PyExaFMM: Designing a high-performance particle fast multipole solver in Python with Numba

S. Kailasa
Department of Mathematics, University College London

T. Betcke
Department of Mathematics, University College London

T. Wang
Department of Mechanical and Aerospace Engineering, The George Washington University

L. A. Barba
Department of Mechanical and Aerospace Engineering, The George Washington University

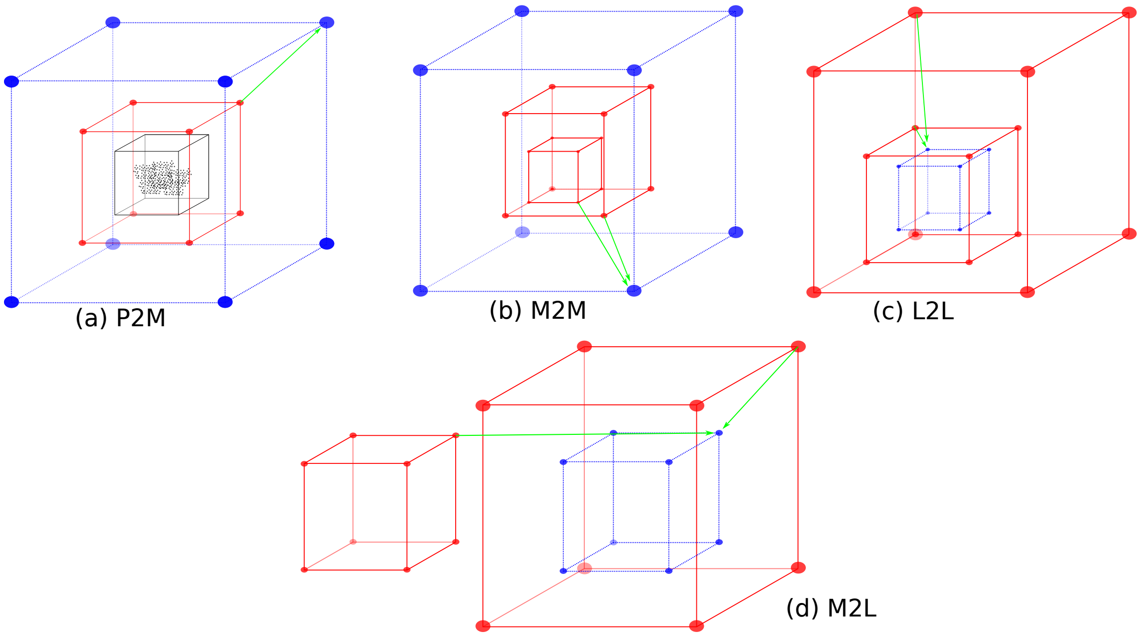
***Abstract*—The particle fast multipole method [FMM] is a good case study for understanding the efficacy of Python for developing high-performance software for non-trivial algorithms, due its reliance on a hierarchical tree data structure. In this paper we explore the mathematical and software design techniques used to extract performance for PyExaFMM, a Python based FMM solver accelerated with Numba, designed to be run on single-node multicore architectures. We report that we achieve runtimes within $\mathcal{O}(N)$ of the state of the art C++ implementation, with comparable accuracy and memory footprint for three dimensional problems in double precision.**

WE INTRODUCE PYEXAFMM[1]

- What's the context of PyExaFMM? - Compiled language implementations exist - Numba JIT compiler for Python can translate hot loops code to efficient machine code. - Python is well understood in SC community. - FMM is a benchmark for complex SC software built using Python alone. - This paper consists of: - Brief intro to FMM algorithm (¡1000 words) - Math methods, software design and parallelization strategies for achieving performance with Numba. (2500 words) - Benchmarks wrt comparable C++ module (1500 words)

[1] https://www.github.com/exafmm/pyexafmm

Published by the IEEE Computer Society



(a) P2M

(b) M2M

(c) L2L

(d) M2L

Figure 1. The operators of the KIFMM. Equivalent surfaces are shown in red, check surfaces in blue, and the charged points in black. Surfaces are plotted with 8 quadrature points, one at each vertex.

FAST MULTIPOLE METHODS

HIGH PERFORMANCE PYTHON WITH NUMBA

TECHNIQUES FOR ACHIEVING PERFORMANCE

Efficient Operators

PERFORMANCE

CONCLUSION

ACKNOWLEDGMENT

SK is supported by EPSRC Studentship 2417009.

REFERENCES

Srinath Kailasa is a Graduate Student at University College London, currently pursuing a PhD in Computational Mathematics. He received an MPhys in Physics (2017) and an MSc in Scientific Computing (2020) from the University of Durham, and University College London respectively, interspersed with time as a Software Engineer in industry. His research interests are in high-performance and scientific computing. Contact him at srinath.kailasa.18@ucl.ac.uk.

Timo Betcke is Professor of Computational Mathematics at University College London. Is the lead investigator of the Bempp project, an open-source boundary element library. He studied Engineering in Germany as Undergraduate and then completed a PhD in Oxford in Numerical Analysis. From 2005 to 2006 he had various research positions until he became a Lecturer at UCL in 2011. Since 2018 he is a full Professor in the Department of Mathematics at UCL. Contact him at t.betcke@ucl.ac.uk.

Tingyu Wang is a PhD student in Mechanical Engineering at the George Washington University. Contact him at twang66@email.gwu.edu.

Lorena. A. Barba is a Professor of Mechanical and Aerospace Engineering at the George Washington University. Contact her at labarba@email.gwu.edu.