MULTIPLE DEVICES

SYCL and the SYCL logo are trademarks of
the Khronos Group Inc.

LEARNING OBJECTIVES

- Learn about contexts and what they are used for
- Learn about how create dependencies across devices
- Learn about moving data between devices

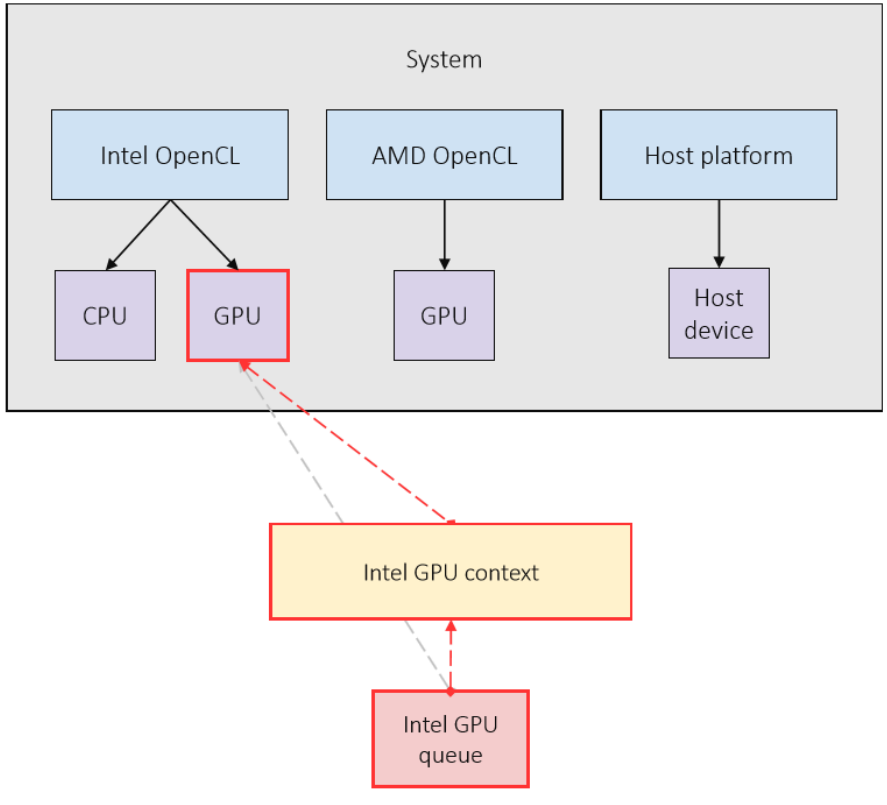
SYCL and the SYCL logo are trademarks of
the Khronos Group Inc.

WHAT IS A CONTEXT

- In SYCL the underlying execution and memory resources of a platform and its devices is managed by creating a context
- A context represents one or more devices, but all devices must be associated with the same platform

SYCL and the SYCL logo are trademarks of the Khronos Group Inc.

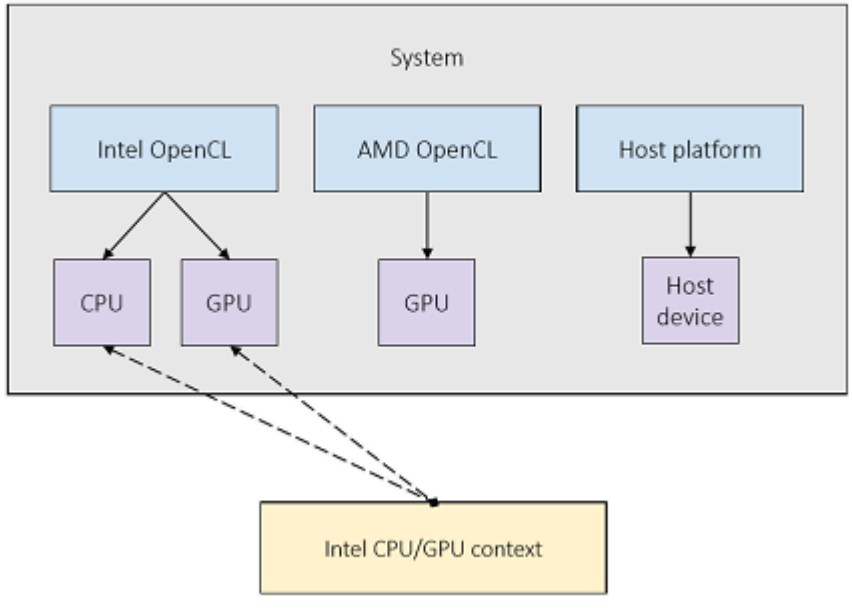
IMPLICIT CONTEXT



- Every `queue` requires a `context` to manage memory allocation and data movement.
- If one is not specified explicitly a `queue` will create a `context` implicitly.

SYCL and the SYCL logo are trademarks of the Khronos Group Inc.

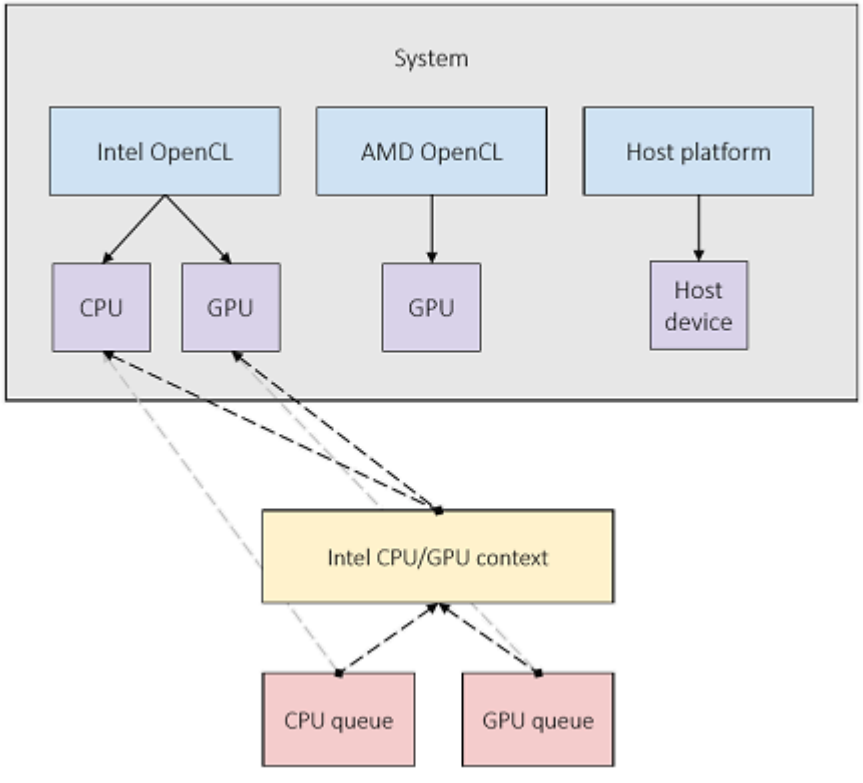
SHARED CONTEXT



- In order to ensure data is efficiently moved between devices in the same platform you can create a common `context`.

SYCL and the SYCL logo are trademarks of the Khronos Group Inc.

CREATING QUEUES



- You can then create a `queue` for each of the devices from the common `context`.

SYCL and the SYCL logo are trademarks of the Khronos Group Inc.

CREATING AN IMPLICIT CONTEXT

```
auto defaultQueue = queue{};
```

- A default constructed queue object will use the `default_selector` to choose a device and create an implicit `context`.

SYCL and the SYCL logo are trademarks of the Khronos Group Inc.

CREATING A CONTEXT FROM DEVICES

```
auto sharedContext = context{{cpuDevice, gpuDevice}};
```

- You can construct a `context` from a `std::vector` of `devices`.

SYCL and the SYCL logo are trademarks of the Khronos Group Inc.

CREATING A CONTEXT FROM A PLATFORM

```
auto sharedContext = context{intelPlatform};
```

- You can construct a `context` from a `platform` in which case it will be associated with all of the devices of that `platform`.

SYCL and the SYCL logo are trademarks of the Khronos Group Inc.

TARGETING MULTIPLE DEVICES

- A single SYCL application will often want to target multiple different devices.
- This can be useful for task level parallelism and load balancing.
- When doing so you want to ensure that data is moved between devices in a `context` efficiently.

SYCL and the SYCL logo are trademarks of the Khronos Group Inc.

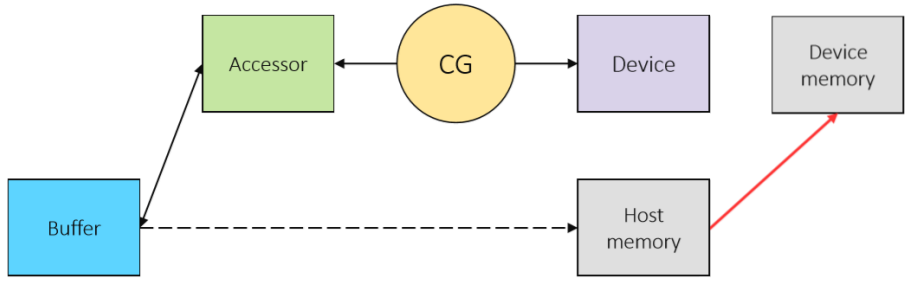
MOVING BETWEEN DEVICES

- Often in heterogeneous applications it's necessary to move data from one device to another.
- In the USM model this is done explicit via `memcpy` as we've seen before.
- In the buffer/accessor model this is done automatically based on dependency analysis.

SYCL and the SYCL logo are trademarks of the Khronos Group Inc.

ACCESSING DATA ON A DEVICE

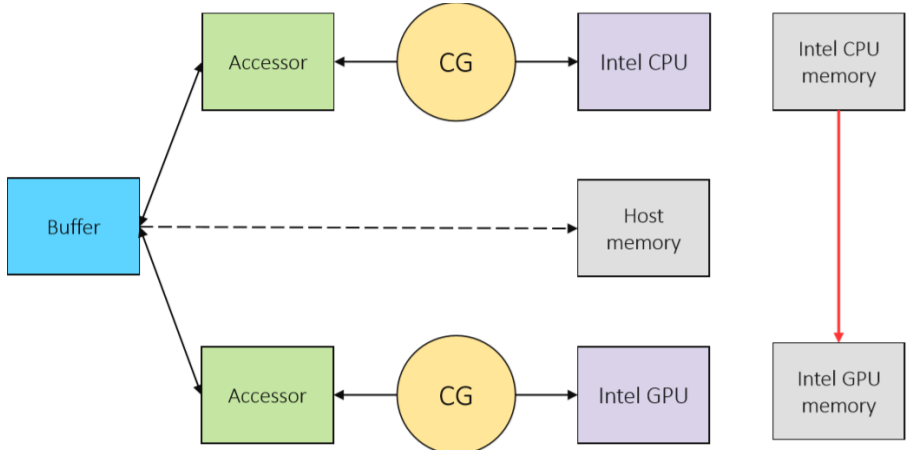
- Remember that a `buffer` will move data to a device when required by an `accessor`.



SYCL and the SYCL logo are trademarks of the Khronos Group Inc.

ACCESSING DATA ON ANOTHER DEVICE (SAME CONTEXT)

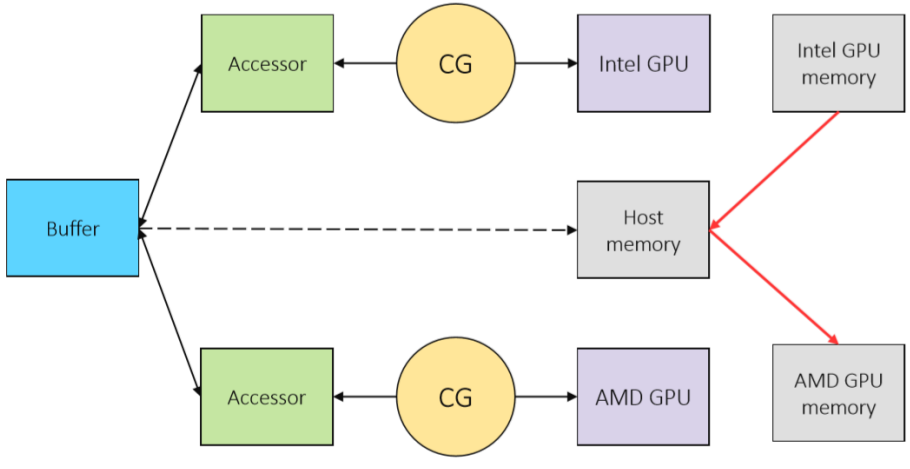
- Now if a `buffer` is accessed on a device when the latest copy of the data is on another device, the data will be moved between the devices.
- If the two devices are of the same context the data can be copied directly.



SYCL and the SYCL logo are trademarks of the Khronos Group Inc.

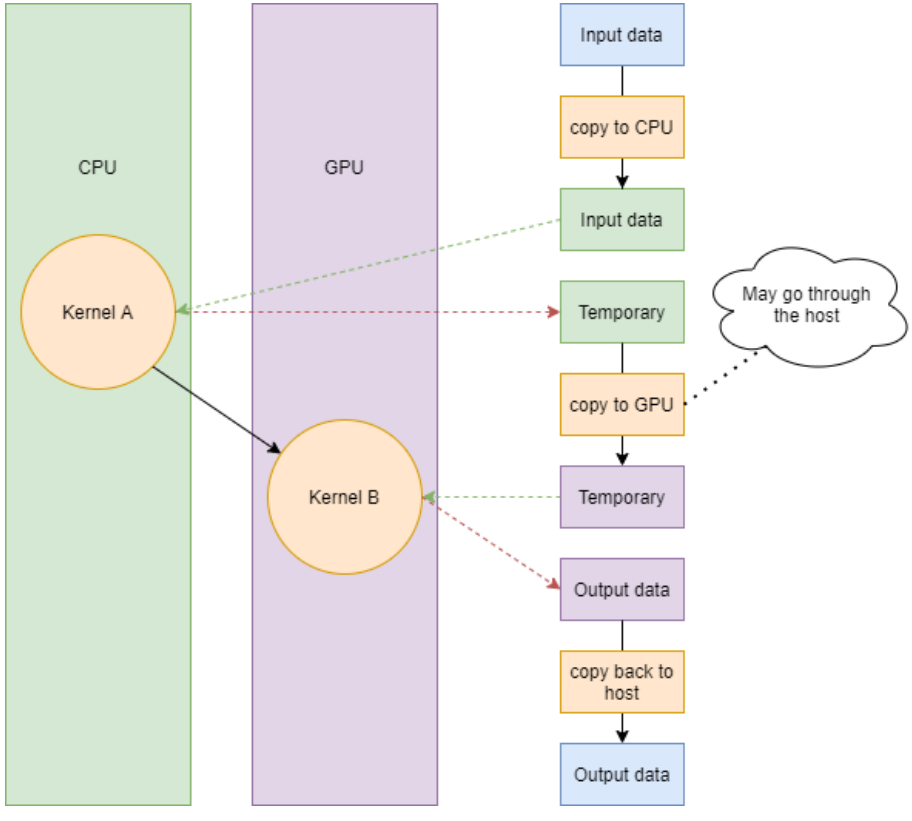
ACCESSING DATA ON ANOTHER DEVICE (DIFFERENT CONTEXT)

- If the devices are of different `contexts` the data must be copied via host memory.
- It's important to consider this as it could incur further overhead when moving data between devices.



SYCL and the SYCL logo are trademarks of the Khronos Group Inc.

MOVING BETWEEN DEVICES (BUFFER/ACCESSOR)



- If a `buffer` is accessed by kernel functions in two different devices commands are enqueued to automatically move the data to the devices it is being accessed on.
- If both of those devices are associated with the same context (i.e. same vendor) then the copy is direct.
- Otherwise the copy will generally go via the host and has additional overhead.

SYCL and the SYCL logo are trademarks of the Khronos Group Inc.

QUESTIONS

SYCL and the SYCL logo are trademarks of
the Khronos Group Inc.

EXERCISE

Code_Exercises/Exercise_13_Load_Balancing/source

Write a SYCL application that splits up a task between two devices.

SYCL and the SYCL logo are trademarks of the Khronos Group Inc.