

PDESoft 2014

Abstract

Extruded Meshes for Large Scale Problems using Firedrake and PyOP2

Gheorghe-Teodor Bercea, Andrew T. T. McRae,
Florian Rathgeber, Fabio Luporini, Lawrence Mitchell, Nicolas Lorient,
David A. Ham, Paul H. J. Kelly

Department of Computing, Imperial College London,
The Grantham Institute for Climate Change, Imperial College London

March 20, 2014

Abstract

Increasingly complex multi-core technology has added a new dimension in writing fast high-performance software, often requiring platform specific tuning. At the highest abstraction level, mathematical concepts need expressive, compose-able specifications that capture the generality implied by a class of problems. Firedrake and PyOP2 bring together the solving of Partial Differential Equations (PDEs) through the Finite Element Method (FEM) and the exploitation of hardware specific optimizations into a single tool chain. By using the Unified Form Language (UFL) in conjunction with the FENICS Form Compiler (FFC), Firedrake is able to automatically generate the computational kernels to be executed over the domain. To support the generality of FEM problems, FIAT is used to generate the meta-data for the different element types which characterize the discretisation. PyOP2 handles the scheduling of the kernel execution over the mesh in order to exploit the parallelism built into various architectures. Meshes, in general, are a machine level discrete approximation of a continuous mathematical domain. The domain is split into non-overlapping cells with homogeneous mathematical properties. A cell has values, also known as degrees of freedom, that characterize the cell's contribution to the overall domain and its relationship with a neighboring cell through the values shared. In an extruded mesh a cell's discretization can be expressed as an outer product of two other mesh element discretizations. Fixing one of the two elements of the product to an interval topology, the final extruded mesh gains a structured direction which can be exploited from a performance point of view. In this way, the performance penalties incurred by extruded meshes with an unstructured base mesh, can be counteracted by having an additional structured direction. In this

paper we will focus on showing the performance of extruded meshes when running PDE solves at scale. The PDEs considered, such as Helmholtz and Burgers Equations, were run on the Imperial College Cluster and the Batch 2 32-core machine in the Computing Department. The results show the behavior of the extruded meshes under varying mesh sizes and discretizations. We looked at the individual performance of the different parallel loops generated for each PDE. At low core counts the computations within the kernel dominate the timing, hence very low valuable bandwidths are achieved. Increasing the number of cores, adds more pressure on the memory bandwidth and therefore the parallel loops shift from being compute bound to being memory bandwidth bound.