KMI - IRM

CMAKE COMPILATION in IAL-BUNDLE

Haumont Denis (RMI), Daan Degrauwe (RMI),
Judicael Grasset (MF), Willem Deconinck (ECMWF)

ACCORD | 5th All Staff Workshop 02.04.2025

1. IAL-BUNDLE

INTRODUCTION

- A **bundle** gathers information on versions of various projects to be assembled in sight of the build of executables, in this case IAL executables
- **Our goal**: Exploit capabilities of ecbundle and ecbuild to drive a **CMake**-based build system for IAL in IAL-Bundle
- In the frame of **DEODE**
- **Collaboration** between RMI, MeteoFrance, ECMWF

MOTIVATIONS

- **Share** build configurations between DEODE **teams**
- Share compilation infrastructure with **ECMWF**
- Bundling facilitates **externalisation** of projects

  (oops, ectrans, phyex, surfex, fiat, atlas, ...)

- Rely on industry **standards** for compilation (maintenance, documentation, CI/CD tools)
- Ensure **portability** on our various HPC's
- **Faster** compilation
- Ensure results **reproducibility**

IAL-BUNDLE PRESENTATION

- IAL-bundle is based on ecbundle from ECMWF

  IAL-bundle is at **ifs-bundle** what 49T1 is at 49R1

  Allows to compiles all our CSC's and Arpege

- ACCORD-NWP/IAL_bundle repository
  - CY49: definition of the IAL bundle only (*)
  - CY50T1: added architecture files and compilation scripts for CMake

(*) Note: The IAL-build can be used for compiling IAL-bundle using GMKPACK

github.com/ACCORD-NWP/IAL-bundle

ANATOMY OF AN IAL-BUNDLE

- One **branch** in IAL-bundle for each **IAL version**
- **bundle.yaml**
  - Contains the repository and a reference (commit, branch or tag) for each project
  - YAML format (file format from ecbundle and shared with ECMWF)
- **ial-bundle:** a driver script to trigger the CMake compilation in ecbundle
- **Architecture folders (like in GMKPACK)**
  - Contains files defining modules and compilation options
  - One folder per platform and per compiler version
  - Can be shared with ECMWF
- **For each project included in the bundle**
  - The CMake compilation script CMakeLists.txt
  - Each project is agnostic of the bundle, and can be compiled in an independent way using **CMake**

IAL-BUNDLE IN PRACTICE

```
$ git clone https://github.com/dhaumont/IAL-bundle

$ git checkout <cycle>

$ ial-bundle create

$ ial-bundle compile –arch <architecture_folder>
```

<cycle> One branch for each version

ial-bundle Bash script driver to ecbundle

- **ial-bundle create** - create *source* folder
  - git clone projects
  - run cmake configuration
- **ial-bundle compile** - create *build* folder
  - compilation all projects

<architecture_folder> platform and compiler settings

See documentation [IAL-bundle/doc/cmake_compilation.md]

IAL-BUNDLE FOR CY50T1

```
$ git clone https://github.com/dhaumont/IAL-bundle

$ git checkout CY50T1

$ ial-bundle create

$ ial-bundle compile --arch \

    \ arch/ecmwf/hpc2020/intel/2023.2.0/hpcx-openmpi/2.9.0
```

*Full setup and compilation of IAL **CY50T1** on **ATOS-Bologna** (hpc2020), **Intel Compiler 2023.2.0***

2. CMAKE

CMAKE OVERVIEW

- De facto industry standard compilation tools

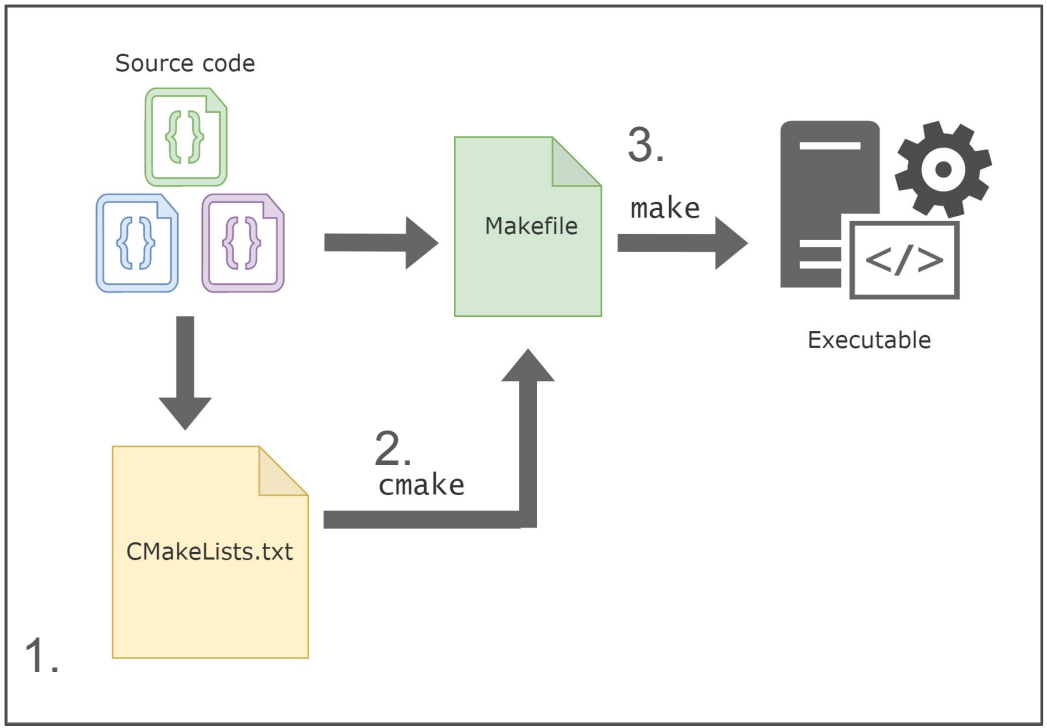
  (2 millions downloads/month)

- Kitware Inc.
- Large open-source community
- Portable
- Well documented

cmake.org

CMAKE - CRASH COURSE

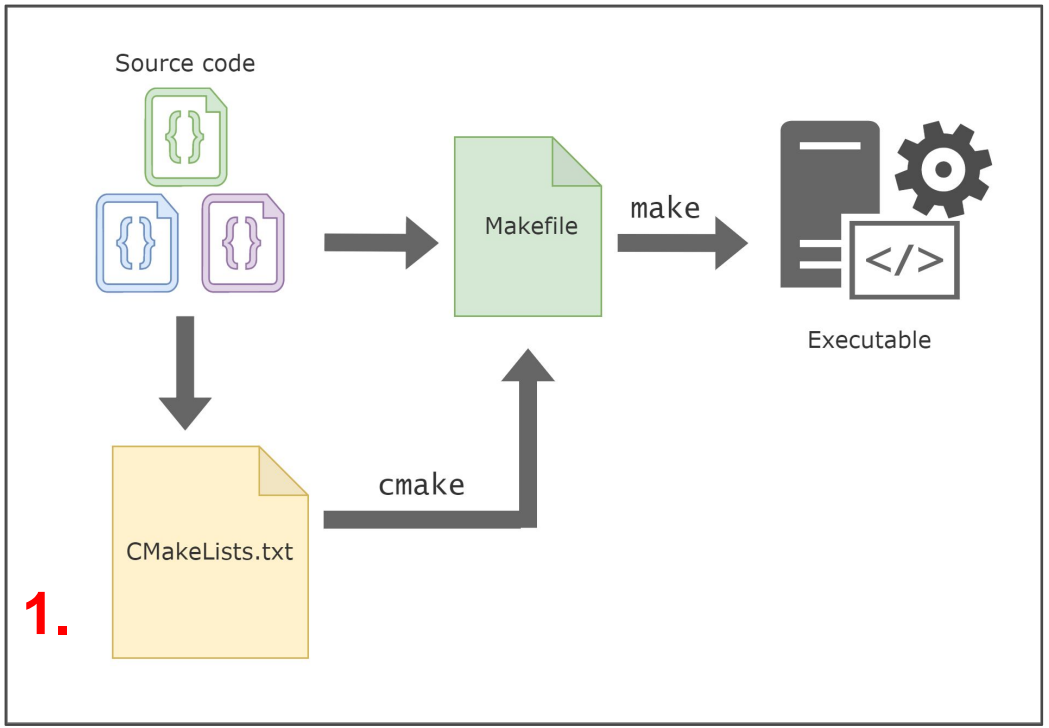
1. Definition of CMakeLists.txt
2. CMake
3. Compilation



CMAKE - CRASH COURSE

1. Definition of CMakeLists.txt

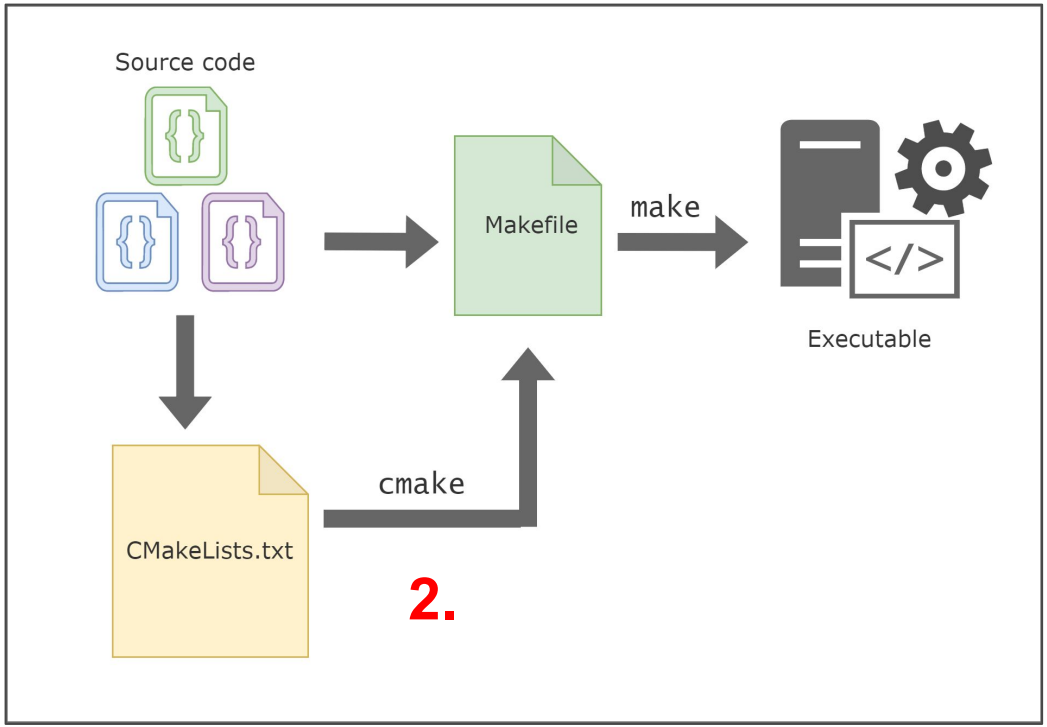
- **CMake build** configuration files
- Code split into modules
  (called targets in CMake)
- Target dependencies definition



CMAKE - CRASH COURSE

2. CMake

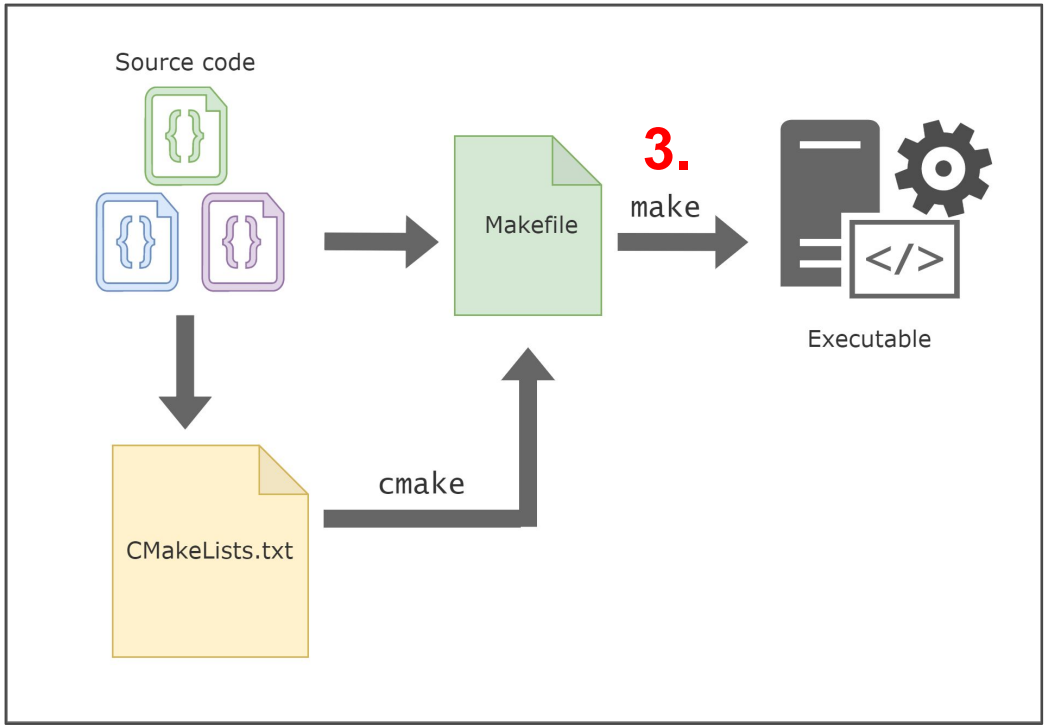
Generation of a single Makefile for the whole bundle combining the CMake files of the bundle projects



CMAKE - CRASH COURSE

3. Compilation

All files of all projects are compiled in **parallel** (fast!)



CMAKE TECHNICALITIES

- Accurate dependencies and minimal rebuilds
- Out-of-source build (*.o in a separated directory than *.F90)
- Cross-platform
- CMake allows to use different build systems
  - make | www.gnu.org/software/make/
  - ninja | ninja-build.org/

ECBUILD TOOLBOX

CMakeLists.txt language: “low-level” and verbose

⇨ “Dialects” developed in our community to reduce boilerplate code

- CMake-Hirlam (HIRLAM): used in Harmonie-Arome and Deode
- ecbuild (ECMWF): used in all ECMWF repositories

⇨ We target is **ecbuild**

github.com/ecmwf/ecbuild

CMAKE for IAL - CONTRIBUTIONS

Adaptation and consolidation of CMake scripts from Hirlam and ECMWF

- Translate the compilation scripts from CMake-Hirlam into ecbuild “dialect”
- Extend the compilation scripts of IFS to handle also IAL
- Make use of the build infrastructure from ECMWF (ecbundle, ecbuild)
- Resolve the circular dependencies introduced in CY50
- Test and validate on different platforms and compilers

PULL REQUESTS (Not merged yet)

- IAL | github.com/ACCORD-NWP/IAL/pull/314
- IAL-BUNDLE | github.com/ACCORD-NWP/IAL-bundle/pull/1

3. RESULTS AND DISCUSSION

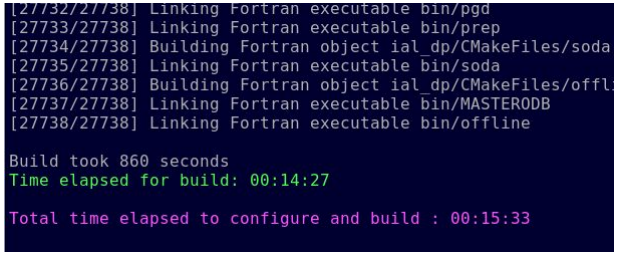
RESULTS (1/2)

- IFS and IAL compiled with the same CMake Files
- Good portability
  - HPC: Belenos (MeteoFrance), Atos_bologna (ECMWF), LUMI, Leonardo
  - Compilers: Intel, NVidia, Cray
- Fast compilation times (full parallel compilation)
  - Full build (including external dependencies)

    ~15 min

  - Re-build, after one file modification

    ~2 min



Intel 2023.0.2, Atos-Bologna (SSD)

RESULTS (2/2)

+

- Out-of-source build
- Easy to manage external packages
- **GMKPACK coexist with CMake**

-

- Dependencies between CMake targets must be defined by hand

  Only once, new contributions must respect existing dependency structure

- Circular dependencies must be eliminated

DISCUSSION POINTS

CIRCULAR DEPENDENCIES

- IAL: arpifs is split into small sub-modules
- IFS: some sources of other directories are included in arpifs (and vice-versa)

INTERFACE FILE GENERATION

- IAL: based on a Perl script (compatible with fypp)
- IFS: based on FCM

4. CONCLUSIONS

IAL-bundle is a **CMake**, future-proof, fast and portable build system for **IAL**

🚀 **Give it a try!**

FUTURE WORK

- More CMake script code **in common** with ECMWF
- **Remove circular** dependencies in Fortran code
- **Incremental pack** (like root pack and local pack in GMKPACK)
- Integration of **source-to-source** translation for GPU porting
- Porting to other **platforms and compilers**
- Integration in **DAVAI** and ACCORD **CI/CD**

KUDOS

- **Yurii Batrak** from METNO, who developed CMake-Hirlam, with the help of Roel Stappers and Ole Vignes

- **Willem Deconinck**, **Olivier Marsden** and collaborators from ECMWF, for the development of ecbuild, ecbundle, and the IFS's CMake files

- All our **colleagues from DEODE** who co-developed IAL-bundle, and who will recognize themselves

QUESTIONS ?

Thanks for your attention!

KMI - IRM

IAL-BUNDLE TL;DR (*)

(*) Too Long; Did not Read

Full setup and compilation of IAL CY50T1 on ATOS-Bologna

```
$ git clone https://github.com/dhaumont/IAL-bundle

$ git checkout CY50T1

$ ial-bundle create

$ ial-bundle compile --arch \

   \ arch/ecmwf/hpc2020/intel/2023.2.0/hpcx-openmpi/2.9.0
```