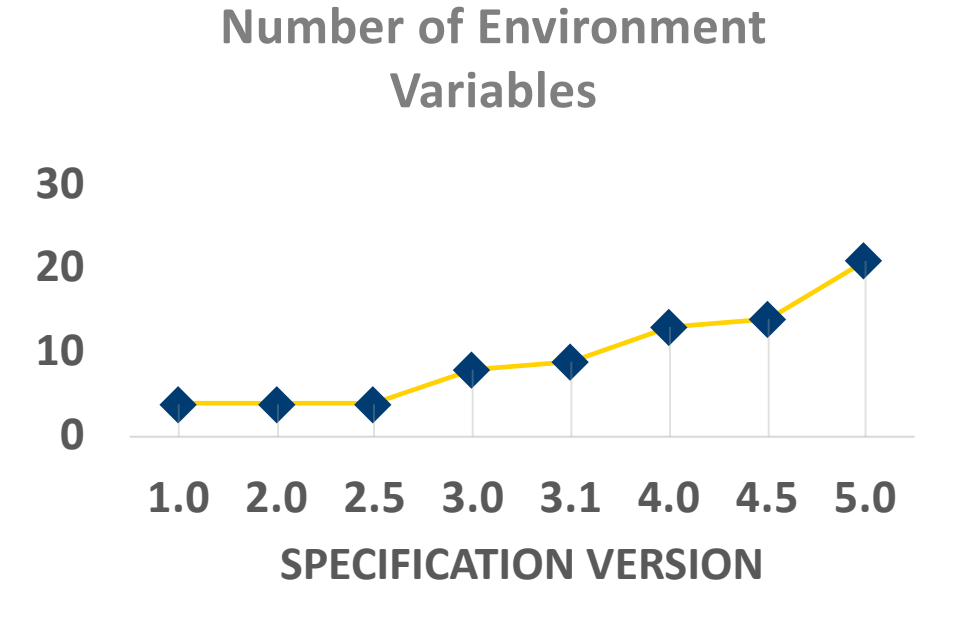
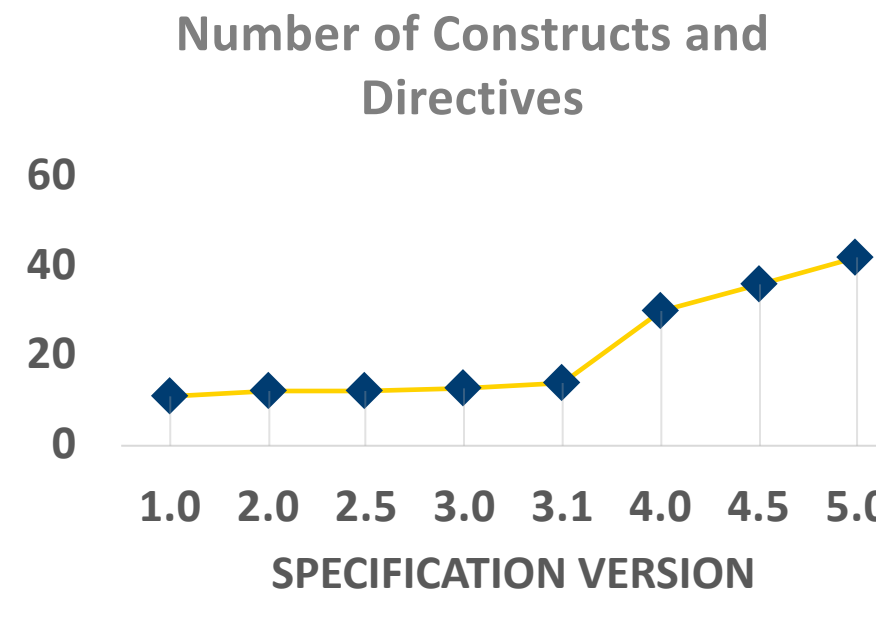
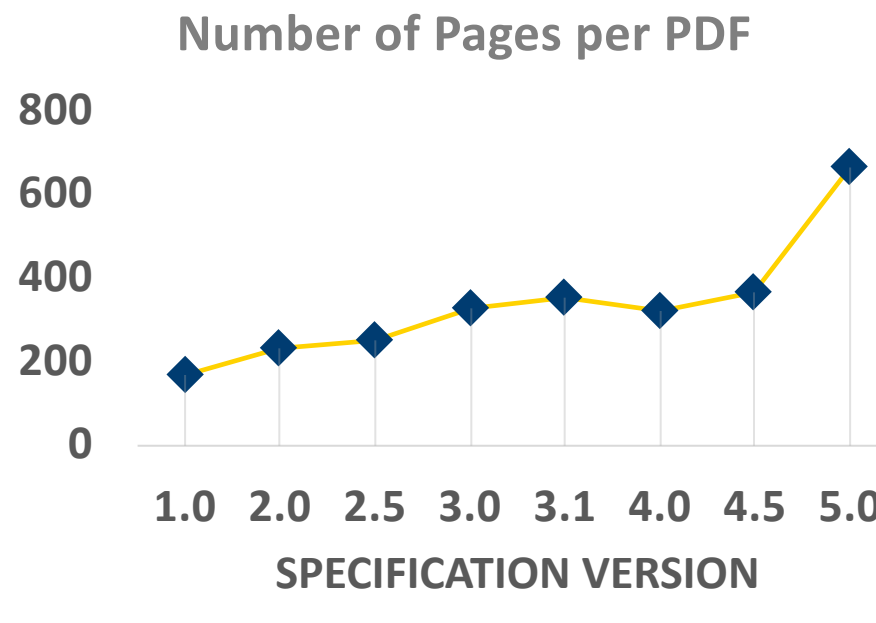
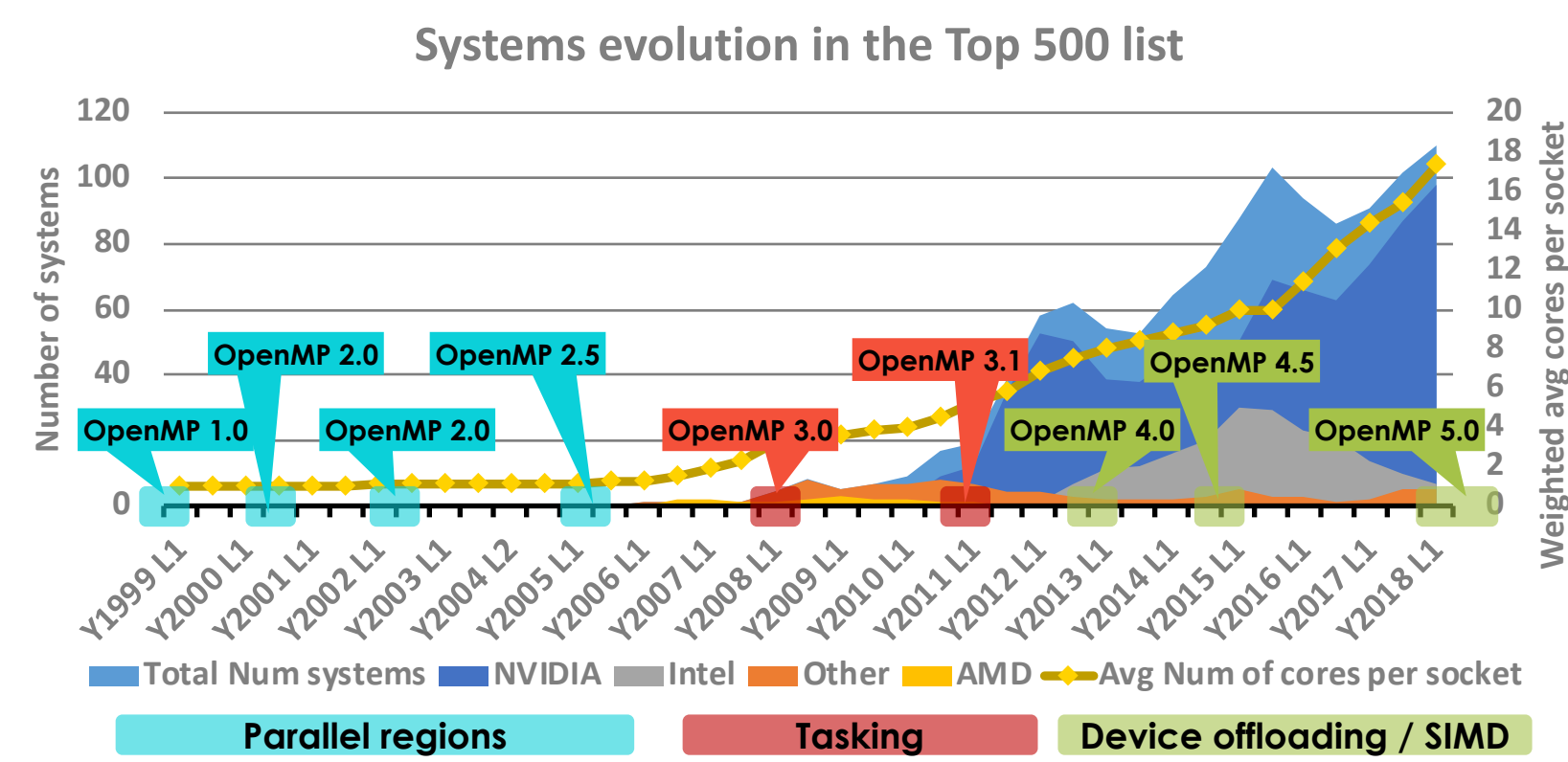


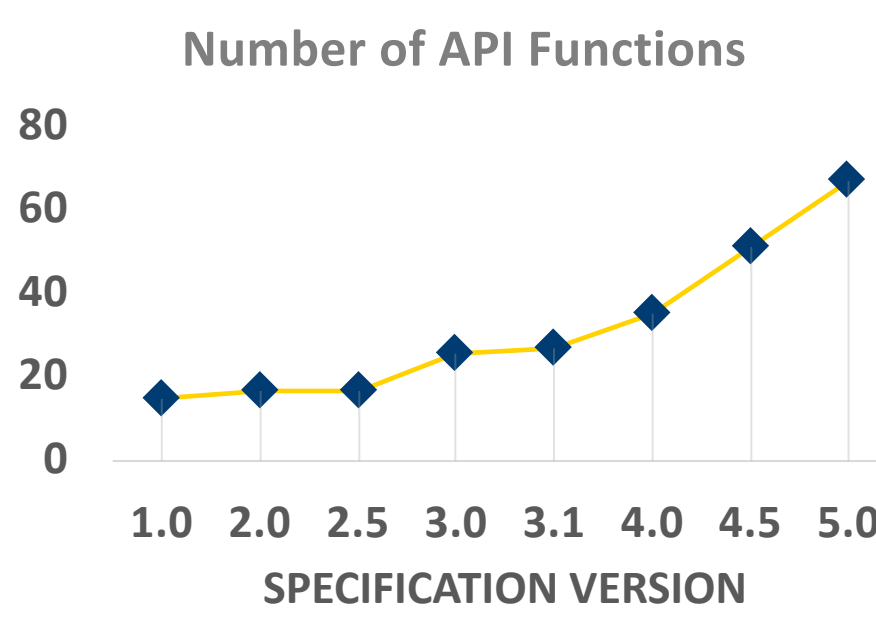


## Evolution of Computer systems and OpenMP



With the evolution of HPC computer systems architectures there are two trends that are evident:

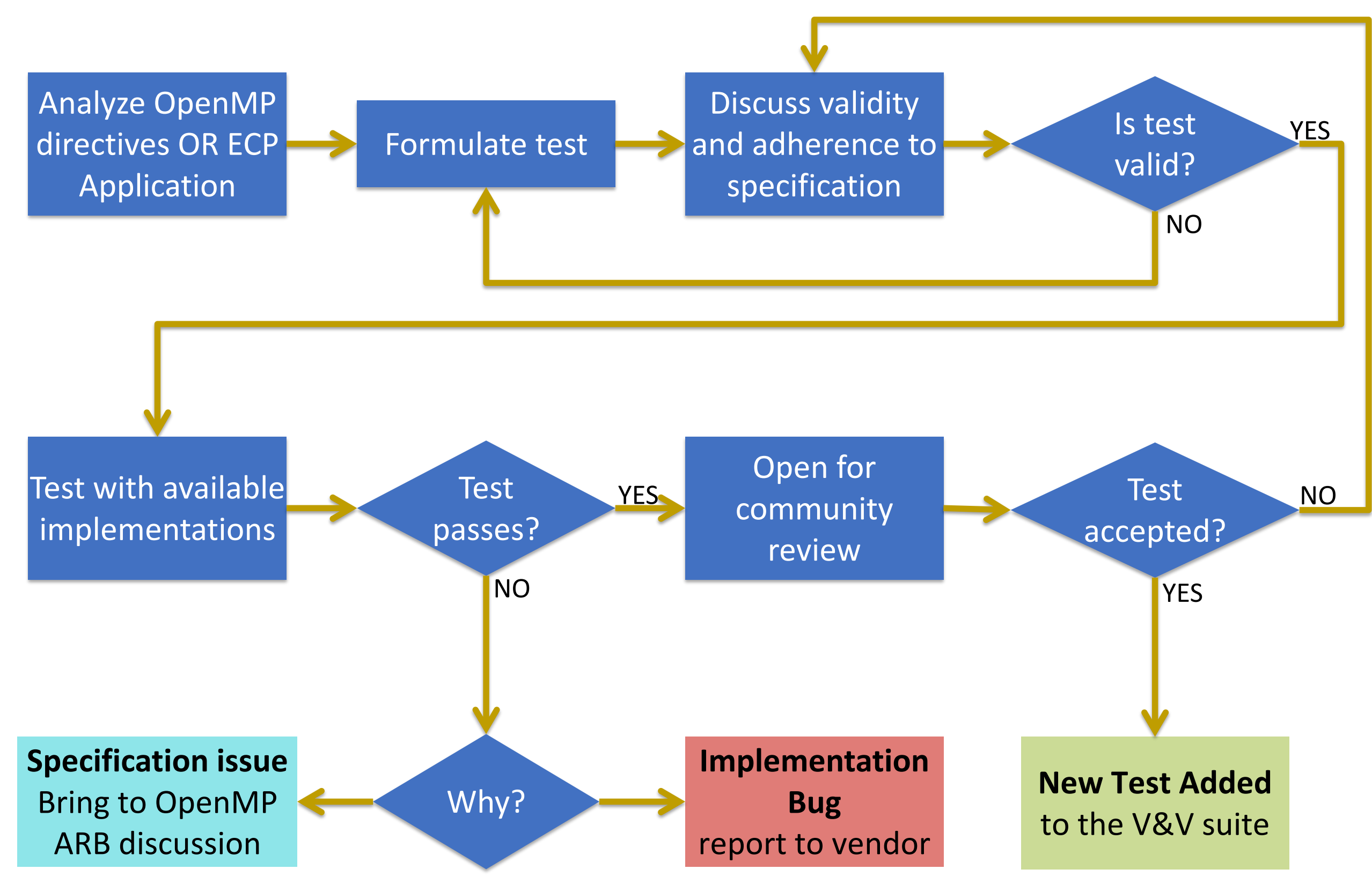
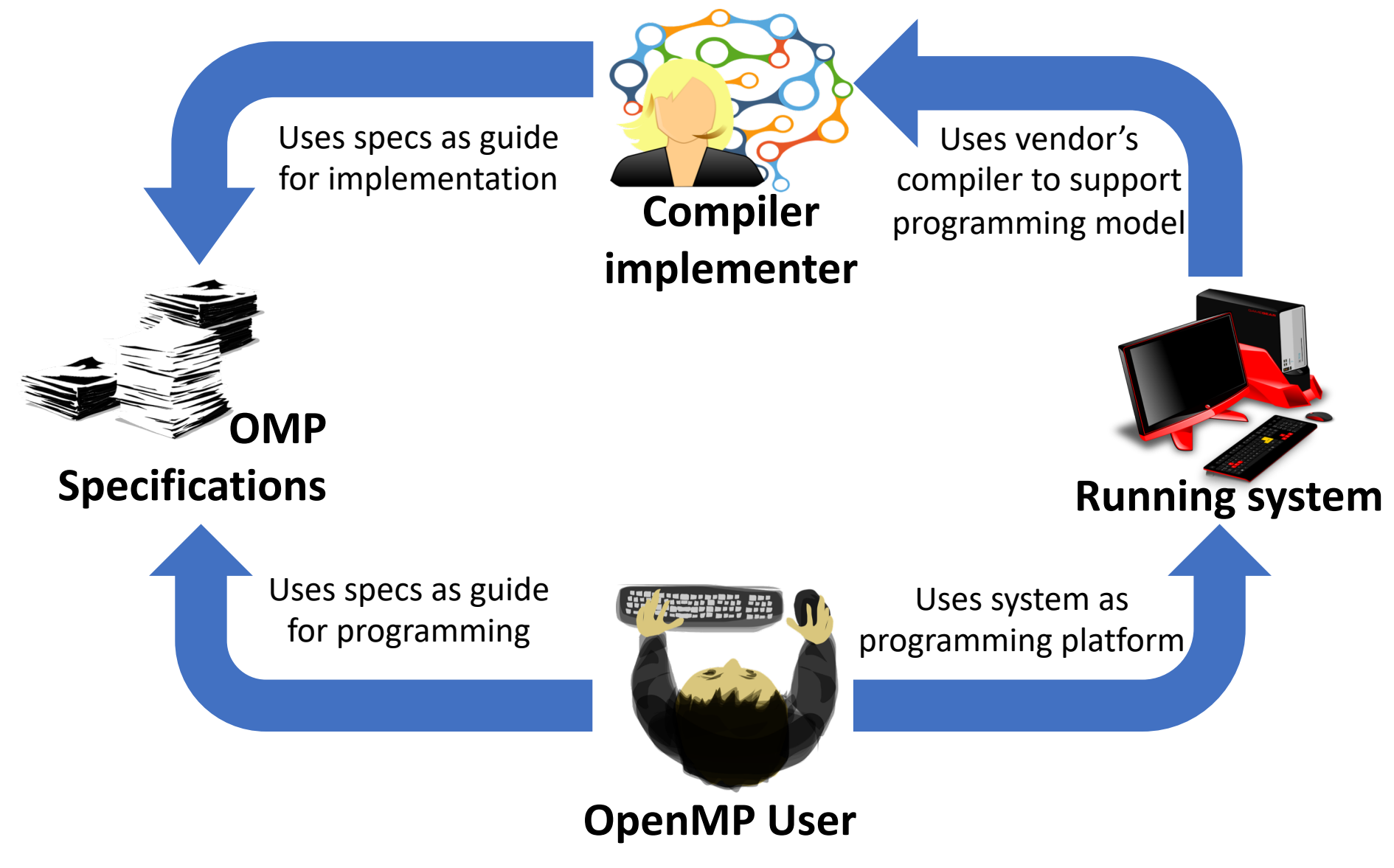
- 1) The increased parallelism
- 2) The evolution of heterogeneous execution environments (e.g. GPGPUs, AI accelerators and others)



OpenMP ARB and implementations work hard to meet the demand of application programmers. OpenMP has constantly evolved to adapt to new HPC systems, allowing programmers to define diverse parallel structures, and programs executing on heterogeneous systems. However, this has led to an increased complexity and an increased length of the specifications. Therefore, there's more opportunities for mistakes which can occur at different levels of this complex ecosystem,

## Testing design, methodology and infrastructure

The OpenMP specifications are a contract between compiler implementers, application developers, and system designers. However, there is a need to assess the quality of an implementation, running on a given system, to serve as a guarantee for the user that the specification is respected. Our team develops tests using a carefully designed methodology that is compiler independent, and which accounts for possible errors in the specification (interpretation and description) as well as possible errors in the implementations. Our tests are open to the community for use and possible re-evaluation.



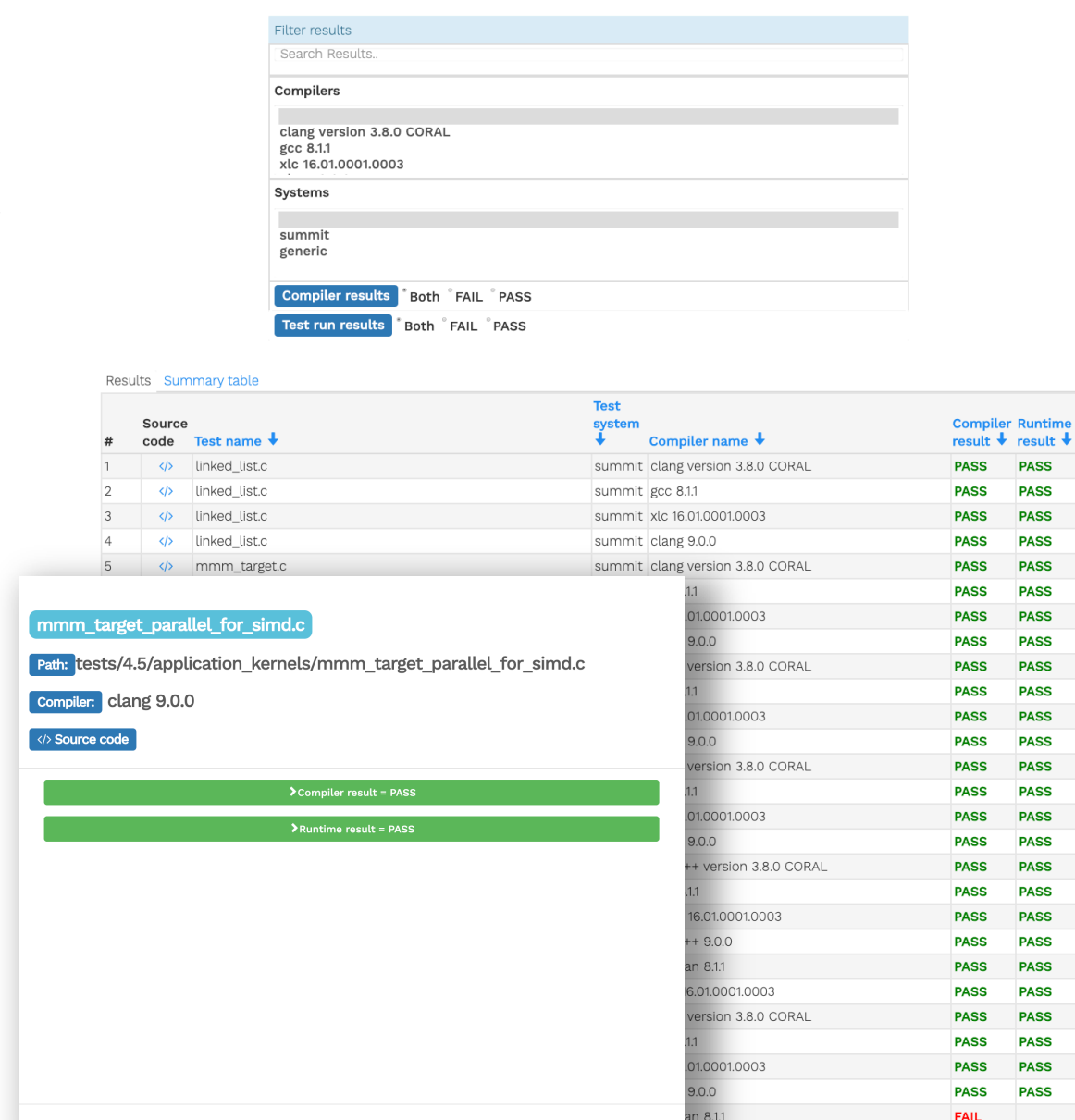
## Software Infrastructure

Our major focus for testing is offloading features, as they are critical to current ECP applications and systems, as well as the future of OpenMP. We currently have out of the box support for GNU GCC, IBM's XL, Clang/LLVM, and CCE compilers, and we are working on including other vendors. However, users can leverage our software by creating their own SYSTEM files, which could allow using custom compilers and infrastructures.

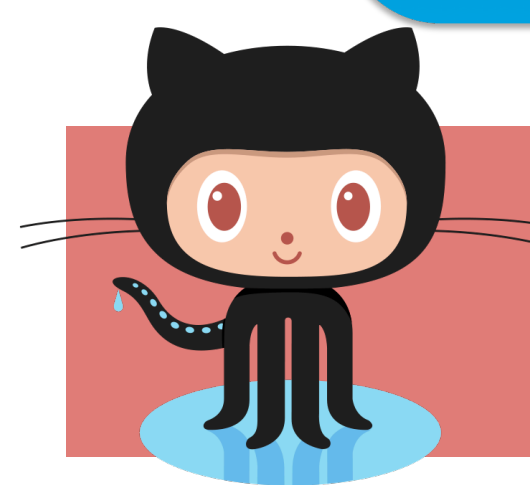
Our tests report their results through command line, to JSON files or by creating an HTML interactive results sheet.

### How to start?

```
> git clone https://github.com/SOLLVE/sollve_vv.git
> cd sollve_vv
> make # For help
> make CC=gcc CXX=g++ \
  LOG_ALL=1 VERBOSE=1 VERBOSE_TESTS=1 all
> make report_html
> # Open results_report/results.html
> # on your favorite browser
```



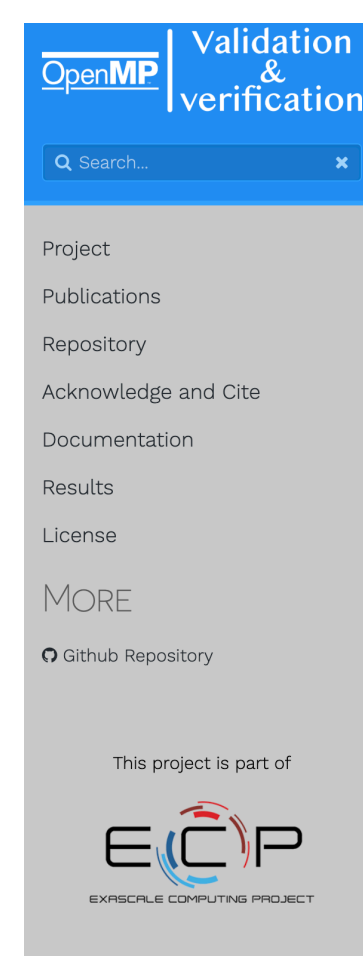
## Our Website



**OUR PROJECT HAS MOVED TO GITHUB! IF YOU ARE A USER OF OUR TEST SUITE PLEASE UPGRADE YOUR UPSTREAM LINKS!**

For More Information visit:

<https://crpl.cis.udel.edu/ompvvsollve/>



### OPENMP VALIDATION AND VERIFICATION

This website contains all related to the OpenMP Validation and Verification suite development. For this project related questions contact us: Jose Diaz (josem@udel.edu), Kyle Frick (kfrick@ornl.gov), Swaroop Pophale, Oscar Hernandez and David Bernholdt.

