Overview of ExaMPI

Derek Schafer  
*University of Tennessee at Chattanooga*

Ryan Marshall  
*University of Tennessee at Chattanooga*

Tony Skjellum  
*University of Tennessee at Chattanooga*

Martin Ruefenacht  
*University of Tennessee at Chattanooga*

Follow this and additional works at: https://scholar.utc.edu/research-dialogues

**Recommended Citation**

Schafer, Derek; Marshall, Ryan; Skjellum, Tony; and Ruefenacht, Martin, "Overview of ExaMPI". *ReSEARCH Dialogues Conference proceedings*. https://scholar.utc.edu/research-dialogues/2020/day2_posters/106.
ExaMPI is an experimental MPI implementation designed to simpler to learn, modify, and use for middleware research.

**Overview of ExaMPI**

Derek Schafer, Ryan Marshall, Anthony Skjellum, Martin Ruefenacht

**Motivation**
- To create an MPI implementation that can be used to experiment with new MPI features with ease
- Enable rapid prototyping of new MPI ideas
- Identify and elucidate opportunities to improve MPI at-large
- Expand MPI's applicability
- Support experimentation and research on:
  - Resource management
  - Fault tolerance
  - New language bindings
  - Elastic MPI
  - MPI Sessions

**Components & Design**
- C++ 17, with a modern development style
- Modular components that facilitate experimentation and the ability to drop in various components rather than having their choice be fixed in the design
- Such components include:
  - Universe
    - A special class to avoid global state
  - Transports
    - Akin to other major MPI implementations.
    - Current transports include support for TCP, UDP, and Libfabric (coming soon)
  - Decider
    - An interface that allows for multiple Algorithms to be used for various communication operations
  - Matcher
    - Object responsible for matching messages
  - Progress Engine
    - Allows for different styles of progress

**Progress Engine**
- ExaMPI implements a strong progress engine that is independent from the user threads, with blocking notification of completion
- The diagram below showcases Dimitrov's Progress and Notification Classification Diagram[1]
- Other modes forthcoming

**Diagrams**
- The left diagram is a partial overview of ExaMPI components, showcasing some of the components that can be changed.
- The above diagram shows the current interface layers in ExaMPI and how ExaMPI can be integrated with different language abstractions by using the PMPI layer. The C interface is specifically shown above.

**Reference Paper**