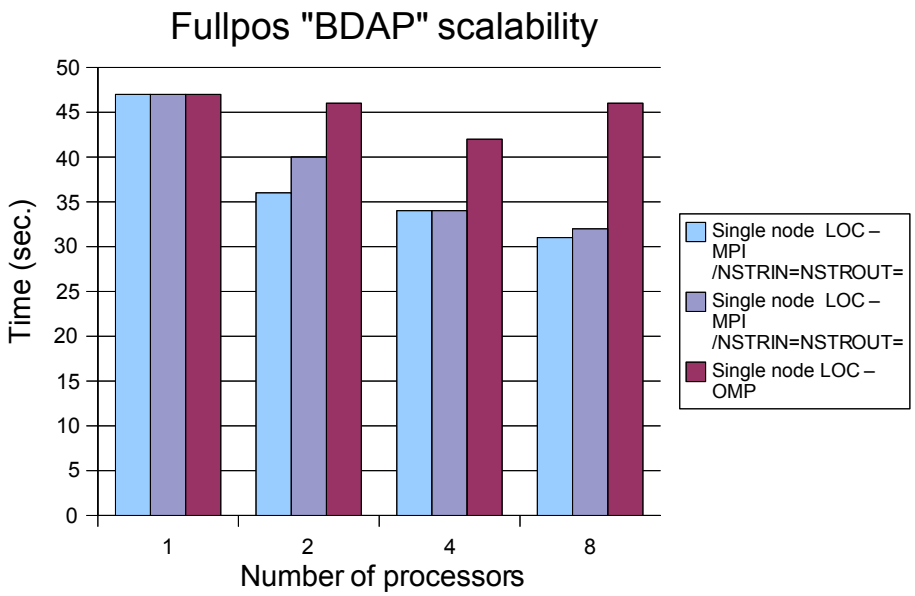
Performance* of AROME vs. ARPEGE

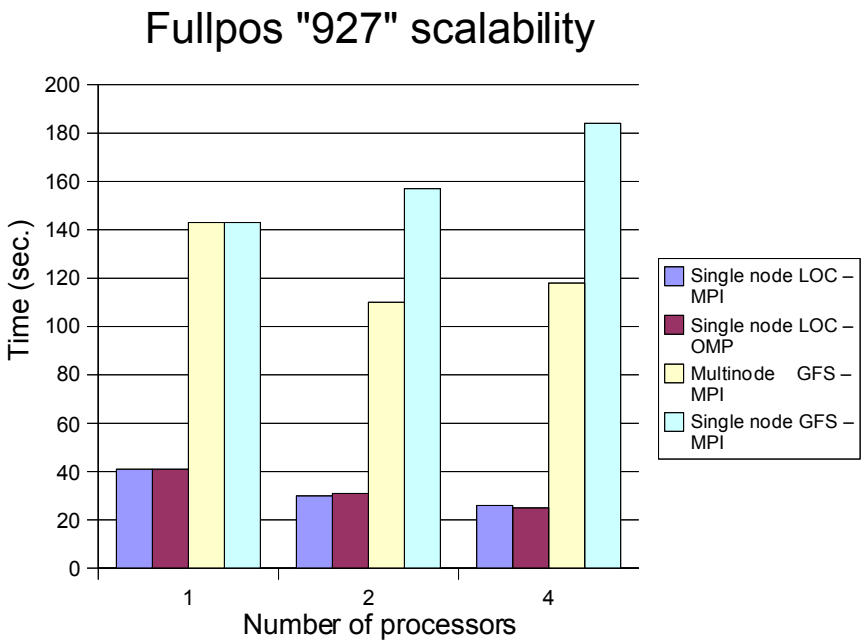
- AROME 32 processors over France** :
  - NGPTOTG=307200 ; NFLEVG=41 ; => 2.85 GFlops/cpu
- AROME 4 processors over Austria :
  - NGPTOTG=64800 ; NFLEVG=41 ; => 3.58 GFlops/cpu
- ARPEGE 32 processors T538 :
  - NGPTOTG=399180 ; NFLEVG=60 ; => 5.36 GFlops/cpu
- ARPEGE 4 processors T538 :
  - NGPTOTG=399180 ; NFLEVG=60 ; => 6.11 GFlops/cpu

* Before optimisation of RRTM and without MPI synchronisation in AROME.
** with predictor-corrector scheme. May not be used.

Fullpos

- Post-processing (on a single file) :
  - Performance driven by large I/Os (mostly matrixes because of stretching)
    => poor scalability





- « 927 » (changing geometries) :
  - Performance driven by the I/Os (mostly the intermediate file ...)
    => poor scalability
  - Better use local disk

Conclusions & Outlooks (1)

- **Investigations on various configurations of ARPEGE and AROME on NEC platform has lead to several positive results :**
  - Better understanding of I/Os performances, improvements, and some ideas to go further
  - Identification of the optimal NPROMA values
  - Reduction of load inbalance and speed-up of communications in AROME
  - Identification of misporting of ODB
  - Optimisation of RRTM, done by NEC team at Météo-France

elapse time : Arpege T358l46/12processors = Arpege T538l60/16processors (= cpu power x 2)

- **Still problems remains :**
  - Limitations in the search for a higher performance in the minimisation
  - Apparently poor performance of AROME, compared to ARPEGE

Conclusions & Outlooks (2)

- **What to do now ?**
  - Priority #1 is now to optimise post-processing :
    - Operations to switch to the in-line mode of Fullpos
  - Finish the work on the miscellaneous identified weaknesses
  - Get more benefits from ECMWF last benchmark (under progress)
  - Update profiles with the last optimisations, and assess next priorities
  - Survey of RTTOV-9 (expected to be even better vectorised, however)
  - Keep the profiles up to date along the code releases *on various platforms*
- **And also :**
  - Investigate more configurations (3DVar Aladin/Arome, 601, CANARI...)
  - Be prepared for optimising on SX9 in early 2009
  - Work on other identified problems :
    - Fullpos « 927 » : remove the intermediate file thanks to the modularity of spectral transforms
    - Further optimisation of Surfex I/Os
    - Support for Open-MP in AROME (at least for portability reasons)

METEO FRANCE

Toujours un temps d'avance