Streams and concurrencies

- Goals :
  - Execute kernels simultaneously
  - Execute kernels and copies DeviceToHost or HostToDevice simulatneously to keep the bus busy

Stream creation

- Declare : cudaStream_t stream;
- Create : cudaStreamCreate(&stream)
- Release : cudaStreamDestroy(stream)

Concurrent copies

- MUST use pinned memory (with cudaHostAlloc)!
  - Non pageable, constant physical location
- No two simultaneous copies (unless duplex PCIe bus)

The Main Stream

- If you don’t specify anything : main stream is used (stream 0 or NULL stream, default).

- All operations issued in a stream are serialized in this stream

Dependencies

- Call issued to GPU are generally non blocking for host
- Exception : implicit synchronization for memory functions : allocation, memset,
- Host - Stream synchronization :
    - cudaError_t cudaStreamSynchronize(cudaStream_t stream); (note the difference with cudaDeviceSynchronize())
    - cudaError_t cudaStreamQuery(cudaStream_t stream);
        - return cudaSuccess if finished or cudaErrorNotReady
- You can also pilot streams via cudaEvents with cudaError_t cudaStreamWaitEvent(cudaStream_t stream, cudaEvent_t event);
- Possibility to add callbacks

False depencies

- HyperQ

- Breadth vs Depth kernel launches