Atmospheric Modelling and HPC

Graziano Giuliani
ICTP – ESP
Trieste, 1 October 2015

NWP and Climate Codes

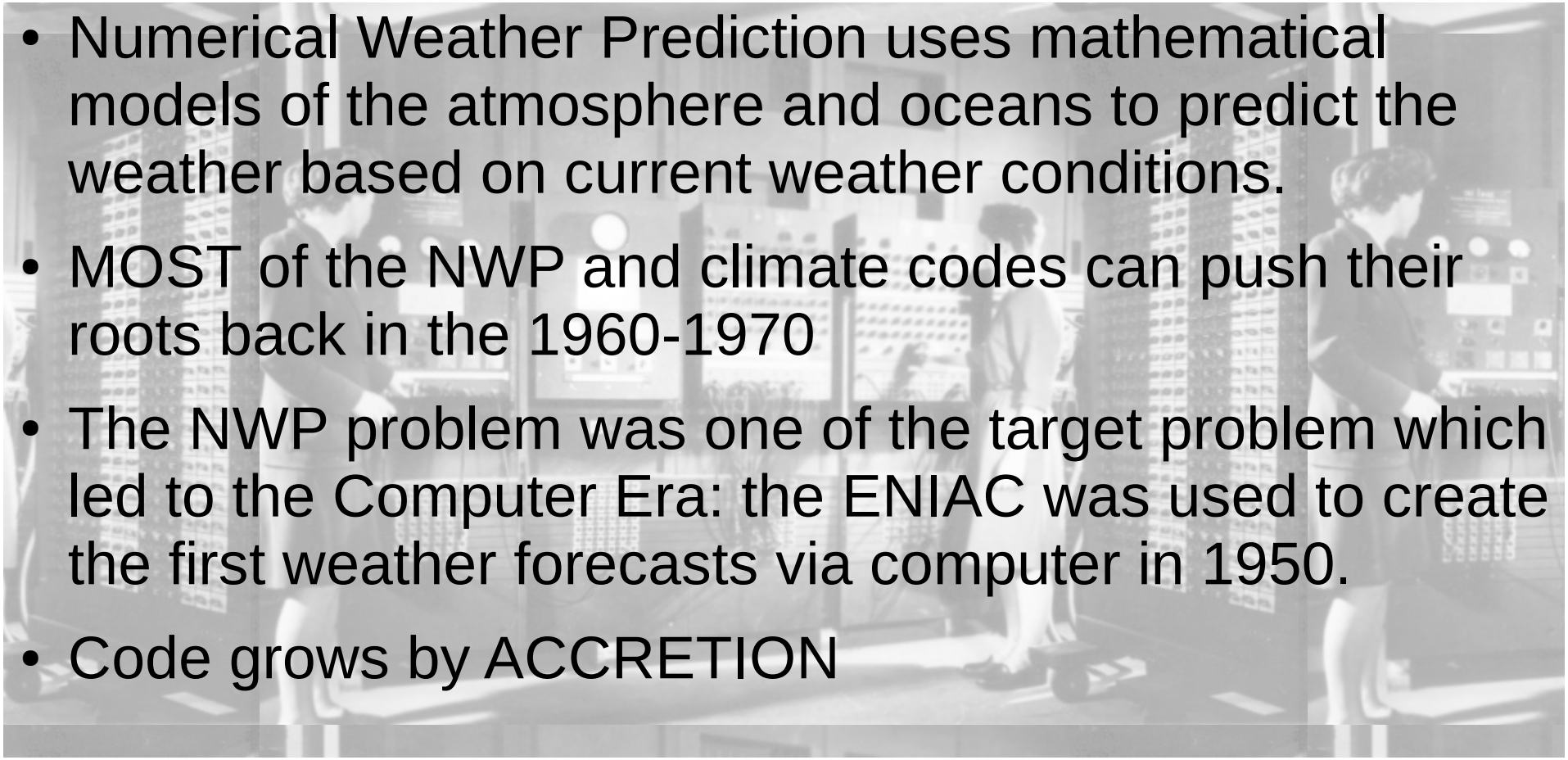
- Legacy code from the '60 : FORTRAN language
- Path and Dynamic Libraries
- Communication Libraries
- I/O format Libraries
- Data analysis tools and the data deluge

http://clima-dods.ictp.it/Workshops/smr2761

Materials

- This presentation:
  - `atmospheric_models_and_hpc.pdf`
- Codes:
  - `codes.tar.gz`
- Download both, uncompress codes:
  - `cp codes.tar.gz /scratch`
  - `cd /scratch`
  - `tar zxvf codes.tar.gz`

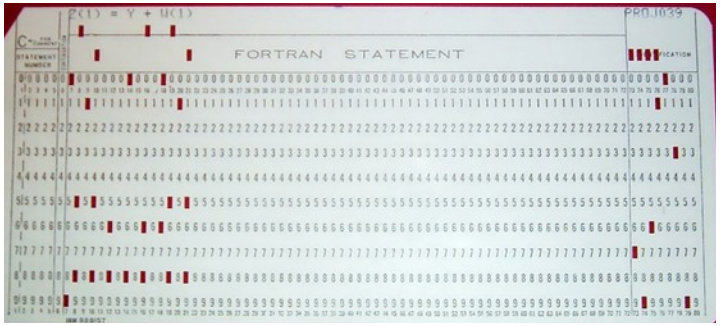
Legacy code



- Numerical Weather Prediction uses mathematical models of the atmosphere and oceans to predict the weather based on current weather conditions.
- MOST of the NWP and climate codes can push their roots back in the 1960-1970
- The NWP problem was one of the target problem which led to the Computer Era: the ENIAC was used to create the first weather forecasts via computer in 1950.
- Code grows by ACCRETION

FORTRAN and FortranXX

- Fortran Programming Language is a general-purpose, imperative programming language that is especially suited to numeric computation and scientific computing.
- Originally developed by IBM in the 1950s for scientific and engineering applications



- Its standard is controlled by an ISO comitee: the most recent standard, ISO/IEC 1539-1:2010, informally known as Fortran 2008, was approved in September 2010.
- Needs a compiler to translate code into executable

Fortran Compilers

- http://fortranwiki.org
- Most of the High Performance Compilers are Commercial, and “SuperComputer” vendors usually provide their HIGH OPTIMIZED version of Fortran Compiler.
- Some part of the Standard are left “COMPILER SPECIFIC”, opening for incompatibility among the binary formats produced by different compilers.

The Fortran .mod files

- Fortran 90 introduced into the language the modules:

```
module my_mod
  contains
    subroutine mysub(a,b)
      real , intent(in) :: a
      real , intent(out) :: b
    end subroutine mysub
 end module my_mod
```

- The compiler creates the object file and a .mod file compiling the above code. Program uses the above with

```
use mymod , only : mysub
call mysub(a,b)
```

The .mod file

- The MOD file is similar to a pre-compiled header in C.
- Each compiler creates its vendor specific format
- Cannot mix
- Requires to use ONE compiler for the whole chain. Compiler specific modules and libraries MUST be kept separate.
- Virtual environment – modules

The SHELL environment

- After checking USER credentials and privileges, the login program run on success a shell for the user
- A shell is a programmable program which receives user input and can execute system program on the user behalf in a user environment.
- The environment has pre-defined variables, and support defining more user variables
- Run a terminal and type
  - `env`

PATH environment variable

- The Operating System searches binary executable programs in pre-defined or configurable in the shell paths.
- Examine the output of the env program.
  - What is the shell path to find executables ?
  - How to change it ?

Dynamic / Static Executable

- Linking phase means taking one or more objects generated by compilers and assemble them into a single executable program
- An executable in a modern OS platform can be:
  - Static : a static build is a compiled version of a program which has been linked against all libraries and does not contains internally unresolved symbols.
  - Dynamic : linking is performed while a program is being loaded (load time) or executed (run time), rather than when the executable file is created and the executable contains unresolved symbols
- `ldd <exec_name>`
- `readelf -a <exec_name>`

Static linked exec

- Pros:
  - Portable : does not dipend on target system library versions
  - Ready to run anywhere given the OS
- Cons:
  - Bigger size of the executable
  - Need recompile if library upgrade
  - Memory consuming

Dynamical link exec

- Pros:
  - Smaller size of executable
  - Upgrade of a library without changing API does not require recompile
  - Memory sharing of library
- Cons:
  - Not directly portable, shared libraries with correct version must come along

Locating Libraries at Run Time

- System specific path, library name, library version number
- On Linux:
  - System paths:
    - `/lib , /usr/lib , /usr/local/lib`
    - `ld.so.config`
  - Compile Time path:
    - `-Wl,-rpath,/search/please/here`
  - Later modification with the `chrpath` program
  - Runtime variables:
    - `LD_LIBRARY_PATH=/here/before:$LD_LIBRARY_PATH`
    - `LD_PRELOAD=/path/to/a/lib.so program`

Static and dynamic libs

- Static libraries are usually ar archives:
  - `ar -rv *.o libmylib.a`
- Dynamic library are ELF executables:
  - `gcc -shared -o libmylib.so *.o`
- http://www.ibm.com/developerworks/library/l-dynamic-libraries

The Stencil and the Communication

- Fluid dynamic model equations contains BOTH 3D space and time derivatives.
- Stencils are the basis for many algorithms to numerically solve partial differential equations (PDE).
- Stencil code problem decomposition requires Communication Library
- Message Passing Interface - MPI

MPI implementation

- Message Passing Interface (MPI) is a standardized and portable message-passing system designed to function on a wide variety of parallel computers
- There are several well-tested and efficient implementations of MPI which follow a STANDARD: latest is MPI-3
- Most NWP and Climate codes require MPI-2

MPI Libraries

- If not Hardware Vendor Provided, some Free Software / Open Source implementation are availble
  - https://www.open-mpi.org
- Some atmospheric or ocean models “prefer” the MPICH implementation.
  - https://www.mpich.org

I/O format libraries

- In the Ocean, Atmosphere and Climate communities, the most used file format is the netCDF.
- It is a binary indexed format which contains BOTH data AND metadata (data about the data).
- Possible other requirement may be the GRIB IO library, but must of the model ship their own implementation for this or require for GRIB2 standard Linux libraries like jasper and png.

netCDF file format

- It has different versions. Recent Climate framework requires netCDF4 file format.
- NetCDF4 file format is HDF5 file format, so you will need also the HDF5 library
- HDF5 file format supports transparent compression, so you need also zlib library
- https://www.unidata.ucar.edu/software/netcdf

Standard Linux System

- We are lucky:
  - Almost any decent Linux System has the required libraries compiled and with the correct dependency chain. For example on *buntu-like :

```
apt-get install netcdf-bin libnetcdf-dev libopenmpi-dev
```

Custom Linux System
Other Compiler

- You must compile all the libraries !
- USER usually expect model code to work out of the box, and rely on “system” software to be JUST AVAILABLE.
- SYSADMIN usually do not want to MAINTAIN complex tool chains.
- Who is going to win ?
- Is the USER or the SYSADMIN to install this or that piece of software?
- Compromise: IF JUST ONE USER, he can go by himself. The overhead may be high, so unless he is the DEAN... BUT MORE THAN ONE, the SYSADMIN should take care. If same piece is needed twice, it will easily be needed three times. A happy user post LESS tickets.

Dependency chain

- Netcdf requires HDF5
- HDF5 requires zlib
- Zlib does not have any other dependency.

Which will you compile first ?

How to make them find each other ?

RegCM prereq_install script

- The RegCM model developed in ICTP is distributed with a simple script that helps users compile their toolchain.
- This is for the USER with simple heavy workstation to run the model on.
- System administrator of a HPC Linux cluster ususally reserve the MPI layer at system level.
- The netCDF library supports Parallel I/O, but the System HW MUST support it !

Let us examine this script.

- http://clima-dods.ictp.it/Workshops/smr2761

Analysis tools

- Most of the tools used by the ESP community require netcdf format support.
- NCO
- CDO
- Python netCDF4
- Ferret
- R
- NCL

Model run vs Data analysis

- Different requirements!
  - Model : run the model, parallel platform, high throughput and big storage access for output data
  - Analysis : Use visualization tools, distille all the data in a simple x,y plot or fancy colour maps
- Analysis tools STILL MOSTLY lack parallel processing capabilities
- USER has not clear in mind from the beginning the expected workflow in data analysis
- Visualization tools require graphic resources
- Data analysis tools tend to use latest environments

Data Deluge

- NWP models usually require heavy computation for a limited period of time, and produce data with a fixed rate. But NEVER stop, and require data produced usually to be stored permanently. A new model or new data usually triggers a reanalysis.
- Climate model usually require heavy computation resources for longer time, produce BIG data in short timeframes, and need to store this data for at least 10 years. But required storage grows exponentially every five years (IPCC report deadlines)

Solution ?

- Still not available...