LLVM/Offload - Meeting Minutes

THIS DOCUMENT IS PUBLIC

Meeting Coordinates

ICS: LLVM Offload meeting calendar inivite

Every second Wednesday, 7:00 - 8:00am PST, starting Jan 24, 2024

Alternates with the OpenMP in LLVM meeting.

Microsoft Teams meeting
Join on your computer, mobile app or room device
Click here to join the meeting
Meeting ID: 272 709 922 544
Passcode: soeJVE
Download Teams | Join on the web

Permalinks

- Development board (GH): https://github.com/orgs/llvm/projects/24/
- RFC: https://discourse.llvm.org/t/rfc-llvm-project-offload-roadmap/

Wednesday, October 1 2025, 9:00 - 10:00 am CDT

Agenda

1. PR Review list
    a.
2. Location for scripts / Infrastructure for testing (JP)
    a. Consolidate scripts that the buildbots run into llvm-project. Makes reproducibility better as it ties the scripts to the LLVM version. Also enables developers to easier recreate what the buildbots do locally.
    b. We already have the container recipes public, CMake cache files are in-tree that are used for the build. What's left is the actual scripts that are run when testing.
    c. Immediate ideas where to put these:
        i. llvm-project/offload/ci
    d. No concerns
3. Liboffload API
    a. Further discussion on device/memory allocation link - in regards to https://github.com/llvm/llvm-project/pull/154733 (closed), new PR: https://github.com/llvm/llvm-project/pull/157478
        i. Joseph: why can't we re-allocate if we got a duplicate address from allocation? Keeping track of platforms adds too much burden on the user.
            1. Ross: We'd like to get the "owner" of each valid address.
            2. Joseph: In case of USM, the linux kernel should manage the address space.
            3. Sergey: Different drivers do not communicate with one another.
            4. Joseph: It's very unlikely to get a conflict, re-allocation would be rare
            5. Piotr: Multiple devices are common with integrated GPUs.
        ii. Concern with memcpy having to do pointer lookup: may not be worth the effort.
            1. If memcpy took device ids the lookup wouldn't be needed
            2. Ross: there is a use-case with bare pointers
            3. Unified RT needs to know if the pointer is on host or on an actual device
            4. Further discussion to be continued in the PR
    b. Callum and Ross will stop working on the liboffload at the end of the month. Intel will pick up the work.
4. https://github.com/jdoerfert/llvm-project/tree/llvm_kernel_languages
    a. Jonas is implementing basic interface, expect more PRs in the next week or so.
        i. https://github.com/llvm/llvm-project/pull/156259
            1. Joseph added comments to the PR, waiting for response

        2. Jonas's internship has ended, but he will continue working on this (at a slower pace, perhaps).
    ii. More PRs coming soon
5. Parallel Thin LTO (WIP)
    a. Some parts are already in upstream. Shilei (?) has been working on that
    b. Second try: https://github.com/jdoerfert/llvm-project/tree/thin_lto
6. Level zero plugin
    a. Review in progress
7. Offload commit/PR github notifications (Alex/Nick)
    a. Should we have a subscriber group like clang?
    b. What if people only care about a subset of files

Wednesday, September 17 2025, 9:00 - 10:00 am CDT

Agenda

8. PR Review list
    a.
9. Location for scripts / Infrastructure for testing (JP)
    a. Consolidate scripts that the buildbots run into llvm-project. Makes reproducibility better as it ties the scripts to the LLVM version. Also enables developers to easier recreate what the buildbots do locally.
    b. We already have the container recipes public, CMake cache files are in-tree that are used for the build. What's left is the actual scripts that are run when testing.
    c. Immediate ideas where to put these:
        i. llvm-project/offload/ci
    d. No concerns
10. Liboffload API
    a. Further discussion on device/memory allocation link - in regards to https://github.com/llvm/llvm-project/pull/154733 (closed), new PR: https://github.com/llvm/llvm-project/pull/157478
        i. Binary search to find memory addresses is not possible or difficult since memory regions can overlap in weird ways for device allocations.
        ii. Using a map has similar problems - we'd need to do a linear search to find the appropriate key anyway.
        iii. We need AllocInfo for a hypothetical olGetMemInfo method which can query an allocations type, base and size.
            1. Sadly it doesn't seem possible to query this information directly from the pointer through HSA.
            2. We could also not provide this information and require the liboffload user to track it themselves if they need it.
        iv. AllocInfo is also used to find the platform used for host/managed allocations.
            1. Wouldn't need this requirement if the "device" parameter to olMemFree was mandatory.
            2. Without olGetMemInfo and with olMemFree only allowing the start of the buffer we wouldn't need to track the size.
            3. Could we allocate an extra 8 bytes and use the start of the buffer to store the device handle?
        v. Piotr is keen on not having olMemCpy (and friends) not accept a device pointer, is this feasible?
            1. Required to identify whether the transfer is host to device, device to host or device to device.

2. Required to identify the device itself, and which platform to use (could be queried through the queue instead?)
3. Specifying both allows the API in the future to be able to transfer data from completely separate devices (using the host as an intermediate).

vi. Pass device id to alloc/free vs pass plugin id
1. Joseph: it's easier for the user to keep track of the platform/plugin
2. Runtime tracks active plugins
3. In free, query the plugin for pointer info first

vii. Joseph: why can't we re-allocate if we got a duplicate address from allocation? Keeping track of platforms adds too much burden on the user.
1. Ross: We'd like to get the "owner" of each valid address.
2. Joseph: In case of USM, the linux kernel should manage the address space.
3. Sergey: Different drivers do not communicate with one another.
4. Joseph: It's very unlikely to get a conflict, re-allocation would be rare
5. Piotr: Multiple devices are common with integrated GPUs.

viii. Concern with memcpy having to do pointer lookup: may not be worth the effort.
1. If memcpy took device ids the lookup wouldn't be needed
2. Ross: there is a use-case with bare pointers
3. Unified RT needs to know if the pointer is on host or on an actual device
4. Further discussion to be continued in the PR

b. Callum and Ross will stop working on the liboffload at the end of the month. Intel will pick up the work.

11. https://github.com/jdoerfert/llvm-project/tree/llvm_kernel_languages
a. Jonas is implementing basic interface, expect more PRs in the next week or so.
i. https://github.com/llvm/llvm-project/pull/156259
1. Joseph added comments to the PR, waiting for response
2. Jonas's internship has ended, but he will continue working on this (at a slower pace, perhaps).
ii. More PRs coming soon

12. Parallel Thin LTO (WIP)
a. Some parts are already in upstream. Shilei (?) has been working on that
b. Second try: https://github.com/jdoerfert/llvm-project/tree/thin_lto

13. Level zero plugin
a. Review in progress

Wednesday, September 3 2025, 9:00 - 10:00 am CDT

Agenda

14. PR Review LIST
    a.
15. Location for scripts / Infrastructure for testing (JP)
    a. Consolidate scripts that the buildbots run into llvm-project. Makes reproducibility better as it ties the scripts to the LLVM version. Also enables developers to easier recreate what the buildbots do locally.
    b. We already have the container recipes public, CMake cache files are in-tree that are used for the build. What's left is the actual scripts that are run when testing.
    c. Immediate ideas where to put these:
        i. llvm-project/offload/ci
        ii. llvm-project/.ci/offload
        iii. llvm-project/offload/infra
16. Liboffload API
    a. Further discussion on device/memory allocation link - in regards to https://github.com/llvm/llvm-project/pull/154733
        i. Binary search to find memory addresses is not possible or difficult since memory regions can overlap in weird ways for device allocations.
        ii. Using a map has similar problems - we'd need to do a linear search to find the appropriate key anyway.
        iii. We need AllocInfo for a hypothetical olGetMemInfo method which can query an allocations type, base and size.
            1. Sadly it doesn't seem possible to query this information directly from the pointer through HSA.
            2. We could also not provide this information and require the liboffload user to track it themselves if they need it.
        iv. AllocInfo is also used to find the platform used for host/managed allocations.
            1. Wouldn't need this requirement if the "device" parameter to olMemFree was mandatory.
            2. Without olGetMemInfo and with olMemFree only allowing the start of the buffer we wouldn't need to track the size.
            3. Could we allocate an extra 8 bytes and use the start of the buffer to store the device handle?
        v. Piotr is keen on not having olMemCpy (and friends) not accept a device pointer, is this feasible?

1. Required to identify whether the transfer is host to device, device to host or device to device.
2. Required to identify the device itself, and which platform to use (could be queried through the queue instead?)
3. Specifying both allows the API in the future to be able to transfer data from completely separate devices (using the host as an intermediate).

vi. Pass device id to alloc/free vs pass plugin id
1. Joseph: it's easier for the user to keep track of the platform/plugin
2. Runtime tracks active plugins
3. In free, query the plugin for pointer info first
4.

b. https://github.com/llvm/llvm-project/pull/155626 - Replace openmp offload info with liboffload one? Would mean losing some vendor specific data.
i. Perhaps improve output format, otherwise good to merge

17. Can we get offload documentation hosted somewhere? Equivalent to openmp.llvm.org?
a. Email Tanya Lattner (Someone from Codeplay)
i. Callum will follow-up, (can also email Brittany Watson bwatson@llvm.org, Tanya's backup)
1. No response yet

18. https://github.com/jdoerfert/llvm-project/tree/llvm_kernel_languages
a. Jonas is implementing basic interface, expect more PRs in the next week or so.
i. https://github.com/llvm/llvm-project/pull/156259
ii. More PRs coming soon

19. Parallel Thin LTO (WIP)
a. Second try: https://github.com/jdoerfert/llvm-project/tree/thin_lto
b. Is this ready for review? Are there public discussions on this?
i. Kevin will ask about the progress about this.

Wednesday, August 20 2025, 9:00 - 10:00 am CDT

Agenda

20. PR Review LIST
    a. Michael: OMPT PR147389 is approved and ready to merge.
21. Liboffload API
    a. How to handle ambiguities in pointers with olMemFree / olGetMemInfo
        i. For example, a system has both AMD and Nvidia cards, olMemAlloc is called on both to create device pointers. Both devices happen to select 0x1000 as the memory address, how should olMemFree free that memory?
            1. The current logic doesn't work for devices that have their own allocators
            2. In unified runtime you have to pass device id to free/getmeminfo.
            3. This is specifically for device_type allocators, so the memory won't be visible on host.
            4. Solution: pass device id to olMemFree / olGetMemInfo. Make it optional for managed allocations if possible.
            5. Callum: it would be useful to have an example of where this issue happens.
            6. Ross: will create a PR
    b. Liboffload does not support multiple active plugins in a single process image. Do other low-level APIs support this and is it planned to add this capability to liboffload?
        i. Joseph: this should work
        ii. Plugin = vendor-specific portion of the library
        iii. For example, SYCL would want to use that functionality.
22. Can we get offload documentation hosted somewhere? Equivalent to openmp.llvm.org?
    a. Email Tanya Lattner (Someone from Codeplay)
        i. Callum will follow-up, (can also email Brittany Watson bwatson@llvm.org, Tanya's backup)
23. https://github.com/jdoerfert/llvm-project/tree/llvm_kernel_languages
    a. Intern at LLNL implementing basic interface, expect PRs in the next week or so.
        i. Jonas: still need some cleanup
        ii. There were some problems, but were resolved. PR expected to be ready this or next week.
24. GPU ASAN (alternative) prototype ready
    a. Testing out "new-new" design to avoid memory allocations/accesses altogether
    b. Are there any PRs available for review?
    c. Is there a design discussion on discourse?
        i. Kevin working on this, but no update yet.
25. Parallel Thin LTO (WIP)

a. Second try: https://github.com/jdoerfert/llvm-project/tree/thin_lto
b. Is this ready for review? Are there public discussions on this?
    i. Kevin will ask about the progress about this.

Wednesday, August 6 2025, 9:00 - 10:00 am CDT

Agenda

26. PR Review LIST
    a. Kevin to review: https://github.com/llvm/llvm-project/pull/143491
    b. ATTACH https://github.com/llvm/llvm-project/pull/149036
        i. Issue with dependency between data transfer and ATTACH
        ii. Joseph to review again
27. Liboffload API
    a. How can we implement out of order queues?
        i. Codeplay looking at this internally, not sure how to implement this
        ii. Any work in the queue can happen in any order
        iii. Currently: pool of queues, rotated in round-robin fashion
            1. Non-blocking queue destruction would make it easier to implement (i.e. making queue easy to create/destroy)
            2. Nobody objects to this.
    b. Implementing host callbacks
        i. Is having a "worker thread" that listens for signal changes on amd acceptable?
        ii. Is "no calling liboffload or libcuda functions in a callback" an acceptable limitation?
        iii. Enqueuing work on host from device: have a worker thread that waits for signal about kernel completing, then executing post-work code.
        iv. Joseph will take a look at PRs
    c. Should olDestroyQueue block until the queue has completed? (https://github.com/llvm/llvm-project/pull/152132)
        i. In OpenCL destroy queue blocks
        ii. In non-blocking case, no more items can be added to the queue, but existing workloads continue to completion.
            1. Decision: non-blocking
28. Auto-generated documentation for liboffload: https://github.com/llvm/llvm-project/pull/147323 (merged)
    a. Can we get the documentation hosted somewhere? Equivalent to openmp.llvm.org?
        i. Email Tanya Lattner (Benny from CodePlay)
29. https://github.com/jdoerfert/llvm-project/tree/llvm_kernel_languages
    a. Intern at LLNL implementing basic interface, expect PRs in the next week or so.
        i. Jonas: still need some cleanup
30. GPU ASAN (alternative) prototype ready

    a. Testing out “new-new” design to avoid memory allocations/accesses altogether
    b. Are there any PRs available for review?
    c. Is there a design discussion on discourse?
        i. Kevin working on this, but no update yet.
31. Parallel Thin LTO (WIP)
    a. Second try: https://github.com/jdoerfert/llvm-project/tree/thin_lto
    b. Is this ready for review? Are there public discussions on this?
        i. Kevin will ask about the progress about this.

Wednesday, July 23 2025, 9:00 - 10:00 am CDT

Agenda

32. PR Review LIST
33. Liboffload API
    a. Should liboffload have a “olLinkProgram” interface? (https://github.com/llvm/llvm-project/pull/148648)
        i. This could get really complicated to support interfaces to all possible compilers.
        ii. Wait until liboffload matures until deciding what to do with this. The PR will be closed for now and revisited later. It may eventually become a separate library.
    b. “OutEvent” vs olCreateEvent (https://github.com/llvm/llvm-project/pull/150217)
        i. If we do use OutEvent, do we want the event to have a field for the entry point that created it (like ur_command_t in UR)?
            1. Fix the name/description
34. Auto-generated documentation for liboffload: https://github.com/llvm/llvm-project/pull/147323 (merged)
    a. Can we get the documentation hosted somewhere? Equivalent to openmp.llvm.org?
        i. Email Tanya Lattner
35. https://github.com/jdoerfert/llvm-project/tree/llvm_kernel_languages
    a. Intern at LLNL implementing basic interface, expect PRs in the next week or so.
36. Second PGO for GPU patches is ready, third under preparation
    a. https://github.com/llvm/llvm-project/pull/93365 (merged)
        i. delete
37. GPU ASAN (alternative) prototype ready
    a. Testing out “new-new” design to avoid memory allocations/accesses altogether
    b. Are there any PRs available for review?
    c. Is there a design discussion on discourse?

38. Parallel Thin LTO (WIP)
    a. Second try: https://github.com/jdoerfert/llvm-project/tree/thin_lto
    b. Is this ready for review? Are there public discussions on this?
39. Performance monitoring
    a. https://crpl.cis.udel.edu/lnt-sollve/
    b. https://gitlab.e4s.io/uo-public/llvm-sollve/-/pipelines
        i. Caught: https://github.com/llvm/llvm-project/pull/96909
40. CUDA on LLVM/Offload
    a. https://github.com/llvm/llvm-project/pull/94821
    b. https://github.com/llvm/llvm-project/pull/95371

Wednesday, July 9, 2025, 9-10am CDT

Agenda

41. PR Review LIST
42. Liboffload API
    a. Unclear how best to implement "kernel handles"
        i. void * handles? Handles with public fields? A type like ol_program_t?
        ii. How best to implement an equivalent to olGetKernelInfo (is a "stat" like function preferred? Should it accept the program as an argument)?
            1. Fields required for UR: Name, number of arguments, string of attributes.
        iii. How best to implement olGetKernelMaxGroupSize. It's an info method that requires the amount of dynamic memory to also be specified.
            1. https://github.com/llvm/llvm-project/pull/142950
43. Auto-generated documentation for liboffload: https://github.com/llvm/llvm-project/pull/147323
    a. Can we get the documentation hosted somewhere? Equivalent to openmp.llvm.org?
44. https://github.com/jdoerfert/llvm-project/tree/llvm_kernel_languages
45. Second PGO for GPU patches is ready, third under preparation
    a. https://github.com/llvm/llvm-project/pull/93365
46. GPU ASAN (alternative) prototype ready
    a. Testing out "new-new" design to avoid memory allocations/accesses altogether
    b. Are there any PRs available for review?
    c. Is there a design discussion on discourse?
47. Parallel Thin LTO (WIP)
    a. Second try: https://github.com/jdoerfert/llvm-project/tree/thin_lto
    b. Is this ready for review? Are there public discussions on this?

48. Performance monitoring
    a. https://crpl.cis.udel.edu/lnt-sollve/
    b. https://gitlab.e4s.io/uo-public/llvm-sollve/-/pipelines
        i. Caught: https://github.com/llvm/llvm-project/pull/96909
49. CUDA on LLVM/Offload
    a. https://github.com/llvm/llvm-project/pull/94821
    b. https://github.com/llvm/llvm-project/pull/95371

Wednesday, May 28 2025, 7:00 - 8:00 am PST

Agenda

50. PR Review LIST
51. Liboffload API
    a. PRs related to improving error handling have all been merged
    b. Currently investigating improvements to memory management
        i. PR up for removing the need to pass the allocation type when freeing memory
        ii. Investigating how to implement SYCL/UR contexts on top of liboffload without adding the concept of a memory context to the plugins (no real equivalent in CUDA and HSA)
    c. We now have a UR adapter for Offload - planning on addressing low-hanging fruit in terms of missing functionality
52. https://github.com/jdoerfert/llvm-project/tree/llvm_kernel_languages
53. Second PGO for GPU patches is ready, third under preparation
    a. https://github.com/llvm/llvm-project/pull/93365
54. GPU ASAN (alternative) prototype ready
    a. Testing out “new-new” design to avoid memory allocations/accesses altogether
    b. Are there any PRs available for review?
    c. Is there a design discussion on discourse?
55. Parallel Thin LTO (WIP)
    a. Second try: https://github.com/jdoerfert/llvm-project/tree/thin_lto
    b. Is this ready for review? Are there public discussions on this?
56. Performance monitoring
    a. https://crpl.cis.udel.edu/lnt-sollve/
    b. https://gitlab.e4s.io/uo-public/llvm-sollve/-/pipelines
        i. Caught: https://github.com/llvm/llvm-project/pull/96909
57. CUDA on LLVM/Offload
    a. https://github.com/llvm/llvm-project/pull/94821
    b. https://github.com/llvm/llvm-project/pull/95371

Wednesday, May 14 2025, 7:00 - 8:00 am PST

Agenda

58. PR Review LIST
59. Liboffload API
    a. Work on better error handling from the plugins is ongoing
        i. https://github.com/llvm/llvm-project/pull/138258
        ii. https://github.com/llvm/llvm-project/pull/139275
    b. Currently investigating improvements to memory management
        i. Liboffload tracks all allocations in a DenseMap - this is a workaround we'd like to get rid of
        ii. WIP changes allow plugins to free memory without knowing the allocation type
        iii. The MemoryManager abstraction is awkward - the only way to check if an allocation belongs to it is either knowing the type or looking up the pointer.
        iv. Would it be possible to decouple the generic MemoryManager from the plugin interface? I.e. make it an optional component that libomptarget can use. Individual plugins would still implement DeviceAllocatorTy.
        v. At the very least we need a way to disable it without relying on env vars
60. Second PGO for GPU patches is ready, third under preparation
    a. https://github.com/llvm/llvm-project/pull/93365
61. GPU ASAN (alternative) prototype ready
    a. Testing out "new-new" design to avoid memory allocations/accesses all together
    b. Are there any PRs available for review?
    c. Is there a design discussion on discourse?
62. Parallel Thin LTO (WIP)
    a. Second try: https://github.com/jdoerfert/llvm-project/tree/thin_lto
    b. Is this ready for review? Are there public discussions on this?
63. Performance monitoring
    a. https://crpl.cis.udel.edu/lnt-sollve/
    b. https://gitlab.e4s.io/uo-public/llvm-sollve/-/pipelines
        i. Caught: https://github.com/llvm/llvm-project/pull/96909
64. CUDA on LLVM/Offload
    a. https://github.com/llvm/llvm-project/pull/94821
    b. https://github.com/llvm/llvm-project/pull/95371

Wednesday, April 30 2025, 7:00 - 8:00 am PST

Agenda

65. PR Review LIST
    a. https://github.com/llvm/llvm-project/pull/137339
66. Liboffload API
    a. PR adding the initial API has merged
    b. Also added a check-offload-unit target for the new unit tests
    c. We have an open PR to better handle plugin errors in liboffload.
    d. The liboffload API introduces some error codes (although not an exhaustive set at this point). Should the PluginInterface return these error codes or its own error codes that we translate?
67. Second PGO for GPU patches is ready, third under preparation
    a. https://github.com/llvm/llvm-project/pull/93365
68. GPU ASAN (alternative) prototype ready
    a. Testing out "new-new" design to avoid memory allocations/accesses all together
    b. Are there any PRs available for review?
    c. Is there a design discussion on discourse?
69. Parallel Thin LTO (WIP)
    a. Second try: https://github.com/jdoerfert/llvm-project/tree/thin_lto
    b. Is this ready for review? Are there public discussions on this?
70. Performance monitoring
    a. https://crpl.cis.udel.edu/lnt-sollve/
    b. https://gitlab.e4s.io/uo-public/llvm-sollve/-/pipelines
        i. Caught: https://github.com/llvm/llvm-project/pull/96909
71. CUDA on LLVM/Offload
    a. https://github.com/llvm/llvm-project/pull/94821
    b. https://github.com/llvm/llvm-project/pull/95371
72. Link to the .ics file for the meeting series has expired - could someone reupload it?

Wednesday, April 16 2025, 7:00 - 8:00 am PST

Agenda

73. PR Review LIST
    a. Implement the remaining initial Offload API: https://github.com/llvm/llvm-project/pull/122106

74. Liboffload API
    a. Second PR is ready for review (see earlier link). Enough to get a simple SYCL program working on offload.
        i. Still working on device/platform discovery design - should be finished soon. All other feedback should be resolved.
75. Second PGO for GPU patches is ready, third under preparation
    a. https://github.com/llvm/llvm-project/pull/93365
76. GPU ASAN (alternative) prototype ready
    a. Testing out "new-new" design to avoid memory allocations/accesses all together
    b. Are there any PRs available for review?
    c. Is there a design discussion on discourse?
77. Parallel Thin LTO (WIP)
    a. Second try: https://github.com/jdoerfert/llvm-project/tree/thin_lto
    b. Is this ready for review? Are there public discussions on this?
78. Performance monitoring
    a. https://crpl.cis.udel.edu/lnt-sollve/
    b. https://gitlab.e4s.io/uo-public/llvm-sollve/-/pipelines
        i. Caught: https://github.com/llvm/llvm-project/pull/96909
79. CUDA on LLVM/Offload
    a. https://github.com/llvm/llvm-project/pull/94821
    b. https://github.com/llvm/llvm-project/pull/95371

Wednesday, April 2 2025, 7:00 - 8:00 am PST

Agenda

80. PR Review LIST
    a. Implement the remaining initial Offload API: https://github.com/llvm/llvm-project/pull/122106
81. Liboffload API
    a. Second PR is ready for review (see earlier link). Enough to get a simple SYCL program working on offload.
        i. Feedback should all be addressed now
82. Second PGO for GPU patches is ready, third under preparation
    a. https://github.com/llvm/llvm-project/pull/93365
83. GPU ASAN (alternative) prototype ready
    a. Testing out "new-new" design to avoid memory allocations/accesses all together
    b. Are there any PRs available for review?
    c. Is there a design discussion on discourse?
84. Parallel Thin LTO (WIP)
    a. Second try: https://github.com/jdoerfert/llvm-project/tree/thin_lto
    b. Is this ready for review? Are there public discussions on this?
85. Performance monitoring

a. https://crpl.cis.udel.edu/lnt-sollve/
b. https://gitlab.e4s.io/uo-public/llvm-sollve/-/pipelines
    i. Caught: https://github.com/llvm/llvm-project/pull/96909

86. CUDA on LLVM/Offload
    a. https://github.com/llvm/llvm-project/pull/94821
    b. https://github.com/llvm/llvm-project/pull/95371

Wednesday, Mar 19 2025, 7:00 - 8:00 am PST

Agenda

87. PR Review LIST
    a. Implement the remaining initial Offload API: https://github.com/llvm/llvm-project/pull/122106
88. Liboffload API
    a. Second PR is ready for review (see earlier link). Enough to get a simple SYCL program working on offload.
        i. Feedback should all be addressed now
89. Second PGO for GPU patches is ready, third under preparation
    a. https://github.com/llvm/llvm-project/pull/93365
90. GPU ASAN (alternative) prototype ready
    a. Testing out “new-new” design to avoid memory allocations/accesses all together
    b. Are there any PRs available for review?
    c. Is there a design discussion on discourse?
91. Parallel Thin LTO (WIP)
    a. Second try: https://github.com/jdoerfert/llvm-project/tree/thin_lto
    b. Is this ready for review? Are there public discussions on this?
92. Performance monitoring
    a. https://crpl.cis.udel.edu/lnt-sollve/
    b. https://gitlab.e4s.io/uo-public/llvm-sollve/-/pipelines
        i. Caught: https://github.com/llvm/llvm-project/pull/96909
93. CUDA on LLVM/Offload
    a. https://github.com/llvm/llvm-project/pull/94821
    b. https://github.com/llvm/llvm-project/pull/95371

Wednesday, Feb 19 2025, 7:00 - 8:00 am PST

Agenda

94. PR Review LIST
    a. Implement the remaining initial Offload API: https://github.com/llvm/llvm-project/pull/122106

95. Liboffload API
    a. Second PR is ready for review (see earlier link). Enough to get a simple SYCL program working on offload.
        i. Some changes based on review feedback - reverted plugin changes and added a new version of memcpy that takes a special host device to represent copies to/from host. Old memcpy versions are still included for comparison. Is everyone happy with the new version?
        ii. Follow-up PR includes testing with device binaries for program & kernel testing. WIP but available here: https://github.com/llvm/llvm-project/pull/127803 (only the last commit)
96. Second PGO for GPU patches is ready, third under preparation
    a. https://github.com/llvm/llvm-project/pull/93365
97. GPU ASAN (alternative) prototype ready

Testing out “new-new” design to avoid memory allocations/accesses all together

Wednesday, Mar 05 2025, 7:00 - 8:00 am PST

Same as below.

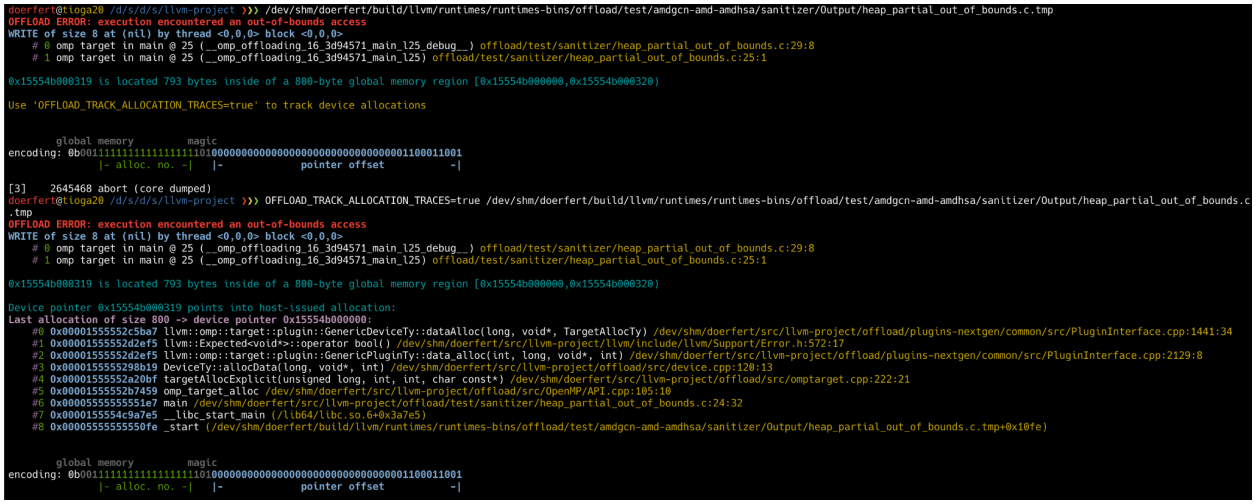
Wednesday, Feb 5 2025, 7:00 - 8:00 am PST

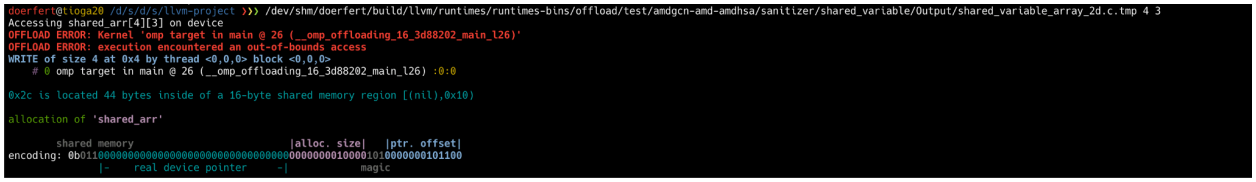
Same as below.

Wednesday, Jan 22 2025, 7:00 - 8:00 am PST

Agenda

98. Port DeviceRTL to C++ https://github.com/llvm/llvm-project/pull/123673
99. PR Review LIST
100. Second PGO for GPU patches is ready, third under preparation
    a. https://github.com/llvm/llvm-project/pull/93365
101. GPU ASAN (alternative) prototype ready
    a. bad/double-free and kernel traces merged
    b. Testing out “new-new” design to avoid memory allocations/accesses all together





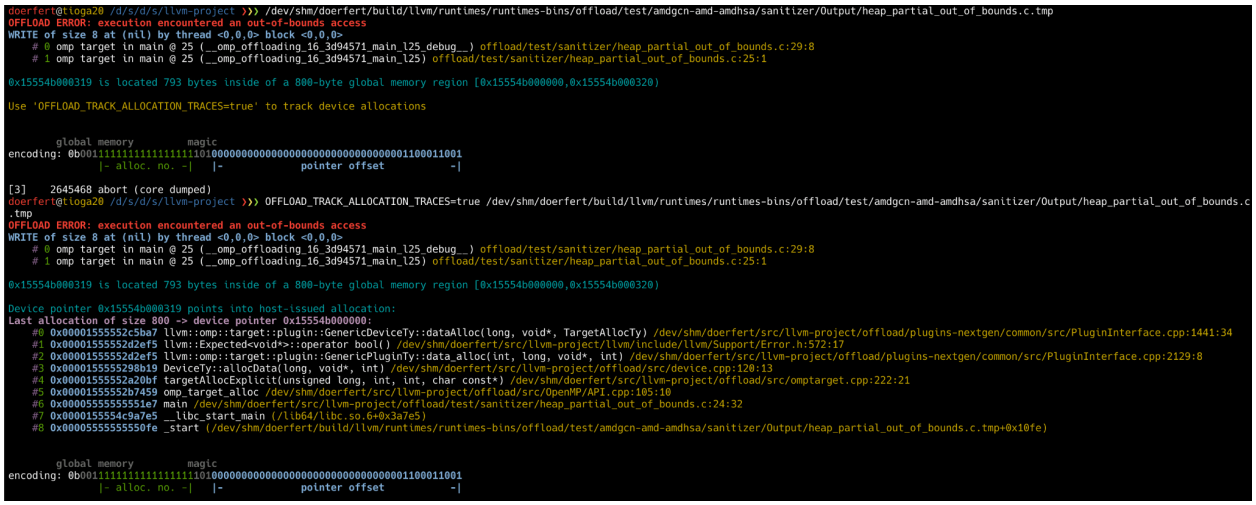
102. Parallel Thin LTO (WIP)
    a. Second try: https://github.com/jdoerfert/llvm-project/tree/thin_lto
103. Performance monitoring
    a. https://crpl.cis.udel.edu/lnt-sollve/
    b. https://gitlab.e4s.io/uo-public/llvm-sollve/-/pipelines
        i. Caught: https://github.com/llvm/llvm-project/pull/96909
104. CUDA on LLVM/Offload
    a. https://github.com/llvm/llvm-project/pull/94821
    b. https://github.com/llvm/llvm-project/pull/95371
105. Initial PR for new API and tablegen tooling has finally merged
    a. Thanks to everyone for the help with the various reviews, reverts and getting it properly landed eventually
    b. Draft follow-up PR to demonstrate the new tooling: https://github.com/llvm/llvm-project/pull/119549
        i. Shows the minimal changes required to add to the API, and the resulting auto-generated files (in the include/generated folder)
    c. Second follow up that implements the remaining initial API: https://github.com/llvm/llvm-project/pull/122106
        i. Still WIP but enough to run a basic SYCL program!
        ii. Intend to finish this soon. Want to confirm - are we ok with (many, ongoing) breaking changes until we reach some kind of stability? Above PRs are

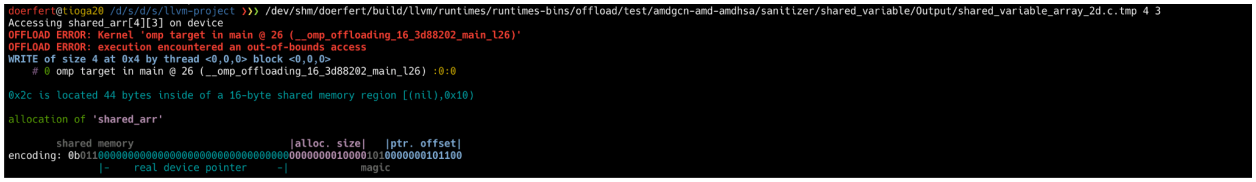
based on the existing plugin design; don't want to be stuck if we decide to change things.

Next Meeting: Wednesday, Jan 08 2025, 7:00 - 8:00 am PST

Agenda

106. PR Review LIST
107. Second PGO for GPU patches is ready, third under preparation
    a. https://github.com/llvm/llvm-project/pull/93365
108. GPU ASAN (alternative) prototype ready
    a. bad/double-free and kernel traces merged
    b. Testing out "new-new" design to avoid memory allocations/accesses all together





109. Parallel Thin LTO (WIP)
    a. Second try: https://github.com/jdoerfert/llvm-project/tree/thin_lto
110. Performance monitoring
    a. https://crpl.cis.udel.edu/lnt-sollve/
    b. https://gitlab.e4s.io/uo-public/llvm-sollve/-/pipelines
        i. Caught: https://github.com/llvm/llvm-project/pull/96909
111. CUDA on LLVM/Offload

   a. https://github.com/llvm/llvm-project/pull/94821
   b. https://github.com/llvm/llvm-project/pull/95371
112. Testing
   a. Keep a list of issues we discuss in the meetings
   b. Make 1-3 people the “bug keepers” (Joseph, Johannes)
113. Initial PR for new API and tablegen tooling has finally merged
   a. Thanks to everyone for the help with the various reviews, reverts and getting it properly landed eventually
   b. Draft follow-up PR to demonstrate the new tooling: https://github.com/llvm/llvm-project/pull/119549
      i. Shows the minimal changes required to add to the API, and the resulting auto-generated files (in the include/generated folder)
   c. Second follow up that implements the remaining initial API: https://github.com/llvm/llvm-project/pull/122106
      i. Still WIP but enough to run a basic SYCL program!

Intend to finish this soon. Want to confirm - are we ok with (many, ongoing) breaking changes until we reach some kind of stability? Above PRs are based on the existing plugin design; don’t want to be stuck if we decide to change things.

Previous Meeting: Wednesday, Nov 13, 2024, 7:00 - 8:00 am PST

No change, same as oct 16 2024

Previous Meeting: Wednesday, Oct 30, 2024, 7:00 - 8:00 am PST

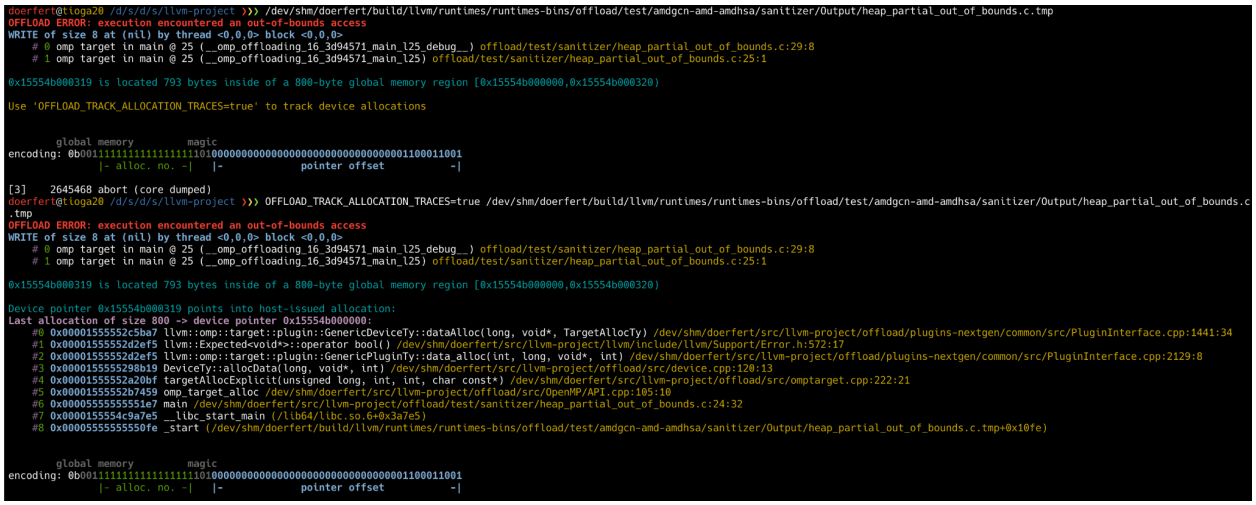
No change, same as oct 16 2024

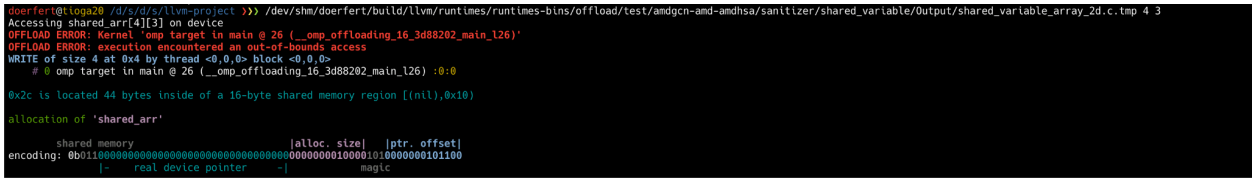
Previous Meeting: Wednesday, Oct 16, 2024, 7:00 - 8:00 am PST

Agenda

114. Second PGO for GPU patches is ready, third under preparation
   a. https://github.com/llvm/llvm-project/pull/93365
115. GPU ASAN (alternative) prototype ready

a. bad/double-free and kernel traces merged
b. Testing out “new-new” design to avoid memory allocations/accesses all together





116. Parallel Thin LTO (WIP)
    a. Second try: https://github.com/jdoerfert/llvm-project/tree/thin_lto
117. Performance monitoring
    a. https://crpl.cis.udel.edu/lnt-sollve/
    b. https://gitlab.e4s.io/uo-public/llvm-sollve/-/pipelines
        i. Caught: https://github.com/llvm/llvm-project/pull/96909
118. CUDA on LLVM/Offload
    a. https://github.com/llvm/llvm-project/pull/94821
    b. https://github.com/llvm/llvm-project/pull/95371
119. Testing
    a. Keep a list of issues we discuss in the meetings
    b. Make 1-3 people the “bug keepers” (Joseph, Johannes)
120. Initial PR for new API and tablegen tooling is up:
    https://github.com/llvm/llvm-project/pull/108413
    a. New updates (Oct 2nd):
        i. Generated files are checked in
        ii. Error handling is improved
        iii. Unit tests added
    b. Open items:

   i. Decide on naming convention (offloadFoo is too verbose, camel case vs snake case)
   ii. API versioning (tied to LLVM version?)
121. New RFC for SYCL upstreaming - https://discourse.llvm.org/t/rfc-sycl-runtime-upstreaming-questions/80323

Previous Meeting: Wednesday, Oct 02, 2024, 7:00 - 8:00 am PST

Agenda

122. Second PGO for GPU patches is ready, third under preparation
   a. https://github.com/llvm/llvm-project/pull/93365
123. GPU ASAN (alternative) prototype ready
   a. bad/double-free and kernel traces merged
   b. Testing out "new-new" design to avoid memory allocations/accesses all together

```
=================================================================
ERROR: OffloadSanitizer out-of-bounds access on address 0x155547402008 at pc 0x155547e064a4
READ of size 8 at 0x155547402008 thread <1, 0, 0> block <0, 0, 0>
    #0 0x155547e064a4 c_rush_larsen_gpu_omp_l1600_debug___omp_outlined in rush_larsen_gpu_hip.cc:1610

0x155547402008 is located 8 bytes inside of a 8-byte region [0x155547402000,0x155547402008)
AMDGPU fatal error 1: Received error in queue 0x15554c1f4000: HSA_STATUS_ERROR_EXCEPTION: An HSAIL operation resulted in a hardware exception.
```

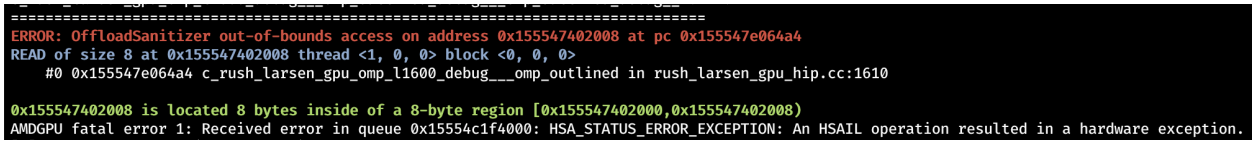
124. Parallel Thin LTO (WIP)
   a. Second try: https://github.com/jdoerfert/llvm-project/tree/thin_lto
125. Performance monitoring
   a. https://crpl.cis.udel.edu/lnt-sollve/
   b. https://gitlab.e4s.io/uo-public/llvm-sollve/-/pipelines
      i. Caught: https://github.com/llvm/llvm-project/pull/96909
126. CUDA on LLVM/Offload
   a. https://github.com/llvm/llvm-project/pull/94821
   b. https://github.com/llvm/llvm-project/pull/95371
127. Testing
   a. Keep a list of issues we discuss in the meetings
   b. Make 1-3 people the "bug keepers" (Joseph, Johannes)
128. Initial PR for new API and tablegen tooling is up: https://github.com/llvm/llvm-project/pull/108413
   a. New updates (Oct 2nd):
      i. Generated files are checked in
      ii. Error handling is improved
      iii. Unit tests added
   b. Open items:
      i. Decide on naming convention (offloadFoo is too verbose, camel case vs snake case)
      ii. API versioning (tied to LLVM version?)

129. New RFC for SYCL upstreaming - https://discourse.llvm.org/t/rfc-sycl-runtime-upstreaming-questions/80323

Previous Meeting: Wednesday, Sep 18, 2024, 7:00 - 8:00 am PST

Agenda

130. Draft PR to enable offload in precommit CI
    a. https://github.com/llvm/llvm-project/pull/109103
131. Second PGO for GPU patches is ready, third under preparation
    a. https://github.com/llvm/llvm-project/pull/76587
    b. https://github.com/llvm/llvm-project/pull/93365
132. GPU ASAN (alternative) prototype ready
    a. bad/double-free and kernel traces merged
    b. Testing out "new-new" design to avoid memory allocations/accesses all together



133. Parallel Thin LTO (WIP)
    a. Second try: https://github.com/jdoerfert/llvm-project/tree/thin_lto
134. Performance monitoring
    a. https://crpl.cis.udel.edu/lnt-sollve/
    b. https://gitlab.e4s.io/uo-public/llvm-sollve/-/pipelines
        i. Caught: https://github.com/llvm/llvm-project/pull/96909
135. CUDA on LLVM/Offload
    a. https://github.com/llvm/llvm-project/pull/94821
    b. https://github.com/llvm/llvm-project/pull/95371
136. Testing
    a. Keep a list of issues we discuss in the meetings
    b. Make 1-3 people the "bug keepers" (Joseph, Johannes)
137. Initial PR for new API and tablegen tooling is up: https://github.com/llvm/llvm-project/pull/108413
138. New RFC for SYCL upstreaming - https://discourse.llvm.org/t/rfc-sycl-runtime-upstreaming-questions/80323

Previous Meeting: Wednesday, Sep 4, 2024, 7:00 - 8:00 am PST

Agenda

139. Second PGO for GPU patches is ready, third under preparation
    a. https://github.com/llvm/llvm-project/pull/76587
    b. https://github.com/llvm/llvm-project/pull/93365
140. GPU ASAN (alternative) prototype ready
    a. bad/double-free and kernel traces merged
    b. Testing out “new-new” design to avoid memory allocations/accesses all together

```
=================================================================================
ERROR: OffloadSanitizer out-of-bounds access on address 0x155547402008 at pc 0x155547e064a4
READ of size 8 at 0x155547402008 thread <1, 0, 0> block <0, 0, 0>
    #0 0x155547e064a4 c_rush_larsen_gpu_omp_l1600_debug___omp_outlined in rush_larsen_gpu_hip.cc:1610

0x155547402008 is located 8 bytes inside of a 8-byte region [0x155547402000,0x155547402008)
AMDGPU fatal error 1: Received error in queue 0x15554c1f4000: HSA_STATUS_ERROR_EXCEPTION: An HSAIL operation resulted in a hardware exception.
```

141. Parallel Thin LTO (WIP)
    a. Second try: https://github.com/jdoerfert/llvm-project/tree/thin_lto
142. Performance monitoring
    a. https://crpl.cis.udel.edu/lnt-sollve/
    b. https://gitlab.e4s.io/uo-public/llvm-sollve/-/pipelines
        i. Caught: https://github.com/llvm/llvm-project/pull/96909
143. CUDA on LLVM/Offload
    a. https://github.com/llvm/llvm-project/pull/94821
    b. https://github.com/llvm/llvm-project/pull/95371
144. Testing
    a. Keep a list of issues we discuss in the meetings
    b. Make 1-3 people the “bug keepers” (Joseph, Johannes)
145. offload-tblgen
    a. Current ongoing work to create a first PR that:
        i. Introduces offload-tblgen
        ii. Implements auto-generated validation and tracing
        iii. Implements minimal new API functions needed to run sycl-ls / urinfo via Unified Runtime (basically just device discovery and device property querying)
    b. Not quite finished yet but intend to present at the next meeting
146. New RFC for SYCL upstreaming - https://discourse.llvm.org/t/rfc-sycl-runtime-upstreaming-questions/80323

Next Meeting: Wednesday, Aug 21, 2024, 7:00 - 8:00 am PST

Agenda

147. LLVM Dev deadline approaching
148. Second PGO for GPU patches is ready, third under preparation
    a. https://github.com/llvm/llvm-project/pull/76587
    b. https://github.com/llvm/llvm-project/pull/93365
149. GPU ASAN (alternative) prototype ready
    a. bad/double-free and kernel traces merged
    b. Testing out "new-new" design to avoid memory allocations/accesses all together

```
=================================================================
ERROR: OffloadSanitizer out-of-bounds access on address 0x155547402008 at pc 0x155547e064a4
READ of size 8 at 0x155547402008 thread <1, 0, 0> block <0, 0, 0>
    #0 0x155547e064a4 c_rush_larsen_gpu_omp_l1600_debug___omp_outlined in rush_larsen_gpu_hip.cc:1610

0x155547402008 is located 8 bytes inside of a 8-byte region [0x155547402000,0x155547402008)
AMDGPU fatal error 1: Received error in queue 0x15554c1f4000: HSA_STATUS_ERROR_EXCEPTION: An HSAIL operation resulted in a hardware exception.
```

150. Parallel Thin LTO (WIP)
    a. Second try: https://github.com/jdoerfert/llvm-project/tree/thin_lto
151. Performance monitoring
    a. https://crpl.cis.udel.edu/lnt-sollve/
    b. https://gitlab.e4s.io/uo-public/llvm-sollve/-/pipelines
        i. Caught: https://github.com/llvm/llvm-project/pull/96909
152. CUDA on LLVM/Offload
    a. https://github.com/llvm/llvm-project/pull/94821
    b. https://github.com/llvm/llvm-project/pull/95371
153. Testing
    a. Keep a list of issues we discuss in the meetings
    b. Make 1-3 people the "bug keepers" (Joseph, Johannes)
154. offload-tblgen
    a. Current ongoing work to create a first PR that:
        i. Introduces offload-tblgen
        ii. Implements auto-generated validation and tracing
        iii. Implements minimal new API functions needed to run sycl-ls / urinfo via Unified Runtime (basically just device discovery and device property querying)
    b. Not quite finished yet but intend to present at the next meeting
155. New RFC for SYCL upstreaming - https://discourse.llvm.org/t/rfc-sycl-runtime-upstreaming-questions/80323
156.

Meeting: Wednesday, Aug 07, 2024, 7:00 - 8:00 am PST

Agenda

157. LLVM Dev deadline approaching
158. Second PGO for GPU patches is ready, third under preparation
    a. https://github.com/llvm/llvm-project/pull/76587
    b. https://github.com/llvm/llvm-project/pull/93365
159. GPU ASAN (alternative) prototype ready
    a. bad/double-free and kernel traces merged
    b. Testing out “new-new” design to avoid memory allocations/accesses all together

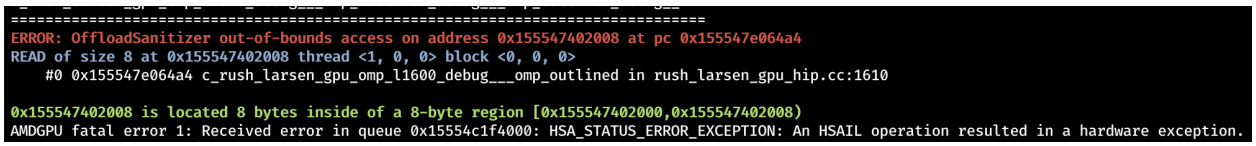
```
=================================================================
ERROR: OffloadSanitizer out-of-bounds access on address 0x155547402008 at pc 0x155547e064a4
READ of size 8 at 0x155547402008 thread <1, 0, 0> block <0, 0, 0>
    #0 0x155547e064a4 c_rush_larsen_gpu_omp_l1600_debug___omp_outlined in rush_larsen_gpu_hip.cc:1610

0x155547402008 is located 8 bytes inside of a 8-byte region [0x155547402000,0x155547402008)
AMDGPU fatal error 1: Received error in queue 0x15554c1f4000: HSA_STATUS_ERROR_EXCEPTION: An HSAIL operation resulted in a hardware exception.
```

160. Parallel Thin LTO (WIP)
    a. Second try: https://github.com/jdoerfert/llvm-project/tree/thin_lto
161. Performance monitoring
    a. https://crpl.cis.udel.edu/lnt-sollve/
    b. https://gitlab.e4s.io/uo-public/llvm-sollve/-/pipelines
        i. Caught: https://github.com/llvm/llvm-project/pull/96909
162. CUDA on LLVM/Offload
    a. https://github.com/llvm/llvm-project/pull/94821
    b. https://github.com/llvm/llvm-project/pull/95371
163. Testing
    a. Keep a list of issues we discuss in the meetings
    b. Make 1-3 people the “bug keepers” (Joseph, Johannes)
164. offload-tblgen
    a. https://github.com/llvm/llvm-project/pull/88923 (Please review)
    b. https://gist.github.com/jhuber6/2117cfd03b7c78d91f5481ac63da682d
        i. API draft Joseph typed up
    c. What are the next steps for introducing API changes?
        i. Design the entire new API before beginning to implement it? (By porting the existing plugins?)
        ii. Move the existing plugins API to the tablegen framework and then introduce changes?
        iii. Ignore the tablegen framework for now and change the existing plugin API directly?
165. New RFC for SYCL upstreaming - https://discourse.llvm.org/t/rfc-sycl-runtime-upstreaming-questions/80323

Meeting: Wednesday, July 24, 2024, 7:00 - 8:00 am PST

Agenda

166. Second PGO for GPU patches is ready, third under preparation
    a. https://github.com/llvm/llvm-project/pull/76587
    b. https://github.com/llvm/llvm-project/pull/93365
167. GPU ASAN (alternative) prototype ready

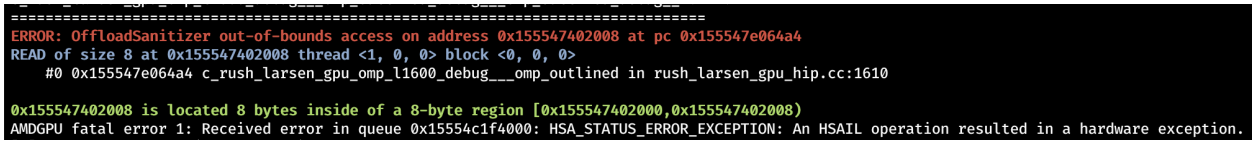


168. Performance monitoring
    a. https://crpl.cis.udel.edu/lnt-sollve/
    b. https://gitlab.e4s.io/uo-public/llvm-sollve/-/pipelines
        i. Caught: https://github.com/llvm/llvm-project/pull/96909
169. CUDA on LLVM/Offload
    a.
    b. https://github.com/llvm/llvm-project/pull/94821
    c. https://github.com/llvm/llvm-project/pull/95371
170. Testing
    a. Keep a list of issues we discuss in the meetings
    b. Make 1-3 people the “bug keepers” (Joseph, Johannes)
171. offload-tblgen
    a. https://github.com/llvm/llvm-project/pull/88923 (Please review)
    b. https://gist.github.com/jhuber6/2117cfd03b7c78d91f5481ac63da682d
        i. API draft Joseph typed up
    c. What are the next steps for introducing API changes?
        i. Design the entire new API before beginning to implement it? (By porting the existing plugins?)
        ii. Move the existing plugins API to the tablegen framework and then introduce changes?
        iii. Ignore the tablegen framework for now and change the existing plugin API directly?
172. Parallel Thin LTO (WIP)
    https://github.com/ggeorgakoudis/llvm-project/tree/hip-omp-parallel-thinlto-clean
173. New RFC for SYCL upstreaming -
    https://discourse.llvm.org/t/rfc-sycl-runtime-upstreaming-questions/80323

Meeting: Wednesday, July 10, 2024, 7:00 - 8:00 am PST

Agenda

174. Second PGO for GPU patches is ready, third under preparation
    a. https://github.com/llvm/llvm-project/pull/76587
    b. https://github.com/llvm/llvm-project/pull/93365
175. GPU ASAN (alternative) prototype ready



176. Performance monitoring
    a. https://crpl.cis.udel.edu/lnt-sollve/
    b. https://gitlab.e4s.io/uo-public/llvm-sollve/-/pipelines
        i. Caught: https://github.com/llvm/llvm-project/pull/96909
177. CUDA on LLVM/Offload
    a. https://github.com/llvm/llvm-project/pull/94549
    b. https://github.com/llvm/llvm-project/pull/94821
    c. https://github.com/llvm/llvm-project/pull/95371
178. Testing
    a. Keep a list of issues we discuss in the meetings
    b. Make 1-3 people the “bug keepers” (Joseph, Johannes)
179. offload-tblgen
    a. https://github.com/llvm/llvm-project/pull/88923 (Please review)
    b. https://gist.github.com/jhuber6/2117cfd03b7c78d91f5481ac63da682d
        i. API draft Joseph typed up
    c. What are the next steps for introducing API changes?
        i. Design the entire new API before beginning to implement it? (By porting the existing plugins?)
        ii. Move the existing plugins API to the tablegen framework and then introduce changes?
        iii. Ignore the tablegen framework for now and change the existing plugin API directly?
180. Parallel Thin LTO (WIP)
    https://github.com/ggeorgakoudis/llvm-project/tree/hip-omp-parallel-thinlto-clean
181. Draft RFC on upstreaming the SYCL runtime with a short term dependency on UR

a. https://docs.google.com/document/d/1QI8opRuabWASxqe8Lu_dGLVO1dXlibzgBi_M-UEHxyw/edit?usp=sharing
b. Feedback welcome before the RFC is finished and shared

Meeting: Wednesday, June 26, 2024, 7:00 - 8:00 am PST

Agenda

182. Second PGO for GPU patches is ready, third under preparation
    a. https://github.com/llvm/llvm-project/pull/76587
    b. https://github.com/llvm/llvm-project/pull/93365
183. Unified Memory Framework (Intel)
184. GPU ASAN (alternative) prototype ready

```
=================================================================================
ERROR: OffloadSanitizer out-of-bounds access on address 0x155547402008 at pc 0x155547e064a4
READ of size 8 at 0x155547402008 thread <1, 0, 0> block <0, 0, 0>
    #0 0x155547e064a4 c_rush_larsen_gpu_omp_l1600_debug___omp_outlined in rush_larsen_gpu_hip.cc:1610

0x155547402008 is located 8 bytes inside of a 8-byte region [0x155547402000,0x155547402008)
AMDGPU fatal error 1: Received error in queue 0x15554c1f4000: HSA_STATUS_ERROR_EXCEPTION: An HSAIL operation resulted in a hardware exception.
```

185. Performance monitoring
    a. https://crpl.cis.udel.edu/lnt-sollve/
    b. https://gitlab.e4s.io/uo-public/llvm-sollve/-/pipelines
186. Testing
    a. Keep a list of issues we discuss in the meetings
    b. Make 1-3 people the “bug keepers” (Joseph, Johannes)
187. offload-tblgen
    a. https://github.com/llvm/llvm-project/pull/88923 (Please review)
    b. https://gist.github.com/jhuber6/2117cfd03b7c78d91f5481ac63da682d
        i. API draft Joseph typed up
    c. What are the next steps for introducing API changes?
        i. Design the entire new API before beginning to implement it? (By porting the existing plugins?)
        ii. Move the existing plugins API to the tablegen framework and then introduce changes?
        iii. Ignore the tablegen framework for now and change the existing plugin API directly?
188. Parallel Thin LTO (WIP)
    https://github.com/ggeorgakoudis/llvm-project/tree/hip-omp-parallel-thinlto-clean
189. Does the meeting invite ICS file need to be updated?
    a. Currently appearing at 8am PST
190. Draft RFC on upstreaming the SYCL runtime with a short term dependency on UR
    a. https://docs.google.com/document/d/1QI8opRuabWASxqe8Lu_dGLVO1dXlibzgBi_M-UEHxyw/edit?usp=sharing

Feedback welcome before the RFC is finished and shared

Meeting: Wednesday, June 12, 2024, 7:00 - 8:00 am PST

Agenda

191. Second PGO for GPU patches is ready, third under preparation
    a. https://github.com/llvm/llvm-project/pull/76587
    b. https://github.com/llvm/llvm-project/pull/93365
192. Unified Memory Framework (Intel)
193. Performance monitoring
    a. https://crpl.cis.udel.edu/lnt-sollve/
    b. https://gitlab.e4s.io/uo-public/llvm-sollve/-/pipelines
194. Testing
    a. Keep a list of issues we discuss in the meetings
    b. Make 1-3 people the “bug keepers” (Joseph, Johannes)
195. offload-tblgen
    a. https://github.com/llvm/llvm-project/pull/88923 (Please review)
    b. https://gist.github.com/jhuber6/2117cfd03b7c78d91f5481ac63da682d
        i. API draft Joseph typed up
    c. What are the next steps for introducing API changes?
        i. Design the entire new API before beginning to implement it? (By porting the existing plugins?)
        ii. Move the existing plugins API to the tablegen framework and then introduce changes?
        iii. Ignore the tablegen framework for now and change the existing plugin API directly?
196. Parallel Thin LTO (WIP)
    https://github.com/ggeorgakoudis/llvm-project/tree/hip-omp-parallel-thinlto-clean
197. Does the meeting invite ICS file need to be updated?
    a. Currently appearing at 8am PST
198. Draft RFC on upstreaming the SYCL runtime with a short term dependency on UR
    a. https://docs.google.com/document/d/1QI8opRuabWASxqe8Lu_dGLVO1dXlibzgBi_M-UEHxyw/edit?usp=sharing
    b. Feedback welcome before the RFC is finished and shared

Meeting: Wednesday, May 29, 2024, 7:00 - 8:00 am PST

Agenda

199. Build regressions:
https://github.com/llvm/llvm-project/issues/75124#issuecomment-2106147001

    Probably a 32bit build issue which is not supported

200. Second PGO for GPU patches is ready, third under preparation
    a. https://github.com/llvm/llvm-project/pull/76587
    b. https://github.com/llvm/llvm-project/pull/93365
201. Unified Memory Framework (Intel)
    a. Intel to present in 2 weeks
202. Performance monitoring
    a. https://crpl.cis.udel.edu/lnt-sollve/
    b. https://gitlab.e4s.io/uo-public/llvm-sollve/-/pipelines
203. Testing
    a. Keep a list of issues we discuss in the meetings
    b. Make 1-3 people the "bug keepers" (Joseph, Johannes)
204. offload-tblgen
    a. https://github.com/llvm/llvm-project/pull/88923 (Please review)
    b. https://gist.github.com/jhuber6/2117cfd03b7c78d91f5481ac63da682d
        i. API draft Joseph typed up
    c. What are the next steps for introducing API changes?
        i. Design the entire new API before beginning to implement it? (By porting the existing plugins?)
        ii. Move the existing plugins API to the tablegen framework and then introduce changes?
        iii. Ignore the tablegen framework for now and change the existing plugin API directly?
205. Parallel Thin LTO (WIP)
https://github.com/ggeorgakoudis/llvm-project/tree/hip-omp-parallel-thinlto-clean
206. Does the meeting invite ICS file need to be updated?
    a. Currently appearing at 8am PST

Meeting: Wednesday, May 15, 2024, 7:00 - 8:00 am PST

Agenda

207. Build regressions:
https://github.com/llvm/llvm-project/issues/75124#issuecomment-2106147001

208. Statically link plugins has landed
209. Offload Buildbot:
    a. https://lab.llvm.org/staging/#/builders/191
    b. Non-buildbot testing pipelines: https://gitlab.e4s.io/uo-public/llvm-sollve/-/pipelines
210. Testing
    a. Keep a list of issues we discuss in the meetings
    b. Make 1-3 people the "bug keepers" (Joseph, Johannes)
211. GPU/Device PGO working, PRs are being prepared
212. offload-tblgen
    a. https://github.com/llvm/llvm-project/pull/88923 (Please review)
213. Parallel Thin LTO (WIP)
    https://github.com/ggeorgakoudis/llvm-project/tree/hip-omp-parallel-thinlto-clean
214. Adding SYCL offload support (Intel : Arvind)
215. Does the meeting invite ICS file need to be updated?

Currently appearing at 6am PST**Meeting:** **Wednesday, May 1, 2024, 7:00 - 8:00 am PST**

Agenda

216. Statically link plugins
    a. https://github.com/llvm/llvm-project/pull/87009
        i. Reviews please :)
217. New PR for the move is done. Zorg patches are up:
    a. Ompt + standalone built issues being looked at
        i. https://github.com/llvm/llvm-project/pull/88957 builds on top of #75125
218. Offload Buildbot:
    a. https://lab.llvm.org/staging/#/builders/191
    b. Non-buildbot testing pipelines: https://gitlab.e4s.io/uo-public/llvm-sollve/-/pipelines
219. Testing
    a. Keep a list of issues we discuss in the meetings
    b. Make 1-3 people the "bug keepers" (Joseph, Johannes)
220. GPU/Device PGO working, PRs are being prepared
221. offload-tblgen
    a. https://github.com/llvm/llvm-project/pull/88923 (Please review)
222. Parallel Thin LTO (WIP)
    https://github.com/ggeorgakoudis/llvm-project/tree/hip-omp-parallel-thinlto-clean
223. Adding SYCL offload support (Intel : Arvind)

Meeting: Wednesday, April 17, 2024, 7:00 - 8:00 am PST

Agenda

224. Statically link plugins
    a. https://github.com/llvm/llvm-project/pull/87009
        i. Reviews please :)
225. New PR for the move is done. Zorg patches are up:
    a. https://github.com/llvm/llvm-project/pull/77154
    b. https://github.com/llvm/llvm-project/pull/75125
        i. llvm org mailing list
    c. Ompt + standalone built issues being looked at
        i. https://github.com/llvm/llvm-project/pull/88957 builds on top of #75125
226. Offload Buildbot:
    a. https://lab.llvm.org/staging/#/builders/191
227. Testing
    a. Keep a list of issues we discuss in the meetings
    b. Make 1-3 people the “bug keepers”
228. GPU/Device PGO working, PRs are being prepared
229. offload-tblgen
    a. https://github.com/llvm/llvm-project/pull/88923

Parallel Thin LTO (WIP)

https://github.com/ggeorgakoudis/llvm-project/tree/hip-omp-parallel-thinlto

Meeting: Wednesday, April 3, 2024, 7:00 - 8:00 am PST

Agenda

230. Statically link plugins
    a. https://github.com/llvm/llvm-project/pull/87009
        i. Reviews please :)
231. New PR for the move is done. Zorg patches are up:
    ~~a. https://github.com/llvm/llvm-zorg/pull/141~~
    ~~b. https://github.com/llvm/llvm-zorg/pull/142~~
    c. https://github.com/llvm/llvm-project/pull/77154
    d. https://github.com/llvm/llvm-project/pull/75125
        i. Llvm org mailing list
    e. Ompt + standalone built issues being looked at
232. Offload Buildbot:
    a. https://lab.llvm.org/staging/#/builders/191

233. Testing
    a. Keep a list of issues we discuss in the meetings
    b. Make 1-3 people the “bug keepers”

GPU/Device PGO working, PRs are being prepared

Meeting: Wednesday, March 20, 2024, 7:00 - 8:00 am PST

Agenda

234. New PR for the move is done. Zorg patches are up:
    a. ~~https://github.com/llvm/llvm-zorg/pull/141~~
    b. ~~https://github.com/llvm/llvm-zorg/pull/142~~
    c. https://github.com/llvm/llvm-project/pull/77154
    d. https://github.com/llvm/llvm-project/pull/75125
        i. Llvm org mailing list
235. PGO for GPUs is a working end-to-end prototype now
    a. https://github.com/EthanLuisMcDonough/llvm-project/tree/gpuprofdriver
    b.
```
# Compile with PGO instrumentation
clang -L/dev/shm/YOURNAME/clang/install/lib
-L/dev/shm/YOURNAME/clang/install/lib/x86_64-unknown-linux-gnu -fopenmp
--offload-arch=native -fprofile-generate-gpu YOUR_FILE.c
/dev/shm/YOURNAME/clang/install/lib/x86_64-unknown-linux-gnu/libomptarget.rtl.YOUR_GPU_TARGET.so
# Reprocess data
llvm-profdata merge YOUR_GPU_TARGET.default.profraw -o YOUR_GPU_TARGET.profdata
# Recompile
clang -L/dev/shm/YOURNAME/clang/install/lib
-L/dev/shm/YOURNAME/clang/install/lib/x86_64-unknown-linux-gnu -fopenmp
--offload-arch=native -fprofile-use-gpu=YOUR_GPU_TARGET.profdata YOUR_FILE.c
```
236. Tablegen tooling update
237. Testing
    a. Keep a list of issues we discuss in the meetings

Make 1-3 people the “bug keepers”

Meeting: Wednesday, March 06, 2024, 7:00 - 8:00 am PST

Agenda

238. New PR for the move
    a. https://github.com/llvm/llvm-project/pull/82459

JP can build liboffload.

Add steps in PR on what was done to help downstream to recreate the steps

239. [Offload][NFC] Add offload subfolder and README #77154
    a. Meeting time needs to be updated (ICS file also?)
    b. Should land well in advance before #82459
    c. Github llvm teams (pr-subscribers, etc)

Meeting: Wednesday, Feb 21, 2024, 7:00 - 8:00 am PST

Agenda

240. API Tooling (auto generation of headers, documentation, etc)
    a. Brief overview of how we do this in Unified Runtime
241. New PR for the move
    a. https://github.com/llvm/llvm-project/pull/82459
242.

Next Meeting: Wednesday, Feb 7, 2024, 7:00 - 8:00 am PST

Agenda

243. Removing OpenMP specific operations from the plugin interface
244. Establishing a consistent init / shutdown order (Stop using global constructors)
245. Removing the dynamically opened plugin interface
    a. Static libraries instead
246. LLVM/libc things to move into LLVM/offload?

Next Meeting: Wednesday, Jan 24, 2024, 7:00 - 8:00 am PST

Agenda (35 people)

1. Welcome
2. Ics file is broken. Johannes will try again.
3. Offload folder PR: https://github.com/llvm/llvm-project/pull/77154
4. Possible API discussions
    a. https://gist.github.com/jhuber6/2117cfd03b7c78d91f5481ac63da682d

   b. Discussion started, see issue: https://github.com/llvm/llvm-project/issues/79304
      i. Stability guarantee?
      ii. Error handling
      iii. Nameing, prefix, suffix
      iv. Source locations
      v. Count API invocations
      vi. Coding style:
         1. Clang format everything
         2. C++, rules of LLVM
         3. API is C?
      vii. “Language” libraries on top of plugin API:
         1. Libllvmomptarget.so, Libllvmkernel.so, Libllvmjulia.so, …
      viii. Unit testing, coverage testing
5. Next steps:
   a. Do we need a separate offload pr subscribe team, something like https://github.com/orgs/llvm/teams/pr-subscribers-offload?
      i. Shilei will take care as soon as the folders exist
      ii. Labeling (libomptarget, offload)
   b. Should we use discourse RFC for key component design discussion instead of llvm issues?
   c. ZORG offload builder (as template for people), in addition to OpenMP offload builder.
   d. Backward compatibility of user exposed env variables (https://openmp.llvm.org/design/Runtimes.html#libopenmptarget-environment-vars)

Previous Meeting: Fri, Jan 12, 2024, 9:00 - 10:00 am PST

Agenda (27 people)

1. Welcome

    - Inline **notes**
2. Future meetings
    - Wednesday 7am pacific, every two weeks.
        - Starting Jan 24, 24
        - Updates to the OpenMP invite will happen too, two separate invites.
    - Alternating with the OpenMP meeting
3. Webpage, docs, etc.
    - Talk to Anton for webpage
    - Offload folder PR: https://github.com/llvm/llvm-project/pull/77154
        - Please review
    - Sphinx webpage, like Clang and OpenMP (.llvm.org)
4. Initial PRs
    - Offload folder PR: https://github.com/llvm/llvm-project/pull/77154
    - Move libomptarget PR: https://github.com/llvm/llvm-project/pull/75125
        - Plan: move then “restore”; Johannes
        - Driver switch to pickup liboffload; Joseph
            - Offload will become the default path for OpenMP shortly after
5. Naming
    - “offload” wins.
6. Working groups
    - API Design
        - Error model, source information, …
    - Testing
        - Buildbots, unit tests, lit tests
7. Rust compiler, offload linking
    - Will this make it easier for the rust compiler
    - Should we include/serve (NVIDIA, AMD, …) libdevice?
        - Will talk to Artem
8.