Impact of Using A Modern Compiler and Optimized Workflow on the Performance of a Complex Nuclear Fusion Application for ITER

Someswar Dutta[1], Deepak Aggarwal[2]

[1]SST-1 Operations Division, Institute For Plasma Research, Gandhinagar, India

[2]University Information Services, University of Cambridge, Cambridge, United Kingdom

Email: someswarduttabhu@gmail.com; da616@cam.ac.uk

Presented at Research Software Engineering Conference Sept 3 - 5, 2024, Newcastle, United Kingdom

P-165

Abstract

Optimizing scientific applications to fully exploit the vast computational power of HPC systems necessitates substantial expertise and long-term dedication to code optimization. This work presents a pragmatic approach tailored to enhance the performance of PARTICLE-3D (P3D), a nuclear fusion application code crucial for simulating runaway electron dynamics - a critical aspect impacting the operational lifespan of nuclear fusion machines like ITER. Transitioning from a legacy compiler dependency, the authors identified performance bottlenecks within the serial version of the code, particularly pertaining to I/O communication between subroutine calls. By meticulously restructuring the code and refining inter-subroutine communication, the work highlights the process adopted by the authors to unlock embarrassingly parallel computation workflow on the HPC cluster. The optimized version of P3D exhibits a performance improvement of several orders of magnitude compared to its base version. These optimizations not only elevated the performance but also improved the code portability, ensuring seamless integration with diverse HPC architectures equipped with modern compilers. The methodology described here can serve as a valuable case study for early researchers and code developers to enhance the performance of any such code designed for problems characterised by the similar algorithmic topologies. This work was completed within a month and underscores the importance of effective communication between users and administrators.

Challenges Addressed: Code portability, Selection of compiler, Role of code profilers, Optimization

1 PARTICLE-3D (P3D) Code : A Nuclear Fusion Application for ITER

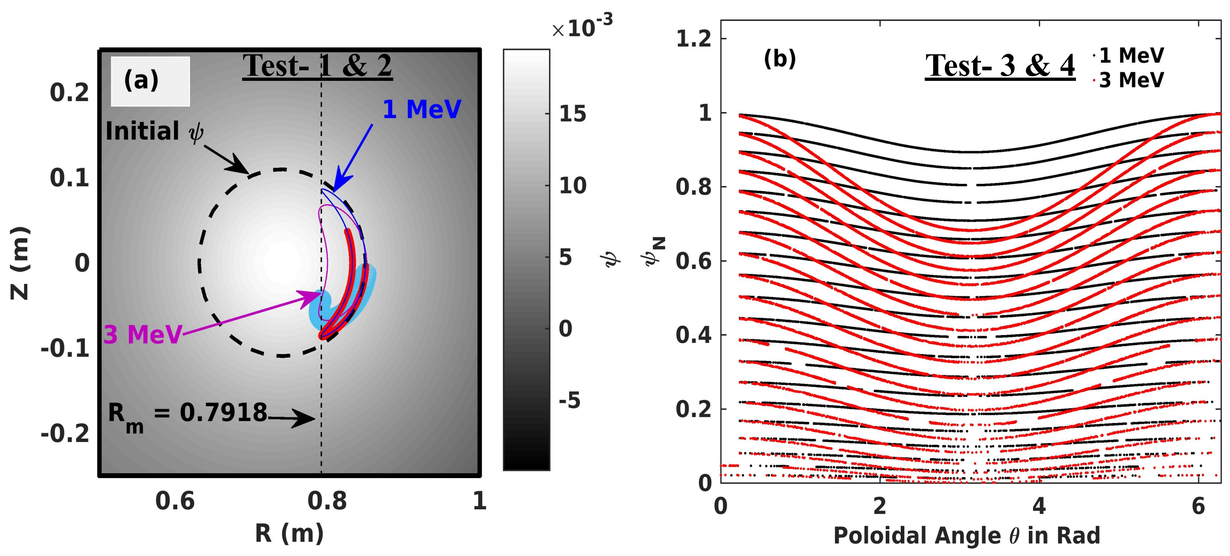
- Runaway electrons (REs), highly energetic (several MeV) electrons that occur in all tokamaks, can significantly impact plasma operation and damage machine components, making their study and subsequent mitigation crucial to nuclear fusion research [1].
- Recently local vertical field (LVF) technique experimentally demonstrated REs mitigation in ADITYA tokamak [2], with numerical modelling conducted with P3D code [1, 3].
- P3D is a relativistic full orbit following code for studying such energetic particles in any tokamak including ITER.

Test Cases To Asses P3D Code Performance

- Test-1: Banana Orbit of a 1 MeV RE with 1 poloidal transit, pitch angle 75°.
- Test-2: Banana Orbit of a 3 MeV RE with 1 poloidal transit, pitch angle 75°.
- Test-3: Poincaré section plot for 20 REs of 1 MeV, 60 poloidal transits.
- Test-4: Poincaré section plot for 20 REs of 3 MeV, 60 poloidal transits.

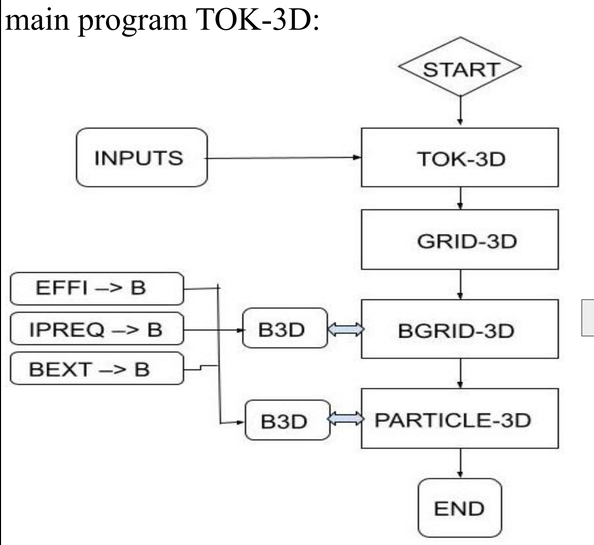
Test-1 and 2 assess performance optimization related to compiler selection (LF95 and IFORT), refactoring and I/O on a single core for single RE. Test-3 and 4 evaluate the overall performance of the P3D code with parallelization.

2 PARTICLE-3D (P3D) Code: Banana Orbits and Poinceré Section Plot of REs

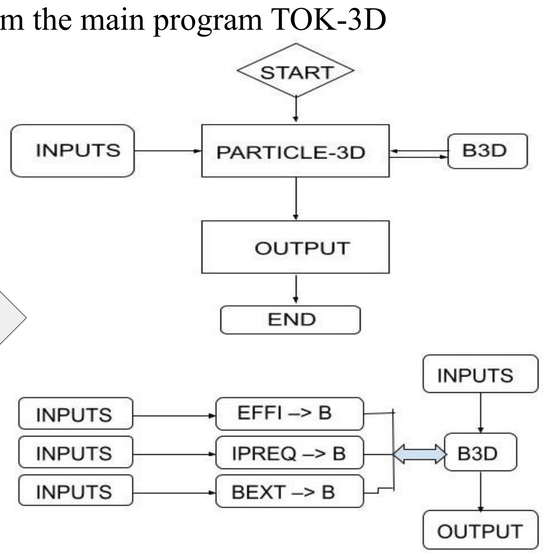


3 Flow Chart of the P3D code Versions (Refactorization)

Base version: Here P3D code is run by a main program TOK-3D:



Modular version: P3D code separated from the main program TOK-3D



4 Profiling of P3D Code Versions Using Intel Vtune Profiler

Modular version: For a test run, this version took large amount of time reading an input file through a runtime library file libpathread.so.0.

Elapsed Time: 976.602s
CPU Time: 964.960s
Total Thread Count: 1
Paused Time: 0s

Function	Module	CPU Time
__read	libpthread.so.0	515.023s
for__read_input	ip3dexe.out	361.910s
__strncpy_chk	libc.so.6	11.479s
for__resource_acquire	ip3dexe.out	7.640s
for__resource_release	ip3dexe.out	7.360s
[Others]		61.548s

I/O Optimized version: Modular version is modified for reading that particular input file, leading to much improved performance.

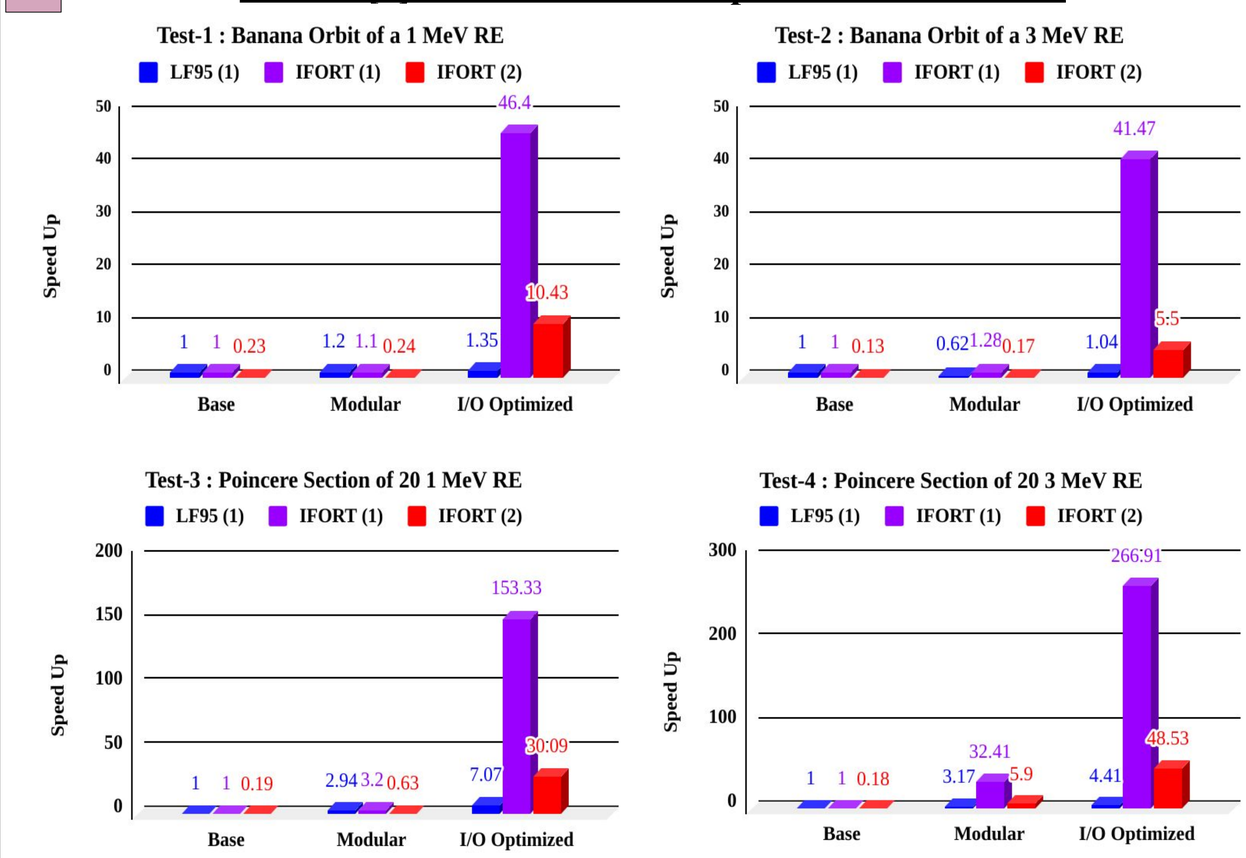
Elapsed Time: 8.934s
CPU Time: 8.410s
Total Thread Count: 1
Paused Time: 0s

Function	Module	CPU Time
ibcdcu	ip3dexe_1g.out	6.311s
open	libpthread.so.0	0.383s
brbz	ip3dexe_1g.out	0.347s
write	libpthread.so.0	0.300s
ftruncate	libc.so.6	0.180s
[Others]		0.888s

5 Performance Enhancement Using Parallelization

- Parallel threads trace individual REs with no inter-thread communication required.
- The Embarrassingly Parallel topology [4] is used to parallelize the P3D code.
- Implementation is done via Array Job deployment using the PBS Job Scheduler.
- Tests 3 and 4 demonstrate the parallel performance [5].

6 Results [6]: Performance Comparison of P3D Code



7 Discussion and Conclusions

- The dynamics of REs are highly nonlinear, with each simulation differing from the others. Four distinct cases were assessed to evaluate code performance.
- The legacy LF95 compiled code showed limited improvement, while the IFORT compilation significantly boosted performance and enhanced code portability.
- Code refactorization and I/O optimization, guided by the Intel VTune profiler, played a crucial role in identifying and rectifying performance hotspots.
- Understanding the algorithmic topology enabled the implementation of an embarrassingly parallel approach, resulting in substantial performance improvements.

Speedup of optimized P3D over the base version

Test cases	Speed up
Test-1	10.43
Test-2	5.50
Test-3	30.09
Test-4	48.53

Speedup values are specific to the simulations and cannot be linearly scaled for comparison. For 20 REs, a 48.53x performance boost was achieved relative to the base version.

References:

1. Someswar Dutta et al 2024 Nucl. Fusion 64 096027
2. Bhatt S.B. et al 1989 ADITYA: the first Indian tokamak Indian J. Pure Appl. Phys. 27 710
3. Someswar DUTTA et al 2019 Plasma Sci. Technol. 21 105101
4. N. Wang et al 2004 tribology transactions, vol. 47, no. 1, pp. 34–42
5. D. Aggarwal et al 2022 IEEE 29th HIPCW pp. 36-43
6. Results: GitHub Repository

Contacts

Someswar Dutta LinkdIn

Deepak Aggarwal LinkdIn