LA-UR-17-30189
Approved for public release; distribution is unlimited.

Title: ATDM LANL FleCSI: Topology and Execution Framework

Author(s): Bergen, Benjamin Karl

Intended for: Report

Issued: 2017-11-06

**Disclaimer:**
Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

## ATDM LANL FleCSI: Topology and Execution Framework

FleCSI is a compile-time configurable C++ framework designed to support multi-physics application development. As such, FleCSI attempts to provide a very general set of infrastructure design patterns that can be specialized and extended to suit the needs of a broad variety of solver and data requirements. This means that FleCSI is potentially useful to many different ECP projects. Current support includes multi-dimensional mesh topology, mesh geometry, and mesh adjacency information, n-dimensional hashed-tree data structures, graph partitioning interfaces, and dependency closures (to identify data dependencies between distributed-memory address spaces).

FleCSI introduces a functional programming model with control, execution, and data abstractions that are consistent with state-of-the-art task-based runtimes such as Legion and Charm++. The model also provides support for fine-grained, data-parallel execution with backend support for runtimes such as OpenMP and C++17. The FleCSI abstraction layer provides the developer with insulation from the underlying runtimes, while allowing support for multiple runtime systems, including conventional models like asynchronous MPI. The intent is to give developers a concrete set of user-friendly programming tools that can be used now, while allowing flexibility in choosing runtime implementations and optimizations that can be applied to architectures and runtimes that arise in the future.

This project is essential to the ECP Ristra Next-Generation Code project, part of ASC ATDM, because it provides a hierarchically parallel programming model that is consistent with the design of modern system architectures, but which allows for the straightforward expression of algorithmic parallelism in a portably performant manner. As such, FleCSI helps to address challenges of heterogeneity, scale, memory and I/O management, and Multiphysics algorithmic consistency. In particular, the FleCSI programming model makes it possible to investigate additive correction models—as opposed to multiplicative models—that expose greater task parallelism and that will allow more efficient resource utilization on exascale computing systems.

**FleCSI Project Overview**

| Scope & Intent | R&D Themes | Delivery Process | Target ECP Users | Support Model |
|---|---|---|---|---|
| Programming model for developing Multiphysics application projects with multiple task-based and data-parallel runtime backends | Parallelism & Concurrency, Data Management, Analysis (*in situ*), and Static Analysis (compiler-aided design). | Regular release of software and documentation, with package manager support (spack, RPM); anytime access to containers (Docker); anytime access to production software from GitHub. | Ristra is the primary target user. We depend on the Legion project, and potentially on Kokkos and Raja. | Email support, mailing lists, issue trackers, web documentation, and user manuals. |

**FleCSI FY18 Milestones**

| Milestone ID | Milestone Title | ECP Users |
|---|---|---|
| STMS12-6 [MS1/Y2, Dec 2017] | Phase-1 release of FleCSI for ECP | Ristra already uses the FleCSI topology data structures and runtime. |
| STMS12-7 [MS2/Y2, March 2018] | Dynamic Data Support & Interoperability | FleCSALE/FULE multi-material hydrodynamics; Interoperability with Trilinos and Hypre. |
| STMS12-8 [MS3/Y2, June 2018] | Static Analysis and Language Extensions | Ristra users are adapting physics codes to use new fine-grained, data-paralllel interface and analysis tools. |
| STMS12-9 [MS4/Y2, Sept 2018] | Data-Parallel Runtime Support | FleCSALE/FUEL Ristra applications. FleCSPH (smoothed-particle hydrodynamics). |

**FleCSI FY19 Milestones**

| Milestone ID | Milestone Title | ECP Users |
|---|---|---|
| NA | Phase-2 release of FleCSI for ECP | Ristra projects |
| NA | Deliver compiler tool (clang extension) for data-parallelism and static analysis. | Ristra projects |
| NA | Updated documentation and examples for data-parallel interface | Ristra projects |
| NA | Phase-3 release of FleCSI for ECP | Ristra projects |

**FleCSI Impact Goals & Metrics**

| Goal | Metric |
|---|---|
| FleCSI framework is used to satisfy Ristra Multiphysics application deliverables. | Delivery of radiation-hydrodynamics simulation code that can solve selected problem set. |
| FleCSI static analysis tool identifies interface calls and emits errors for incorrect syntax. | Number of interface calls detected (of total), number of errors detected (of total), for a selected test input. |