Introduction to OpenCL

John Poole, Primate Labs

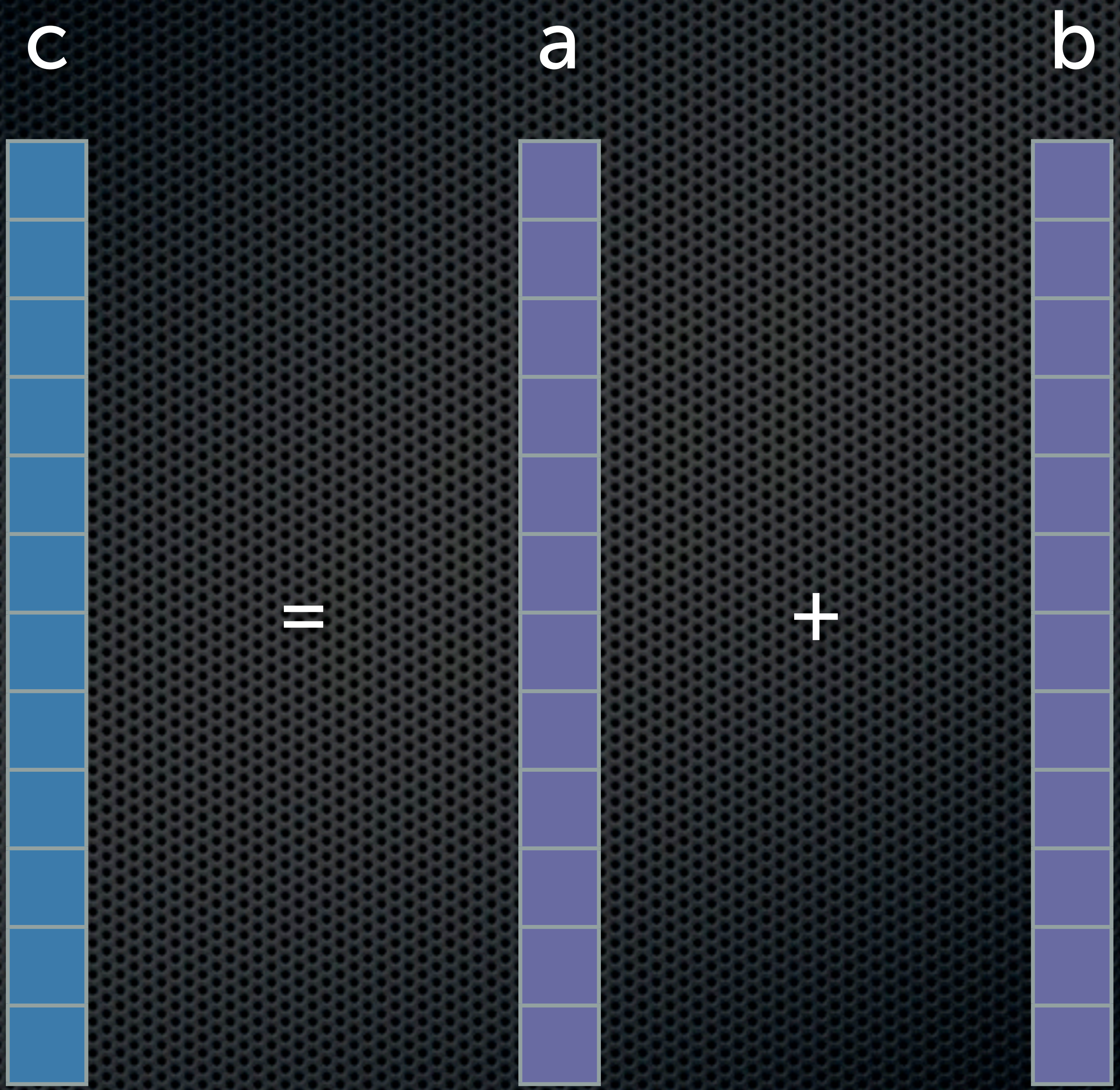
john@primatelabs.ca

@jfpoole

What is OpenCL?

- Open Computing Language
- Parallel programming API and language
  - API based on OpenGL and CUDA
  - Language based on C
- Designed for heterogeneous systems
  - Write parallel code for CPUs and GPUs

$c = a + b$



c a b

= +

C Function

```
void add(int* c, int* a, int* b, int N)
{
  int i;
  for (i = 0; i < N; i++) {
    c[i] = a[i] + b[i];
  }
}
```

OpenCL Kernel

```
__kernel void add(
  __global int* c,
  __global int* a,
  __global int* b)
{
  int i = get_global_id(0);
  c[i] = a[i] + b[i];
}
```

OpenCL Execution Model

- Hosts and devices
- Contexts and command queues
- Programs and kernels
- Work items and work units

OpenCL Memory Model

- Host and devices have separate address spaces
- Memory management is explicit

Demo

https://github.com/jfpoole/opencl_example

Performance Tips

- Compilation can be expensive
- Don’t block unless necessary
  - Keep both host and devices busy
- Don’t transfer unless necessary
  - Use OpenGL integration

Other Languages

- C++ API from Khronos
  - http://www.khronos.org/registry/cl/
- Objective-C API from SuperMegaUltraGroovy
  - https://bitbucket.org/liscio/smugopencl/

Questions?

john@primatelabs.ca

@jfpoole