Lowering the “Cost of Entry” to using the UFS

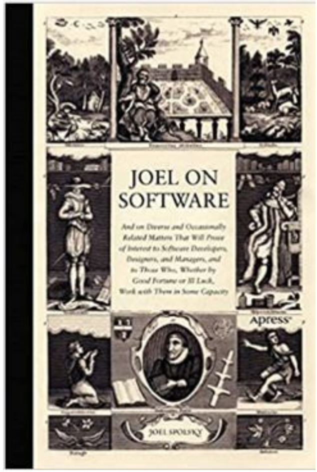
Louis J. Wicker NOAA/OAR/NSSL
Curtis Alexander NOAA/OAR/GSL
Jacob Carley, NOAA/NWS/NCEP/EMC
Christiane Jablonowski University of Michigan

Purpose of this talk is to think outside the box and generate discussion…

Thanks to Patrick Burke@NSSL for his suggestions and help with talk

“Cost of Entry”?

- From *Joel on Software:* A story of Microsoft Excel vs. Lotus 1-2-3
- late 1980’s: Lotus 1-2-3 was THE spreadsheet software
- Bill Gates was building Microsoft Excel (and Windows)
  - How to get people to try Excel?
  - build a converter for Lotus 1-2-3 to be imported into Excel…and…
  - build a converter to import Excel back into Lotus 1-2-3!
  - one could do work in Excel **and if one did not like it**, convert back!
- Why provide both conversions?



It lowered the cost to TRY EXCEL! More people tried Excel; End result: selling Windows 2.x!

- Does the UFS “cost” too much too try? Maybe…

Current UFS

- UFS success story: the system can run on NOAA HPC, singularity containers, and on Mac and Linux. ***(thank you all for that!)***
  - How long can software be supported across this many platforms?
  - How much resource will be devoted to help desks, debugging, answering questions etc?

- While the concept behind UFS is to simplify and unite modeling, UFS is currently a complex system from user perspective
  - Fully coupled model provided
  - Keeping close to the operational version is a worthy goal - is that the only goal?
  - Need balance between complexity of the full system versus accessibility to people without NOAA HPC access.

- In academia, many people don't need the full system for:
  - teaching / testing / fully **coupled model**
  - many types of research for NWP, data assimilation, or process studies.

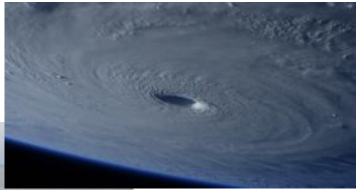
Should UFS consider a "Lite" version ?

Pros of "UFS-Lite"

- Reduce cost of user support mechanisms
- Broaden community use: this pays off!
- UFS needs a user base similar to that which WRF and MPAS have gathered through years of interaction
- UFS-lite could be used to get community to on specific issues by releasing "targeted" versions
- Lite version helps driver ML applications?

Primary Concern with "UFS-Lite"?

- How would user support would there be for both a lite and full system?
- Concern over divergence of code bases; however, there are ways to maintain links.
- Would this effort "work"?
- Need more info (community survery?)



What do we mean by “Lite?”[1]

Model System	Libraries
UFS	50*
WRF	~ 10
MPAS	10

* Includes 23 EMC libraries and ESMF

These libraries must be included before the UFS SRW or MRW app can be compiled

Model System	Files to Compile	Directories
UFS SRW	1400	1100
UFS HPC	27,000	1900
WRF	9000	262

Many of these UFS files are NCEP I/O and conversion libs that are not needed for research and development - netCDF is standard there.

[1] *A “lite” system is less filling, but still works great!*

GFDL Solo Code: already a UFS-lite?

GFDL Solo

GFDL's open source FV3 code for atmos. E.g., model driver routine is 680 lines (instead of ~ 3000 for SHiELD or UFS)

Pros of GFDL Solo

- Simplified FV3+physics system
- uses FMS and netCDF
- Fewer than 2000 files, 150 directories
- Global model but can be used for simple CAM tests
- Physics choices are limited, but Solo could be amenable to physics suite sets

Still less than ideal

- IPD, not CCPP
- While number of directories is low, they are organized as a deep hierarchy
- Build and source directories are independent
- No visualization tools, verification, etc.
- Other details (e.g., Cannot dump out history file at T=0)

Bottom Line: GFDL Solo is an example of how a UFS-lite could be designed…

UFS-lite Strawman

NWP Capabilities:
- Global, regional, local,
- Idealized tests available for global and CAM scales
- Hurricane test case?

Main Priorities:
- Ease of use (remember Joel on Software!)
- File I/O flexibility
- Ability to restart from GFS and RRFS files
- **Provide interface with JEDI**

Note: ease of use and portability, not computational speed, are main priorities.

Other Thoughts:
- Combine and flatten directory structures
- NetCDF files should be COARDs compliant
- Borrow from WRF concepts: multiple output
- streams are available for different fields and output geometry (soundings, etc.)
- Emphasis on providing UFS via containers (docker, singularity, etc.) not raw hardware
- More cloud support!

Questions:

Include nesting capability?
Drop NCEP and ESMF libraries?
CCPP or not?
How do we decide what makes sense?

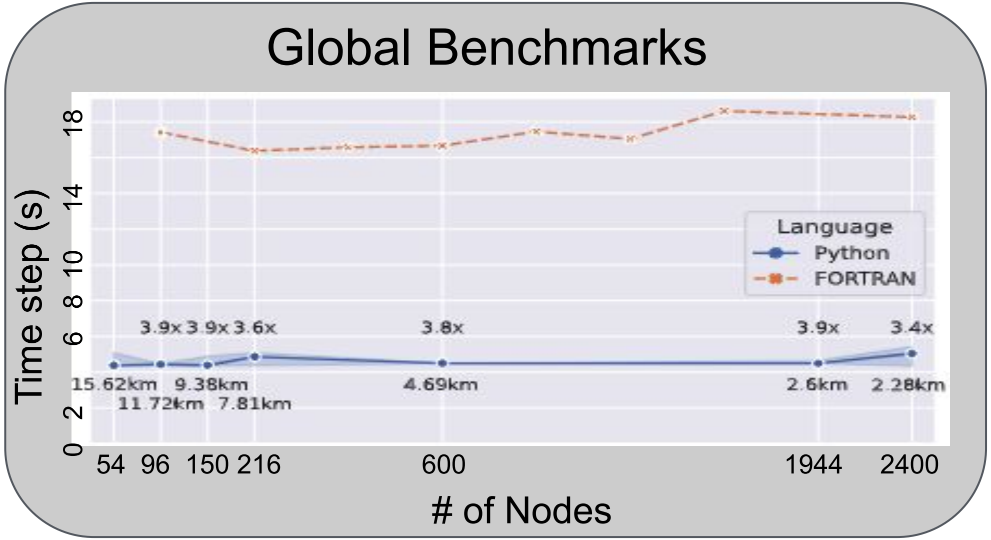
Looking Ahead…

- FV3 dycore employs a climate model approach to its formulation, particularly for mass, thermodynamic and moisture variables.
  - This approach does not translate easily to use at CAM scales
  - use of full condensate density and pressure is complicated, especially for physics development.
  - FV3 is an elegant formulation but very complicated to learn, particularly when documentation is minimal.
- A dry mass/pressure formulation should be considered (Lauritzen et al. 2018)
  - would simplify code for dycore and physics development
  - offers an opportunity for code refactoring / reorganization
- Consider the FV3-Pace version from **ai2cm** (private sector)?
  - They rewrote FV3 dycore in python (code is 55% shorter)
  - FV3-Pace is open source: https://github.com/ai2cm
  - Read more here: https://arxiv.org/abs/2205.04148
  - By using a domain specific language (GT4Py), FV3-Pace is already GPU-ready.

A python version of FV3 code

running on conventional hardware

is >3.5x faster !



Summary

- Great progress made in pulling together UFS for users!
- Works well on NOAA HPC - facilitates mostly NOAA researchers
- Need to facilitate more academic and private sector
  - a simpler UFS with limited capabilities would lower the cost of entry for all user bases
  - needs to pass the "*middle-aged faculty advisor*" test
    (not the grad student test)
- Investments between now and 2025 will still take 3-5 years to bear fruit.
- UFS and EPIC do an extensive community survey to determine barriers and needs!
  (are you using FV3? Why or why not? What could help make you try it? What to improve support, etc.)
- If not now, when?