



# **MPI-4: Standardization VS What Users Need for Exascale**

**Tony Skjellum**

**[tony-skjellum@utc.edu](mailto:tony-skjellum@utc.edu)**

**April 24, 2019**



# What MPI-4.0 will deliver

- Persistent Collective Communication APIs
  - New tools interfaces (QMPI, events)
  - Enhanced documentation and rules for behavior after an error occurs (pre FT)
  - Semantics/naming conventions clarity [non-functional]
  - Miscellaneous improvements
-

# What MPI-4.0 may deliver

- 64-bit clean interfaces (“Big MPI”)
    - C / FORTRAN
  - Sessions-based dynamic process management
    - Build-up scalable enclaves
    - Sparse connectivity of communication
-

# What MPI-4.0 might deliver

- Topology-awareness at runtime
  - Thread-interface/partitioned communication (Finepoints) – better than `MPI_THREAD_MULTIPLE`
  - New, native C++ and Python Interfaces
-

# What might users/programmers want?

- Better thread interoperability
  - Heterogeneous Device Support
  - Mechanisms for extreme scale
  - Resource-consciousness
  - Ability to get external events
  - Some type of fault awareness/tolerance
  - Security Model
  - Stronger Progress Guarantees
  - Differentiated Service (QoS)
-

# Summary

- MPI-4 standardization is on-going
- Goal is to have standard by 2020, aligned with release of first Exascale machine in 2021
- Connection to the features/difficulties of exascale are limited
- Understanding user pain points needed
- Some activities are driven by ECP, EPIGRAM or other programs (e.g., Sessions, Finepoints, Persistent ...)