ARCHER eCSE Final Report

eCSE Number:	CSE01-009
eCSE Title:	Scabable and interoperable I/O for Fluidity
Date of Submission:	12/06/15
PI and Co-Is	Gerard Gorman and Jon Hill
Technical staff member(s)	Michael Lange
Author of this document	Michael Lange
Project start date	01/07/14
Project completion date	31/12/14
Total number of funded project months	6

1 Publishable Report

1.1 *Achievement of objectives*

The key objectives for the Fluidity CFD code, mainly improved file I/O, format interoperability have been achieved, while significant additional contributions to the underlying PETSc library have been made. These contributions include performance optimisation and feature additions that have been made available to the scientific to a large user base.

1.2 Achievements and Impact

1.2.1 Performance improvements

As part of this work significant performance and scalability improvements have been achieved for PETSc's DMPlexDistribute() interface, which facilitates run-time mesh partitioning and distribution [1]. The resulting performance increase is then utilised to improve the mesh initialisation performance of the Fluidity CFD code through run-time distribution and more efficient data migration [2].

1.2.2 Additional functionality

In addition to performance improvements the capabilities of PETSc's DMPlexDistribute interface have been extended to include load-balancing and re-distribution of parallel meshes, as well as the ability to generate multiple levels of partition overlap [1]. Furthermore, support for additional mesh file formats has been added to DMPlex and Fluidity, including binary Gmsh, Fluent-Case [2].

1.2.3 Other achievements

Academic impact / submitted papers:

[1] - Unstructured Overlapping Mesh Distribution in Parallel; M. Knepley, M. Lange, G. Gorman; submitted to ACM Transactions on Mathematical Software; http://arxiv.org/abs/1506.06194; 2015

[2] - Flexible, Scalable Mesh and Data Management using PETSc DMPlex; M. Lange, M. Knepley, G. Gorman; submitted for EASC2015 conference proceedings; http://arxiv.org/abs/1505.04633; 2015

1.3 *Publishable summary*

Scalable file I/O and efficient domain topology management present important challenges for many scientific applications if they are to fully utilise future exascale computing resources. Designing a scientific software stack to meet next-generation simulation demands, not only requires scalable and efficient algorithms to perform data I/O and mesh management at scale, but also an abstraction layer that allows a wide variety of application codes to utilise them and thus promotes code reuse and interoperability. PETSc provides such an abstraction of mesh topology in the form of the DMPlex data management API.

During the course of this project PETSc's DMPlexDistribute API has been optimised and extended to provide scalable generation of arbitrary sized domain overlaps, as well as efficient mesh and data distribution, contributing directly to improving the flexible and scalable domain data management capabilities of the library. The ability to perform parallel load balancing and re-distribution of already parallel meshes was added, which enables further I/O and mesh management optimisations in the future. Moreover, additional mesh input formats have been added to DMPlex, including a binary Gmsh and a Fluent-CAS file reader, which improves the interoperability of DMPlex and all it's dependent user codes. A key aspect of this work was to maximise impact by adding features and applying optimisations at a library level in PETSc, resulting in benefits for a several application codes.

The main focus of this project, however, was the integration of DMPlex into the Fluidity mesh initialisation routines. The new Fluidity version utilises DMPlex to perform on-the-fly domain decomposition to significantly improve simulation start-up performance by eliminating a costly I/O-bound pre-processing step and improved data migration (see Illustration 1). Moreover additional new mesh input formats have been added to the model via new reader routines available in the public DMPlex API. Due to the resulting close integration with DMPlex, mesh renumbering capabilities, such as the Reverse Cuthill-McKee (RCM) algorithm provided by DMPlex, can now be leveraged to improve the cache coherency of Fluidity simulations. The performance benefits of RCM mesh renumbering for velocity assembly and pressure solve are show in Illustration 2 and 3.

1.4 *Summary of the software*

Most of the relevant DMPlex additions and optimisations are available in the current master branch of the PETSc development version, as well as the latest release version 3.6. A access to a central developer package for tracking petsc-master on Archer has been provided by the support team and is being maintained by the lead developer until the required features and fixes are available in the latest cray-petsc packages.

The current implementation of the DMPlex-based version of the Fluidity CFD code is available through a public feature branch:
https://github.com/FluidityProject/fluidity/tree/dmplexreader
Efforts in providing the full set of Fluidity features through DMPlex are ongoing and integration of this new feature into the next Fluidity release is actively being prepared.

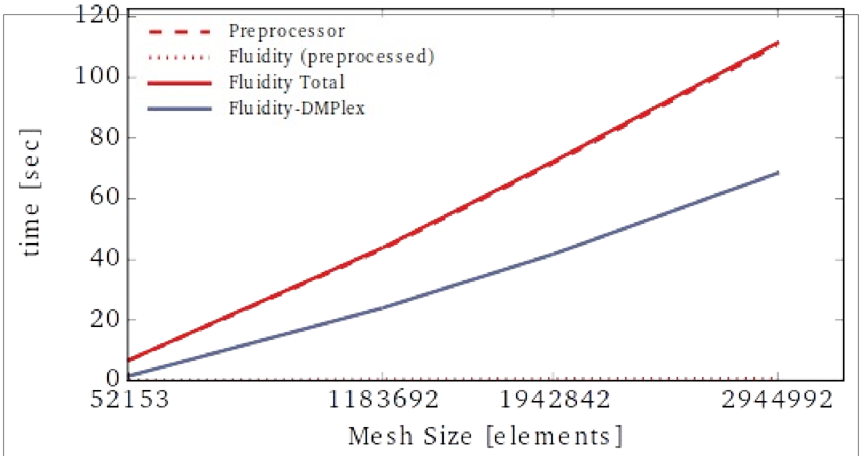

Illustration 0: Fluidity start-up improvement through domain decomposition via DMPlex.

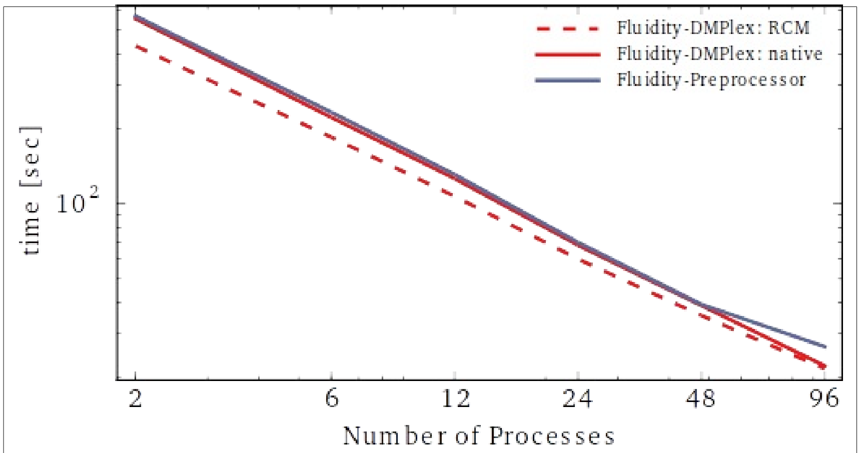

Illustration 0: Fluidity performance increase for velocity assembly from RCM mesh renumbering.

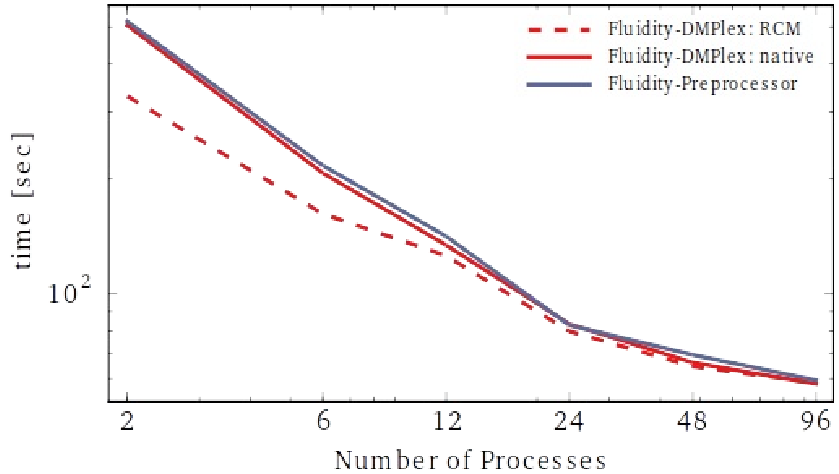

Illustration 0: Fluidity performance increase for pressure assembly from RCM mesh renumbering.