Working group summary report

ECMWF Scalability Workshop, ECMWF Reading, 14-15 April 2014

Peter Bauer[1], Alain Joly[2], Mike Hawkins[1], John Michalakes[3], Deborah Salmond[1], Paul Selwood[4], Stephan Siemen[1], Yannick Trémolet[1], Nils Wedi[1]

[1] ECMWF, Reading UK; [2] Météo-France, Toulouse France; [3] NOAA/NCEP, College Park, MD USA; [4] Met Office, Exeter UK

1. Introduction

In 2013, ECMWF has initiated a Programme on Scalability that aims at developing the next-generation forecasting system addressing the challenges of future exa-scale high-performance computing and data management architectures. The programme relies on external partnerships with numerical weather prediction centres, high performance computing centres, academia and hardware providers.

To help defining the programme and to prepare partnerships ECMWF organised a 2-day workshop on 14-15 April 2014 at ECMWF for which over 50 external participants were registered. The workshop included presentations covering weather and climate science applications at scale, as well as numerical algorithms and hardware/coding aspects.

Following the presentations, three working groups discussed topics inspired by a list of guiding questions (see Appendix 1) aiming to produce recommendations for ECMWF and the community. This report summarizes the main discussions and provides direction for future research and joint projects.

2. Discussion

2.1 General

The general development towards Earth-system modelling at fine scale for both weather and climate science imposes scalability and operability limits on NWP and climate centres that need to be addressed through fundamentally new scientific and technical methods.

For computing, the key figure is the electric power consumption per floating point operation per second (Watts/FLOP/s) while for I/O it is the absolute data volume to archive and the bandwidth available for transferring the data to the archive during production, and dissemination to multiple users. Both aspects are subject to hard limits, i.e. capacity and cost of power, networks and storage, respectively.

The working groups agreed that the urgency of adaptation to highly parallel computing is different for each component of the forecasting system, namely data assimilation, forecasting and data post-processing/archiving. Regarding ECMWF, the working groups recommended keeping the integrated aspect of the IFS alive, which means maintaining the approach of a single model and data assimilation system for all applications as opposed to promoting separate components tailored to forecast range and application.

The participants recognized that, despite ambitious targets being set for model resolution, complexity and ensemble size, today the bulk of the calculations are not performed with configurations that utilize the maximum possible number of processors. Data assimilation, extended range prediction and research experimentation mostly operate at relatively lower resolutions predominantly for technical and affordability reasons. However, the forecast suites always contain a cutting-edge component that fully exploits capabilities.

Firstly, two main data assimilation development streams are being pursued at operational NWP centres, namely long-window 4D-Var and EnVar. Both have scientific and technical advantages and disadvantages (see 2.3) but 4D-Var reaches efficiency limits very soon. Secondly, the next-generation forecast models are being developed now (ICON, Gung-HO, NICAM, GEM), their scientific and computing performance still needs to be established, and they may only be needed in full operations in 10 years. Lastly, I/O limitations will become effective in the short term, for example linked to data bandwidth not growing at the same rate as computing, and data dissemination becoming impossible for large productions in NWP and climate. I/O optimization and the scientific component in data assimilation research should therefore assume very high priority in the Scalability Programme. For climate research and production the task of data assimilation for model initialization is only emerging now. Model integrations are substantially longer than in NWP and need to be completed in a realistic time frame even though there in no critical production path as in NWP. Data storage and dissemination to a large user community are of fundamental concern for climate prediction.

It was recognized that, while scientific choices differ quite substantially between centres, a more coordinated effort to develop common tools can be made and will produce benefits for the community, e.g. regarding libraries, work-flows, or efficiency monitoring tools. This also holds for common developments between NWP and climate prediction communities. While this could be challenging it may offer the only opportunity for co-development with hardware/software providers and for gaining access to external funding. Other areas identified for collaboration were benchmarking, bit-reproducibility versus fault tolerance and common strategies for I/O.

2.2 Workflows

The working groups agreed that there has been little investment in the assessment and optimization of workflows compared to code design. An efficient workflow is important at overall production and scientific component level. Data assimilation, in particular, has a complicated and mostly sequential workflow through the general use of hybrid systems with many inter-dependencies of suites as well as the pre- and post-processing of observations. Workflows fostering data processing by streaming are preferable because they distribute this task better along the critical path without much need for I/O and file management. Intermediate storage can be limited to parts where resilience is needed (e.g. restart files).

At the workshop, the question was raised in how far workflow can be automated and controlled in complex systems. One area of collaboration between NWP centres and with the climate community may be in the use and design of community workflow control and also in procedures for data provision to users. Error resilience is an important requirement here as well.

2.3 Scientific flexibility/choices

The ratio between scientific and technical developments required for enhancing code scalability varies largely between data assimilation, forecast model and data pre-/post-processing. It is also evident that the period for taking full advantage of new model or data assimilation formulations takes up to 10 years. Therefore, investment in the scientific element has to start well before implementation.

For data assimilation, the Environment Canada example demonstrates that a higher-resolution EnVar can be run in shorter time than 4D-Var. Other centres also invest in EnVar, for example Met Office and Météo-France. It was recognized that OOPS will provide the technical framework for testing different algorithms but that it does not buy scalability per se. ECMWF needs to invest more resources in the science of data assimilation, e.g. the formulation of model error crucial for long-window 4D-Var, and EnVar, which should not wait for the full readiness of OOPS. But OOPS will facilitate testing of scalability options within the same code framework.

Regarding ocean modelling a pragmatic choice has been made at ECMWF to eliminate couplers but there is potential to reconsider parallelized solutions. In how far future ocean model and ocean data assimilation efforts will be managed by ECMWF or left to the existing external consortia needs to be decided at a R&D strategic level, before scalability developments can be defined. For example, the Met Office collaborates with Navy, NCAR, Argonne National Laboratory and others to employ ESMF as a core model configuration and coupling tool including a high degree of parallelism. This approach could be taken into consideration to avoid major in-house developments.

It was discussed whether the OOPS example should be extended to the model such that different dynamical cores could be interfaced with flexible top-level structures. The concept of a set of unified equations as proposed by PantaRhei may be a way to address some of this and the generalization of OOPS as a template for the IFS is part of the Scalability Programme.

2.4 Numerical techniques/libraries

Traditionally, the development of numerical methods and libraries are performed at NWP and climate centres since they are considered fundamentally embedded in the entire forecasting system and because flexibility is not the highest priority once the main choices for scientific algorithms have been made.

The workshop presentations and discussions clearly indicated that more flexibility is needed and that the development of shared libraries containing numerous kernels serving many different applications is becoming important. Examples exist such as PETSc, STELLA or LFric but their adaptation to the entire range of currently available system needs to be improved and their suitability more closely assessed. A fundamental advantage of the shared approach is that libraries can be co-developed with HPC-centres and vendors and thus optimization for specific architectures is greatly facilitated. Sharing would also remove pressure from NWP and climate centres to perform all library development in house. For library development emphasis needs to be put on algorithm development though, not only on low-level code optimization and refactoring. ECMWF's non-IFS software strategy envisages ECMWF to directly contribute to open source software projects in the future.

An important question in this context was how far into the future these libraries need to be interfaced with Fortran since most science codes heavily rely on it. Further, the use of libraries requires more general software structures in the code calling libraries to be able to employ families of algorithms rather than today's single forecast application codes.

The working groups proposed to revisit the selection of numerical algorithms and favour those that are compute intensive but require little data communication. Compute intensive algorithms may have been a reason to discard algorithms in the past but may become attractive for implementation on accelerators. Examples are spectral element, finite element and discontinuous Galerkin methods. Parallel-in-time algorithms were discussed but it is not clear at present whether they are useful for complex problems in data assimilation and model integration.

Error resilience was identified as an issue with increasing relevance on exa-scale systems. Fault tolerant algorithms (also MPI) and techniques to compensate for missing calculations and data need to be developed. Failure detection is an important component and has a strong dependence on hardware and compilers. Ensembles are less critically affected since ensemble statistics can be derived from fewer than nominal members. Identification of more/less sensitive parts in the model to (machine) error can be started already.

In the discussions, the role of the IFS as a community model (Arpege, Arome, Hirlam, OpenIFS) received special attention. As for I/O related software development (e.g. grib_api, bufr2odb, see also 2.6) a sufficiently large user and development community already exists. The question in how far future components will be co-developed within this community and code can be managed as open source needs to be addressed.

2.5 Hardware/compilers

Given the expected stagnation of processor frequency and memory the main effort needs to be spent on distributed (load balanced) calculations with as much overlap between computing and communication as possible. This requires, most likely, hybrid architectures comprising CPUs and accelerators aiming at optimizing performance against energy consumption.

It was demonstrated that energy efficiency gains are not easy to achieve with hybrid architectures since CPUs consume 80% of their peak energy when idle and fully exploiting accelerator capacity requires optimal compute-communication configurations.

There are a number of efforts on code refactoring in the community aiming at improving the Watts/FLOP/s ratio for NWP codes or code components. Depending on the available hardware, these are either based on new compiler directives (OpenACC for GPUs) or hybrid methods (MPI/OpenMP for CPUs), compiler options (MIC), the use of language specific features (e.g. PGAS), recoding in hardware specific languages (e.g. CUDA for GPUs). The latter was seen rather critically by some of the working groups except if they are limited to low-level code components (see 2.4).

It was concluded that the most important starting point for efficiency gains is a rigorous and continuous optimization, for example, with MPI/OpenMP, better vectorization, employing more flexible data structures, and cache optimization. Experience from code adaptation efforts showed that code optimized for accelerators also improves performance on CPUs. Additional gain through specific accelerator architecture and code adaptation requires major developments and, with existing technologies, needs to include large parts of the code to be effective in terms of overall Watts/FLOP/s budgets. The recommendation is therefore for ECMWF to continue detailed code analysis before committing to refactoring. NWP model components attractive for refactoring are spectral transforms, selected physical parameterizations (radiation, cloud scheme) as well as radiative transfer calculations in data assimilation.

A source of uncertainty at present is the inconsistent support of directives (e.g. OpenACC, OpenMP4), Fortran language features (e.g. CAF) and vectorization across compilers. Dynamic task parallelism represents an interesting feature but it could impose load imbalance on the calculations. However, limited experience exists with this feature in our community. An important recommendation of the working groups was therefore the formulation of joint NWP-climate community requirements for future compiler development in view of the hardware evolution. Through a 'white paper' representing the NWP-climate community's view, hardware vendors would be encouraged to invest in co-development together with scientists.

Bit-reproducibility received wide attention during the workshop. It was recognized that bit-reproducibility for a fixed processor configuration is of crucial importance for code debugging and operational error tracing, and the only means for distinguishing between hardware differences and code issues. However, reduced or part bit-reproducibility may be crucial for operating on future architectures with acceptable fault tolerance and, e.g., for running large ensembles stably over long time periods. The working groups encouraged the formation of a working group that continues the discussion of this topic in strong collaboration with hardware providers.

2.6 I/O

The participants agreed that I/O scalability issues are much less prominently exposed than computing issues. It was proposed to invest in internationally agreed rankings of I/O systems similar to top500 lists for HPC. NWP and climate communities can support this by defining a unified set of requirements including benchmarks employing realistic sets of NWP and climate model output.

There are several categories of issues related to I/O, namely data volume and the number for files that need to be communicated between tasks as well as archived, the time allocated to I/O along the critical path and the diversity of data, e.g. ranging from the large set of observational data to direct model output and post-processed information. At ECMWF, archive contents grow exponentially and 1/3 of this growth is related to resolution increase and 2/3 to increasing diversity of product types. While the growth of observational data is expected to stabilize after the launch of spectrometers on geostationary satellites, model data output will not, in particular due to the increasing size and complexity of ensembles. It was also demonstrated that recomputing instead of storing does not present a viable alternative due to cost in most cases.

The groups recommended investigating the need for storing model output at its native resolution at which the true information content (degrees of freedom) is likely to be small. Ensemble output can be reduced to information characterizing the ensemble probability distribution rather than each ensemble member's output.

Price and capacity of storage media are still expected to decrease but storage cost increases faster than compute cost. The more data needs to be stored and managed on disks the more complex the task to manage error resilience becomes (e.g. RAID systems). However, bandwidth represents potentially the most important limitation at present. Again (see 2.2), enhancing data streaming during production can help distributing the I/O workload. A sub-aspect of streaming is to abandon storing and distributing model output as global fields but to localize this task as it is computed.

Data formats and data compression need to be addressed by the community as well. Fewer formats and accepted standards of compression can alleviate many bottlenecks when model output and observational data are shared. WMO standards (BUFR and GRIB) may show the way; however, acceptance procedures are slow.

The working groups suggested investing in shared projects on CDI-PIO, XIOS, ADIOS, SSDalloc. ECMWF developments such as grib_api, bufr2odb and odb should turn into community projects, at least among partners already sharing IFS components (see also 2.4). In this context, the lack of parallelism in the initialization of grib_api was mentioned as a specific problem.

2.7 Benchmarking

Various metrics exist for performance assessment. For production total runtime is important and thus entails the evaluation of tasks along the critical path. For hardware architecture assessment the total power envelope is important. CPU-accelerator comparisons should focus on socket-socket or node-node (e.g. 2 CPUs vs 1 CPU + 1 GPU).

The employment of profiling tools is crucial for code optimisation, and this needs to be part of benchmarking exercises. A number of tools developed at HPC centres exist (e.g. Vampir TU Dresden, Paraver BSC) and their applicability for the IFS should be tested.

As in Section 2.6, the benchmarking of I/O code components needs to be more systematically addressed and be considered a core element in future procurements.

An important message is that for NWP the ultimate benchmark remains wall-clock time.

3. Main recommendations

- It is important to maintain the integrated forecasting system approach serving the range of applications run at ECMWF. Code adaptation to the evolving hardware should be performed keeping portability in mind. ECMWF should invest in:
    - identifying code components that promise compute/energy savings when optimized for specific hardware solutions;
    - testing solutions in collaboration with HPC centres and vendors giving access to such hardware.
- The development of next-generation data assimilation and forecast models coincides with scalability challenges. The diversity of approaches pursued at different NWP centres favours code implementations that implement a high degree of algorithmic flexibility. Concepts such as OOPS and PantaRhei are promising and need to be extended.
- Common and shared development of software should be pursued with partners in NWP and climate. Examples are:
    - assessing compute intensive numerical methods with a low level of data movement;
    - developing data structures supporting flexibility for horizontal/vertical discretization;
    - implementing profiling and optimization tools for workflows;

    - developing software optimizing data formats and I/O management;
    - developing data compression options, e.g., from assessment of information content of (ensemble) forecasts,
    - benchmarking core components (forecast model, physics, data assimilation, I/O);
    - assessing fault-aware software and the requirement for bit reproducibility.
- I/O performance is expected to become insufficient in the near future, of which the bandwidth available for data transfer between suites as well as between computing and archiving is a key element. Benchmarking community standards need to be defined and become more strongly emphasized in procurements. Archive growth rates are mostly driven by product diversity and thus model output and product generation strategies need to be examined.

The participants further recommended community actions, namely:

- to establish a working group dealing with the issues of bit-reproducibility and fault tolerance;
- to produce a 'White Paper' defining the NWP requirements for future compiler development addressing HPC scalability, in collaboration with hardware vendors.

Appendix 1: Working group questions

List of guiding questions distributed to working groups.

1. General:
    a. Is the opportunity of Exa-scale computing power fundamentally changing the way we do NWP forecast and analysis? If yes, at what anticipated time-scale do we expect the change?
    b. What are common components of the NWP system that may be shared between ECMWF, other centres, the Climate community, regional applications?
    c. What should be the European approach to strengthen industry, HPC centre, science, application chain?
    d. Which partnerships will optimize funding opportunities in Horizon 2020?

2. Workflows:
    a. What are the main bottlenecks in current workflows:
        i. Climate: long time series, NWP: critical-path production and dissemination schedules?
        ii. Analysis, forecasts, ensembles?
        iii. Pre-/post-processing?

3. Scientific flexibility/choices:
    a. Which are the priorities between complexity, resolution, ensembles given scalability limitations?
    b. What are common components to be shared between centres and disciplines; what is the role of libraries such as PETSC?

4. Numerical techniques/libraries:
    a. What are examples of emerging algorithms and procedures that are fundamentally different to what we do and explore today?
    b. What defines a "power-aware" numerical algorithm?
    c. What are examples of "fault-aware" numerical algorithms?
    d. What is the trade-off between accuracy and energy efficiency (e.g. double vs single precision)?

5. Hardware/compilers:
    a. What are hardware requirements of big-business users driving co-design and where are similarities with weather/climate?
    b. What are the defining characteristics of the (exa-scale) future computers that affect developers? What can be done up to critical power limit 10-20 MW?
    c. How important is bit-reproducibility at hardware and software level?
    d. Should the emphasis be on compiler rather than hardware co-design?
    e. Is it preferable to employ new languages or compiler directives?
    f. What are the future projections for memory/core and memory/node in 2020 and 2030?
    g. How can dynamic task parallelism be effectively used in an operational NWP environment?
    h. How will the other components of an exa-scale system cope, e.g. Operating System, resource scheduler, workflow management, file system?

6. I/O:
    a. Are we prepared for the data and I/O expected from high-resolution simulations

(e.g., a standard 10-day forecast at 2.5 km resolution produces ~ 20 Terabytes)? Can these be supported by the hardware and file systems?

   b. Can we characterize our needs? Are they different from other exa-scale applications? What are the performance characteristics and access patterns?
   c. What needs to be archived/disseminated, what can be post-processed on the fly or recalculated?
   d. Is there a community approach to data compression?

7. Benchmarking:
   a. What should be the metrics for CPU-accelerator comparison: e.g. node-node, socket-socket?
   b. How do models and algorithms compare when forecast days per Watt is used as a metric?
   c. How can error resilience best be addressed? Is this only a question for operating system and hardware developers?
   d. Which components of the workflow should be benchmarked separately and how?
   e. Which code profiling tools are easy to implement and most profitable?
   f. Can I/O performance benchmarking be shifted from measuring the peak performance to sustained performance in the same way that code performance is benchmarked?