# AOMP: OpenMP Target Offloading for AMD GPUs

Jan-Patrick Lehr Member of Technical Staff Software Development Engineer

> AMD together we advance\_

## Disclaimer

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

THIS INFORMATION IS PROVIDED 'AS IS." AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

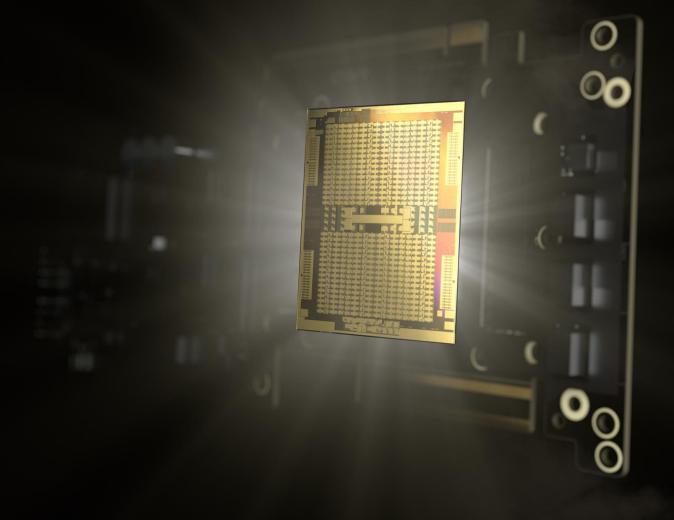
© 2023 Advanced Micro Devices, Inc. All rights reserved.

AMD, the AMD Arrow logo, EPYC, Instinct and combinations thereof are trademarks of Advanced Micro Devices, Inc. PCIe is a registered trademark of PCI-SIG Corporation. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

# Agenda

- 1. A high-level overview of the ROCm<sup>™</sup> software stack
- 2. AOMP and its (software) dependencies
- 3. AOMP compilation/linking process
- 4. AOMP architecture
- 5. Example OpenMP target offload

# **ROCm™** Software Stack

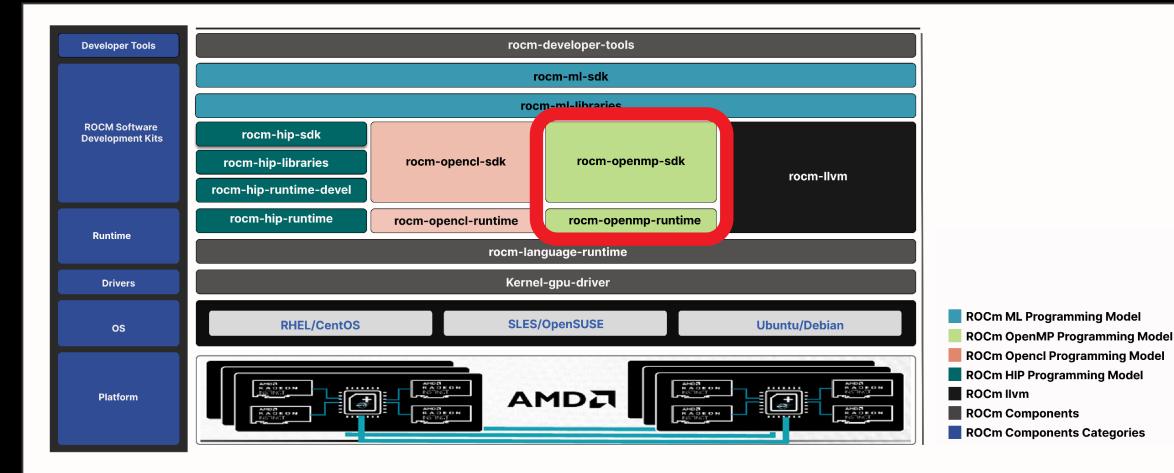




With AMD ROCm<sup>™</sup> open software platform built for flexibility and performance, the HPC and AI communities can gain access to open compute languages, compilers, libraries and tools designed to accelerate code development and solve the toughest challenges in the world today.

AMD together we advance\_

# **ROCm<sup>™</sup> Software Stack (Meta Packages)**



# **ROCm<sup>™</sup> Associated Packages**

	-				1				
rocm-developer-tools		hsa-amd-aqlprofile	rocn	rocm-debug-agent		cer-devel	rocprofiler-devel		
			m-language-runtime rocm-dbgapi		bgapi	rocm-gdb			
rocm-ml-sdk	]	rocm-core	rocm-core rocm-hip-sdk			es	miopen-hip-devel		
rocm-ml-libraries			rocm-llvm				miopen-hip		
rocm-mi-noraries		-	- rocm-core				rocm-hip-libraries		
		rocprim-devel	rocprim-devel hipcub-devel		roc	offt-devel	rocm-hip-runtime-devel		
		rocalution-devel	rocalution-devel hipblas		devel rocsparse-deve		rocblas-devel	rocwmma-devel	
rocm-hip-sdk		hipfft-devel	ro	ocrand-devel	rccl	I-devel	hipsparse-devel	hipsolver-devel	
		rocm-core	rocm-core rocm-hip-l		rocthrust-devel		hipfort-devel	rocsolver-devel	
		- roct	rocblas		rocalution		rcci		
rocm-hip-libraries	H	hipfort	rocsolver	roc	rocrand			rocsparse	
		rocm-core	rocm-hip-runt	time hir	ipblas	hipfft	hipsparse	hipsolver	
rocm-openmp-sdk	]	rocm-core	. rocm-core r		m-llvm	openmp-extras-devel		rocm-language-runtime	
rocm-opencl-sdk	-	rocm-core	rocm-core hsa-rocr-devel		rocm-openc	cm-opencl-runtime rocm-opencl-devel		hsakmt-roct-devel	
rocm-opencl-runtime	]	rocm-core	rocm-core re		nguage-runtime		rocm-opencl	rocm-ocl-icd	
tic susting doubl		hip-devel	· · · · · · · · · · · · · · · · · · ·	hipify-clang			rocm-cmake	hsakmt-roct-devel	
rocm-hip-runtime-devel		rocm-core		rocm-hip-runtime		r	rocm-llvm	hsa-rocr-devel	
rocm-hip-runtime		- rocm-core	rocm-core		rocm-language-runtime		rocminfo	hip-runtime-amd	
		- hsa-rocr rocm-core				comgr			
rocm-language-runtime						openmp-extras-runtime			
		k							

AMD together we advance\_

# **Software System Terminology**

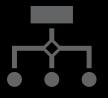
# **ROCm**<sup>™</sup>



AMD ROCm<sup>™</sup> is an open software platform for GPU compute consisting of compilers, libraries, tools, etc.

https://www.amd.com/en/graphics/servers-solutionsrocm

# AOMP



AOMP is an open-source Clang/LLVMbased compiler with added support for the OpenMP® API on Radeon<sup>™</sup> GPUs that builds on top of ROCm<sup>™</sup>

Releases are more frequent than **ROC**m<sup>™</sup> (https://github.com/ROCm-Developer-Tools/aomp)



Since AOMP is Clang/LLVM based, the high-level software architecture of OpenMP target offload support is identical

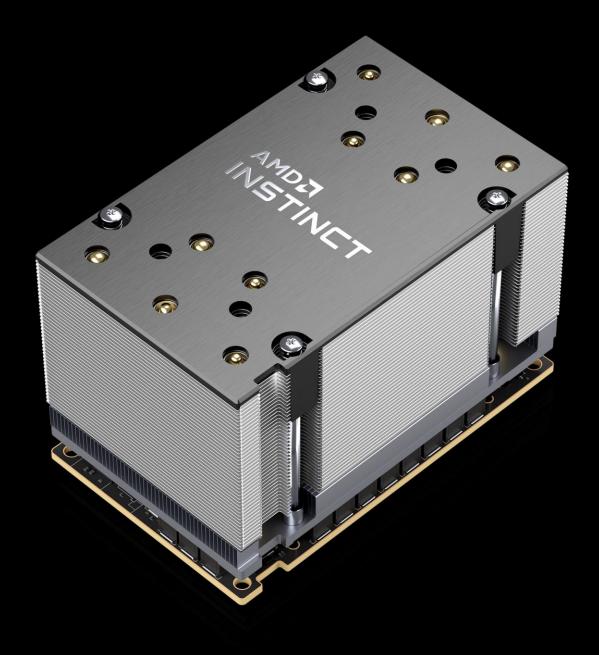
AOMP is both ahead and behind LLVM mainline trunk (upstream)



# Jargon

- Upstream: LLVM mainline trunk (i.e., the code in their GitHub repository)
- Host: The host machine (e.g., AMD Epyc processor-based server)
- (Target) Device: An attached accelerator (e.g., AMD Instinct MI200)
- (Host) Runtime: The OpenMP runtime implementation for the host
- Libomptarget: The OpenMP target offload runtime (running on the host)
- (Target or Device) Runtime: The OpenMP runtime implementation for the target device
- Plugin: The device-specific target-runtime plugin
- Kernel: A target region executable on the device
- Queue: An HSA (hardware) queue
- Signal: An HSA signal

Open Source OpenMP Compiler for AMD GPUs



- **Open Source** OpenMP compiler for AMD GPUs
  - Available from <u>https://github.com/ROCm-Developer-Tools/aomp</u>
- Based on Clang/LLVM
  - AOMP tracks LLVM upstream closely, typically only a few hours behind
  - Includes additional optimizations (e.g., faster reductions)
  - Includes additional features (e.g., OMPT support)
  - Fixes are commonly submitted to upstream LLVM
- Fortran support
  - Currently via classic Flang
  - Active development on LLVM Flang



Getting, building and installing AOMP (and its dependencies)

- AOMP is "standalone"  $\rightarrow$  requires only the kernel module (dkms) and libdrm
- AOMP is isolated from ROCm<sup>™</sup> installations by installing in /usr/lib/aomp and uses RPATH on runtime libs
- AOMP includes builds of related ROCm<sup>™</sup> components
- Packages are provided on Github for
  - CentOS 7, 8, and 9
  - SLES 15 SP 4
  - Ubuntu 20.04 and 22.04

# **Components from ROCm**<sup>™</sup>

- AOMP includes various ROCm<sup>™</sup> components
  - The components are built from source
  - Versions and repositories are encoded in manifest file in the Github repository
  - Build scripts for individual components build the dependencies
  - All components are installed in AOMP install directory

#### ROCm<sup>™</sup> Components

- rocm-compiler-support
- rocm-device-libs
- ROCprofiler
- ROCtracer
- ROCdbgapi
- ROCgdb
- Hipamd
- Hip
- ROCclr
- ROCm-OpenCL-runtime
- ROCminfo
- ROCm-cmake
- ROCR-Runtime
- ROCT-Thunk-Interface
- hipfort

# **AOMP Manifest File**

Code	Blame 33 lines (28 loc) · 2.94 KB	Raw [] 🕹 🖉 🔻 📢								
	xml version="1.0" encoding="UTF-8"?									
	<manifest></manifest>									
	Manifest for AOMP 17.0-0 which uses ROCM 5.4 release branches of ex</th <th>xternal repositories&gt;</th>	xternal repositories>								
	<pre>5 <remote fetch="ssh://gerritgit/" name="gerritgit" review="git.amd.com:8080"></remote></pre>									
	<pre><default remote="gerritgit" revision="release/rocm-rel-5.4" sync-c="true" sync-j="4"></default></pre>									
	<remote fetch="https://github.com/ROCm-Developer-Tools/" name="roctools"></remote>									
	<remote fetch="https://github.com/RadeonOpenCompute/" name="roc"></remote>									
	<remote fetch="https://github.com/ROCmSoftwarePlatform/&lt;/th&gt;&lt;th&gt;" name="rocsw"></remote>									
	<project name="llvm-project" path="llvm-project" remote="roc" revision<="" th=""><th>n="aomp-17.0-0" groups="unlocked" /&gt;</th></project>	n="aomp-17.0-0" groups="unlocked" />								
	<project <="" name="ROCm-Device-Libs" path="rocm-device-libs" pre="" remote="roc" revision="38d950e7e7&lt;/th&gt;&lt;/tr&gt;&lt;tr&gt;&lt;th&gt;&lt;/th&gt;&lt;th&gt;&lt;project remote=" roc"="" upstream="amd-stg-open"></project>	upstream="amd-stg-open" revision="07b347366e								
	<project name="flang" path="flang" remote="roctools" rev<="" th=""><th>vision="aomp-17.0-0" groups="unlocked" /&gt;</th></project>	vision="aomp-17.0-0" groups="unlocked" />								
	<project name="aomp-extras" path="aomp-extras" r<="" remote="roctools" revolutions="" th=""><th>vision="aomp-17.0-0" groups="unlocked" /&gt;</th></project>	vision="aomp-17.0-0" groups="unlocked" />								
	<project name="aomp" path="aomp" remote="roctools" rev<br="">rev</project>	vision="aomp-17.0-0" groups="unlocked" />								
	<project <="" name="rocprofiler" path="rocprofiler" remote="roctools" th=""><th><pre>revision="rocm-5.4.3" groups="unlocked" /&gt;</pre></th></project>	<pre>revision="rocm-5.4.3" groups="unlocked" /&gt;</pre>								
	<project <="" name="roctracer" path="roctracer" remote="roctools" th=""><th><pre>revision="rocm-5.4.3" groups="unlocked" /&gt;</pre></th></project>	<pre>revision="rocm-5.4.3" groups="unlocked" /&gt;</pre>								
	<project <="" name="ROCdbgapi" path="ROCdbgapi" remote="roctools" th=""><th><pre>revision="rocm-5.4.3" groups="unlocked" /&gt;</pre></th></project>	<pre>revision="rocm-5.4.3" groups="unlocked" /&gt;</pre>								
	<project <="" name="ROCgdb" path="ROCgdb" pre="" remote="roctools"></project>	<pre>revision="rocm-5.4.3" groups="unlocked" /&gt;</pre>								
	<project <="" name="hipamd" path="hipamd" remote="roctools" th=""><th><pre>revision="rocm-5.4.3" groups="unlocked" /&gt;</pre></th></project>	<pre>revision="rocm-5.4.3" groups="unlocked" /&gt;</pre>								
	<project <="" name="hip" path="hip" remote="roctools" th=""><th><pre>revision="rocm-5.4.3" groups="unlocked" /&gt;</pre></th></project>	<pre>revision="rocm-5.4.3" groups="unlocked" /&gt;</pre>								
	<project <="" name="ROCclr" path="ROCclr" pre="" remote="roctools"></project>	<pre>revision="rocm-5.4.3" groups="unlocked" /&gt;</pre>								
	<project "="" "<="" name="RoCm-OpenCL-Runtime" path="ROCm-OpenCL-Runtime" remote="roc" th=""><th>ntime" revision="rocm-5.4.3" groups="unlocked" /&gt;</th></project>	ntime" revision="rocm-5.4.3" groups="unlocked" />								
	<project <="" name="rocminfo" path="rocminfo" pre="" remote="roc"></project>	<pre>revision="rocm-5.4.3" groups="unlocked" /&gt;</pre>								
	<project <="" name="rocm-cmake" path="rocm-cmake" pre="" remote="roc"></project>	<pre>revision="rocm-5.4.3" groups="unlocked" /&gt;</pre>								

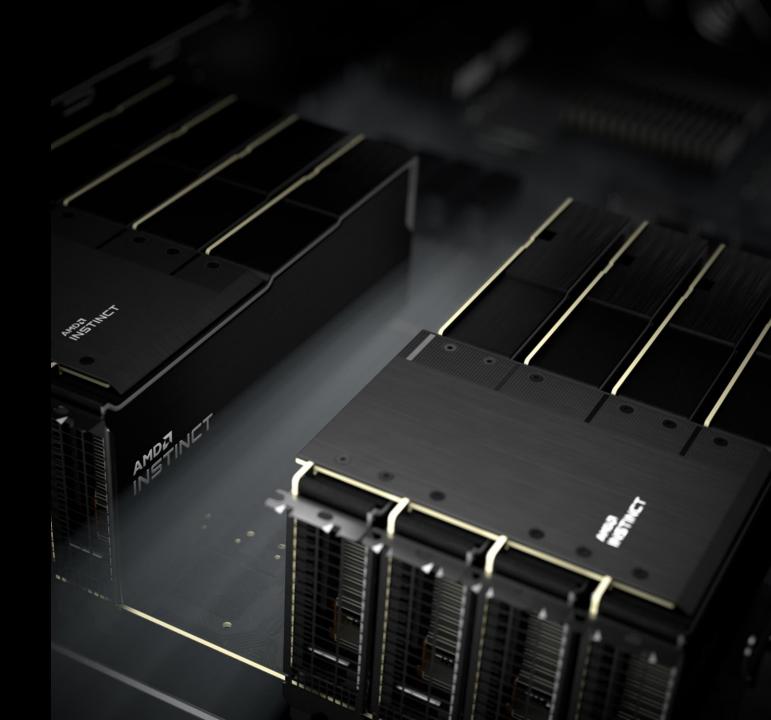
Used to tag dependencies for each AOMP release

- Used by AOMP team for development on most recent development branches
- Specific versions for each public release

**Compilation Process** 

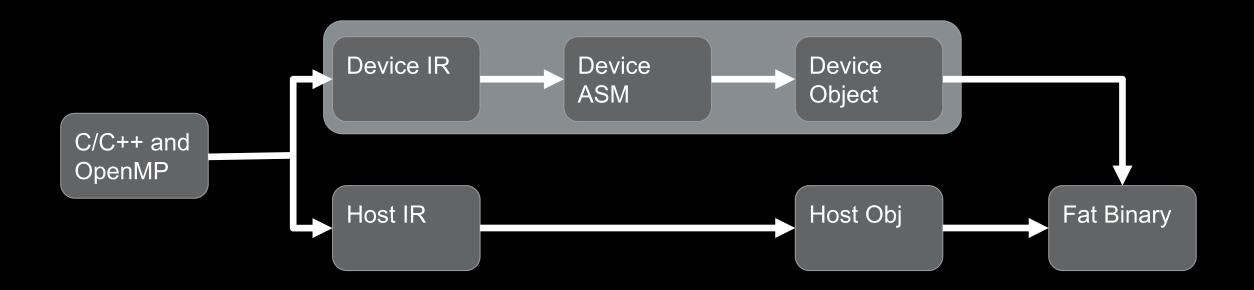
#### Execution

Device Plugin Architecture



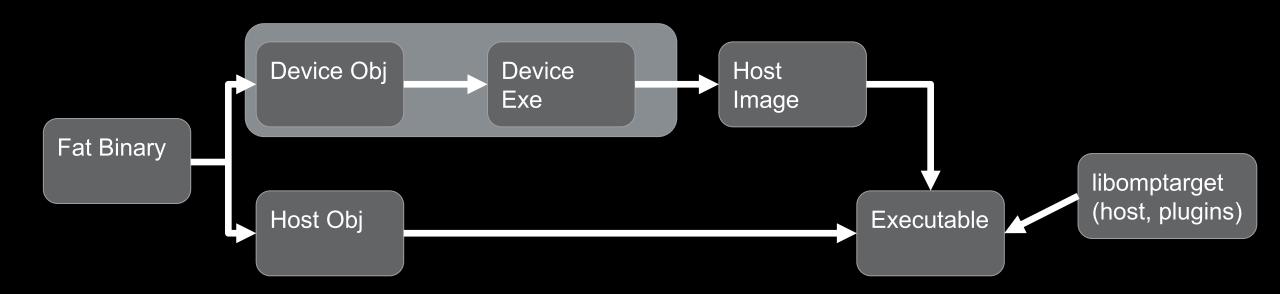
# **AOMP Compilation Process**

- Compilation requires code generation for host and for device
  - Invoke compiler twice for the different target triples
  - Object code for both targets is output into a single fat binary

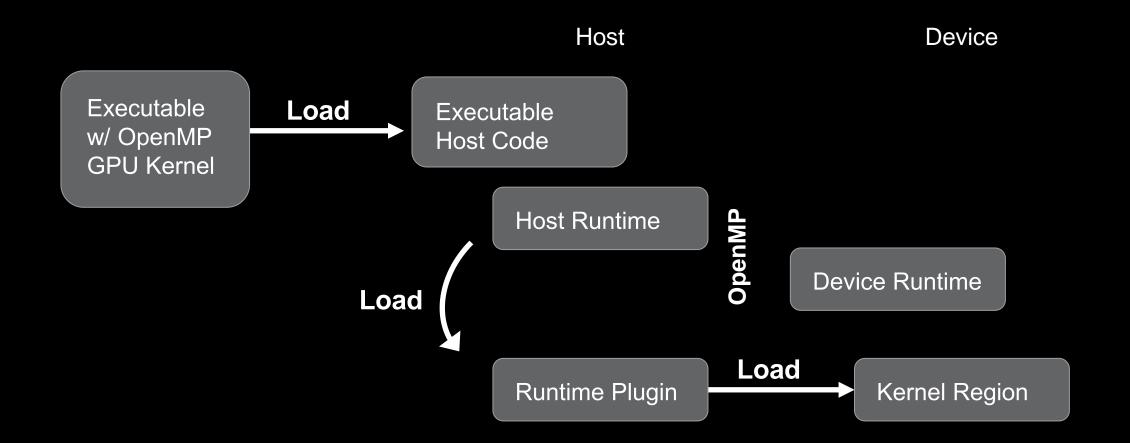


# **AOMP Linking Process**

- Linking requires generating one host image and an embedded device executable
  - Start with fat binary and *unbundle* into device and host objects again
  - Create a correctly linked host executable with embedded device ELF objects for offload kernels
- Moving towards a simplified toolchain, closer to today's upstream LLVM



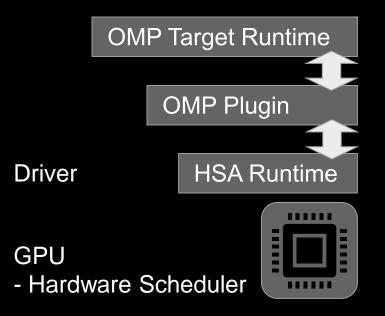
# **High-level Runtime Process Overview**



#### Public]

# **AOMP Plugin Implementation using HSA**

- HSA is the Heterogenous System Architecture, managed by the HSA foundation
  - Provides standardized interface to interact with heterogenous components
  - AMD provides vendor-specific extensions to the HSA standard as part of the ROCm<sup>™</sup> stack
- AMDGPU Plugin builds on top of HSA making use of
  - HSA locked or pinned memory (i.e., non-migratable)
  - HSA Signals and Queues for (asynchronous) operation dispatch
  - HSA extensions to obtain profiling information for HSA signals
- HSA is low level!
  - hsa\_signal\_create( ... )
  - hsa\_signal\_wait\_relaxed( ... )
  - hsa\_queue\_load\_write\_index\_acquire( ... )
  - hsa\_queue\_store\_write\_index\_relaxed( ... )



# **Example of OpenMP Target Offload**



# **OpenMP Target Offload Code Example**

```
int main(int argc, char **argv) {
    int vals[1024] = {0};
```

```
#pragma omp target teams distribute parallel for map(vals)
for(int i = 0; i < 1024; ++i) {
  vals[i] = 1;
}</pre>
```

```
for(const auto vi : vals) {
   std::cout << vi << '\n';
}
return 0;</pre>
```





# **OpenMP Target Offload Code Example**

```
int main(int argc, char **argv) {
    int vals[1024] = {0};
```

```
#pragma omp target teams distribute parallel for map(tofrom: vals)
for(int i = 0; i < 1024; ++i) {
   vals[i] = 1;
}
for(const auto vi : vals) {
   std::cout << vi << `\n`;
}</pre>
```



return 0;



# **OpenMP Target Offload Code Example**

```
int main(int argc, char **argv) {
    int vals[1024] = {0};
```

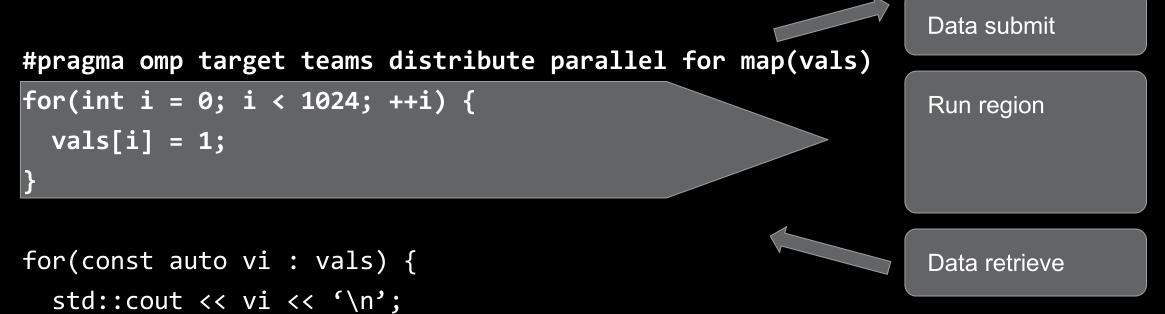
```
#pragma omp target teams distribute parallel for map(vals)
for(int i = 0; i < 1024; ++i) {
  vals[i] = 1;
}</pre>
```

Executes on the device

```
for(const auto vi : vals) {
   std::cout << vi << '\n';
}
return 0;</pre>
```

# **Libomptarget Plugin Entry Points**

```
int main(int argc, char **argv) {
    int vals[1024] = {0};
```



}

return 0;

# **AOMP Summary**

- Open-source compiler for OpenMP target offload based on Clang/LLVM
  - Support for C/C++/Fortran
  - Features are merged into ROCm<sup>™</sup> releases
- Additional optimizations compared to upstream Clang/LLVM
- Provides standalone build and installation from source using provided scripts
  - Requires the kernel driver to be installed on the system
  - Pulls-in the required ROCm<sup>™</sup> dependencies

#