

Extending just-in-time compilation for OP2

Nathan Dunne - u1604486

October 3, 2019

Problem Statement

OP2 is an Embedded Domain Specific Language for solving unstructured mesh based applications. It exists to provide a level of abstraction to scientists and domain application developers, allowing utilisation of the high performance benefits of advancements in hardware, without requiring them to maintain an high level of knowledge on any new technologies and architectures. As the high-level application written using the OP2 API no longer requires any platform specific optimisations, it can be used on any platform supported by OP2; saving the time and financial costs of rewriting the application with different hardware optimisations.

One way to improve the performance of an application such as the unstructured meshes targetted by OP2 is using Just-In-Time (JIT) compilation, whereby performance sensitive sections of the application are re-compiled at runtime to apply additional optimisations once the parameters of execution are known. The performance gain must be sufficient to offset the extra time taken to recompile in order for this technique to be effective. There is existing work implemented already for some JIT optimisations, however there is still space for further performance gain. The aim of this project is to implement additional optimisations in this area, such as replacing library operations with constants, hard coding loop bounds, and applying loop fission/fusion.

There are a number of High Performance Computing applications already written in the OP2 api, from smaller example projects to production applications. Since the API itself will not be altered, it will be easy to apply my additions to the OP2 translator, and determine whether there is performance benefit using benchmarks. The goal of adding further optimisations to the OP2 library is that the performance of these applications is improved with no cost to accuracy.