TECHNICAL MEMORANDUM

616

Report from IFS scalability project

Mats Hamrud

Research Department

February 2010

European Centre for Medium-Range Weather Forecasts
Europäisches Zentrum für mittelfristige Wettervorhersage
Centre européen pour les prévisions météorologiques à moyen

Series: ECMWF Technical Memoranda

A full list of ECMWF Publications can be found on our web site under:
http://www.ecmwf.int/publications/

Contact: library@ecmwf.int

European Centre for Medium Range Weather Forecasts
Shinfield Park, Reading, Berkshire RG2 9AX, England

Abstract.

The scalability of the data assimilation system and forecast model at ECMWF is an area of growing concern. Current industry trends indicate that the core count on High Performance Computers will increase dramatically in the future and ECMWF must be ready to meet this challenge. Here we discuss the scaling properties of the main components of ECMWF's forecasting system and reach some tentative conclusions. The forecast model seem to scale reasonably well when trying to solve bigger problems on increasing number of cores. In contrast, the data assimilation system is only able to efficiently use a fairly small fraction of the High Performance Computer currently installed at ECMWF. It would seem that this is a fundamental feature of the algorithm used rather than a technical implementation issue. Any further improvement must come from a thorough review of the algorithms employed, adopting solutions that can be seen to scale to an order of magnitude more cores than currently employed.

1 Introduction

The issue of scalability of meteorological applications have been receiving increased attention within both the weather forecasting and climate community in recent years. The background is the rapidly increasing core counts on the computers procured within the community to run its models on and the predictions from the HPC community that this trend will continue in the future. Most existing forecast models and analysis systems were originally designed to run on shared memory vector computers and then adapted to use a limited number of distributed memory processors. The algorithms chosen and the implementation of these algorithms often put limits on the number of cores that can be usefully employed. At ECMWF this limit has already been reached for the data assimilation system whereas the forecast model is less severely affected.

With the upgrades to the High performance Computer (HPC) that has taken place over the last decade, the fraction of the installed system that can be usefully utilized for the operational data assimilation at ECMWF has steadily decreased. This has been in a period when both the number of cores in the systems and the speed of the individual cores have increased. If the predictions we have from HPC vendors are accurate there will be little or no increase in the speed of individual cores in the future, mainly due to the increased power consumption required by higher frequencies. What we will have instead is an increase in the number of cores on individual chips. This will put increased demands on the scalability of applications to efficiently utilize these massively parallel computers. It would seem from previous experience that the incremental 4D-Var data assimilation scheme, see Courtier et al. (1994), Rabier et al. (2000), currently implemented at ECMWF would struggle to make good use of a vastly increased core count.

This decrease in the fraction of the HPC that can efficiently be used for the incremental 4D-Var data algorithm has occurred although significant effort, se e.g. Isaksen and Barros (1994), Saarinen et al. (1996), Isaksen and Hamrud (1996), Hamrud(1998), Saarinen(1998), Mozdzynski (2006), has been spent through the years to improve the parallel implementation, especially with a view to load balancing issues. This effort has been partially offset by the increasing complexity of the 4-D var algorithm employed, driven by scientific developments aimed at improving the quality of the analysis and by a large increase in the number of observational data sources, mainly new satellite instruments.

The "IFS scalability project" was set up at ECMWF in August 2008. The aim of the project was to prepare and test proposals on how to adapt the IFS code to run more efficiently on massively parallel computers, with particular emphasis on the 4D-Var data assimilation at future operational resolutions. This project was initiated to study this issue now, as any outcome requiring drastic changes to the algorithms would take a significant time to implement and evaluate. Similar projects are up and running or being considered within other organizations involved in weather forecasting and climate research.

Initial work within this project has been focused on a better understanding of why incremental 4D-Var scales badly and on addressing some weak points in the technical implementation. In the following the outcome of this work is presented and some tentative conclusions are drawn regarding future directions of work.

2 Scaling properties of 4D-var data assimilation and deterministic forecast model

After addressing a number of technical issues affecting the scaling of 4D-Var (see appendix A), the scaling properties of the system scheduled for implementation with the increase in resolution to T1279 has been assessed. The assessment has been done on our current IBM Power6 system. Due to the difficulties of transferring the environment needed for running data assimilation jobs, it has not been possible to assess the generality of the findings by running on other architectures. For the forecast model, which is much more portable, we have done some comparisons of the strong scaling properties of a low-resolution model between the Power6 and a Cray XT4. These results indicate qualitatively similar scaling properties although the details will depend on significant properties of the systems like processor speed, interconnect bandwidth and interconnect topology.

The scaling experiments done with 4D-Var are *strong scaling* experiments, i.e. we are running the same experiment at the same resolution using different number of processors. It would also be interesting to know the *weak scaling* properties by running different size problems in the same wall clock time. This is difficult to do in practice as there is a priori no reason for a fixed ratio between the resolution used in the inner and outer loops, the convergence rate of the minimizations may change with resolution, you can vary the time-step used in the inner and outer loops and so on. In the case of the forecast model this is more straightforward for the range of resolutions we are considering, in principle you can just change the horizontal resolution and adapt the time-step. Thus for the forecast model we will present both strong and weak scaling properties.

The headline numbers for the scaling of 4D-Var can be seen in Figures 1 and 2. The figures show the behaviour of the 12 hour 4D-Var, not the early delivery 6 hour version. The early delivery version scales slightly worse.

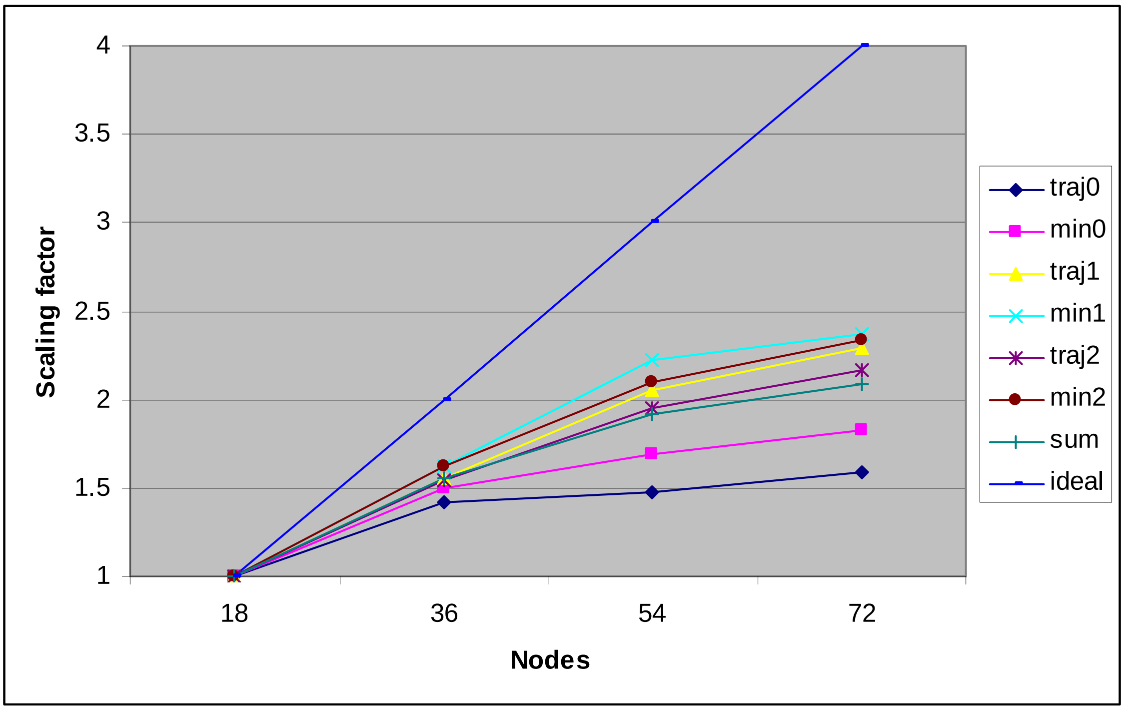

Figure 1. Scaling as a function of node count of the different components of 4D-Var. Ideal stands for perfect scaling and sum for the sum of all the jobs. The resolutions used for various components are T1279 (traj0, traj1, traj2), T159 (min0) and T255(min1 and min2).

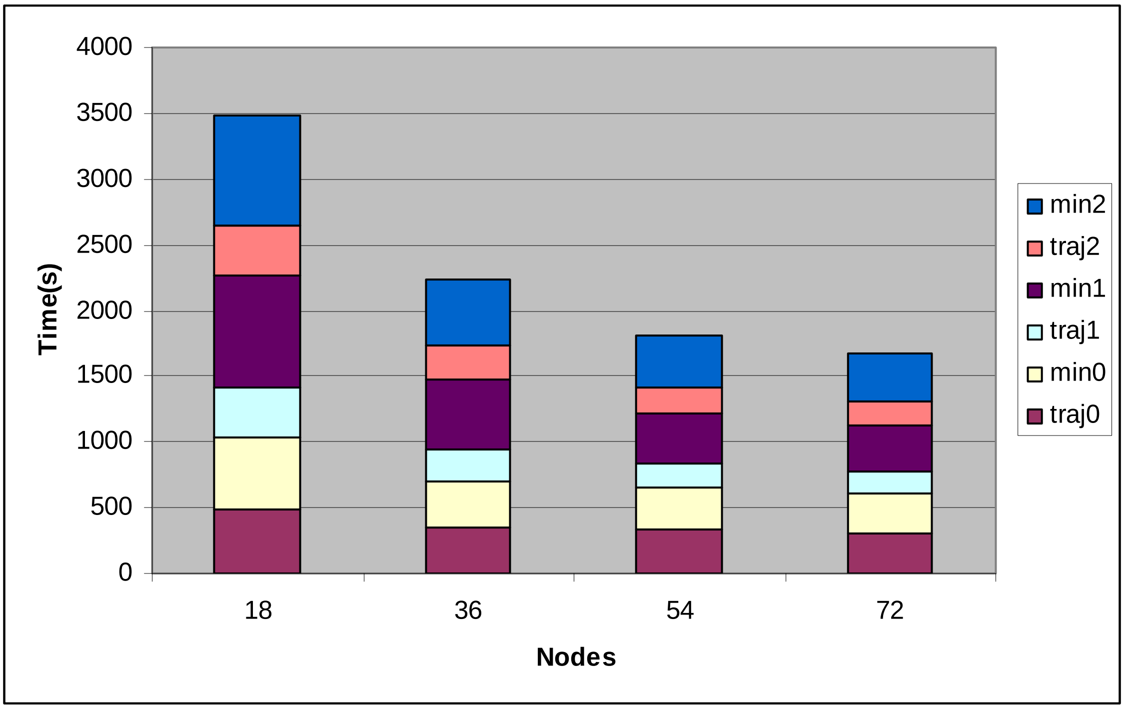

Figure 2. Run-times of 4D-Var for different node counts with 4D-Var broken down into its constituents.

The smallest job using 18 nodes is close to the limit imposed by memory restrictions. It is clear that by 54 nodes the useful limit for increasing the speed has been reached although the scaling has not yet turned negative. The element of the analysis that behaves worst is the first trajectory which includes the screening of the observations followed by the first T159 minimization. The best behaving elements are the two T255 minimizations. The T159 minimization differs from the T255 minimizations not only by the resolution but also by the use of a very simplified linearized physics, only including vertical diffusion.

Taking a closer look at one of the minimizations we compare two runs at 18 and 72 nodes. The ideal speedup would be a factor of four, what we achieve is instead a factor of around 2.3. The computational characteristics change fairly dramatically between the two runs (see Figure 3), the most obvious being the decrease in the proportion of time spent in OpenMP regions and the increase in the load imbalance. The proportion of time spent doing IO increases by a factor of four which indicates no scaling and the proportion spent in message passing almost doubles. If the proportion spent in the OpenMP regions is used as an indicator of the proportion of "useful computations" this measure decreases from 73% to 47%.

Another way of looking at the same problem is to view it from a more scientific point of view (see Figure 4). Here an attempt has been made to categorize the different detailed timers into parts of the computations a scientist working with 4D-Var would recognize like adjoint model, background term etc. This mapping is not always very straightforward especially in correctly attributing load imbalance which is the reason why we see fairly large chunks of time attributed to un-assignable load imbalance or simply not classified. With these caveats it is still clear that the proportion of time spent in the model part decreases with increasing node count and some others increase, notably the background term. This means that the model part of the computations scales better than the background term evaluation. It is also worth noting that the observation calculation is always a small part of the cost and that this part decreases slightly with increasing node count. The conclusion from this is that the overall scaling properties would improve with more observations.

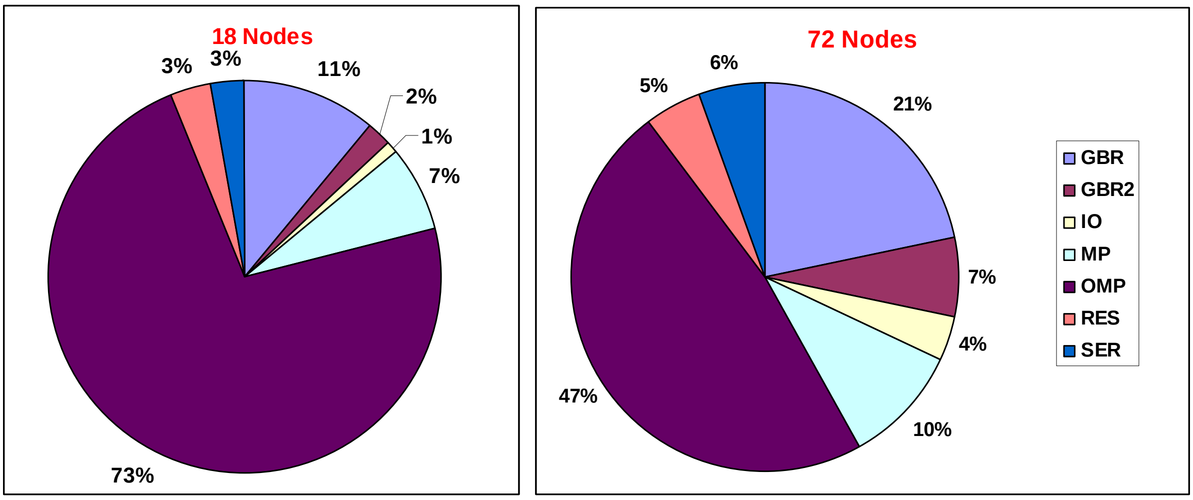

Figure 3. Change in computational characteristics between an 18 and 72 node executions of the second minimization step. Here GBR denotes computational load imbalance, GBR2 load imbalance in the communications, IO Input/Output operations, MP message passing, OMP OpenMP regions, RES unclassified and SER serial computations.

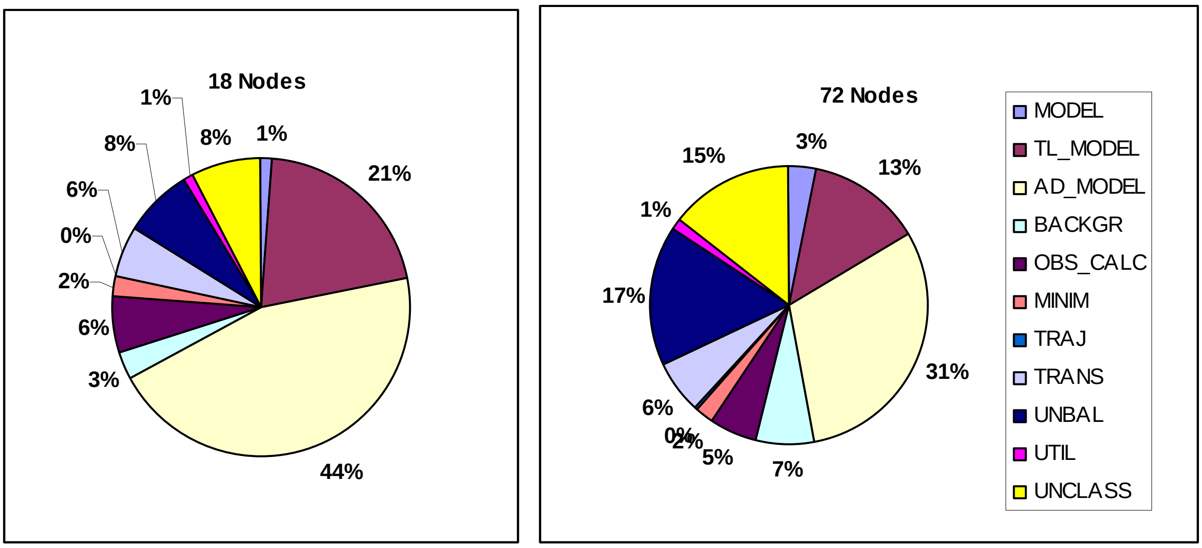

Figure 4. Change in distribution of time spent from a scientific point of view going from 18 to 72 nodes in the second minimization. Here MODEL stands for the non-linear model, TL_MODEL for the tangent-linear model, AD_MODEL for the adjoint model, BACKGR for the background terms, OBS_CALC for the observation related computations, MINIM for the minimization calculations, TRAJ for handling the linearization state, TRANS for the spectral transforms, UNBAL for load imbalance that can not be directly attributed to any other term, UTIL for general utility routines and UNCLASS for unclassified time.

The basis for the detailed analysis of the minimization is the GSTATS package, developed specifically for timing the IFS. The package consists of simple wrappers for calling the basic timing routines (wall-clock, CPU time etc.) available on all systems. The use of a wrapper enables us to produce basic statistics like the standard deviation of the time taken in specific code segments. The intention is to have all OpenMP, message passing and significant serial parts of the timed by these calls to GSTATS. The fact that these have to be manually added implies quite a significant initial investment and a continuous maintenance task to make sure code additions are covered. Another part of the package is (optional) barrier synchronizations before and after message passing regions in order to be able to measure load imbalance in different parts of the code. There are to our knowledge no generally available tools that could be used for this type of study. A significant amount of time was spent in the early stages of the project to ensure that the GSTATS timers adequately covered all aspects of the 4D-Var algorithm. We currently have well over a thousand individual regions within the IFS timed by the GSTATS package.

An example output from a post-processed GSTATS output comparing the second minimization on 18 and 72 nodes can be found in appendix B. These kinds of outputs form the basis for all the detailed assessments of the scaling properties. The last column (lost) in the output indicates how many seconds were “lost” in a code segment compared to what we would expect from perfect scaling. The technical scaling issues we have addressed (see appendix A) were selected for further study on the basis of these numbers. It can be seen that now there are no individual timers where the loss is more than 7 percent of the total time lost. Thus progressing along this road we would see sharply diminishing returns even if we knew how to improve the scalability of these regions.

The high resolution trajectory runs scale in similar fashion to the minimizations (see Figure 1) but a closer analysis (see Figure 5 and Figure 6) shows a rather different picture. At low node count the run time is dominated by the cost of the atmospheric model whereas at higher node count other aspects like the writing out of the trajectory and the post-processing start taking a significant portion. For the first and the last trajectory we have again additional parts of the code becoming active with the screening of observations and the match-up for the creation of the observation feedback data (not shown).

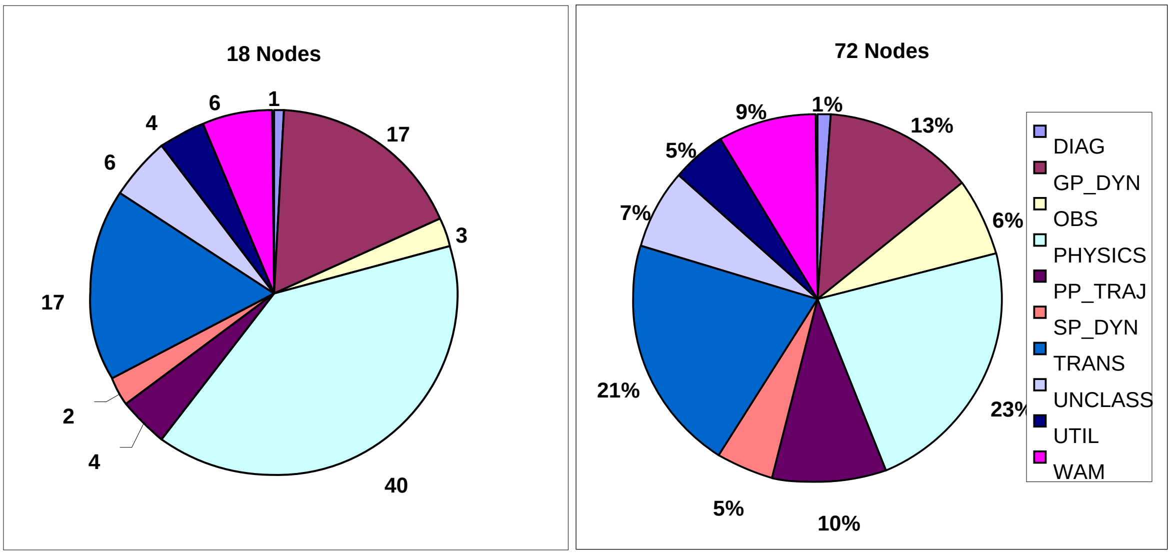

Figure 5. Change in distribution of time from a scientific point of view going from 18 to 72 nodes for the second trajectory. Here DIAG stands for diagnostics, GP_DYN grid-point dynamics, OBS observation related tasks, PHYSICS the model parameterization, PP_TRAJ post-processing and trajectory handling, SP_DYN the spectral part of the dynamics, TRANS spectral transforms, UNCLASS un-classified, UTIL general utility routines and WAM for the wave model.

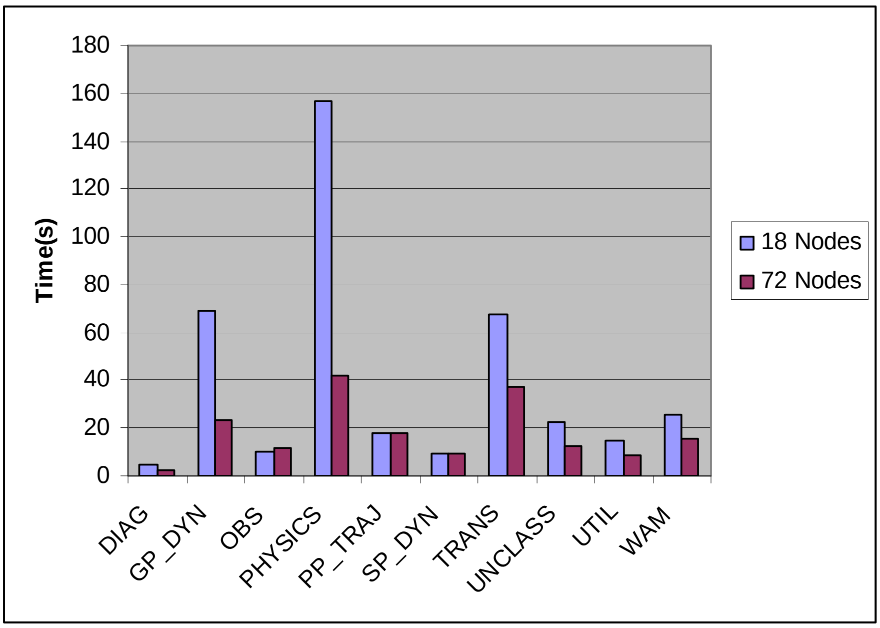

Figure 6. Change in wall clock time going from 18 to 72 nodes for the second trajectory from a scientific point of view. Here DIAG stands for diagnostics, GP_DYN grid-point dynamics, OBS observation related tasks, PHYSICS the model parameterization, PP_TRAJ post-processing and trajectory handling, SP_DYN the spectral part of the dynamics, TRANS spectral transforms, UNCLASS un-classified, UTIL general utility routines and WAM for the wave model.

The forecast model at the operational resolution scales significantly better than 4D-Var (see Figure 7). It continues scaling positively up to a whole cluster, with the efficiency decreasing only by a factor of two from 10% at 32 nodes to 5% at 256 nodes. For a run time that complies with the operational schedule it is enough to run the 10-day forecast on the order of 30-40 nodes. The efficiencies achieved for the forecast model can be compared with 4D-Var which achieves an efficiency of 3.2% on 48 nodes for the 12 hour window case.

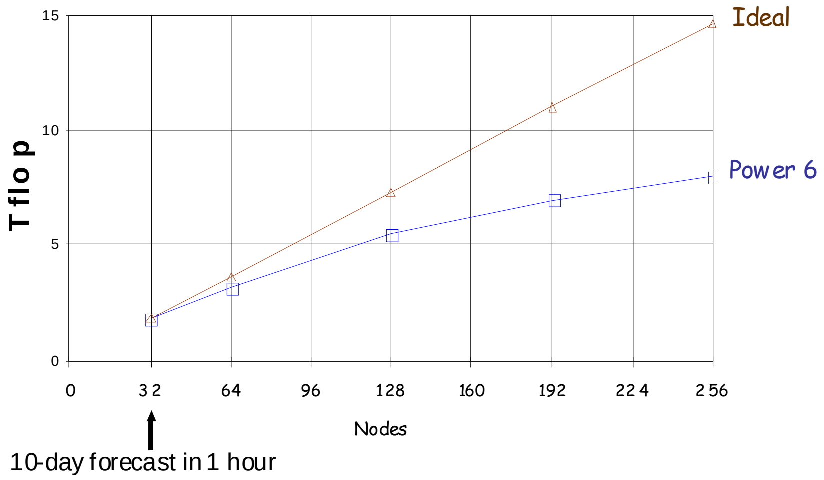

Figure 7. Scaling of the T1279 forecast model on the Power6. [Figure provided by Deborah Salmond]

Another way of considering the scaling of the forecast model is to look at its weak scaling capabilities. In Table 1 we see the run times and percentage of peak achieved when running a 10 day forecast at different resolutions, always aiming at a run time of around one hour. The number of nodes needed for each run was estimated from the T1279 base taking into account only the changing number of grid-points and the time-step. As we can see the efficiency, as measured by the percentage of peak obtained, stays more or less constant at around 10 percent. This means that for the deterministic forecast model the weak scaling, which is the one we are primarily interested in, is not currently an area of concern. The slightly longer runtime for the T2047 may be attributed to the quadratic cost of the Legendre transform although there are also other parts of the code where the operation count is not linearly dependent on the number of grid-points of the model.

Resolution	Nodes	Cores	Time (s)	% of peak
T511	3	96	3343	9.6
T799	10	320	3323	9.6
T1279	32	1024	3370	10.1
T2047	110	3520	3765	9.9

Table 1. Executions of a 10-day forecast on the Power6 at varying resolution

The scaling properties of the forecast model do however resemble closely those of 4D-Var if we consider running a forecast in the same way we require 4D-Var to perform. This requirement is more akin to running a relatively low resolution (T255) forecast so that the time taken to perform a model time-step has to be 25-50 times shorter than what is required when running the high resolution forecast. This would be the same as running a 30 day forecast in 5 minutes. Figure 8 below illustrates this; it is the same as Figure 1 above but with an added curve showing the scaling properties of a 30-day T255 forecast with the same node count as those in the 4D-Var scaling study.

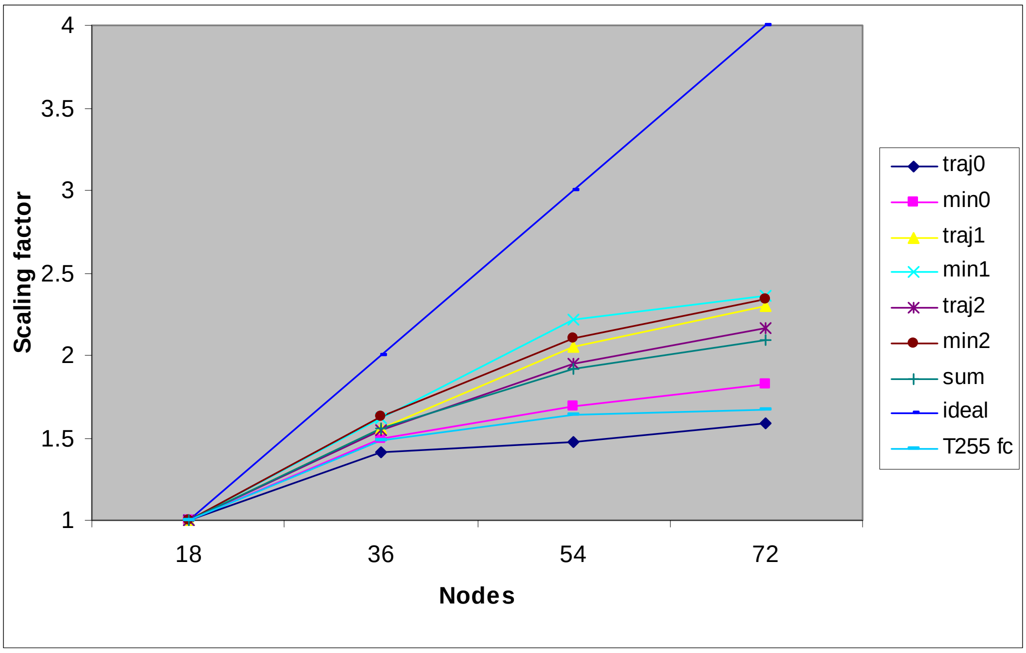

Figure 8. Scaling as a function of node count of the different components of 4D-Var and a T255 forecast model. Ideal stands for perfect scaling and sum for the sum of all the 4D-Var jobs.

As the forecast model is much more portable than the 4D-Var system we have been able to compare the scaling properties of a low resolution forecast between the Power6 at ECMWF and a Cray XT4 (see Figure 9 below). The overall impression is one of very similar behaviour (the kink in the IBM curve is explained by a disturbed run). The actual run -times on the IBM are lower (see Figure 10) and this does not take account of the use of SMT on the IBM, the corresponding runs on the IBM are performed with half the number of physical cores compared to the Cray runs.

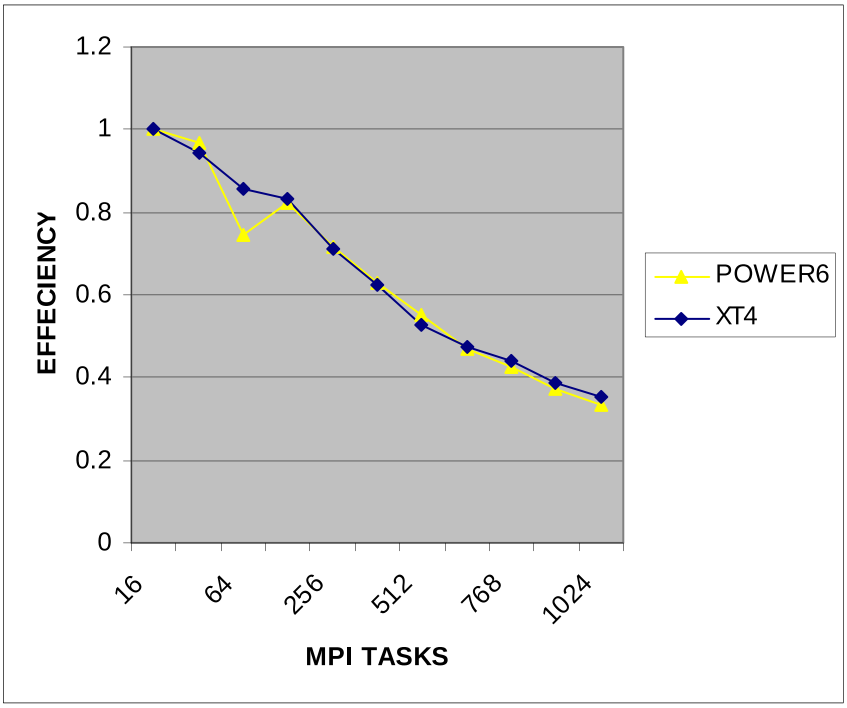

Figure 9. Efficiency of running a 10-day T255 forecast using varying number of MPI tasks. All runs use 4 OpenMP threads.

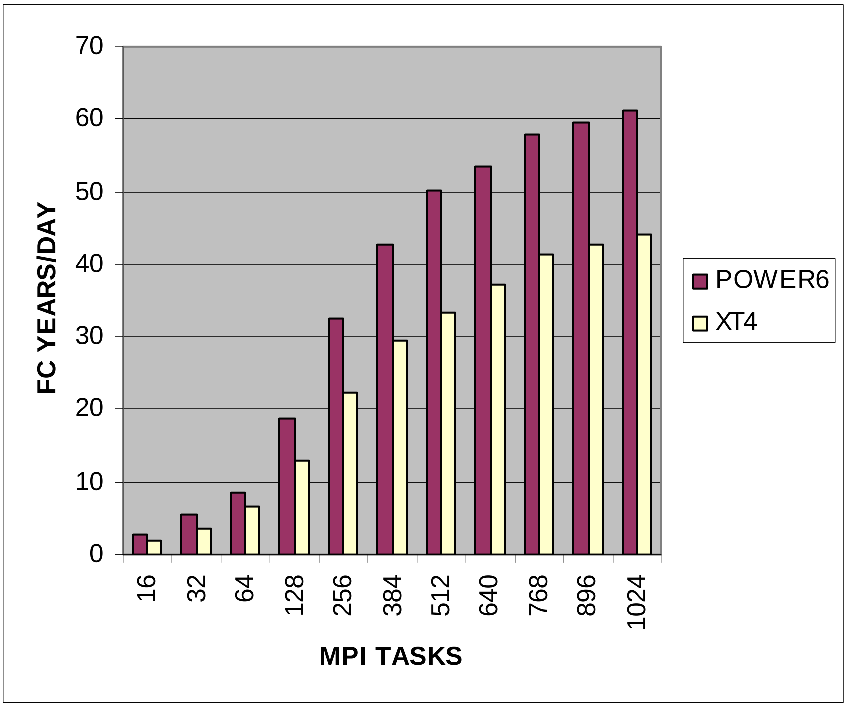

Figure 10. Number of forecast years per day achieved using different number of MPI tasks. All runs use 4 OpenMP threads.

3 Current understanding of problem

The main problem with the scaling of incremental 4D-Var comes from the fact that what we have is basically a sequential algorithm. The iterations have to be done in a sequential manner. Within the minimization step the use of the Conjugate Gradient Method implies a sequential process. The time integrations of the tangent linear and adjoint models are also sequential. Thus, the parallelisation is only at the lowest level, for the model part based on a space decomposition and for the observations an equal distribution for all observation types. This leads to a fine granularity of work with frequent implicit synchronizations arising from the message passing (each time step implies at least 6 transpositions + the Semi-Lagrangian communications). For example in the first minimization we call the tangent linear physics 1700 times spending on the average 1.5 milliseconds in each call. The individual packages of data exchanged in the message passing also decrease in size when using more processors, increasing the relative importance of the message passing latency and exposing ourselves to any “jitter” present in the message passing fabric or operating system.

When we run 4D-Var on 72 nodes (2304 cores) we only have 64 vertical columns in grid-point space per MPI task in the first minimization (T159). For each MPI task we use 8 threads so this implies only 8 columns per thread. This very small number of columns means that we will not get any help from dynamical load balancing between columns in different meteorological situations, we will for each time step asymptote towards the most expensive area on the globe. Also the efficiency of the computations decreases with decreasing number of vertical columns per thread (see Figure 11). The main reason for this is that shorter loops leave less scope for the compiler to optimize, e.g. by un-rolling, as seen by the difference between the “opt” and the “nonopt” curve in Figure 11. Another reason is the fixed overhead of traversing the calling tree and loop start-ups. This problem of decreased work package size is most easily illustrated in the case of the model grid-point calculations but it extends to all other areas of the 4D-Var algorithm like the observation calculations and the background term. In some parts of the problem we are running out of degrees of freedom to keep all processors busy.

In theory the forecast model should suffer from the same fundamental problem as 4D-Var. It is also a sequential problem in the time dimension. When we increase the resolution we normally also decrease the time step proportionally. This means the cost of the problem is proportional to the resolution to the power of four (the three spatial dimensions plus time). The implication is that if processor speed does not increase and we want to solve a bigger problem using the increased number of processors of a future computer, the number of grid columns per processor would have to decrease like the square of the resolution change.

In practice the difference between the scaling properties of 4D-Var and the forecast model is where we are on the curve. At T1279 running on 32 nodes, which is fast enough for the operational schedule, we still have over 1000 grid columns per processor.

In addition to the fundamental scaling problem for 4D-Var stated above there are other, more technical problems. The incremental 4D-Var algorithm is currently implemented as a sequence of jobs, each with one or more executions of the IFS. This causes overheads in repeated startup costs that normally do not scale well and additional IO to communicate between job steps. The advantage of this approach is the creation of simple restart points where individual jobs can be re-run in case of a failure.

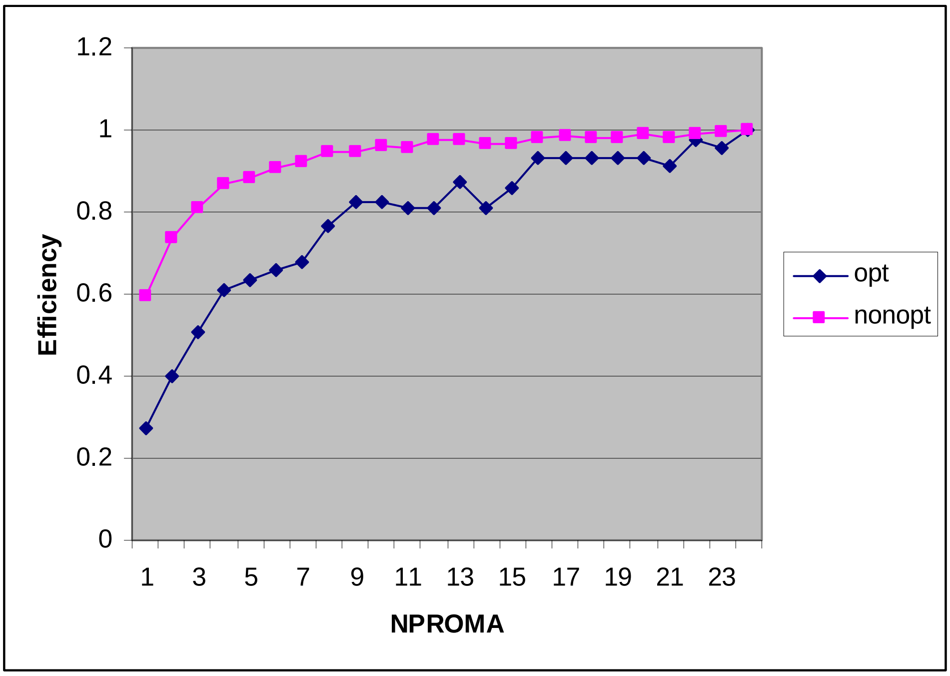

Figure 11. Efficiency of grid-point computations as a function of the number of vertical columns in each work packet. The efficiency is defined as the time taken using a packet size of 24 divided by the time taken using the NPROMA packet size. Larger package sizes than 24 give no significant further benefit on the Power6. The label "opt" stands for the code compiled at our normal optimization level, "nonopt" stands for the code compiled without any optimization enabled.

4 Discussion and conclusions

There is limited scope for further useful work on improving scaling properties in the low-level implementation of incremental 4D-Var. The overall scaling improvements achieved by addressing individual problem areas is decreasing and the effort involved is increasing as the problems that could be solved quickly have already been addressed. Some recent attempts, following ideas that superficially seemed likely to lead to an improvement, have lead nowhere although considerable thought and effort was spent.

Substantial improvements can only be made by reviewing the algorithms used within incremental 4D-Var or by considering alternative Data Assimilation methods. Weak constraint 4D-Var with time-varying forecast error in conjunction with Long window 4D-Var possibly offers further scope for parallelism and improved scaling properties. This has to be balanced against the large increase in the size of the control vector and general increase of the problem size. Much depends on the scientific progress in providing a good pre-conditioner for the problem ensuring fast convergence of the minimizations.

The Ensemble Kalman Filter Technique seems to offer a scalable approach to the Data Assimilation problem but there are still many open questions as to whether this approach can provide an analysis of comparable quality to the one provided by 4D-Var. From the technical point of view there is also the question of the total cost of the system. The scaling properties may be good but the total cost, dependent on the size of the ensemble required, may be prohibitive, especially for further research work. Currently, because of the poor scaling properties of 4D-Var, the research workload runs much more efficiently than when it is run under the constraint of the operational schedule.

In conclusion, any further work in this area must be done in even closer collaboration with the scientists involved, concentrating on changes in algorithms that can improve scalability.

Acknowledgments

This work has been done in collaboration with George Mozdzynski, Deborah Salmond, Tomas Wilhelmsson, Peter Towers and Anne Fouilloux. It has also benefited from useful discussions with, among others, Lars Isaksen, Mike Fisher, Yannick Tremolet and John Hague from IBM.

Appendix A –Problems addressed in the scaling of 4D-Var

A number of technical scaling problems in running IFS in the 4D-Var configuration has been identified and addressed in the course of this project. Some of the solutions have already been incorporated in IFS releases, some are pending incorporation and some are still work in progress. What follows is a brief description of these items.

- Computation of grid-point norms using a more efficient and scalable approach
- Optimization of conserving interpolation scheme
- Optimisation of routines gatherspec (used for gathering a spectral field to one processor) and updtim (initialization of data structures used in the parameterization)
- Better load balance in the observation processing
- Distribution of the encoding and decoding of GRIB data messages with the aid of "helper" processors when reading and writing using one MPI-task only
- Re-distribution by the ODB of observations used in the rain assimilation
- Parallelisation of the distribution of pre-conditioning vectors
- Parallelisation and simplification of dot-products for control vectors
- Revised parallelisation of the semi-implicit dynamics in conjunction with the use of implicit Coriolis
- Improvements in distributing wavelet Jb correlation matrices (using OpenMP)
- Improvement in Semi-Lagrangian communication in 4D-var using the fact that the columns that need to be communicated are the same for each iteration of the minimization.

Appendix B – Example output

```
Output file 1 - f8fi/uptraj_2/ifsmin.1
Output file 2 - f8hi/uptraj_2/ifsmin.1
Hoped for speedup factor -  4
Date and time of job1
Summed time of job 1 845.302777777778 s out of 874.527777777777 s
141.162025316456 (144) outputs
Date and time of job2
Summed time of job 2 357.996180555555 s out of 375.187500000003 s
561.86329113924 (576) outputs
Achieved speedup factor 2.36120613484199


Id Descriptor Calls Time1(s) Time2(s) Speedup Lost time(s)
   0  ALL CNT0     - COMPLETE EXECUTION     1  874.5  375.2   2.33  156.6
1001  OMP PHYSICS                          25    1.3    0.4   3.66    0.0
1004  OMP CALL_SL      1                   25    0.2    0.1   2.00    0.1
1005  OMP CALL_SL      2                   25    0.2    0.1   2.00    0.1
1006  OMP GP_MODEL_TL 1                   720    9.1    2.7   3.37    0.4
1007  OMP GP_MODEL_TL 2                   720  144.1   37.4   3.86    1.3
1008  OMP GP_MODEL_TL 3                   720    0.4    0.1   3.82    0.0
1009  OMP CALL_SL_TL  1                   720    9.8    3.1   3.15    0.7
1010  OMP CALL_SL_TL  2                   720    7.7    2.1   3.59    0.2
1011  OMP GP_MODEL_AD 1                   720    1.6    0.5   3.14    0.1
1012  OMP GP_MODEL_AD 2                   720    0.4    0.1   3.92    0.0
1013  OMP GP_MODEL_AD 3                   720  325.5   87.3   3.73    5.9
1016  OMP GP_MODEL_AD 6                   720    9.7    2.9   3.34    0.5
1017  OMP CALL_SL_AD  1                   720    5.0    1.5   3.29    0.3
1018  OMP CALL_SL_AD  2                   720   25.1    8.5   2.95    2.2
1019  OMP MKGLOBSTAB                        2    0.1    0.1   1.00    0.1
1020  OMP SLINT LAID*IOBS                 800    0.4    0.3   1.69    0.1
1022  OMP SPCM         1                  744    0.1    0.0   5.10   -0.0
1024  OMP SPCM         3                  744    0.1    0.0   5.49   -0.0
1025  OMP CPG          1                   25    0.3    0.1   2.78    0.0
1028  OMP SPCIMPFSOLVE                    744    0.2    0.1   1.81    0.1
1029  OMP SPCMAD       1                  720    0.1    0.0 288.00   -0.0
1031  OMP SPCMAD       3                  720    0.1    0.0   5.19   -0.0
1035  OMP SPCHORAD                        720    0.7    0.3   2.30    0.1
1037  OMP SPCHOR                          744    0.2    0.1   2.00    0.1
1039  OMP SPCIMPFSOLVEAD                  720    0.2    0.1   1.68    0.1
1041  OMP SCAN2MTL     1                  750    0.5    0.2   2.69    0.1
1042  OMP SCAN2MAD     1                  750    0.7    0.5   1.38    0.3
1043  OMP SCAN2MAD     2                  750    0.7    0.3   2.32    0.1
1044  OMP SCAN2MAD     3                 2220    0.5    0.2   2.99    0.0
1047  OMP SCAN2MAD     5                  750    0.1    0.0 576.00   -0.0
1051  OMP TRAJ_MAIN_MOD    2             1772    1.1    0.3   3.64    0.0
1061  OMP TASKOBTL     1                   30   18.3    5.2   3.51    0.6
1062  OMP TASKOBAD     1                   30   24.8    7.2   3.47    1.0
1070  OMP MPOBSEQ      1                  800    0.3    0.3   0.86    0.2
1071  OMP MPOBSEQ      2                   31    0.1    0.0  23.41   -0.0
1073  OMP MPOBSEQ      4                  800    0.0    0.0  30.00   -0.0
1075  OMP MPOBSEQAD    1                   30    0.2    0.1   2.03    0.0
1076  OMP MPOBSEQAD    2                  750    0.1    0.1   2.26    0.0
1077  OMP MPOBSEQAD    3                  750    0.2    0.2   0.90    0.2
1085  OMP LAIDLIOBSAD                     750    0.5    0.2   3.10    0.0
1090  OMP BALVERT                          30    0.3    0.1   3.34    0.0
1092  OMP BALVERTAD                        30    0.4    0.1   4.05   -0.0
1094  OMP DIGFIL                          750    0.2    0.1   1.67    0.1
1109  OMP TRSTOM       1                 3048    0.3    0.1   2.99    0.0
1110  OMP TRSTOM       2                 3048    1.8    0.8   2.35    0.3
1111  OMP TRSTOM       3                 3048    0.6    0.2   3.03    0.0
1112  OMP TRMTOS       1                 3048    0.1    0.1   1.01    0.1
1113  OMP TRMTOS       2                 3048    1.3    0.6   2.21    0.3
1114  OMP TRMTOS       3                 3048    0.8    0.3   2.67    0.1
1115  OMP SLCOMM       1                   26    0.1    0.0   6.27   -0.0
```

1116	OMP	SLCOMM 2	26	0.1	0.1	1.00	0.1
1118	OMP	SLCOMM2a 1	2185	1.6	0.8	1.97	0.4
1120	OMP	SLCOMM1 2	73	0.1	0.0	4.61	-0.0
1121	OMP	SLCOMM2a 2	2185	2.8	1.3	2.16	0.6
1131	OMP	SLEXTPOL2	897	0.0	0.0	1.52	0.0
1132	OMP	SLEXTPOLAD	1470	0.0	0.0	0.50	0.0
1202	OMP	RADDRV 2	13	0.1	0.0	28.00	-0.0
1209	OMP	RADINTG-INPUT LOOP	65	0.1	0.1	1.08	0.1
1210	OMP	RADINTG-RADLSW	13	1.8	0.8	2.38	0.3
1220	OMP	SL2_PACK	1544	1.2	0.7	1.80	0.4
1221	OMP	SL2_UNPACK	1544	0.7	0.3	2.19	0.1
1222	OMP	GP_MODEL_AD INIT	720	2.4	1.4	1.70	0.8
1224	OMP	COPYGOM5T0	30	0.2	0.1	2.07	0.0
1227	OMP	GP_MODEL_TL INIT	720	0.4	0.1	3.60	0.0
1230	OMP	PRECOND	61	1.1	0.4	2.68	0.1
1231	OMP	CONGRAD OMP	29	0.3	0.1	3.05	0.0
1233	OMP	CONGRAD OMP 1	632	0.7	0.2	3.42	0.0
1234	OMP	DOT_PRODUCT_CTLVEC OMP 2	89	2.2	0.6	3.71	0.0
1235	OMP	PREPPCM	975	2.2	0.4	5.27	-0.1
1427	OMP	GPNORM1	805	0.1	0.1	1.00	0.1
1441	OMP	JBVCOR_WAVELETIN	1020	1.1	0.4	2.78	0.1
1442	OMP	JBVCOR_WAVELETINAD	1530	1.2	0.4	2.86	0.1
1443	OMP	WAVXFORM	1020	0.3	0.1	2.75	0.0
1448	OMP	CVAR2INAD	210	0.1	0.0	576.00	-0.0
1450	OMP	TRANSDIR_WAVELET	1080	0.2	0.1	2.00	0.1
1451	OMP	TRANSDIR_WAVELETAD	1080	0.1	0.1	1.53	0.1
1602	OMP	TRGTOL PACK	2489	1.0	0.3	3.04	0.1
1603	OMP	TRGTOL UNPACK	2489	1.0	0.3	3.01	0.1
1604	OMP	TRLTOG 1	2398	0.1	0.1	1.92	0.0
1605	OMP	TRLTOG PACK	2431	0.9	0.3	3.11	0.1
1606	OMP	TRLTOG UNPACK	2431	1.3	0.5	2.88	0.1
1609	OMP	TRGTOL 1	2458	0.1	0.1	1.91	0.0
1639	OMP	FTINV_CTL	1111	1.5	0.5	3.20	0.1
1640	OMP	FTDIR_CTL	1378	0.7	0.2	3.38	0.0
1641	OMP	FTINV_CTLAD	930	1.4	0.4	3.59	0.0
1642	OMP	FTDIR_CTLAD	1320	0.8	0.3	3.00	0.1
1645	OMP	LTDIR_CTL - DIRECT LEGENDRE	1378	1.7	0.7	2.33	0.3
1646	OMP	LTDIR_CTLAD - ADJ. DIRECT LEGE	1320	1.4	0.6	2.42	0.2
1647	OMP	LTINV_CTL - INVERSE LEGENDRE	1078	1.8	0.7	2.67	0.2
1648	OMP	LTINV_CTLAD - ADJ. INVERSE LEG	930	1.9	0.8	2.45	0.3
1663	OMP	DIST_GRID_CTL	218	0.1	0.0	356.00	-0.0
1704	IO-	GRIBEX WRITE	274	0.0	0.0	4.00	0.0
1706	IO-	SUJBBAL	2	0.0	0.0	12.00	-0.0
1718	IO-	PREGRBENC Modify GRIB headers	274	0.0	0.0	4.00	0.0
1752	IO-	OPEN IN IOSTREAM	35	0.2	0.0	6.00	-0.0
1753	IO-	CLOSE IN IOSTREAM	33	0.0	0.0	8.00	-0.0
1764	IO-	WRITE IN IOSTREAM	290	0.0	0.0	4.00	0.0
1765	IO-	READ IN IOSTREAM	0	1.0	0.3	3.79	0.0
1766	IO-	RTSETUP I/O	1	3.2	4.2	0.76	3.4
1791	IO-	DB in READOBA	3	2.4	7.8	0.30	7.2
1792	IO-	DB in WRITEOBA	3	2.1	1.6	1.31	1.1
1798	IO-	SUJQDATA	1	0.0	0.0	3.60	0.0
1805	SER	TRGTOL init	2489	0.2	0.6	0.31	0.6
1806	SER	TRLTOG init	2431	0.2	0.7	0.29	0.6
1811	SER	SUPHEC	1	0.1	0.1	1.00	0.1
1815	SER	SLEXTPOLAD array	1470	0.0	0.0	1.92	0.0
1818	SER	SUECRAD	5	1.6	1.5	1.07	1.1
1821	SER	COMMJBBAL zbuf	1	0.0	0.0	4.00	0.0
1822	SER	COMMJBBAL array	1	0.0	0.0	0.00	0.0
1826	SER	GATHERCOSTO array	17250	0.0	0.0	0.92	0.0
1832	SER	CNT4AD OBSPREP	720	0.0	0.0	4.00	0.0
1834	SER	SLEXTPOL1 doloop	2987	0.1	0.1	1.19	0.1
1846	SER	SCATTER_CTLVEC pack	1	0.0	0.0	2.18	0.0
1847	SER	MULTISCATTER_CTLVEC pack	1	0.1	0.1	2.30	0.0
1869	SER	SLINTAD	750	0.0	0.0	88.00	-0.0
1874	SER	SUJQCOR	5	15.0	15.0	1.00	11.3
1876	SER	BCASTCOV PACK/UNPACK	2	0.0	0.0	1.00	0.0

1878	SER IOSTREAM GRIBEX ENC	10	0.1	0.0	1.59	0.0
1879	SER IOSTREAM GRIBEX DEC	1725	0.1	0.0	4.00	0.0
1880	SER SUOBS STAGE 1	1	0.8	0.7	1.23	0.5
1890	SER BRPTOB array1	720	0.9	0.7	1.27	0.5
1898	SER DIGFILAD	750	0.3	0.1	3.01	0.0
1904	SER UPDTIM setup	1495	0.8	0.3	2.37	0.1
1931	SER COMMSPNORM DOLOOP	260	0.0	0.0	2.00	0.0
1934	SER SU0YOMA	4	0.1	0.0	6.78	-0.0
1962	SER DOT_PRODUCT_CTLVEC 1	1128	4.0	1.1	3.61	0.1
1999	SER PRTGOM array	325	0.3	0.1	3.29	0.0
2503	GB2 GBAR IN TRMTOL	2431	1.5	1.5	0.98	1.1
2504	GB2 GBAR IN TRLTOG	2431	1.8	1.8	1.04	1.3
2506	GB2 GBAR IN SUJBWAVALLO	255	0.0	0.3	0.00	0.3
2507	GB2 GBAR IN SUJBWAVALLO MAT	85	2.7	2.5	1.07	1.8
2508	GB2 GBAR IN IOSTREAM READ_RECORD	5	0.1	0.1	0.68	0.1
2510	GB2 GBAR IN DIST_SPEC_CONTROL	109	0.0	0.0	1.28	0.0
2511	GB2 GBAR IN DIST_GRID_CTL	109	0.2	0.3	0.60	0.3
2513	GB2 GBAR IN TRGTOL	2489	1.7	2.0	0.82	1.6
2514	GB2 GBAR IN TRLTOM	2308	1.5	1.7	0.87	1.3
2517	GB2 GBAR IN GATHERBDY	456	0.1	0.2	0.44	0.2
2519	GB2 GBAR IN MULTISCATTER_CTLVEC	1	0.0	0.0	18.40	-0.0
2520	GB2 GBAR IN GATH_GRID_CTL	80	0.1	0.2	0.50	0.2
2522	GB2 GBAR IN DOT_PRODUCT_CTLVEC	1217	0.0	0.0	0.17	0.0
2523	GB2 GBAR IN SCATTER_CTLVEC	1	0.3	0.7	0.47	0.6
2524	GB2 GBAR IN COMMSPNORM	260	0.1	0.2	0.81	0.1
2525	GB2 GBAR IN SLCOMM2 PART1	1544	0.1	0.4	0.28	0.4
2526	GB2 GBAR IN SLCOMM2 PART2	1544	1.1	3.0	0.37	2.7
2527	GB2 GBAR IN SLCOMM	26	0.1	0.3	0.42	0.3
2528	GB2 GBAR IN SLCOMM1	73	0.1	0.3	0.46	0.2
2529	GB2 GBAR IN SLCOMM2A	2185	0.2	0.5	0.42	0.4
2530	GB2 GBAR IN SLCOMM2A PART2	2185	1.7	2.9	0.57	2.5
2531	GB2 GBAR IN TRMTOS	3048	1.0	1.0	1.01	0.8
2532	GB2 GBAR IN TRSTOM	3048	1.0	1.0	0.96	0.8
2535	GB2 GBAR IN MPOBSEQ	31	0.3	0.2	2.11	0.1
2538	GB2 GBAR IN MPOBSEQAD	30	0.7	0.6	1.12	0.4
2539	GB2 GBAR IN BRPTOB	720	1.0	0.7	1.40	0.5
2541	GB2 GBAR IN GATHERCOSTO	30	0.5	1.4	0.32	1.3
2544	GB2 GBAR IN IOSTREAM WRITE_RECORD	290	0.4	0.4	0.99	0.3
501	MPL SLCOMM2_COMMS PART1	1544	0.2	0.1	1.38	0.1
502	MPL SLCOMM2A_COMMS PART1	2185	0.3	0.2	1.20	0.1
507	MPL TRSTOM_COMMS	3048	3.3	1.7	1.93	0.9
508	MPL TRMTOS_COMMS	3048	3.2	1.7	1.87	0.9
509	MPL SLCOMM1_COMMS	99	0.9	0.7	1.22	0.5
511	MPL SLCOMM2_COMMS PART2	1544	2.3	1.4	1.63	0.8
512	MPL SLCOMM2A_COMMS PART2	2185	6.1	3.7	1.64	2.2
513	MPL MPOBSEQ_COMMS	31	2.4	0.9	2.49	0.4
514	MPL MPOBSEQAD_COMMS	30	2.3	1.3	1.82	0.7
518	MPL CVARBCINAD	30	0.1	0.1	0.98	0.1
524	MPL SCATTER_CTLVEC	1	1.1	0.9	1.26	0.6
525	MPL MULTISCATTER_CTLVEC	1	1.7	1.8	0.95	1.4
602	MPL BRPTOB	1440	4.7	2.7	1.77	1.5
607	MPL COMMSPNORM GATH	260	0.0	0.2	0.20	0.2
609	MPL GATHERBDY GATH	456	0.7	0.7	1.07	0.5
610	MPL GATHERCOSTO	30	0.0	0.5	0.10	0.5
626	MPL BROADCAST GETMINI	1	0.1	0.1	1.10	0.1
635	MPL SUJBWAVELET	106	0.0	0.1	0.00	0.1
636	MPL SUJBWAVALLO MATRICES	8	4.1	4.3	0.96	3.2
637	MPL BROADCAST SUJBBAL	1	0.0	0.1	0.28	0.1
640	MPL SUJBWAVALLO INDXL2G	85	0.1	0.5	0.19	0.5
649	MPL IOSTREAM WRITE_RECORD	290	0.0	0.0	4.00	0.0
650	MPL IOSTREAM READ_RECORD	5	0.1	0.0	3.16	0.0
657	MPL ALLGATHERV IN DOT_PRODUCT_CTLV	1217	0.2	0.2	1.00	0.2
664	MPL BROADCAST IN COMMJBBAL	1	0.1	0.1	1.00	0.1
665	MPL BROADCAST IN COMMFCE2	1	0.2	0.2	0.87	0.1
666	MPL SEND and RECV IN GATHERGOM	2875	0.1	0.3	0.33	0.3
667	MPL BROADCAST IN SUECRAD	1	0.6	1.6	0.35	1.5
691	MPL MKGLOBSTAB	2	0.1	0.1	0.73	0.1

705	GBR GBAR IN MULTISCATTER_CTLVEC	1	0.7	1.2	0.53	1.1
707	GBR GBAR IN IOSTREAM_MIX:IO_INQUIR	3	0.3	0.3	0.99	0.2
714	GBR GBAR IN SCATTER_CTLVEC	1	3.7	4.4	0.84	3.5
725	GBR GBAR IN EVCOST	30	0.3	1.3	0.23	1.2
727	GBR GBAR IN IOSTREAM_MIX:GRID_IN	781	0.3	0.3	0.99	0.2
729	GBR GBAR IN IOSTREAM_MIX:IOSTREAM_	1	0.0	0.1	0.00	0.1
730	GBR BARRIER IN EC_PHYS	25	0.3	0.3	1.30	0.2
731	GBR BARRIER IN EC_PHYS_TL	720	8.6	4.9	1.74	2.8
732	GBR BARRIER IN EC_PHYS_AD	720	19.1	13.1	1.45	8.3
738	GBR GBAR IN IOSTREAM WRITE_RECORD	290	0.6	0.6	0.97	0.5
739	GBR GBAR IN IOSTREAM READ_RECORD	5	0.2	0.1	1.20	0.1
745	GBR GBAR IN IOSTREAM CLOSE	8	3.4	4.3	0.78	3.5
746	GBR GBAR IN DOT_PRODUCT_CTLVEC	1217	4.1	2.2	1.87	1.1
747	GBR GBAR IN DOT_PRODUCT_CTLVEC	30	0.0	0.1	0.00	0.1
748	GBR GBAR IN SLCOMM2 PART2	1544	1.2	1.4	0.83	1.1
749	GBR GBAR IN SLCOMM2A PART2	2185	2.2	2.7	0.79	2.2
755	GBR GBAR IN TASKOBTL	30	5.9	1.9	3.04	0.5
756	GBR GBAR IN TASKOBAD	30	8.1	2.6	3.12	0.6
757	GBR GBAR IN SLCOMM1	73	0.3	0.3	1.10	0.2
758	GBR GBAR IN SLCOMM2A	2185	3.8	3.0	1.27	2.0
759	GBR GBAR IN SLCOMM2 PART1	1544	4.7	5.0	0.95	3.8
760	GBR GBAR IN SLCOMM	26	0.4	0.2	1.99	0.1
761	GBR GBAR IN TRGTOL	2489	9.4	7.1	1.32	4.8
762	GBR GBAR IN TRLTOG	2431	1.5	2.3	0.65	2.0
763	GBR GBAR IN TRLTOM	2308	1.7	2.4	0.72	2.0
764	GBR GBAR IN TRMTOL	2431	2.4	3.0	0.82	2.4
765	GBR GBAR IN TRMTOS	3048	2.0	3.8	0.52	3.3
766	GBR GBAR IN TRSTOM	3048	3.6	5.0	0.72	4.1
767	GBR GBAR IN MPOBSEQ	31	0.1	0.1	1.40	0.1
768	GBR GBAR IN MPOBSEQAD	30	0.2	0.1	1.80	0.1
771	GBR GBAR IN GATHERBDY	456	0.0	0.1	0.00	0.1
772	GBR GBAR IN GATHERGOM	5	0.3	0.1	3.34	0.0
773	GBR GBAR IN COMMSPNORM	260	0.6	0.5	1.21	0.4
775	GBR GBAR IN COMMJBBAL	1	0.4	0.2	2.21	0.1
780	GBR GBAR IN BRPTOB	720	0.7	0.6	1.04	0.5
781	GBR GBAR IN DOT_PRODUCT_CTLVEC	30	0.5	0.2	2.34	0.1
783	BAR BARRIER IN SUTRLE	9	0.2	0.2	0.79	0.2
786	BAR BARRIER IN DIST_GRID_CTL	109	0.0	0.1	0.00	0.1
789	GBR GBAR IN GATH_GRID_CTL	80	0.2	0.1	1.99	0.1
790	GBR GBAR IN DIST_SPEC_CONTROL	109	1.0	1.0	0.99	0.8
791	GBR GBAR IN DIST_GRID_CTL	109	2.1	2.3	0.93	1.7
795	GBR BARRIER IN SUSTAONL	10	0.0	0.7	0.00	0.7
803	MPL TRGTOL_COMMS	2489	7.0	2.8	2.45	1.1
805	MPL TRLTOG_COMMS	2431	7.4	2.9	2.55	1.0
806	MPL TRLTOM_COMMS	2308	5.6	2.3	2.46	0.9
807	MPL TRMTOL_COMMS	2431	6.0	2.4	2.53	0.9
809	MPL GATH_GRID_CTL_COMMS	80	0.0	0.0	1.33	0.0
811	MPL DIST_GRID_CTL_COMMS	109	0.4	0.6	0.63	0.5
812	MPL DIST_SPEC_CONTROL_COMMS	109	0.1	0.0	2.09	0.0
815	MPL GPNORM_TRANS	181	0.0	0.0	0.00	0.0
920	MPL BCAST IN BCASTCOV	1	0.2	0.2	0.99	0.2
921	GBR GBAR IN BCASTCOV	1	1.0	1.1	0.90	0.8

```
Sums in categories for job 1
      Sum for BAR =     0.2  as percent of total    0.0
      Sum for GB2 =    18.4  as percent of total    2.2
      Sum for GBR =    95.9  as percent of total   11.3
      Sum for IO- =     8.8  as percent of total    1.0
      Sum for MPL =    61.7  as percent of total    7.3
      Sum for OMP =   633.6  as percent of total   75.0
      Sum for RES =    29.2  as percent of total    3.5
      Sum for SER =    24.9  as percent of total    3.0


Sums in categories for job 2
      Sum for BAR =     0.3  as percent of total    0.1
      Sum for GB2 =    24.4  as percent of total    6.8
      Sum for GBR =    81.3  as percent of total   22.7
      Sum for IO- =    13.9  as percent of total    3.9
      Sum for MPL =    37.6  as percent of total   10.5
      Sum for OMP =   179.3  as percent of total   50.1
      Sum for RES =    17.2  as percent of total    4.8
      Sum for SER =    21.3  as percent of total    5.9

Total time lost=157.010590277782 s out of 654.530729166665 s hoped gain
Total lost for BAR =     0.3  223.5 % out of     0.1     0.2 %, speedup    0.5
Total lost for GB2 =    19.8  143.8 % out of    13.8    12.6 %, speedup    0.8
Total lost for GBR =    57.3   79.7 % out of    71.9    36.5 %, speedup    1.2
Total lost for IO- =    11.7  176.4 % out of     6.6     7.4 %, speedup    0.6
Total lost for MPL =    22.2   47.9 % out of    46.3    14.1 %, speedup    1.6
Total lost for OMP =    20.9    4.4 % out of   475.2    13.3 %, speedup    3.5
Total lost for RES =     9.9   45.1 % out of    21.9     6.3 %, speedup    1.7
Total lost for SER =    15.0   80.3 % out of    18.7     9.6 %, speedup    1.2
```

References

P. Courtier, J.-N. Thepaut and A. Hollingsworth, 1994. A strategy for operational implementation of 4D-Var, using an incremental approach. *Q.J.R. Meteorol. Soc*., **120**, 1367-1388.

Hamrud, M. Parallel aspects of ECMWF's Integrated Forecasting System (IFS) with special emphasis on data assimilation. In: *Towards Teracomputing, Proceedings of the eighth ECMWF workshop on the use of parallel processors in meteorology,* W. Zwieflhofer and N. Kreitz eds, World Scientific, 1998, 61-66.

Isaksen, L. and S.R.M. Barros. IFS 4D-Variational Analysis - Overview and Parallel Strategy. In: *Coming of Age, Proceedings of the sixth ECMWF workshop on the use of parallel processors in meteorology*, G.-R. Hoffmann and N. Kreitz eds, World Scientific, 1994, 337-251.

Isaksen, L. and M.Hamrud. ECMWF operational forecasting on a distributed memory platform: Analysis. In: *Making its mark, Proceedings of the seventh ECMWF workshop on the use of parallel processors in meteorology*, G.-R. Hoffmann and N. Kreitz eds, World Scientific, 1996, 22-35.

Mozdzynski, G. A new partitioning approach for ECMWF's Integrated Forecasting System (IFS). In: *Proceedings of the twelfth ECMWF workshop on the use of parallel processors in meteorology*, G.Mozdzynski ed, World Scientific, 2006, 148-166.

Rabier,F., H. Jarvinen, E. Klinker, J.-F. Mahfouf and A. Simmons, 2000. The ECMWF operational implementation of four-dimensional variational assimilation. Part I: Experimental results with simplified physics. *Q.J.R. Meteorol. Soc*., 126, 1143-1170.

Saarinen, S. , D. Vasiljevic, H. Jarvinen. Parallelization of observation processing.. In: *Making its mark. Proceedings of the seventh ECMWF workshop on the use of parallel processors in meteorology,* G.-R. Hoffmann and N. Kreitz eds, World Scientific, 1996, 52-67.

Saarinen, S. A distributed database for observation handling. In: *Towards Teracomputing, Proceedings of the eighth ECMWF workshop on the use of parallel processors in meteorology*, W. Zwieflhofer and N. Kreitz eds, World Scientific, 1998, 61-66.