University of Tennessee at Chattanooga

UTC Scholar

Apr 15th, 1:00 PM - 3:00 PM

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

Overview of ExaMPI

Derek Schafer, Ryan Marshall, Anthony Skjellum, Martin Ruefenacht

SimCenter Center of Excellence in Applied Computational Science & Engineering

THE UNIVERSITY OF TENNESSEE AT CHATTANOOGA

THE UNIVERSITY of EDINBURGH

Tennessee TECH

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

> ExaMPI is an **experimental MPI implementation** designed to simpler to **learn**, **modify**, and **use** for middleware research.
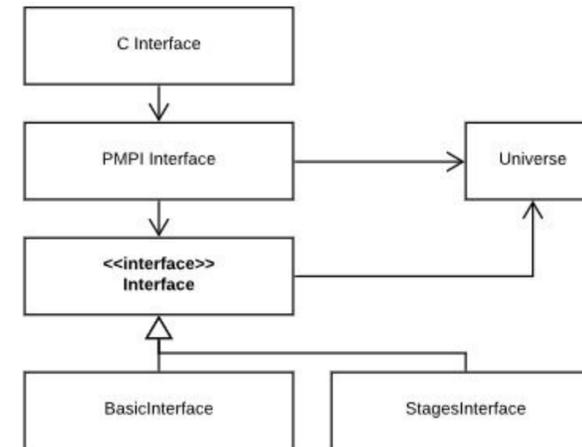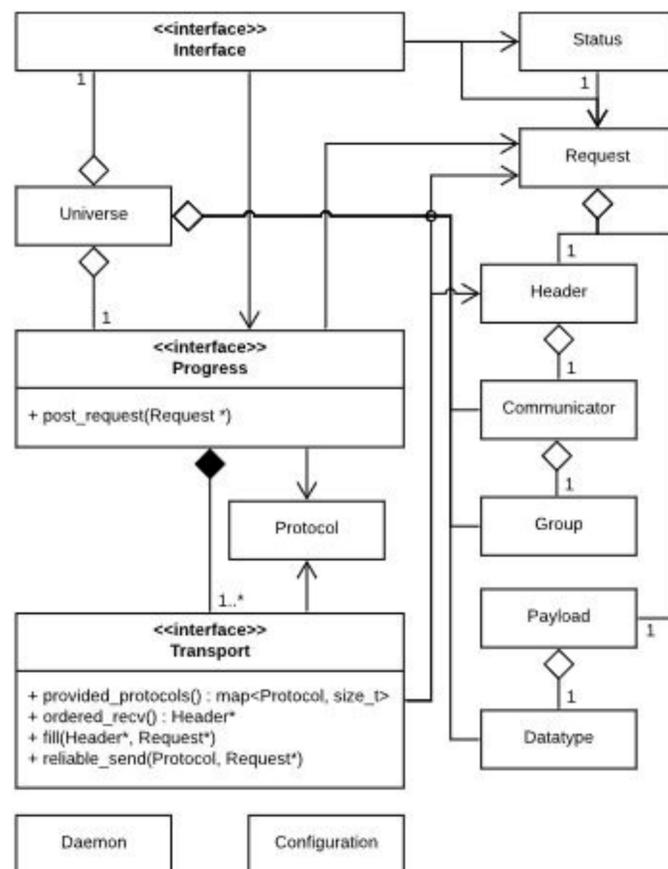
Diagrams

- 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.

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
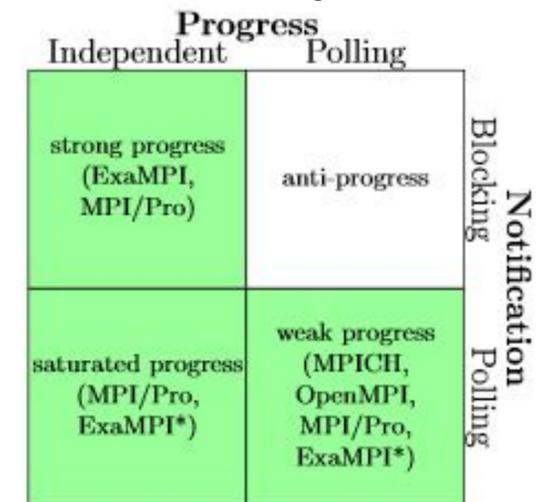
Featured Works

1. Dimitrov, Rossen. "Overlapping of Communication and Computation and Early Binding: Fundamental Mechanisms for Improving." (2001).
2. Sultana, Nawrin, et al. "MPI stages: Checkpointing MPI state for bulk synchronous applications." Proceedings of the 25th European MPI Users' Group Meeting. 2018.
3. Schafer Derek, et al. "User-Level Scheduled Communications for MPI." 2019 IEEE 26th International Conference on High Performance Computing, Data, and Analytics (HiPC), Hyderabad, India, 2019, pp. 290-300.

Reference Paper

- Skjellum, Anthony, et al. "ExaMPI: A Modern Design and Implementation to Accelerate Message Passing Interface Innovation." Latin American High Performance Computing Conference. Springer, Cham, 2019.