WASI CG

wasi-parallel

Andrew Brown, Johnnie Birch, Enrico Galli, Petr Penzin, Mingqiu Sun

Motivation

- WebAssembly lacks support for parallel execution in general, with performance lagging far behind native for many workloads (ML & HPC)
  - 128-bit SIMD provides limited parallelism in comparison to native
  - Domain-specific solutions: wasi-nn solves the ML problem for major frameworks; community asks for more general acceleration support
  - Many programs benefit from parallel execution—no standard way to do so in standalone WebAssembly engines (unlike browser Web Workers)
- wasi-parallel intends to provide low-level parallel execution support
  - Initial scope: “parallel for,” which is more amenable to heterogeneous execution
- Looking for partners to work with: the right API benefits many parties

Direction

Goals:

- Access system parallelism capabilities using WASI
- Execute on heterogeneous devices
- Match abstraction of many parallel programming frameworks

Non-goals:

- Modify the Wasm specification, if possible
- Limit execution to a specific class of device (e.g. GPU-only APIs)
- Support all programming models (e.g. pthreads)

API (Draft)

```
get_device: function(hint: device_kind) -> expected<device,
   err>
create_buffer: function(device: device, size: u32, access:
   access) -> expected<buffer, err>
write_buffer: function(data: push_buffer<u8>, buffer:
   buffer) -> err
read_buffer: function(buffer: buffer, data: pull_buffer<u8>)
   -> err
for: function(kernel: funcref, num_threads: u32, block_size:
   u32, in_buffers: list<buffer>, out_buffers: list<buffer>)
   -> err
```

Current State

- Built a prototype using Wasmtime that executes on the CPU
  - Can execute certain PRK kernels in the prototype
- GPU execution using OpenCL is in-progress
  - Can execute simple kernels (blocking on all Wasm calls), need to benchmark
- Working on toolchain support to compile existing OpenMP programs to wasi-parallel
  - Can compile OpenMP to Wasm, working on a shim for OpenMP runtime calls

Future Directions

- Check direction with WASI community
- Refine API based on community feedback
- More benchmarking; interested in partner use cases
- Investigate compiling from other frameworks: SYCL, OpenCL
- Investigate compiling from a high-level library: MediaPipe, RAJA