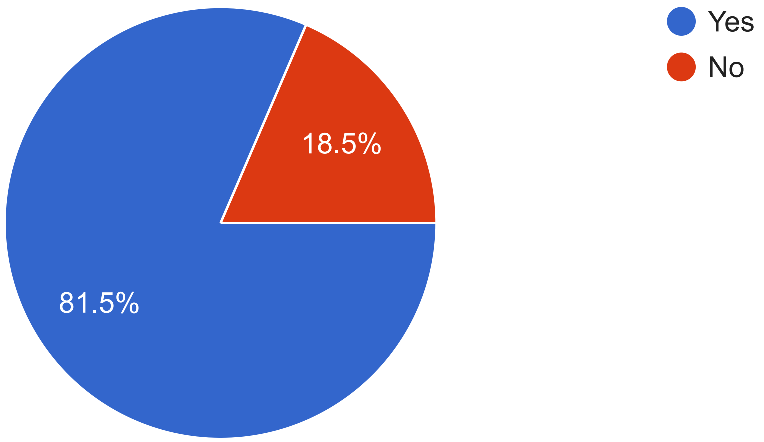
HWRF and HAFS Community Survey

27 responses

Publish analytics

Are you a HWRF user and/or developer?

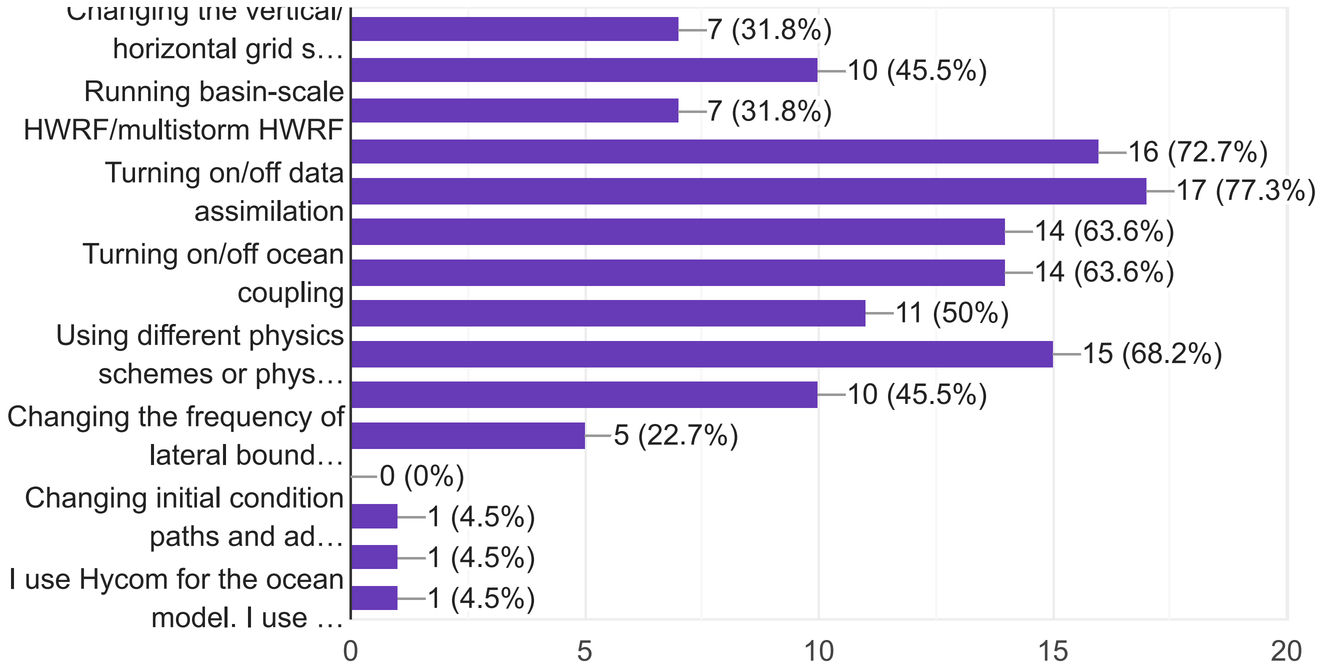
27 responses



Your experience with HWRF

What HWRF configuration options do you use most frequently, if any? Select all that apply.

22 responses



What features of the HWRF workflow make your work easier to conduct?

22 responses

Changing configuration option thought rocoto command line

default values for settings are easy to find, and easy to change/override in the param files

I find having the configuration options at the parm level is very straightforward rather than hunting down namelists for all components in separate locations. Having archival tasks in the workflow is appreciated. The build system has pros and cons, but in general, I find the build system useful.

Being able to configure/modify various aspects of the end-to-end modeling in a set a configuration files

All tunable or changeable parameters are in one or two files, or at least in one directory

independent components

What features of the HWRF workflow do you find difficult or cumbersome to use or develop?

22 responses

Changing the time step or the model size to meet certain requirement

portability to new platforms and/or compilers is challenging, requiring a high level of expertise

I would like the logs and error messages to be more descriptive - in particular for I/O issues. The portability of the system is cumbersome. Having the ability to run a specific component in a standalone mode is useful for many developers. Making modifications to the system, in particular the scripting layer, can be hard to follow. The storage locations and naming conventions for the input data is somewhat frustrating.

It can be hard to execute just a single task in the workflow, e.g., just the postprocessing.

Too many control files

python scripting with hidden scripts and libs and workflow manager

If you could change one thing about the HWRF workflow to make it more user-friendly, what would it be?

22 responses

It would be nice if the workflow can re-start any job from $COM directory. e.g. re-run only post or track job from $COM.

Simplify the input dataset specifications

Make the scripting layer less complex to follow.

More documentation about the HWRF datastore.

Simplify it

re-organize the python scripts and required libs

An overarching wrapper script to set simulation settings, input paths, configuration settings, etc. This is sort of already built but could be improved for efficiency and user-friendliness.

Make the workflow more modular. Components should be added/subtracted without

Consider the design of the entire HWRF system's workflow, including the workflow manager (Rocoto, ecFlow), configuration files (parm/*.conf), scripts, and the data management system (the HWRF datastore). What strengths of the HWRF workflow should be made available in HAFS? What should be done differently in HAFS?

22 responses

The current HWRF workflow is easy to use. Would like to see a similar workflow for HAFS. GUI based workflow?

Strength: defaults and overrides for param settings; Differently: Simplify the platform/compiler settings with good defaults, so that porting is more straightforward; allow running without a workflow manager and batch system (i.e. docker container, simple linux cluster, AWS)

Strengths: workflow manager, configuration files, archival tasks. Needs improvement from current state: HWRF datastore and scripts.

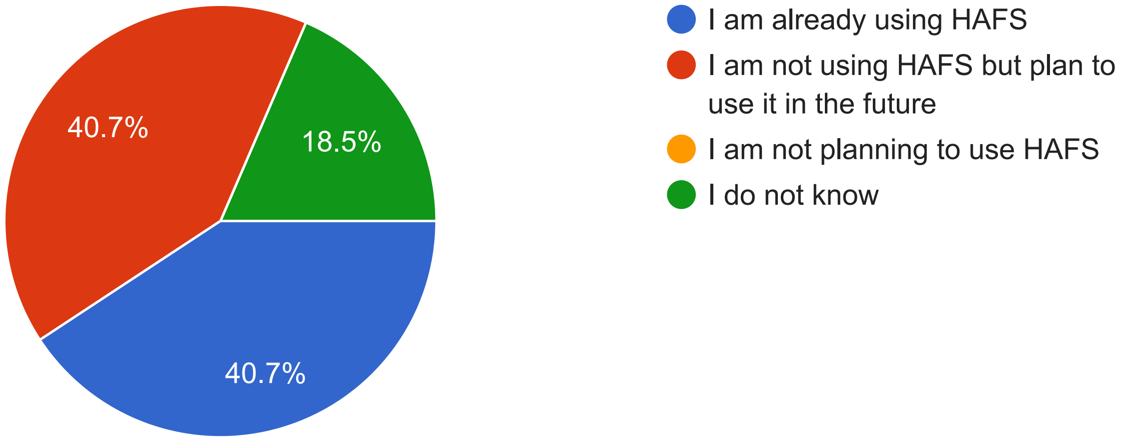
Strengths: availability of codes through community repositories, well-documented and well-supported system; ability to use more than one workflow manager; ability to configure many parts of the system through configuration files. What could be different: an experiment database to store various configurations that users may find useful.

Easy to check where the problems are.

Transitioning to HAFS

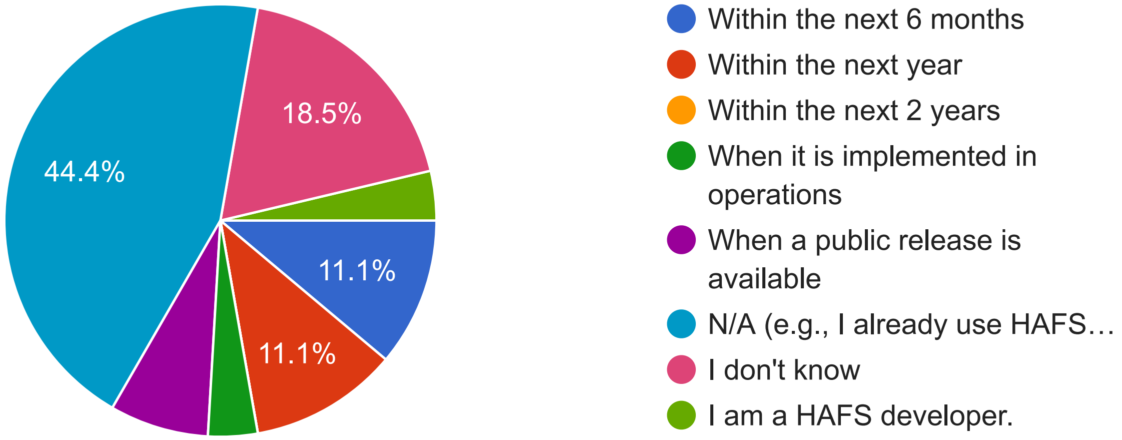
Do you plan to use HAFS?

27 responses



If you plan to use HAFS in the future, when do you plan to start using it?

27 responses



If you do not plan to use HAFS or are reluctant to use it, why is that the case? What capabilities, if any, would convince you to use it?

7 responses

I work with FV3 CAM, and am gathering information about the pros/cons of CROW, specifically.

sufficiently low implementation threshold

seem to customize for NCEP system specifically, which makes it hard for outsider users

N/A

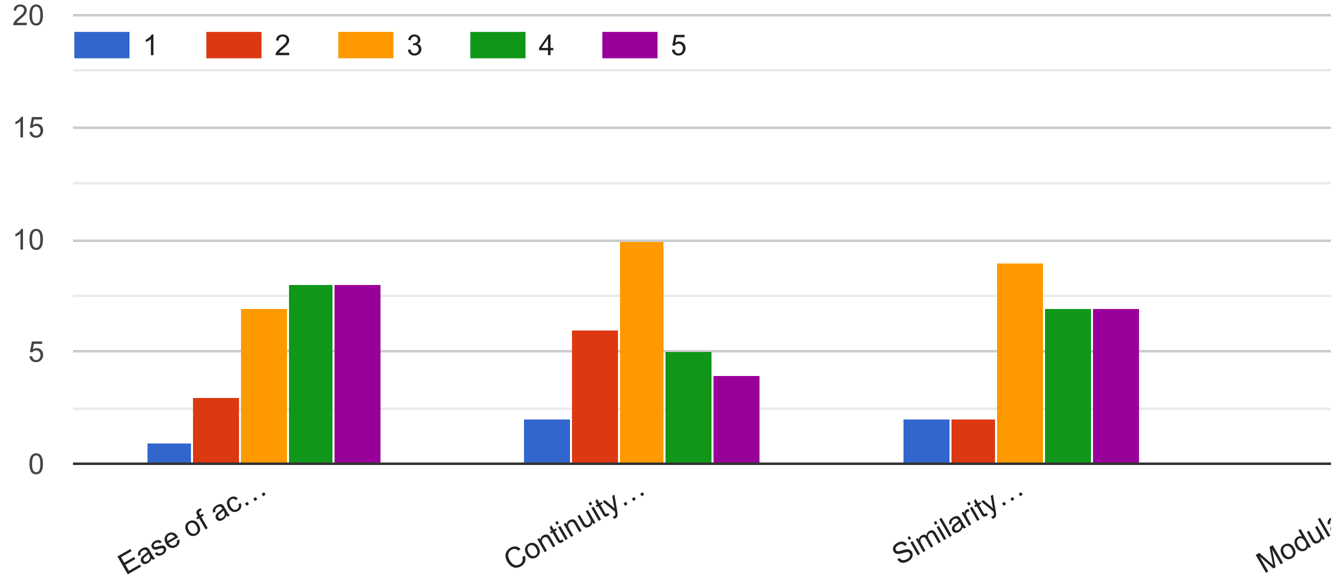
If HWRF is no longer supported.
If HAFS is supported.

I just learned recently how to use HWRF. It would be difficult and time-consuming to change model now. Maybe in the future.

not actively using it but could be in future since I have worked with HWRF before

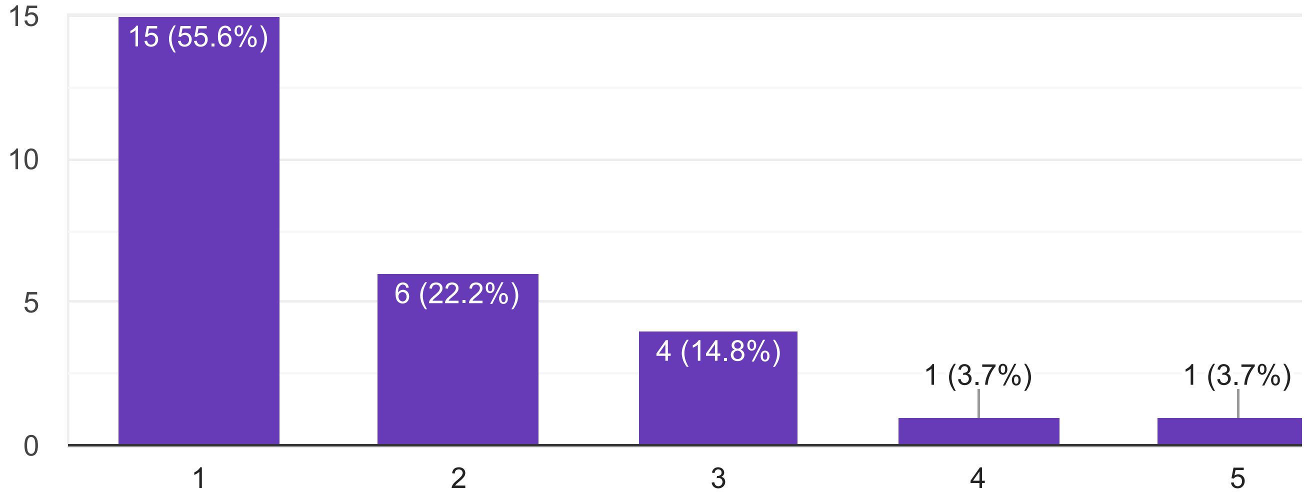
Design choices in HAFS

What criteria should inform the choice of programming language for the HAFS scripts? Give each criterion a score from 1 to 5, with 1 being not important at all, and 5 being extremely important. You may use a number more than once.



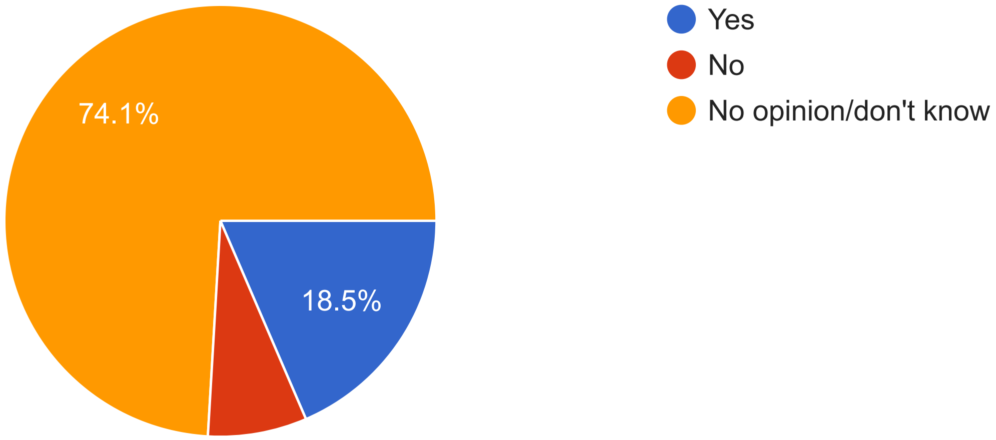
How familiar are you with CROW?

27 responses



Based on the requirements for your work in HAFS, do you believe CROW would be a suitable tool for HAFS?

27 responses



If you answered yes or no, please explain why.

6 responses

HAFS will need a way to configure experiments (cases); CROW can do this; There may be other suitable tools as well, of course, CROW is one of these

CROW allows for truly modular workflows, so I think it would be suitable. Other options could work as well.

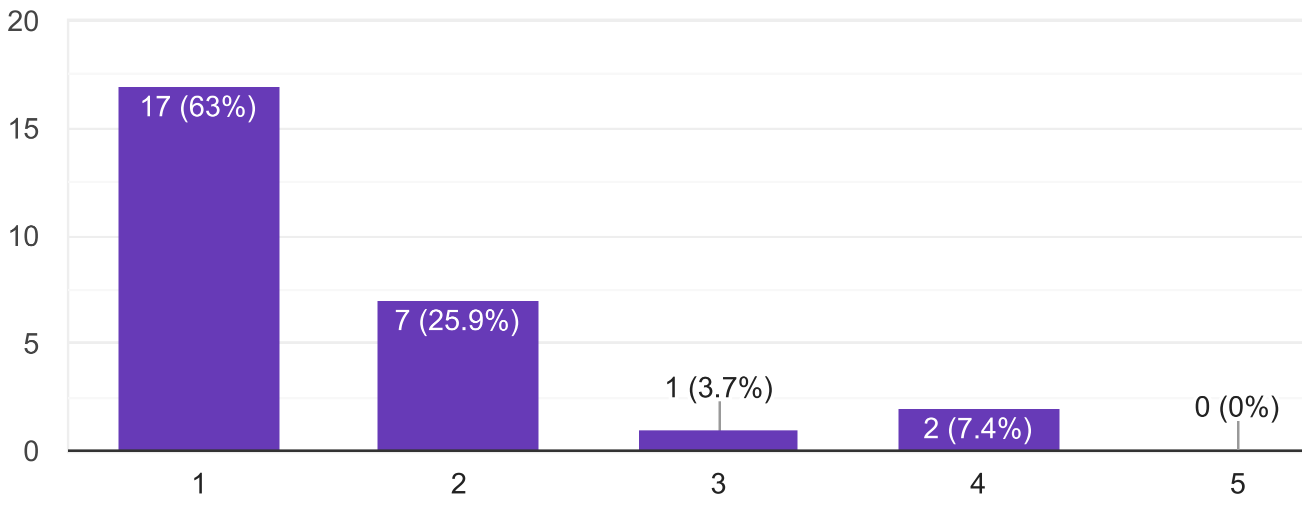
Even if CROW is not selected for the HAFS application, the future HAFS workflow development can still borrow lots of useful capabilities from CROW's Python-yaml based configuration system.

I'm sure CROW would work. But is it accepted to run in operations ?

I answered "don't know," but I'd still like to mention some reservations I have. I am concerned that the use of CROW will make it more difficult for HAFS developers to modify the system. If we are going to be asked to learn a new software system to do our work, there should be a very compelling reason/high bar to do it. Is there something about CROW that will make developing HAFS substantially easier than it is now? If not, I don't think it should be used.

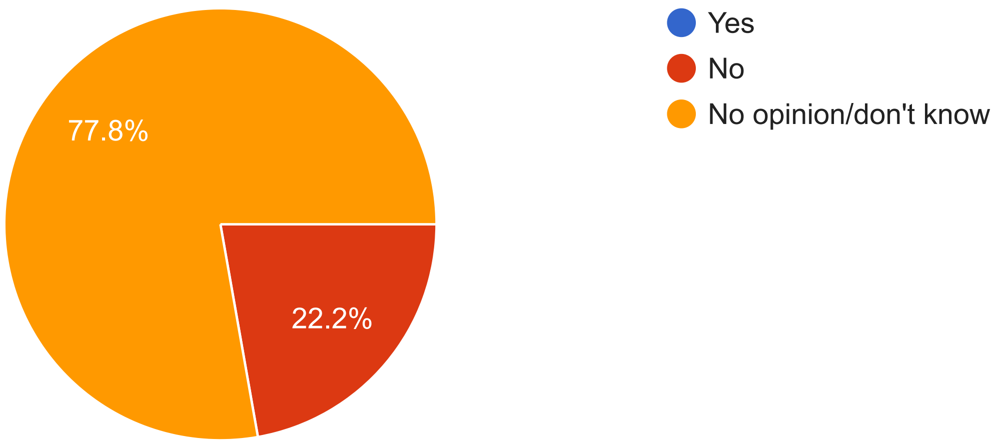
How familiar are you with the CIME Case Control System?

27 responses



Based on the requirements for your work in HAFS, do you believe the CIME Case Control System would be a suitable tool for HAFS?

27 responses



If you answered yes or no, please explain why.

6 responses

CIME focuses on the coupled forecast model; HAFS requires many other tasks (DA, vortex init, post-procs); while these can be done "outside" of CIME, it makes for a complex hybrid system that is hard to understand; also, CIME does not utilize a workflow management tool (has some internal capabilities) which works well for the coupled model, but not so well for more complex task dependencies; another possible issue is a real-time trigger - to run real-time forecasts triggered by a clock (or data arrival) vs a set case study - I'm not sure of CIME's capabilities in this area

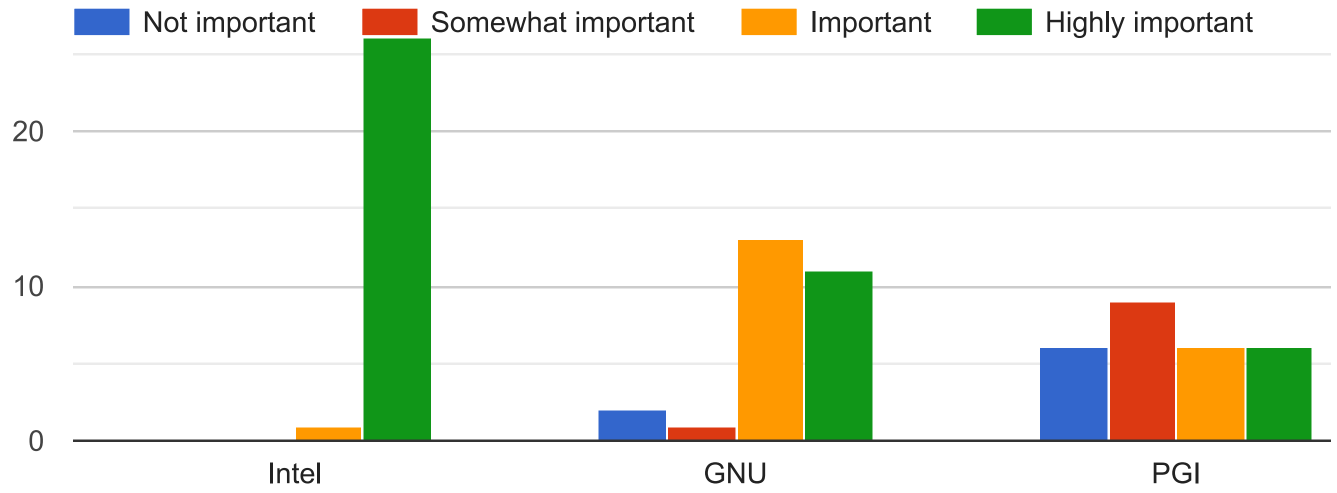
I believe the CIME as is today would not be suitable for HAFS. Perhaps CIME could be expanded to become suitable for HAFS. It would be necessary to have a way to run over multiple forecasts for both realtime experiments and large retrospective datasets. Also, a capability to run many tasks as part of a forecast, and of cycling (using results from one forecast to feed onto the next) would be required.

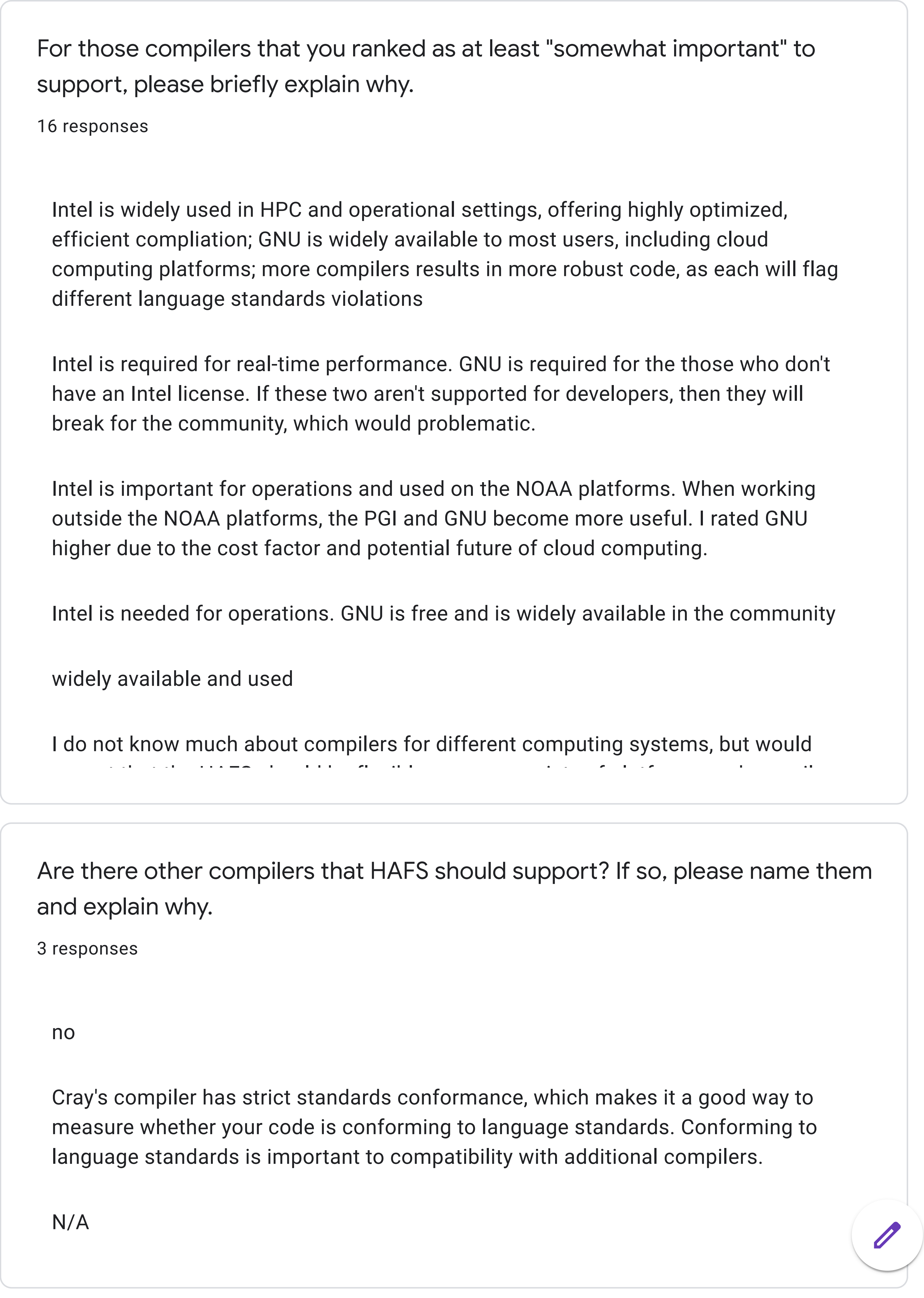
CIME's Case Control System may be useful for HAFS configurable build system. But, it might not be suitable for scripting system to support both research and operational hurricane applications.

I again answered "don't know," but I do have some reservations. Will CIME make it

HAFS build system

How important is it for users and developers to be able to build HAFS with the following compilers?





For those compilers that you ranked as at least "somewhat important" to support, please briefly explain why.

16 responses

Intel is widely used in HPC and operational settings, offering highly optimized, efficient compliation; GNU is widely available to most users, including cloud computing platforms; more compilers results in more robust code, as each will flag different language standards violations

Intel is required for real-time performance. GNU is required for the those who don't have an Intel license. If these two aren't supported for developers, then they will break for the community, which would problematic.

Intel is important for operations and used on the NOAA platforms. When working outside the NOAA platforms, the PGI and GNU become more useful. I rated GNU higher due to the cost factor and potential future of cloud computing.

Intel is needed for operations. GNU is free and is widely available in the community

widely available and used

I do not know much about compilers for different computing systems, but would

Are there other compilers that HAFS should support? If so, please name them and explain why.

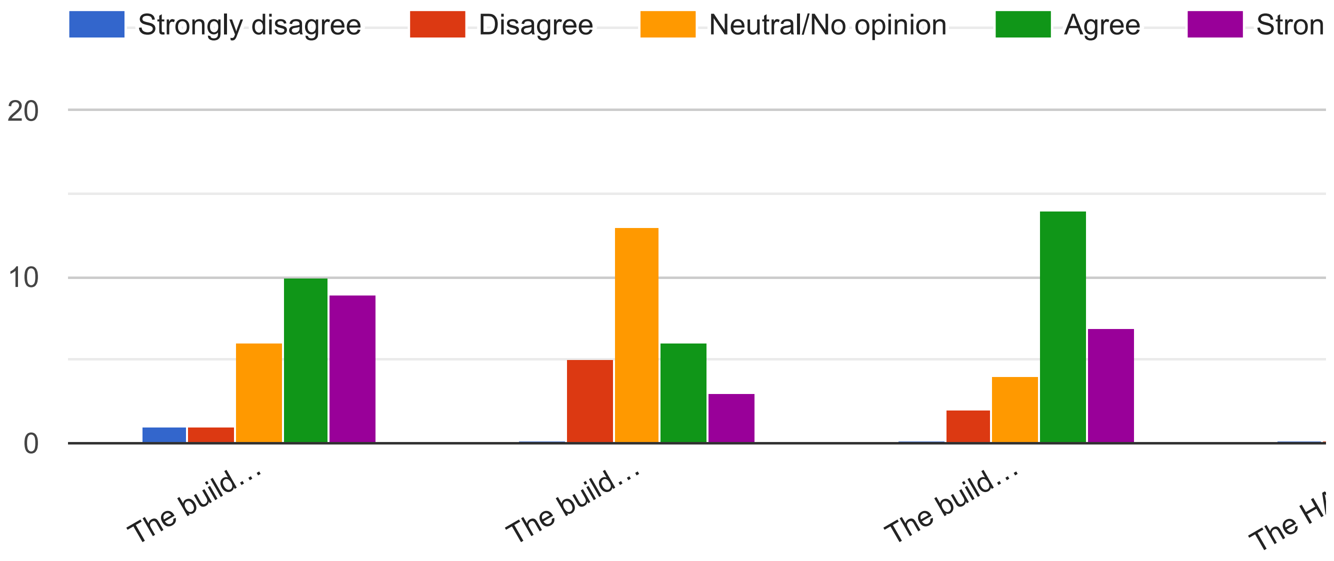
3 responses

no

Cray's compiler has strict standards conformance, which makes it a good way to measure whether your code is conforming to language standards. Conforming to language standards is important to compatibility with additional compilers.

N/A

To what extent do you agree or disagree with the following statements about the design of the HAFS build system?



If you have other comments about the design of the HAFS build system, please add them here.

6 responses

I want to clarify - I do not support autodetection of system location of files, but do support being able to fully configure a system's default locations and having the user choose such an option, e.g. for a Make-based build "make jet" or "make hera" would be preferable to a "if /lfs2 exists, then Jet" type of auto-detection.

HAFS should take advantage of other UFS applications and public releases to support of all the compilers supported by UFS.

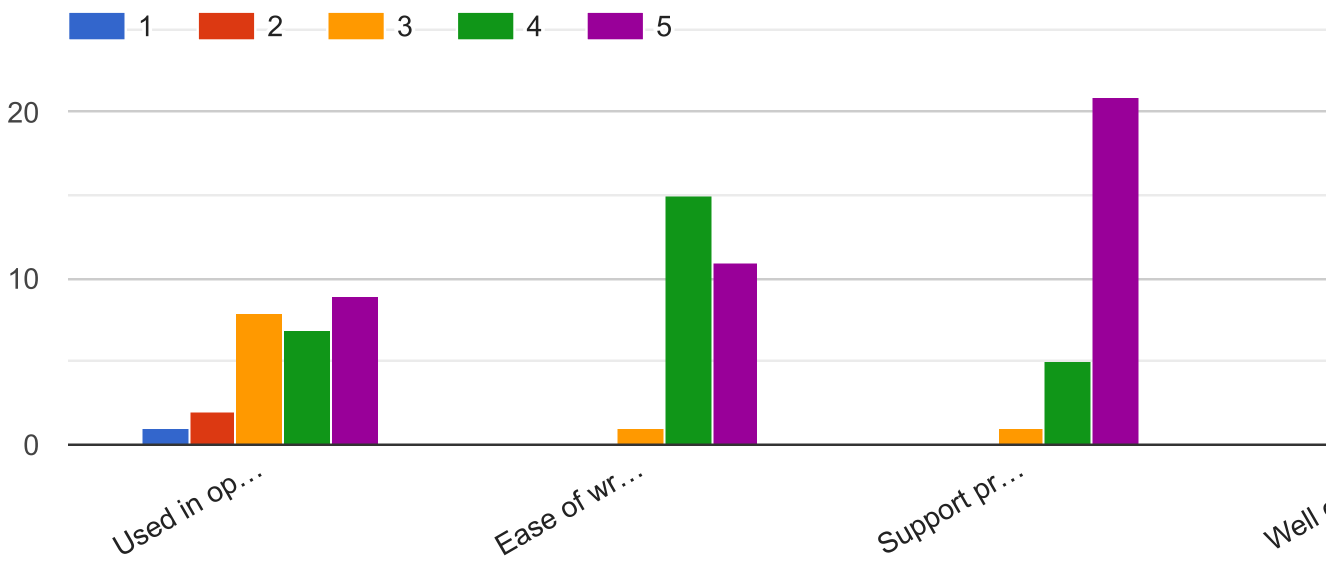
I'm not so sure If the build system should only compile the components based on the configuration. I think this could lead to more maintenance support and user confusion.

It should be easy to port the automated build system to new machines, for example, to university clusters. No one wants to have to compile all of the HAFS components by hand.

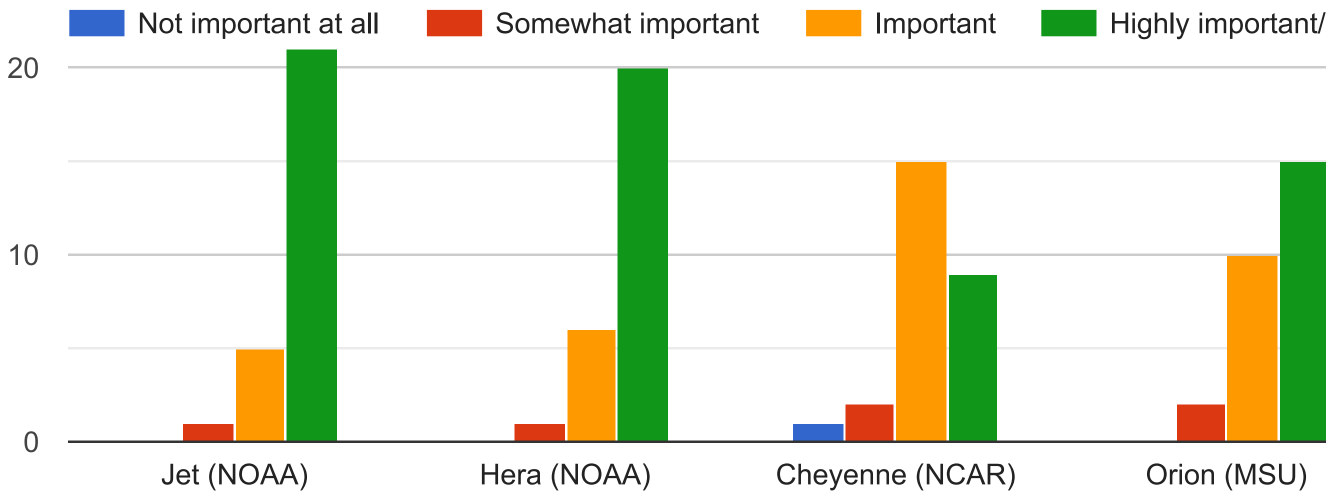
Consider using a package management system instead of a build system, or in addition to a build system.

Supported workflow managers and platforms in HAFS

What criteria should be used to pick supported workflow managers (e.g., Rocoto, ecFlow, cylc) in HAFS? Give each criterion a score from 1 to 5, with 1 being not important at all, and 5 being extremely important. You may use a number more than once.



How important is it for users to be able to run HAFS using workflow automation on the following supercomputers? By workflow automation, we mean using a workflow manager like Rocoto or ecFlow to submit jobs without user intervention when upstream dependencies are satisfied.



Are there other machines that HAFS should be able to run on using workflow automation? If so, provide the names of these machines and why.

9 responses

cloud computing services (AWS, etc) and docker/singularity-type containers

Cloud-based systems?

HAFS should be able to run in generic Linux clusters with a common batch system. That would make it more usable by university students.

NSF supported systems including those at TACC

The WCOSS system, where the operational hurricane models are being running.

NCEP WCOSS

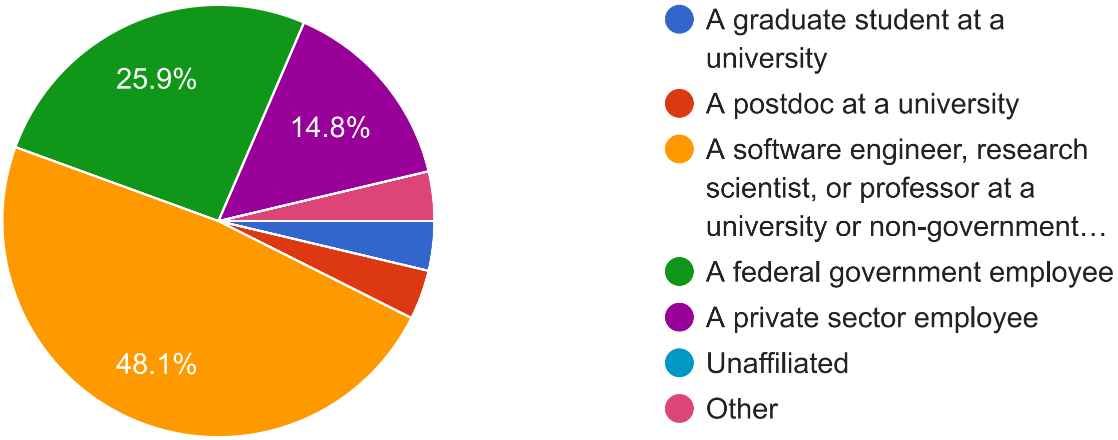
NCEP operational machine. Even if you don't use the same automation suite as NCEP operations, developers need to run there to test the new system during operational delivery, or reproduce operational failures.

perhaps NASA HPC (e.g., Pleiades)

About you

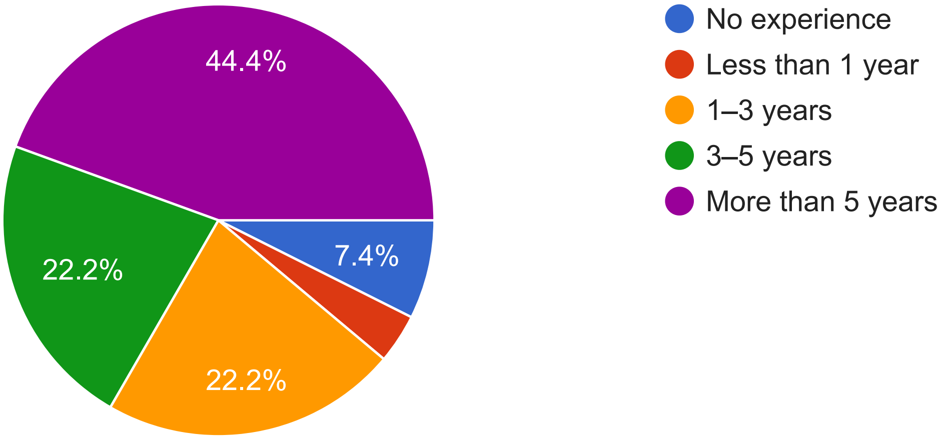
What best describes your employee affiliation?

27 responses



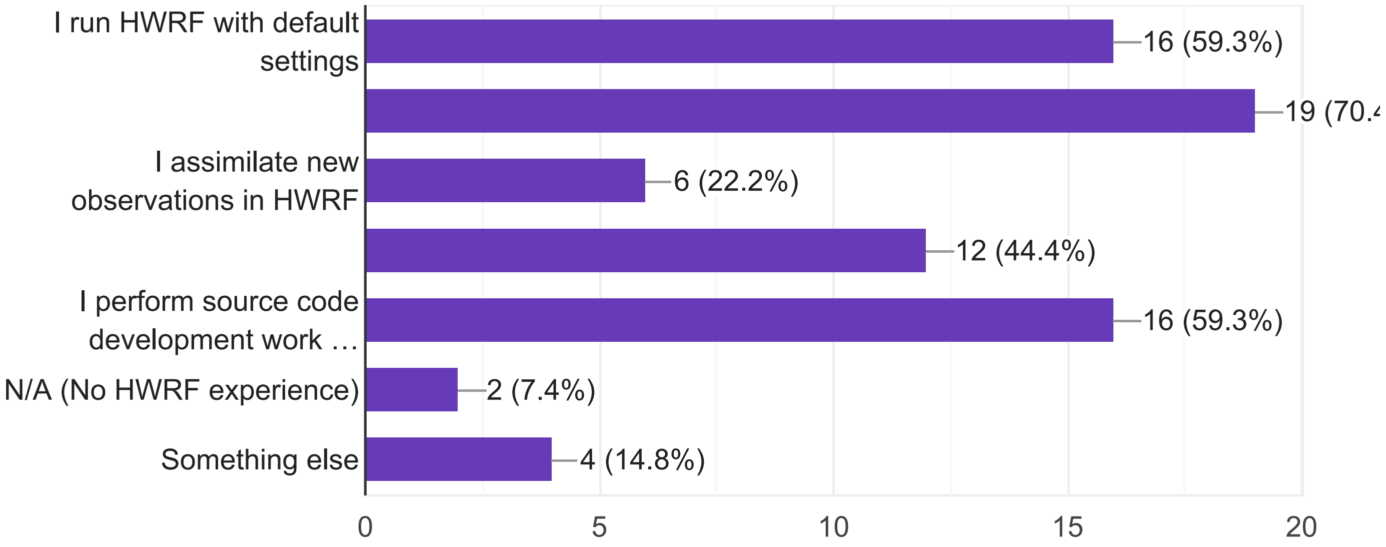
How much experience do you have using HWRF?

27 responses



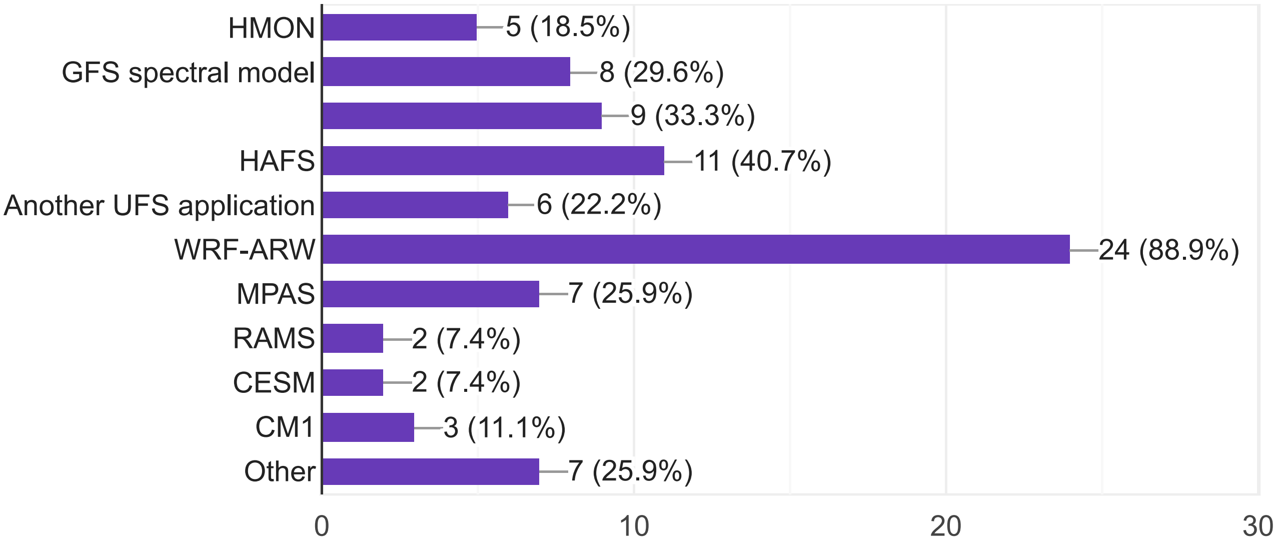
If you have HWRF experience, how do you typically use it (select all that apply)?

27 responses



What other modeling systems do you run or have you run in the past, if any? Check all that apply.

27 responses


HMON
GFS spectral model
HAFS
Another UFS application
WRF-ARW
MPAS
RAMS
CESM
CM1
Other
5 (18.5%)
8 (29.6%)
9 (33.3%)
11 (40.7%)
6 (22.2%)
24 (88.9%)
7 (25.9%)
2 (7.4%)
2 (7.4%)
3 (11.1%)
7 (25.9%)