WG2: Report

General Questions

a. Fundamental changes to Analysis and model?

- 2 directions in data assimilation:
    - ECMWF 4DVar (long window, weak constraint etc.)
    - EnVar by many centres

    → Both have advantages and drawbacks
- Model: can adapt progressively

    → Evolution not Revolution

b. Sharing more NWP components

- Already sharing of software like grib_api and odb as well as European-wide use of models close to IFS.
- Recommendation – to build a community to co-develop tools like grib_api, bufr2odb, etc. This requires to use the experience of co-development of the main IFS.

c and d

- The group took good note of the opportunities and constraints for preparing a project for funding within Horizon 2020

Recode or adapt IFS

- Examples from other groups
  - Environment Canada
  - DWD, MPI and Meteo Swiss
  - Met Office
- 5-10 years effort which should be driven by science as well as technology
- Try to continue OOPS and extend some of its ideas to interface alternative dynamical cores in IFS
- No reason to break the ‘Integration’ of IFS

Bit-reproducibility

- Two aspects
  - Same configuration
  - Different configuration (nproma, MPI tasks etc.)
- Only way to separate application errors from compiler and hardware problems
- Small dedicated working group from IFS developers and vendors to discuss in detail

I/O

- Whole I/O workflow would benefit from well though coding and data organization

→ Improve WMO standards – ECMWF and member states may show the way.

- System level aspects that could be addressed by associating application and operating system developers
- Maybe we should follow the example of other communities (astronomy, particle physics) – by defining a unified set of requirements (including new I/O benchmarks)
- At ECMWF the initialization of grib_api should be re-written so it can run at scale

Other points

- Re-visit old Algorithms?
- Fault tolerance becoming more important
- Continuous optimisation of IFS for CPUs (many aspects: data layout, reduce copies, memory efficiency, vectorisation etc.)
- Software architecture allowing for families of algorithms rather than just one

Code Adaptation to new architectures

- Should we re-write IFS or ‘adapt’ it?
  - Is now the right time?
  - Can we put part of it on accelerators – and/or run Hybrid?
  - Can we have ‘portability’ between CPU, GPU, MIC etc
  - Is there any other we should consider?
  - Role of OpenACC

- What language?
  - Do we need DSL?
  - Fortran with C++ at control level like OOPS?

What can we do to improve the existing code to eventually get to ‘many core’?

- Optimisation for CPUs
  - data layout more flexible
  - memory copies, recalculate and not store.
  - better vectorisation
  - better OpenMP
  - FLOPs per Byte accessed from memory
  - Cache optimisation
  - Instrumentation and profiling tools
  - CAF – pretty much limited to Cray?

- How do we get past the fact T255 doesn’t have enough work to distribute?