

oneMKL Technical Advisory Board

Session 4

July 1, 2020

Agenda

- Welcoming remarks – 5 minutes
- Exceptions/Error handling – Maria Kraynyuk (30 minutes)
- Open discussion on oneMKL Specification and open source oneMKL interfaces GitHub project – all (15 minutes)
- Wrap-up and next steps – 5 minutes

Exceptions/Error handling

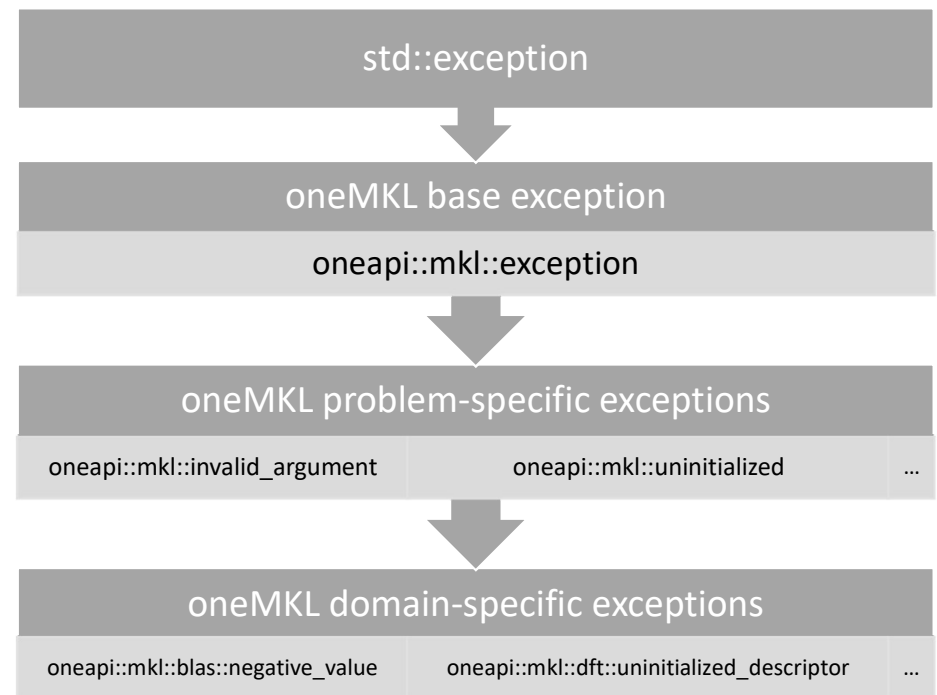
oneMKL exceptions

oneMKL can be used in wide range of applications and plugins, so it's important to provide exceptions with several levels of details.

- Standard exceptions: no need to handle oneMKL-specific problems.
- oneMKL specific exceptions: identify oneMKL related exceptions.
- oneMKL problems-specific exceptions: identify oneMKL problems to handle them at runtime.
- oneMKL domain-specific exceptions: identify problems in particular oneMKL domain to handle them at runtime.

oneMKL exception classes hierarchy

1. oneMKL base exception class can be inherited from STD exception, it's up to implementation, based on the product requirements.
2. All problem-specific oneMKL exception classes should be inherited from oneMKL base exception class.
3. All oneMKL domain-specific exception classes should be inherited from oneMKL base class or problem-specific exception classes.



oneMKL base class and list of problem-specific exceptions

Exception	Exception class name	Example
Base class for all problem-specific exceptions below	oneapi::mkl::exception	General unspecific problem, not applying to specific exceptions below
HOST allocation problem	oneapi::mkl::host_allocation_failed	malloc() returns NULL
Device allocation problem	oneapi::mkl::device_allocation_failed	malloc_shared() returns NULL
Targeted device is not supported	oneapi::mkl::unsupported_device	AMD GPU is not supported
Not implemented yet	oneapi::mkl::unimplemented	foo() is not implemented for complex type
Incorrect input	oneapi::mkl::invalid_argument	Number of columns is negative
Descriptor/handle is not initialized	oneapi::mkl::uninitialized	DFT descriptor wasn't initialized before commit step
Computation problem	oneapi::mkl::computation_error	FFT invalid descriptor at commit step LAPACK calculation failure Sparse BLAS conditions are not satisfied

problem-specific exceptions

Open discussion on oneMKL
specification and open source
oneMKL interfaces GitHub project

Open discussion

- Anything missing from the spec you'd expect to see?
- Anything surprising in the spec?
- What domains are you most eager to see finalized in the spec?
- Any issues trying out the open source interfaces?
- Are you interested in contributing/encouraging contributions to the open source interfaces?

Next Steps

- Look over current oneMKL Spec v. 0.8.5
 - Focus most on oneMKL Architecture (section 1) and BLAS/LAPACK APIs
- Focuses for next meeting(s):
 - Asynchronous execution
 - Member function APIs

Resources

- oneAPI Main Page: <https://www.oneapi.com/>
- Latest release of oneMKL Spec (currently v. 0.8.5):
<https://spec.oneapi.com/versions/latest/elements/oneMKL/source/index.html>
- GitHub for oneAPI Spec: <https://github.com/oneapi-src/oneAPI-spec>
- GitHub for oneAPI TAB: <https://github.com/oneapi-src/oneAPI-tab>
- Latest build of oneAPI Spec: <http://staging.spec.oneapi.com.s3-website-us-west-2.amazonaws.com/exclude/ci/branches/refs/heads/master/versions/latest/index.html>
- GitHub for open source oneMKL interfaces (currently BLAS domain):
<https://github.com/oneapi-src/oneMKL>