

# **An implementation independent Validation** and Verification Test Suite for OpenMP

Jose M Monsalve Diaz<sup>\*</sup>, Joshua Davis<sup>\*</sup>, Swaroop Pophale<sup>†</sup>, Oscar Hernandez<sup>†</sup>, David E. Bernholdt<sup>†</sup>, Sunita Chandrasekaran<sup>\*</sup>

<sup>†</sup>Oak Ridge National Laboratory <sup>\*</sup>University of Delaware



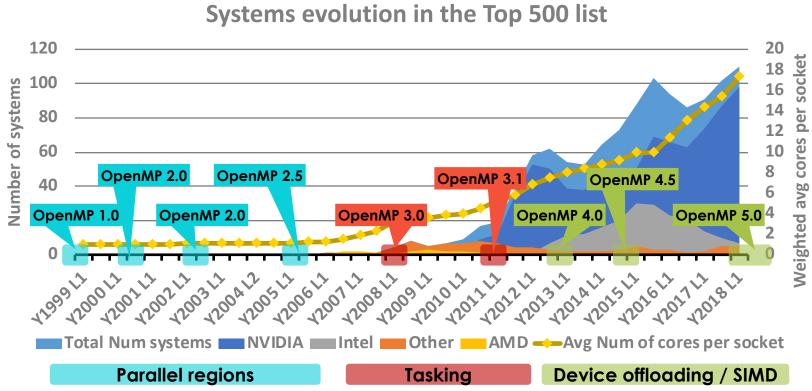
Is test

valid?

NO

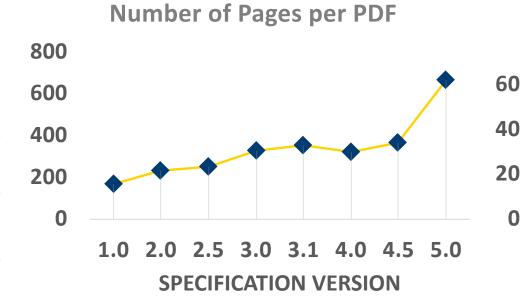
YES

## **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
- The evolution of heterogenous execution environments (e.g. GPGPUs, AI accelerators and others)



**Number of API Functions** 

80

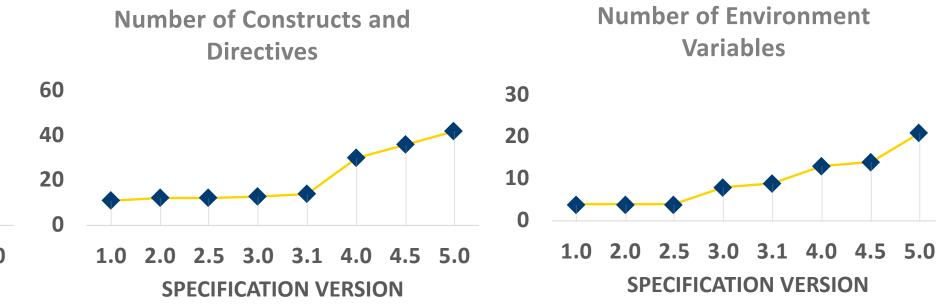
60

40

20



3.0 3.1 4.0 4.5 5.0 1.0 2.0 2.5 SPECIFICATION VERSION



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 heterogenous 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,

**Discuss validity** 

specification

and adherence to

Formulate test

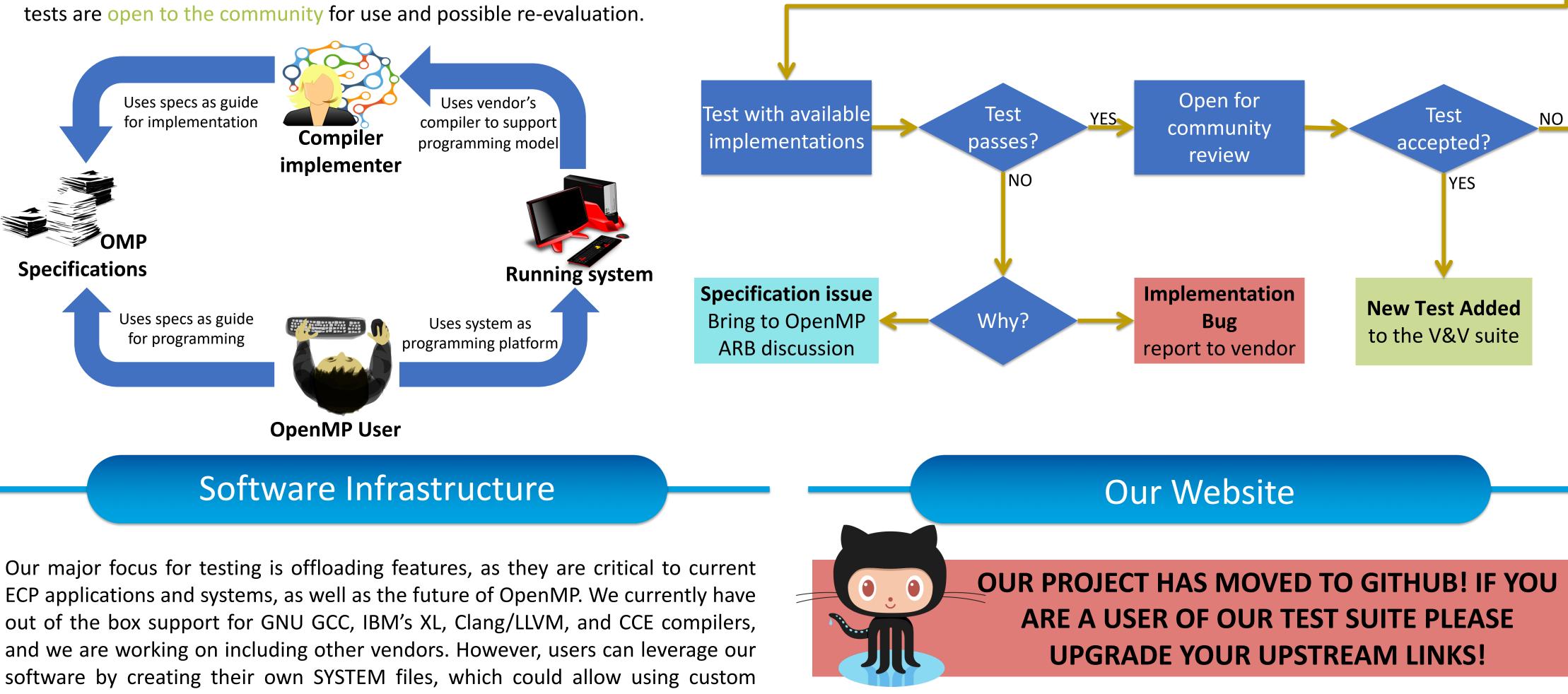
### Testing design, methodology and infrastructure

Analyze OpenMP

directives OR ECP

Application

The OpenMP specifications are a contract between compiler application developers, and system designers. implementers, 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



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

Filter results		
Search Results		
Compilers		
clang version 3.8.0 CORAL gcc 8.1.1 xlc 16.01.0001.0003		
Systems		
summit generic		
Compiler results * Both * F	IL <sup>°</sup> PASS	
Test run results Both F	L PASS	

#	Source code	Test name 🖊	Test system ↓	Compiler name 🕈	Compiler result ↓	Runtim result
1		linked_list.c	summit	clang version 3.8.0 CORAL	PASS	PASS
2		linked_list.c	summit	gcc 8.1.1	PASS	PASS
3		linked_list.c	summit	xlc 16.01.0001.0003	PASS	PASS
4		linked_list.c	summit	clang 9.0.0	PASS	PASS
5		mmm_target.c	summit	clang version 3.8.0 CORAL	PASS	PASS
				.1.1	PASS	PASS
mm_target_parallel_for_simd.c			PASS	PASS		
			9.0.0	PASS	PASS	
h: tests/4.5/application_kernels/mmm_target_parallel_for_simd.c		version 3.8.0 CORAL	PASS	PASS		
		.1.1	PASS	PASS		
ipiter. Cu	ang 5.0.0			.01.0001.0003	PASS	PASS
ource code	•			9.0.0	PASS	PASS
				version 3.8.0 CORAL	PASS	PASS
		Compiler result = PASS			PASS	PASS
►Runtime result = PASS		.01.0001.0003	PASS	PASS		
		9.0.0	PASS	PASS		
				++ version 3.8.0 CORAL	PASS	PASS
				.1.1	PASS	PASS
				16.01.0001.0003	PASS	PASS
				++ 9.0.0	PASS	PASS
				an 8.1.1	PASS	PASS
				6.01.0001.0003	PASS	PASS
				version 3.8.0 CORAL	PASS	PASS
				.1.1	PASS	PASS
				.01.0001.0003	PASS	PASS
				9.0.0	PASS	PASS
				an 8.1.1	FAIL	

# For More Information visit: https://crpl.cis.udel.edu/ompvvsollve/

Open <b>MP</b>	verification			
<b>Q</b> Search	×			
Project				
Publications				
Repository				
Acknowledge and Cite				
Documentation				
Results				
License				
More				
O Github Repository				
	ect is part of			
EXASCALE CO	MPUTING PROJECT			

#### OPENMP VALIDATION AND VERIFICATION

This website contains all related to the OpenMP Validation and Verification suite de Scaling OpenMP Via LLVM for Exascale Performance and Portability (SOLLVE) project

For this project related questions contact us: Jose Diaz (josem@udel.edu), Kyle F Swaroop Pophale, Oscar Hernandez and David Berndholt

See Acknowledgement and citation of this project

This project is a co

**NIVER** ELA

COMPUTING







