LINMA2710 - Scientific Computing
Graphics processing unit (GPU)

P.-A. Absil and B. Legat

☐ Full Width Mode ☐ Present Mode

Table of Contents

Sources

- OpenCL.jl
- HandsOnOpenCL
- Optimizing Parallel Reduction in CUDA
- Parallel Computation Patterns (Reduction)
- Profiling, debugging and optimization
- How to use TAU for Performance Analysis

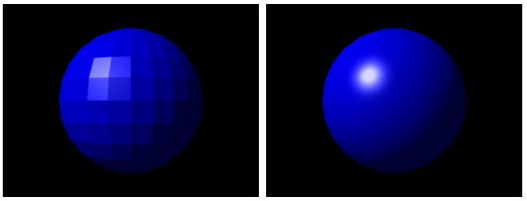
Introduction

Context

- Most *dedicated* GPUs produced by NVIDIA and AMD
- *Integrated* GPUs by intel used in laptops to reduce power consumption
- Designed for 3D rendering through ones of the APIs : DirectX, OpenGL, WebGL, WebGPU, Vulkan or Apple's Metal
- Illustration on the right is from Charge's film, it show how 3D modeling work.

General-Purpose computing on GPU (GPGPU)

Also known as *compute shader* as they abuses the programmable shading of GPUs by treating the data as texture maps.

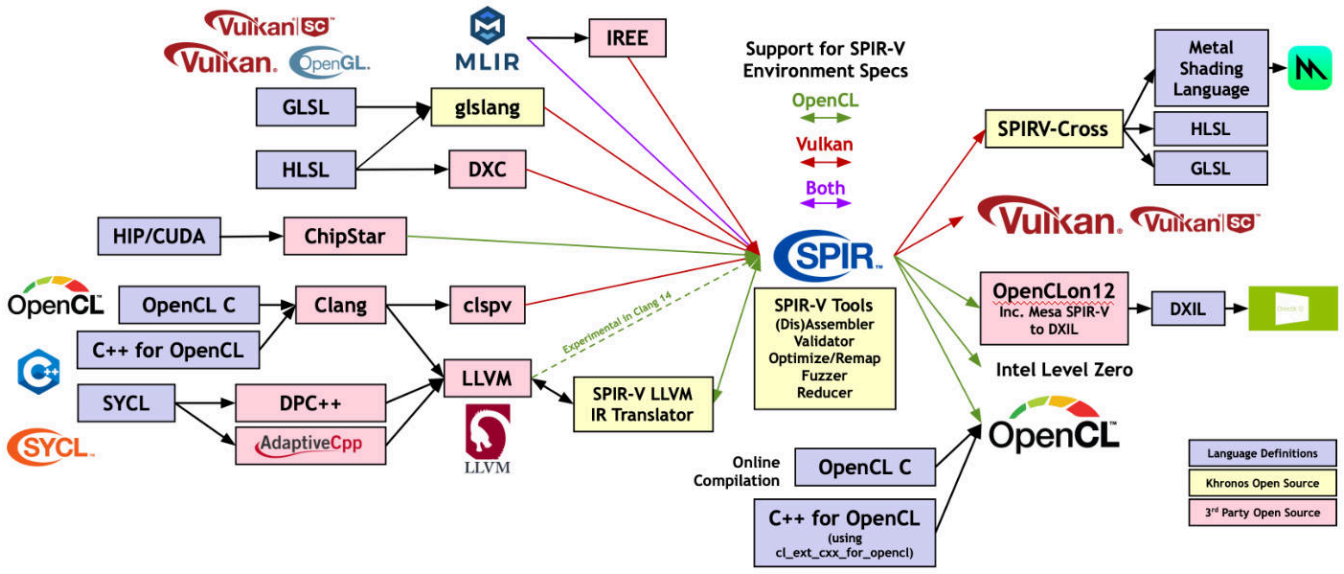


Hardware-specific

Common interface

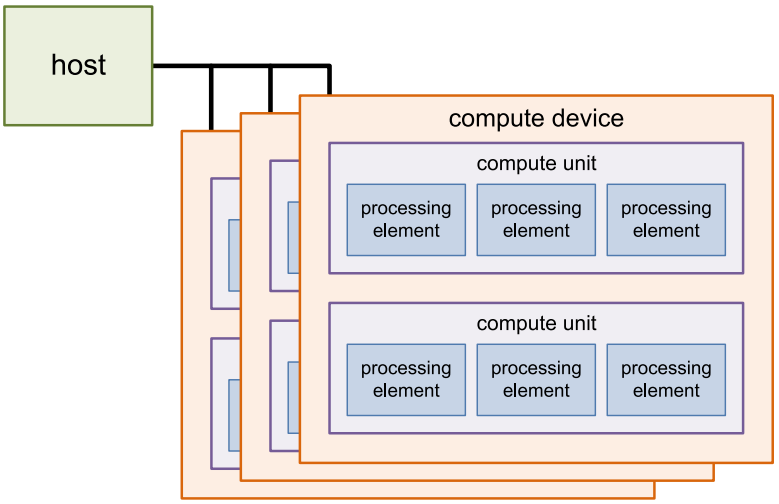
Standard Portable Intermediate Representation (SPIR)

Similar to LLVM IR : Intermediate representation for accelerated computation.



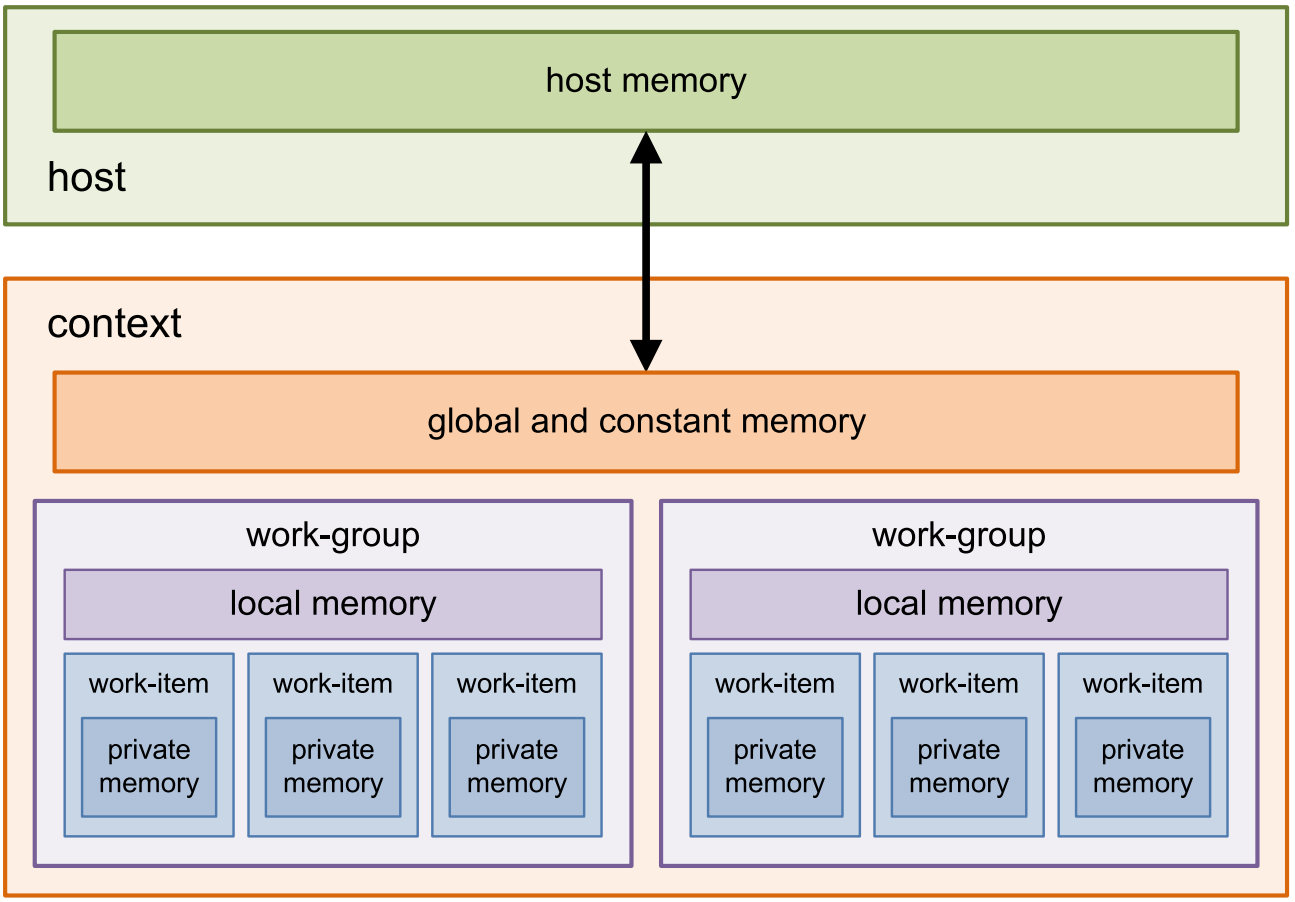
Hierarchy

- CPUs:
  - All CPUs part of same device
  - 1 Compute Unit per core
  - Number of processing elements equal to SIMD width
- GPUs:
  - One device per GPU



compute device	compute unit	processing element
`get_global_id`	`get_group_id`	`get_local_id`
`get_global_size`	`get_num_groups`	`get_local_size`

Memory



OpenCL Platforms and Devices

- Platforms are OpenGL implementations, listed in `/etc/OpenCL/vendors`
- Devices are actual CPUs/GPUs
- ICD allows to change platform at runtime

```
1 OpenCL.versioninfo()
```

```
No OpenCL drivers available, either system-wide or provided by a JLL.

Please install a system-wide OpenCL driver, or load one together with OpenCL.jl, e.g., by doing `using OpenCL, pocl_jll`.
```

```
OpenCL.jl version 0.10.2

Toolchain:
 - Julia v1.11.5
 - OpenCL_jll v2024.5.8+1

Available platforms: 0
```

See also `clinfo` command line tool and `examples/OpenCL/common/device_info.c`.

Tip

▶ **tl;dr To refresh the list of platforms, you need to quit Julia and open a new session**

Important stats

Error message

> *Another cell defining info_platform contains errors.*

```
1  md"""
2  * Platform
3    - name: $(info_platform.name)
4    - profile: $(info_platform.profile)
5    - vendor: $(info_platform.vendor)
6    - version: $(info_platform.version)
7  * Device
8    - name: $(info_device.name)
9    - type: $(info_device.device_type)
10
11
```

```
  | ['clGetDeviceInfo']
(https://registry.khronos.org/OpenCL/sdk/3.0/docs/man/html/clGetDeviceInfo.html) |
Value |
  | ---- | ---- |
  | `CL_DEVICE_GLOBAL_MEM_SIZE` |
$(BenchmarkTools.prettymemory(info_device.global_mem_size)) |
  | `CL_DEVICE_MAX_COMPUTE_UNITS`   | $(info_device.max_compute_units) |
  | `CL_DEVICE_LOCAL_MEM_SIZE` |
$(BenchmarkTools.prettymemory(info_device.local_mem_size)) |
  | `CL_DEVICE_MAX_WORK_GROUP_SIZE` | $(info_device.max_work_group_size) |
  | `CL_DEVICE_NATIVE_VECTOR_WIDTH_HALF` |
$(get_scalar(cl.CL_DEVICE_NATIVE_VECTOR_WIDTH_HALF, cl.cl_uint)) |
  | `CL_DEVICE_NATIVE_VECTOR_WIDTH_FLOAT` |
$(get_scalar(cl.CL_DEVICE_NATIVE_VECTOR_WIDTH_FLOAT, cl.cl_uint)) |
  | `CL_DEVICE_NATIVE_VECTOR_WIDTH_DOUBLE` |
$(get_scalar(cl.CL_DEVICE_NATIVE_VECTOR_WIDTH_DOUBLE, cl.cl_uint)) |
  | `CL_DEVICE_MAX_CLOCK_FREQUENCY` | $(info_device.max_clock_frequency) MHz |
  | `CL_DEVICE_PROFILING_TIMER_RESOLUTION` |
$(BenchmarkTools.prettytime(i
```

👀 *Reading hidden*

Error message from Main

```
BoundsError: attempt to access 0-element Vector{Pair} at index [1]
```

Stack trace

Here is what happened, the most recent locations are first:

1. **throw_boundserror**(A::Vector{…}, I::Tuple{…}) ...show types...
   from **julia** → *essentials.jl:14*
2. **getindex**
   from *essentials.jl:916*
3. **first**(a::Vector{Pair})
   from **julia** → *abstractarray.jl:452*
4. **get**(select::PlutoUI.BuiltinsNotebook.Select)
   from **PlutoUI** → *Builtins.jl:667*
5. **macro expansion**
   from *bonds.jl:127*
6. from This cell: *line 1*

```
aside((@bind info_platform Select([p => p.name for p in cl.platforms()])), v_offset = -300)
```


```
aside((@bind info_platform Select([p => p.name for p in cl.platforms()])), v_offset = -300)
```


👀 Reading hidden code

Error message

Another cell defining info_platform contains errors.

```
aside((@bind info_device Select([d => d.name for d in cl.devices(info_platform)])), v_offset = -300)
```

👀 Reading hidden code

Examples

Vectorized sum

```
__kernel void vadd(
    __global const float *a,
    __global const float *b,
    __global float *c,
    int verbose) {
  int i = get_global_id(0);
  c[i] = a[i] + b[i];
```

vadd_size = 512

vadd_verbose = 0

Error message

```
UndefVarError: `vadd_device` not defined in `Main.var"workspace#3"`
Suggestion: check for spelling errors or missing imports.
```

Stack trace

Here is what happened, the most recent locations are first:

1. **vadd**(len::Int64, verbose::Int64)
   from Other cell: *line 6*

```
4  c = similar(a)
5
6  cl.device!(vadd_device)
7  vadd_kernel = cl.Kernel(cl.Program(; source = vadd_source.code) |> cl.bui
ld!, "vadd")
8
```

cell preview

2. Show more...

```
1 evt = vadd(vadd_size, vadd_verbose);
```

Reading hidden

Error message from Main

```
BoundsError: attempt to access 0-element Vector{Pair} at index [1]
```

Stack trace

Here is what happened, the most recent locations are first:

1. throw_boundserror(A::Vector{…}, I::Tuple{…}) ...show types...
   from julia → essentials.jl:14
2. getindex
   from essentials.jl:916
3. first(a::Vector{Pair})
   from julia → abstractarray.jl:452
4. get(select::PlutoUI.BuiltinsNotebook.Select)
   from PlutoUI → Builtins.jl:667
5. macro expansion
   from bonds.jl:127
6. from This cell: line 1

```
1 aside((@bind vadd_platform Select([p => p.name for p in cl.platforms()])), v_offset = -250)
```

```
1 aside((@bind vadd_platform Select([p => p.name for p in cl.platforms()])), v_offset = -250)
```

👀 Reading hidden

Error message

Another cell defining vadd_platform contains errors.

```
1 aside((@bind vadd_device Select([d => d.name for d in cl.devices(vadd_platform)])), v_offset = -250)
```

👀 Reading hidden code

```
vadd (generic function with 1 method)
```

Mandelbrot

```
__kernel void mandelbrot(__global float2 *q,
    __global ushort *output, ushort const maxit) {

  int gid = get_global_id(0), it;
  if (gid == 0)
    printf("%d\n", get_num_groups(0));
  float tmp, real = 0, imag = 0;
  output[gid] = 0;
  for(it = 0; it < maxit; it++) {
    tmp = real * real - imag * imag + q[gid].x;
    imag = 2 * real * imag + q[gid].y;
    real = tmp;
    if (real * real + imag * imag > 4.0f)
        output[gid] = it;
```

mandel_size = 512

maxiter = 100

```
1 q = [ComplexF32(r,i) for i=1:-(2.0/mandel_size):-1, r=-1.5:(3.0/mandel_size):0.5];
```

Error message

> *Another cell defining mandel_platform contains errors.*

```
1 mandel_image = mandel(q, maxiter, mandel_device; global_size=length(q));
```

Error message from Main

> **BoundsError: attempt to access 0-element Vector{Pair} at index [1]**

Stack trace

Here is what happened, the most recent locations are first:

1. **throw_boundserror**(A::Vector{…}, I::Tuple{…}) ...show types...

from julia → essentials.jl:14

2. getindex
   from essentials.jl:916
3. first(a::Vector{Pair})
   from julia → abstractarray.jl:452
4. get(select::PlutoUI.BuiltinsNotebook.Select)
   from PlutoUI → Builtins.jl:667
5. macro expansion
   from bonds.jl:127
6. from This cell: line 1

```
1 aside((@bind mandel_platform Select([p => p.name for p in cl.platforms
()])), v_offset = -400)
```

```
1 aside((@bind mandel_platform Select([p => p.name for p in cl.platforms()])),
  v_offset = -400)
```

👀 Reading hidden code

Error message

Another cell defining mandel_platform contains errors.

```
1 aside((@bind mandel_device Select([d => d.name for d in
  cl.devices(mandel_platform)])), v_offset = -400)
```

👀 Reading hidden code

Error message

Another cell defining mandel_platform contains errors.

```
1 aside(CairoMakie.image(CairoMakie.rotr90(mandel_image)), v_offset = -400)
```

👀 Reading hidden code

```
mandel (generic function with 1 method)
```

```
function mandel(q::Array{ComplexF32}, maxiter::Int64, device; kws...)
    cl.device!(device)
    q = CLArray(q)
    o = CLArray{Cushort}(undef, size(q))

    prg = cl.Program(; source = mandel_source.code) |> cl.build!
    k = cl.Kernel(prg, "mandelbrot")

    timed_clcall(k, Tuple{Ptr{ComplexF32}, Ptr{Cushort}, Cushort},
           q, o, maxiter; kws...)

    return Array(o)
end
```

```
mandel_source = code(Example("OpenCL/mandelbrot/mandel.cl"));
```

Compute π

Error message

```
UndefVarError: `π_device` not defined in `Main.var"workspace#3"`
Suggestion: check for spelling errors or missing imports.
```

Stack trace

Here is what happened, the most recent locations are first:

1. **mypi(; niters::Int64, in_nsteps::Int64)**
   from Other cell: *line 2*

```
function mypi(; niters = 262144, in_nsteps = 512*512*512)
 cl.device!(π_device)

    prg = cl.Program(; source = π_code.code) |> cl.build!
```

cell preview

2. Show more...

```
mypi()
```

▶ **How to compute π with a kernel ?**

Error message from Main

```
BoundsError: attempt to access 0-element Vector{Pair} at index [1]
```

Stack trace

Here is what happened, the most recent locations are first:

1. throw_boundserror(A::Vector{…}, I::Tuple{…}) ...show types...
   from julia → essentials.jl:14
2. getindex
   from essentials.jl:916
3. first(a::Vector{Pair})
   from julia → abstractarray.jl:452
4. get(select::PlutoUI.BuiltinsNotebook.Select)
   from PlutoUI → Builtins.jl:667
5. macro expansion
   from bonds.jl:127
6. from This cell: line 1

```
1 aside((@bind π_platform Select([p => p.name for p in cl.platforms()])), v_offset = -200)
```

```
1 aside((@bind π_platform Select([p => p.name for p in cl.platforms()])), v_offset = -200)
```

👀 Reading hidden

Error message

Another cell defining π_platform contains errors.

```
aside((@bind π_device Select([d => d.name for d in cl.devices(π_platform)])),
v_offset = -200)
```

👀 *Reading hidden code*

```
mypi (generic function with 1 method)
```

First element

Let's write a simple kernel that returns the first element of a vector in global memory.

```
__kernel void first_el(__global float* glob, __global float* result) {
  int item = get_local_id(0);
  if (item == 0)
    *result = glob[item];
}
```

Error message

```
UndefVarError: `first_el_device` not defined in `Main.var"workspace#3"`
Suggestion: check for spelling errors or missing imports.
```

Stack trace

Here is what happened, the most recent locations are first:

1. **first_el**(x::Vector{Float32})
   from Other cell: *line 2*

```
function first_el(x::Vector{T}) where {T}
  cl.device!(first_el_device)
  result = CLArray(zeros(T, 1))
```

cell preview

2. Show more...

```
first_el(rand(Float32, first_el_len))
```

```
first_el (generic function with 1 method)
```

Error message from Main

> BoundsError: attempt to access 0-element Vector{Pair} at index [1]

Stack trace

Here is what happened, the most recent locations are first:

1. throw_boundserror(A::Vector{…}, I::Tuple{…}) ...show types...
   from julia → essentials.jl:14
2. getindex
   from essentials.jl:916
3. first(a::Vector{Pair})
   from julia → abstractarray.jl:452
4. get(select::PlutoUI.BuiltinsNotebook.Select)
   from PlutoUI → Builtins.jl:667
5. macro expansion
   from bonds.jl:127
6. from This cell: line 1

```
1 aside((@bind first_el_platform Select([p => p.name for p in cl.platforms
()])), v_offset = -400)
```

```
1 aside((@bind first_el_platform Select([p => p.name for p in cl.platforms()])),
  v_offset = -400)
```

👀 Reading hidden

Error message

> *Another cell defining first_el_platform contains errors.*

```
1 aside((@bind first_el_device Select([d => d.name for d in
  cl.devices(first_el_platform)])), v_offset = -400)
```

👀 Reading hidden

first_el_len = 16

Copy to local memory

```
__kernel void copy_to_local(__global float* glob, __local float* shared) {
  int global_size = get_global_size(0);
  int local_size = get_local_size(0);
  int item = get_local_id(0);
  shared[item] = 0;
  for (int i = 0; i < global_size; i += local_size) {
    shared[item] += glob[i + item];
  }
}
```

Error message

> **UndefVarError: `first_el_device` not defined in `Main.var"workspace#3"`**
> **Suggestion: check for spelling errors or missing imports.**

Stack trace

Here is what happened, the most recent locations are first:

1. **copy_to_local**(global_size::Int64, local_size::Int64)
   from Other cell: *line 2*

```
1 function copy_to_local(global_size, local_size)
2   cl.device!(first_el_device)
3   T = Float32
4   x = rand(T, global_size)
```

cell preview

2. Show more...

```
1 copy_to_local(copy_global_len, copy_local_len)
```

```
copy_to_local (generic function with 1 method)
```

Error message from Main

> BoundsError: attempt to access 0-element Vector{Pair} at index [1]

Stack trace

Here is what happened, the most recent locations are first:

1. throw_boundserror(A::Vector{…}, I::Tuple{…}) ...show types...
   from julia → essentials.jl:14
2. getindex
   from essentials.jl:916
3. first(a::Vector{Pair})
   from julia → abstractarray.jl:452
4. get(select::PlutoUI.BuiltinsNotebook.Select)
   from PlutoUI → Builtins.jl:667
5. macro expansion
   from bonds.jl:127
6. from This cell: line 1

```
1 aside((@bind copy_to_local_platform Select([p => p.name for p in cl.platfo
rms()])), v_offset = -300)
```

```
1 aside((@bind copy_to_local_platform Select([p => p.name for p in cl.platforms()])),
  v_offset = -300)
```

👀 Reading hidden

Error message

> *Another cell defining copy_to_local_platform contains errors.*

```
1 aside((@bind copy_to_local_device Select([d => d.name for d in
  cl.devices(copy_to_local_platform)])), v_offset = -300)
```

👀 Reading hidden

copy_global_len = 16

copy_local_len = 16

Reduction on GPU

Many operations can be framed in terms of a MapReduce operation.

- Given a vector of data
- It first map each elements through a given function
- It then reduces the results into a single element

The mapping part is easily embarassingly parallel but the reduction is harder to parallelize. Let's see how this reduction step can be achieved using arguably the simplest example of `mapreduce`, the sum (corresponding to an identity map and a reduction with `+`).

Sum

Error message

> *Another cell defining local_platform contains errors.*

```
1 local_sum(global_len, local_len, local_code, local_device)
```

▶ **How to compute the sum an array in local memory with a kernel ?**

```
local_sum (generic function with 1 method)
```

Error message from Main

```
BoundsError: attempt to access 0-element Vector{Pair} at index [1]
```

Stack trace

Here is what happened, the most recent locations are first:

1. throw_boundserror(A::Vector{…}, I::Tuple{…}) ...show types...
   from julia → essentials.jl:14
2. getindex
   from essentials.jl:916
3. first(a::Vector{Pair})
   from julia → abstractarray.jl:452
4. get(select::PlutoUI.BuiltinsNotebook.Select)
   from PlutoUI → Builtins.jl:667
5. macro expansion
   from bonds.jl:127
6. from This cell: line 1

```
1 aside((@bind local_platform Select([p => p.name for p in cl.platforms
()])), v_offset = -400)
```

```
1 aside((@bind local_platform Select([p => p.name for p in cl.platforms()])),
  v_offset = -400)
```

👀 Reading hidden

Error message

Another cell defining local_platform contains errors.

```
1 aside((@bind local_device Select([d => d.name for d in
  cl.devices(local_platform)])), v_offset = -400)
```

👀 Reading hidden

global_len = 16

local_len = 16

Blocked sum

Error message

```
UndefVarError: `block_local_device` not defined in `Main.var"workspace#3"`
Suggestion: check for spelling errors or missing imports.
```

Stack trace

Here is what happened, the most recent locations are first:

1. **block_local_sum**(global_size::Int64, local_size::Int64, factor::Int64)
   from Other cell: *line 2*

```
1 function block_local_sum(global_size, local_size, factor)
2   cl.device!(block_local_device)
3   T = Float32
4   Random.seed!(0)
```

cell preview

2. Show more...

```
1 block_local_sum(block_global_len, block_local_len, factor)
```

▶ **How to reduce the amount of `barrier` synchronizations ?**

▶ **Was it beneficial in terms of performance for GPUs like in the case of OpenMP ?**

```
block_local_sum (generic function with 1 method)
```

Error message from Main

```
BoundsError: attempt to access 0-element Vector{Pair} at index [1]
```

Stack trace

Here is what happened, the most recent locations are first:

1. **throw_boundserror**(A::Vector{…}, I::Tuple{…}) ...show types...
   from **julia** → *essentials.jl:14*
2. **getindex**
   from *essentials.jl:916*
3. **first**(a::Vector{Pair})
   from **julia** → *abstractarray.jl:452*
4. **get**(select::PlutoUI.BuiltinsNotebook.Select)
   from **PlutoUI** → *Builtins.jl:667*
5. **macro expansion**
   from *bonds.jl:127*
6. from This cell: *line 1*

```
1 aside((@bind block_local_platform Select([p => p.name for p in cl.platform
s()])), v_offset = -400)
```

```
1 aside((@bind block_local_platform Select([p => p.name for p in cl.platforms()])),
  v_offset = -400)
```

👀 Reading hidden

Error message

> *Another cell defining block_local_platform contains errors.*

```
1 aside((@bind block_local_device Select([d => d.name for d in
  cl.devices(block_local_platform)])), v_offset = -400)
```

👀 Reading hidden

block_global_len = 16

block_local_len = 16

factor = 16

Back to SIMD

- Also called Single Instruction Multiple Threads (SIMT)
- CUDA Warp : width of 32 threads
- AMD wavefront : width of 64 threads
- In general : `CL_KERNEL_PREFERRED_WORK_GROUP_SIZE_MULTIPLE`
- Consecutive `get_local_id()` starting from 0
    - So the thread of local id from 0 to 31 are in the same CUDA warp.
- Threads execute the **same instruction** at the same time so no need for `barrier` .

Warp divergence

Suppose a kernel is executed on a nvidia GPU with `global_size` threads. How much time will it take to execute it ?

```
__kernel void diverge(n)
{
  int item = get_local_id(0);
  if (item < n) {
    do_task_A(); // `a` ns
  } else {
    do_task_B(); // `b` ns
  }
}
```

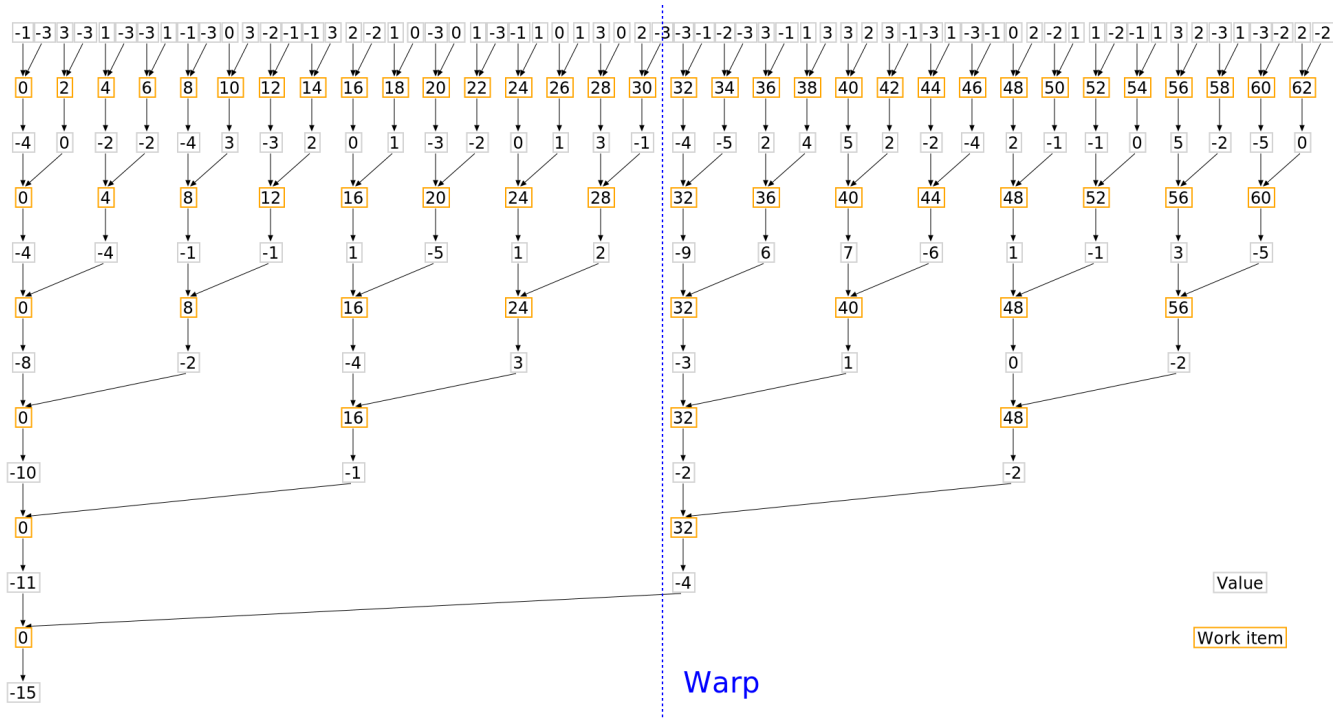
▶ **How much time will it take to execute it if `global_size` is 32 and `n` is 16 ?**

▶ **How much time will it take to execute it if `global_size` is 64 and `n` is 32 ?**

Are the threads that are still active in the same warp for you sum example ?

Warp diversion for our sum

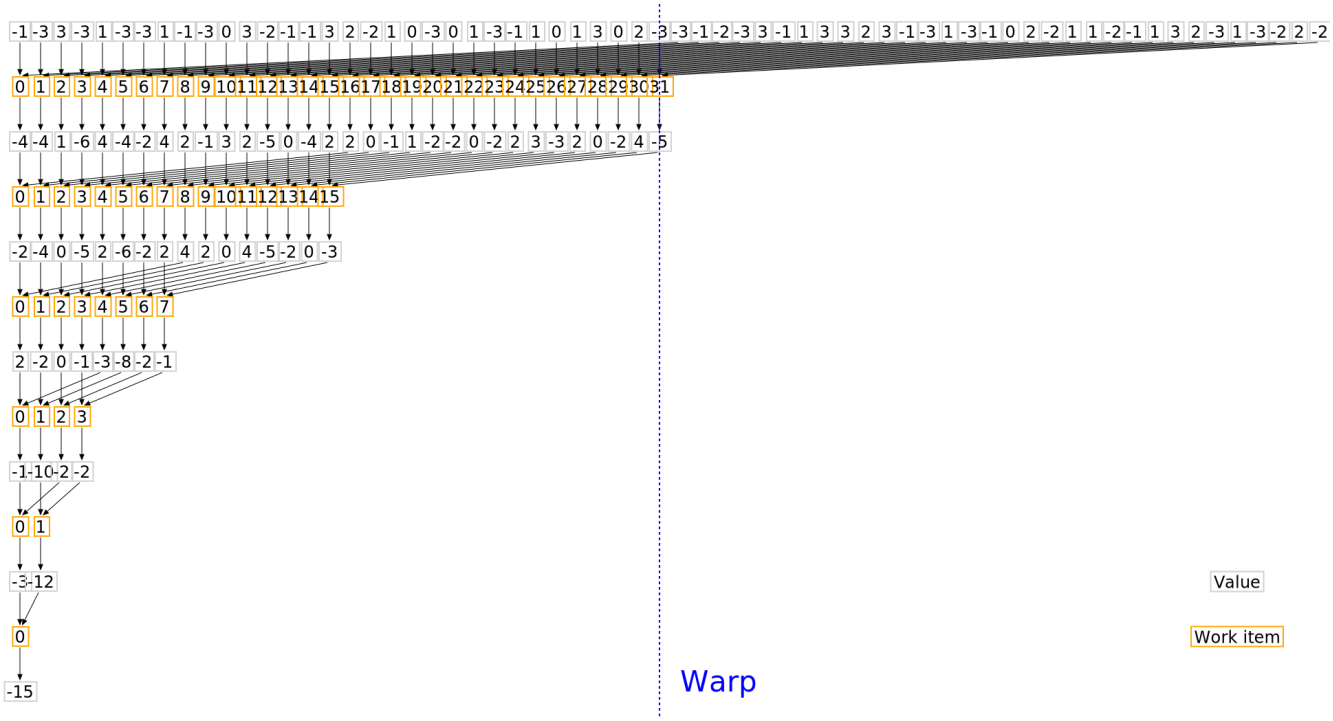
▶ **We are still using different warps until the end. Is that a good thing ?**



How should we change the sum to keep the working threads on the same warp ?

No warp divergence

Now the same warp is used for all threads so we don't need `barrier` and it frees other warps to stay idle (reducing power consumption) or do other tasks.



Reordered local sum

```
__kernel void local_sum(__local float* shared)
{
  int items = get_local_size(0);
  int item = get_local_id(0);
  int stride = items / 2;
  float other_val = 0;
  while (stride > 0) {
    barrier(CLK_LOCAL_MEM_FENCE);
    if (item < stride) {
      other_val = 0;
      if (item + stride < items)
        other_val = shared[item+stride];
      shared[item] += other_val;
    }
    stride /= 2;
  }
}
```

Error message

> *Another cell defining reordered_local_platform contains errors.*

```
local_sum(reordered_global_size, reordered_local_size, reordered_local_sum_code, reordered_local_device)
```

👀 Reading hidden

Error message from Main

> **BoundsError: attempt to access 0-element Vector{Pair} at index [1]**

Stack trace

Here is what happened, the most recent locations are first:

1. **throw_boundserror**(A::Vector{…}, I::Tuple{…}) ...show types...
   from **julia** → *essentials.jl:14*
2. **getindex**
   from *essentials.jl:916*
3. **first**(a::Vector{Pair})
   from **julia** → *abstractarray.jl:452*
4. **get**(select::PlutoUI.BuiltinsNotebook.Select)
   from **PlutoUI** → *Builtins.jl:667*
5. **macro expansion**
   from *bonds.jl:127*
6. from This cell: *line 1*

```
aside((@bind reordered_local_platform Select([p => p.name for p in cl.platforms()])), v_offset = -400)
```

```
aside((@bind reordered_local_platform Select([p => p.name for p in cl.platforms()])), v_offset = -400)
```

👀 Reading hidden

Error message

> *Another cell defining reordered_local_platform contains errors.*

```
aside((@bind reordered_local_device Select([d => d.name for d in
cl.devices(reordered_local_platform)])), v_offset = -400)
```

👀 Reading hidden

reordered_global_size = 16

reordered_local_size = 16

SIMT sum

```
__kernel void simt_sum(volatile __local float* shared)
{
  int items = get_local_size(0);
  int item = get_local_id(0);
  barrier(CLK_LOCAL_MEM_FENCE);
  while (items > 1) {
    items /= 2;
    shared[item] += shared[item + items];
  }
}
```

Error message

> *Another cell defining simt_platform contains errors.*

```
local_sum(simt_global_size, simt_local_size, simt_code, simt_device)
```

▶ **Why don't we check any condition on `item`, aren't some thread computing data that won't be used ?**

Error message from Main

> BoundsError: attempt to access 0-element Vector{Pair} at index [1]

Stack trace

Here is what happened, the most recent locations are first:

1. throw_boundserror(A::Vector{…}, I::Tuple{…}) ...show types...
   from julia → essentials.jl:14
2. getindex
   from essentials.jl:916
3. first(a::Vector{Pair})
   from julia → abstractarray.jl:452
4. get(select::PlutoUI.BuiltinsNotebook.Select)
   from PlutoUI → Builtins.jl:667
5. macro expansion
   from bonds.jl:127
6. from This cell: line 1

```
1 aside((@bind simt_platform Select([p => p.name for p in cl.platforms()])), v_offset = -400)
```

```
1 aside((@bind simt_platform Select([p => p.name for p in cl.platforms()])), v_offset = -400)
```

👀 Reading hidden

Error message

> *Another cell defining simt_platform contains errors.*

```
1 aside((@bind simt_device Select([d => d.name for d in cl.devices(simt_platform)])), v_offset = -400)
```

👀 Reading hidden

simt_global_size = 16

simt_local_size = 16

Beware!

POCL does not synchronize, even for `simt_len <= 8`

▶ **Why do we need `volatile` ?**

Unrolled sum

▶ **How to get even faster performance by assuming that `items` is a power of 2 smaller than 512 and that the SIMT width is 32 ?**

Error message

> *Another cell defining unrolled_platform contains errors.*

```
1 local_sum(unrolled_global_size, unrolled_local_size, unrolled_code, unrolled_device)
```

▶ **How to have portable code using unrolling ?**

Error message from Main

```
BoundsError: attempt to access 0-element Vector{Pair} at index [1]
```

Stack trace

Here is what happened, the most recent locations are first:

1. throw_boundserror(A::Vector{…}, I::Tuple{…}) ...show types...
   from julia → essentials.jl:14
2. getindex
   from essentials.jl:916
3. first(a::Vector{Pair})
   from julia → abstractarray.jl:452
4. get(select::PlutoUI.BuiltinsNotebook.Select)
   from PlutoUI → Builtins.jl:667
5. macro expansion
   from bonds.jl:127
6. from This cell: line 1

```
1 aside((@bind unrolled_platform Select([p => p.name for p in cl.platforms
()])), v_offset = -400)
```

```
1 aside((@bind unrolled_platform Select([p => p.name for p in cl.platforms()])),
  v_offset = -400)
```

👀 Reading hidden

Error message

> *Another cell defining unrolled_platform contains errors.*

```
1 aside((@bind unrolled_device Select([d => d.name for d in
  cl.devices(unrolled_platform)])), v_offset = -400)
```

👀 Reading hidden

unrolled_global_size = 16

unrolled_local_size = 16

Utils

```
_pretty_time (generic function with 1 method)
```

```
1 _pretty_time(x) = BenchmarkTools.prettytime(minimum(x))
```

```
timed_clcall (generic function with 1 method)
```

```
function timed_clcall(kernel, args...; kws...)
    info = cl.work_group_info(kernel, cl.device())
    # See https://registry.khronos.org/OpenCL/sdk/3.0/docs/man/html/clGetKernelWorkGroupInfo.html
    println("CL_KERNEL_WORK_GROUP_SIZE                    | ", info.size)
    println("CL_KERNEL_COMPILE_WORK_GROUP_SIZE            | ", info.compile_size)
    println("CL_KERNEL_LOCAL_MEM_SIZE                     | ", BenchmarkTools.prettymemory(info.local_mem_size))
    println("CL_KERNEL_PRIVATE_MEM_SIZE                   | ", BenchmarkTools.prettymemory(info.private_mem_size))
    println("CL_KERNEL_PREFERRED_WORK_GROUP_SIZE_MULTIPLE | ", info.prefered_size_multiple)

    # `:profile` sets `CL_QUEUE_PROFILING_ENABLE` to the command queue
    queued_submit = Float64[]
    submit_start = Float64[]
    start_end = Float64[]
    cl.queue!(:profile) do
        for _ in 1:num_runs
            evt = clcall(kernel, args...; kws...)
            wait(evt)

            # See https://registry.khronos.org/OpenCL/sdk/3.0/docs/man/html/clGetEventProfilingInfo.html
            push!(queued_submit, evt.profile_submit - evt.profile_queued)
            push!(submit_start, evt.profile_start - evt.profile_submit)
            push!(start_end, evt.profile_end - evt.profile_start)
        end
    end
    println("Send command from host to device  | $(_pretty_time(queued_submit))")
    println("Including data transfer           | $(_pretty_time(submit_start))")
    println("Execution of kernel               | $(_pretty_time(start_end))")
end
```

num_runs = 1

```
  Activating project at `~/work/LINMA2710/LINMA2710/Lectures`
```

Error message from Main

```
InitError: could not load library
"/home/runner/.julia/artifacts/ebb51f9798bd29ec4f6f64a8b26ce9ca5c84a72f/lib/libpocl.so"
/home/runner/.julia/artifacts/ebb51f9798bd29ec4f6f64a8b26ce9ca5c84a72f/lib/libpocl.so: undefined symbol: _ZN4llvm10CallbackVH6anchorEv
during initialization of module pocl_jll
```

Stack trace

Here is what happened, the most recent locations are first:

1. **dlopen**(s::String, flags::UInt32; throw_error::Bool)
   from **julia** → *libdl.jl:120*
2. **dlopen**(s::String, flags::UInt32)
   from **julia** → *libdl.jl:119*
3. macro expansion
   from *library_generators.jl:63*
4. **__init__**()
   from **pocl_jll** → *x86_64-linux-gnu-cxx11.jl:17*
5. **run_module_init**(mod::Module, i::Int64)
   from **julia** → *loading.jl:1378*
6. **register_restored_modules**(sv::Core.SimpleVector, pkg::Base.PkgId, path::String)
   from **julia** → *loading.jl:1366*
7. **_include_from_serialized**(pkg::Base.PkgId, path::String, ocachepath::String, depmods::Vector{Any}, ignore_native::Nothing; register::Bool)
   from **julia** → *loading.jl:1254*
8. _include_from_serialized
   from *loading.jl:1210*
9. **#_require_search_from_serialized#1105**(pkg::Base.PkgId, sourcepath::String, build_id::UInt128, stalecheck::Bool; reasons::Dict{…}, DEPOT_PATH::Vector{…})
   ...show types...
   from **julia** → *loading.jl:2057*
10. **_require**(pkg::Base.PkgId, env::String)
    from **julia** → *loading.jl:2527*

11. __require_prelocked(uuidkey::Base.PkgId, env::String)
    from julia → loading.jl:2388
12. #invoke_in_world#3
    from essentials.jl:1089
13. invoke_in_world
    from essentials.jl:1086
14. _require_prelocked(uuidkey::Base.PkgId, env::String)
    from julia → loading.jl:2375
15. macro expansion
    from loading.jl:2314
16. macro expansion
    from lock.jl:273
17. __require(into::Module, mod::Symbol)
    from julia → loading.jl:2271
18. #invoke_in_world#3
    from essentials.jl:1089
19. invoke_in_world
    from essentials.jl:1086
20. require(into::Module, mod::Symbol)
    from julia → loading.jl:2260
21. from This cell: line 1

```
1 using OpenCL, pocl_jll # `pocl_jll` provides the POCL OpenCL platform for CPU devices
```

```
1 using OpenCL, pocl_jll # `pocl_jll` provides the POCL OpenCL platform for CPU devices
```