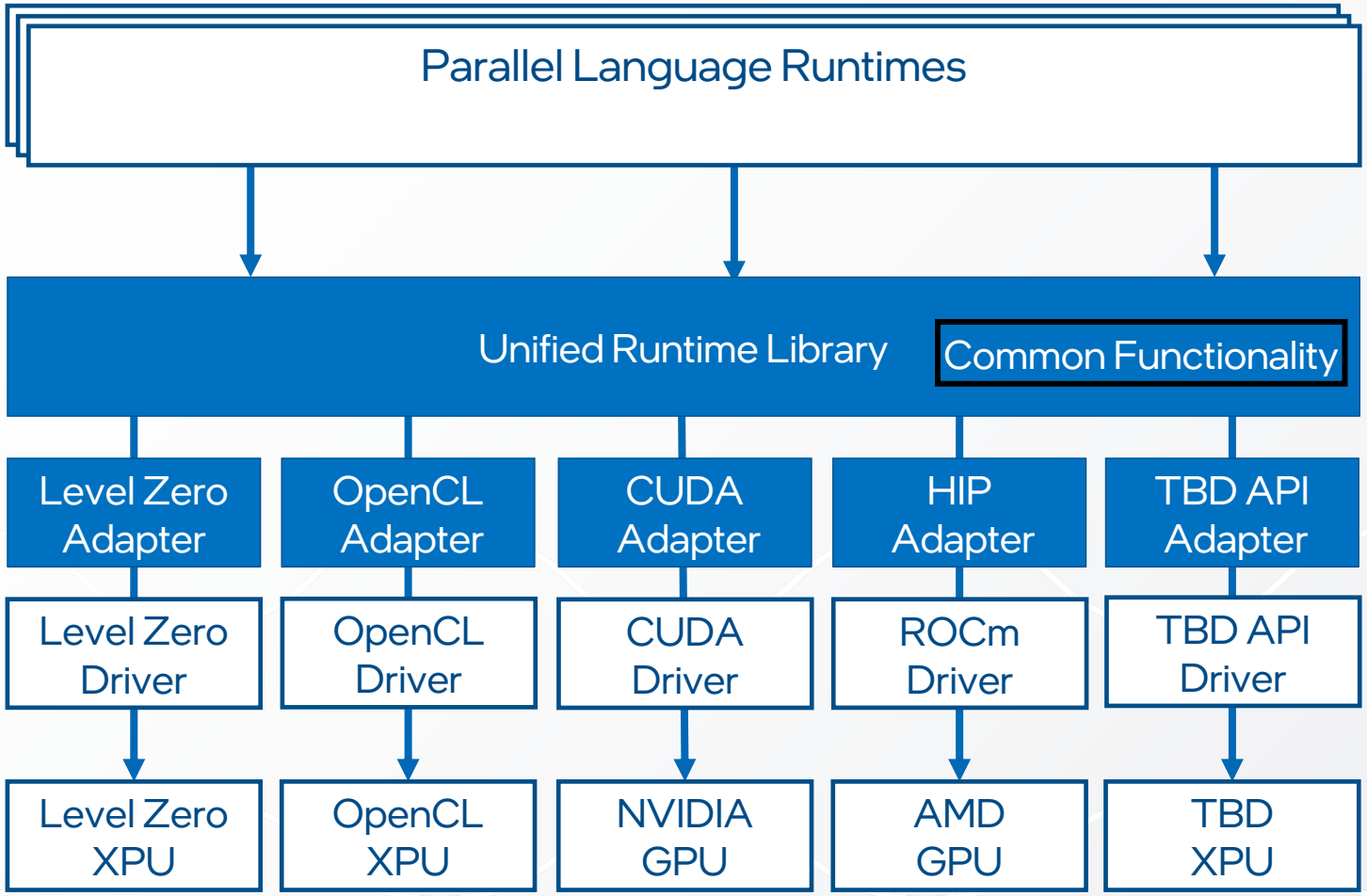
Unified Runtime

Topics

- Recap of Unified Runtime
- License update
- One slide specification and implementation update since previous SIG

30,000 Foot View of Unified Runtime

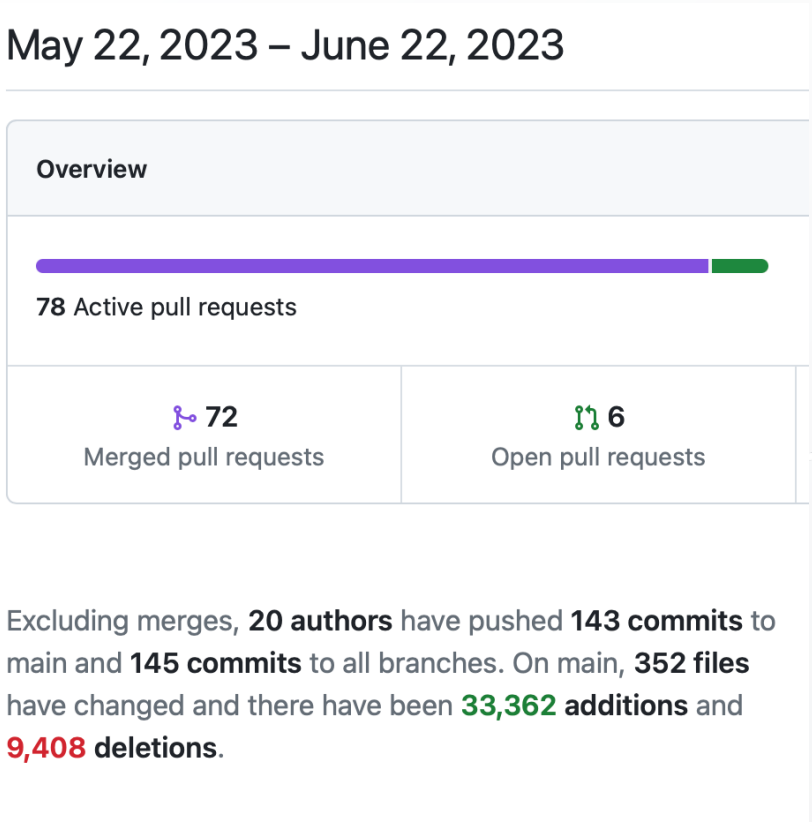


- Define a “Unified Runtime”, usable by any Parallel Language Runtime, with a well-defined interface
- Support a range of low-level or implementation specific adapters
- Specification:
  - https://spec.oneapi.io/unified-runtime/latest/

License update

- Requested feedback at previous SIG
- License for unified-runtime repo is now "Apache 2.0 with LLVM Exceptions"
  - I.e. same as upstream LLVM
  - Easier to move code to/from both Intel LLVM and upstream LLVM
- https://github.com/oneapi-src/unified-runtime/blob/main/LICENSE.TXT

Specification v0.7 (will be tagged soon)



- Specification work happens on GitHub directly
  - https://github.com/oneapi-src/unified-runtime
- Adapter and runtime porting in DPC++ codebase
  - https://github.com/intel/llvm
  - CUDA and L0 adapters are now unified-runtime under the hood, will expose fully later
- Many many additions since v0.6, headlines:
  - Reserve virtual memory regions
  - Introduced consistent extensibility mechanisms
  - Exposed control of USM pooling
  - Experimental command buffers (for graphs)
  - Experimental USM import/release
  - Experimental bindless image support

Summary

- Unified runtime specification and implementation still under current, active development

- Great time to give us any feedback!
  - Comments on recent specification changes:
    - https://spec.oneapi.io/unified-runtime/latest/
  - Thoughts on building language runtimes on top of unified runtime
  - Can file discussion points at:
    - https://github.com/oneapi-src/unified-runtime/issues

- We think that Unified Runtime is now ready to build high-level languages on top
  - We'll be looking at SYCL and OpenMP
  - But goal is to support a much broader set
  - Anyone can start to investigate this

Questions