

# LLOV: A FAST STATIC DATA-RACE CHECKER FOR OPENMP PROGRAMS

Utpal Bora<sup>1</sup> and Santanu Das<sup>1</sup> and Pankaj Kukreja<sup>1</sup> and  
Saurabh Joshi<sup>1</sup> and Ramakrishna Upadrasta<sup>1</sup> and Sanjay Rajopadhye<sup>2</sup>

<sup>1</sup>Computer Science and Engineering, IIT Hyderabad  
<sup>2</sup>Computer Science and Engineering, Colorado State University



## Data Race Definition

**Theorem 1 (Data Race)** An execution of a concurrent program is said to have a data race when two different threads access the same memory location, these accesses are not protected by a mutual exclusion mechanism (e.g., locks), the order of the two accesses is non-deterministic and one of these accesses is a write.

## Problem Statement

Data races are common source of bugs in parallel programs. Specifying parallelism with frameworks such as OpenMP is relatively easy, but data races in these programs are an important source of bugs. There exists dynamic bug detection tools for these programs. However, they take a lot of time to report bugs or miss data races altogether. In this paper, we propose LLOV, a fast, language agnostic, and static data race checker for OpenMP programs.

## Methodology

```
for (i=0; i<10; i++) {
  #pragma omp parallel for
  for (j=1; j<10; j++) {
    b[i][j]=b[i][j-1];
  }
}
```

Iteration Domain :  $I = \{SO(i, j) : 0 \leq i \leq m-1 \wedge 1 \leq j \leq n-1\}$   
Schedule :  $S = \{SO(i, j) \rightarrow (i, j)\} \cap_{dom} I$   
Access Map :  $A = \{SO(i, j) \rightarrow M(i, j); SO(i, j) \rightarrow M(i, j-1)\}$   
Dependences :  $D = \{SO(i, j) \rightarrow (i, j-1) : 0 \leq i \leq m-1 \wedge 1 \leq j \leq n-1\}$

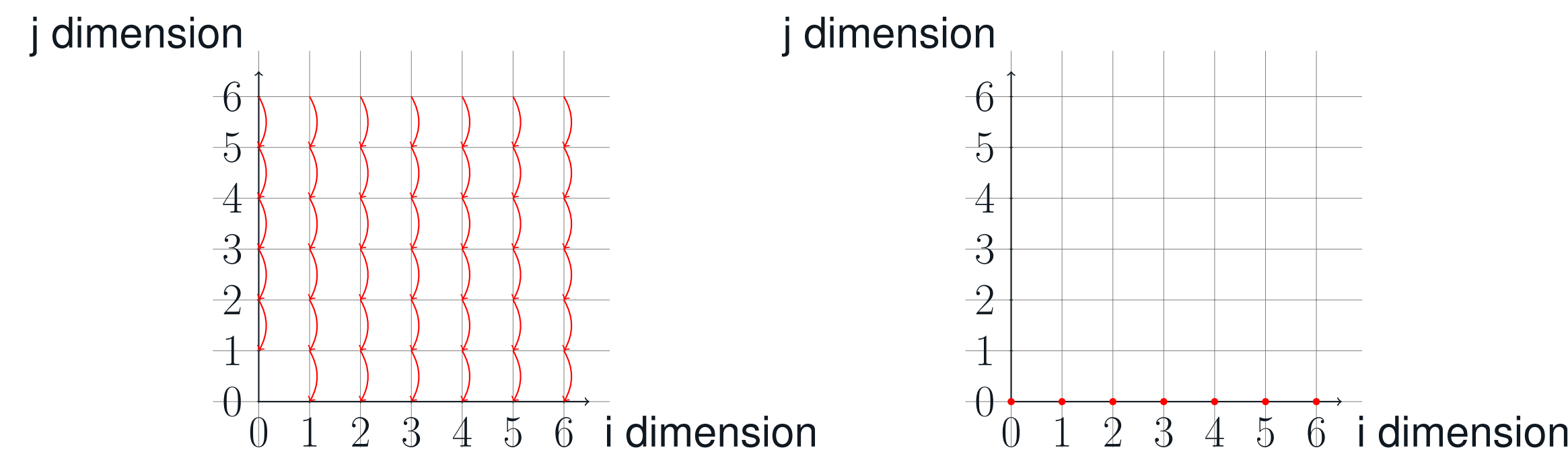


Figure 1: Dependence Polyhedra

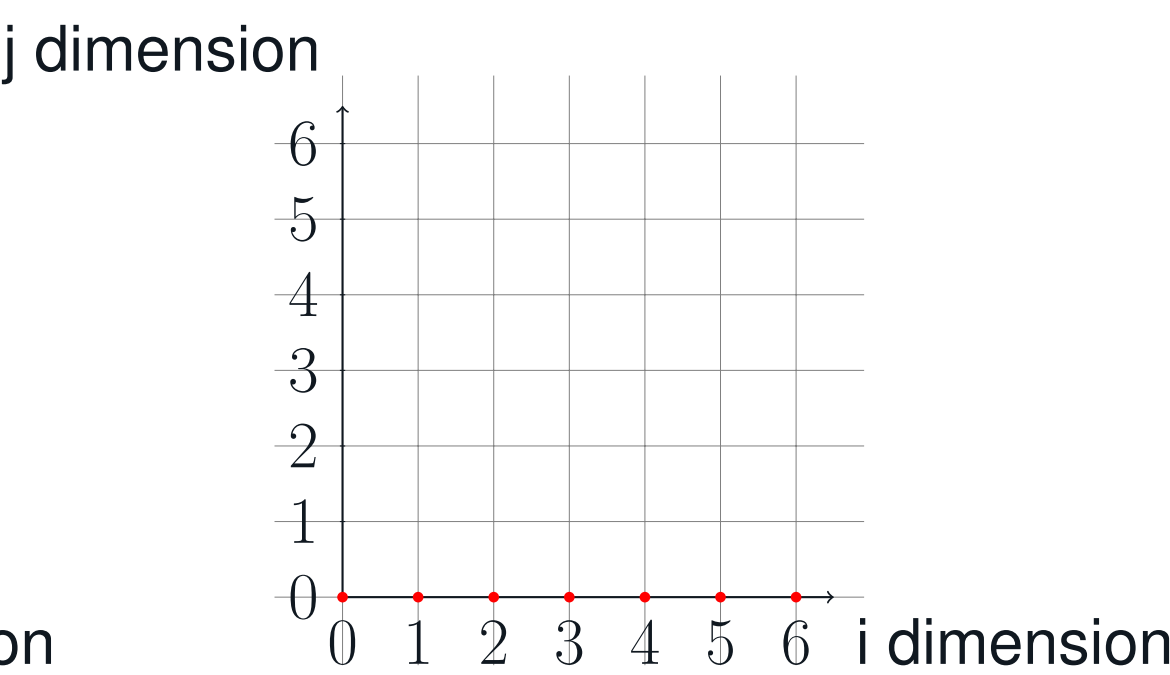


Figure 2: Projection of the Dependence Polyhedra on i-dimension

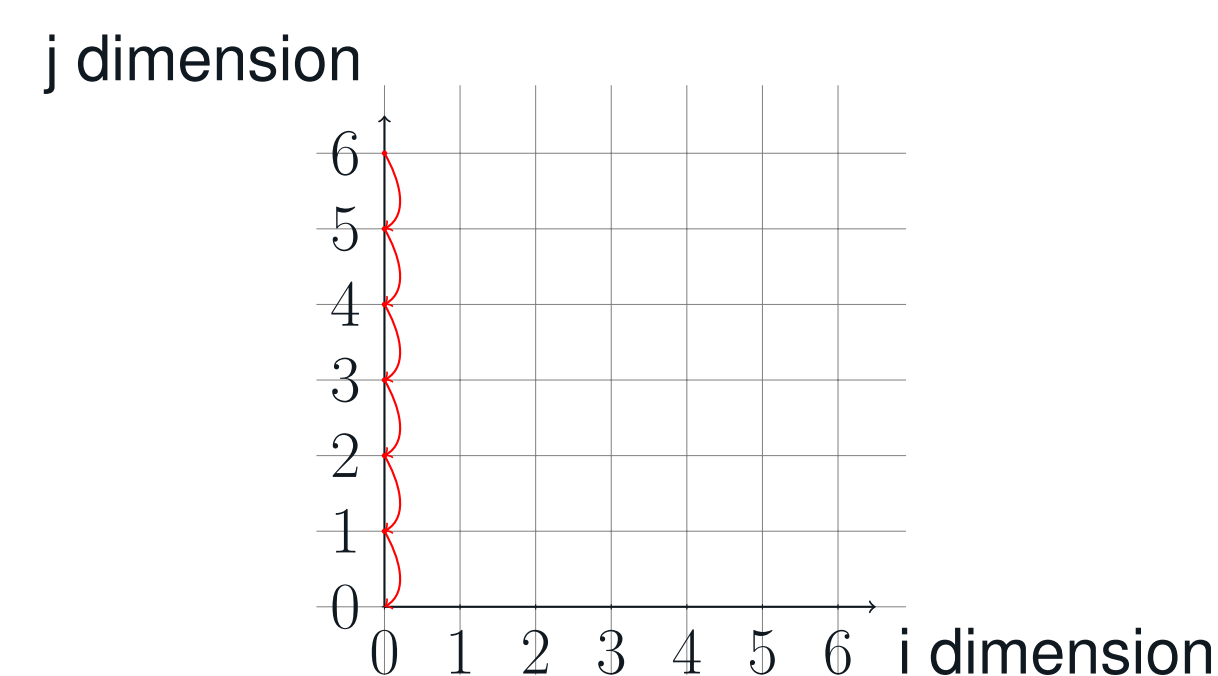


Figure 3: Projection of the Dependence Polyhedra on j-dimension

## Workflow

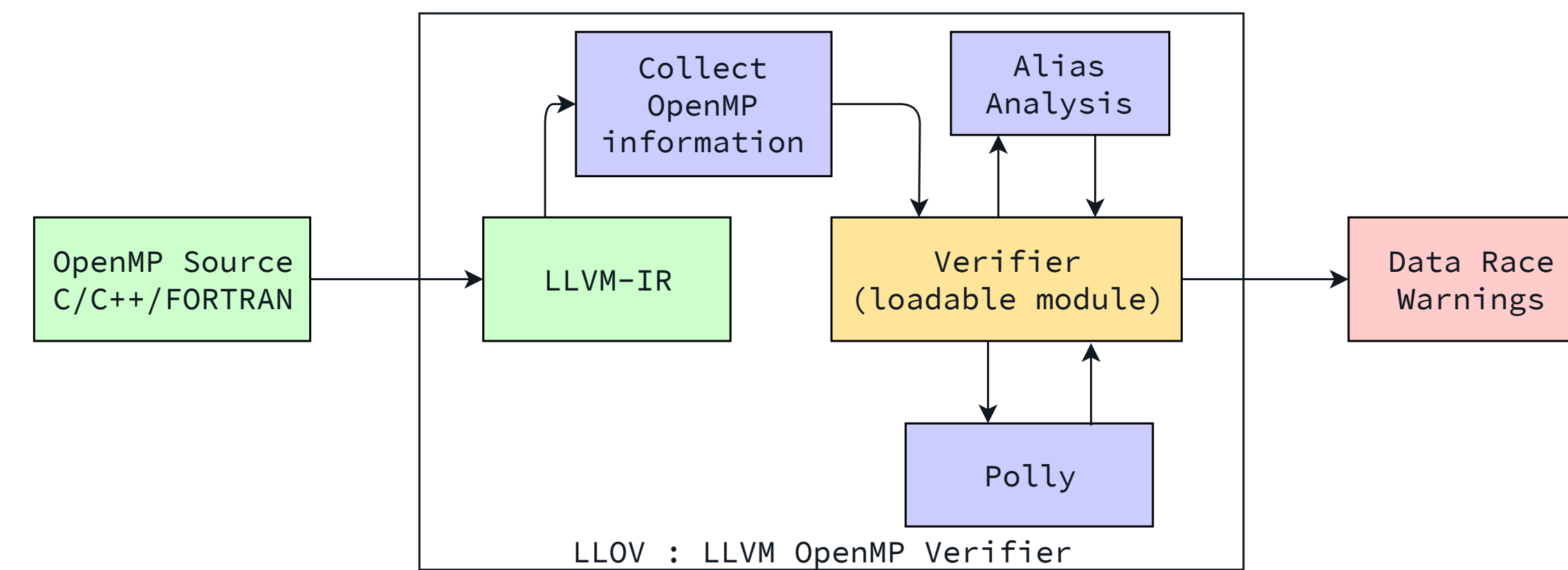


Fig. 1: LLOV Flow Diagram.

## Results

Table 1: Race detection tools with the version numbers used for comparison

| Tools            | Source            | Version / Commit |
|------------------|-------------------|------------------|
| HELGRIND [5]     | Valgrind          | 3.13.0           |
| VALGRIND DRD [4] | Valgrind          | 3.13.0           |
| TSAN-LLVM [3]    | LLVM              | 6.0.1            |
| ARCHER [1]       | git master branch | fc17353          |
| SWORD [2]        | git master branch | 7a08f3c          |

Table 2: Maximum number of Races reported by different tools in DataRaceBench 1.2

| Tools        | Race: Yes |    | Race: No |    | Coverage/116 |
|--------------|-----------|----|----------|----|--------------|
|              | TP        | FN | TN       | FP |              |
| HELGRIND     | 56        | 3  | 2        | 55 | 116          |
| VALGRIND DRD | 56        | 3  | 26       | 31 | 116          |
| TSAN-LLVM    | 57        | 2  | 2        | 55 | 116          |
| ARCHER       | 56        | 3  | 2        | 55 | 116          |
| SWORD        | 47        | 4  | 24       | 4  | 79           |
| LLOV         | 48        | 2  | 36       | 5  | 91           |

Table 3: Performance of the tools on DataRaceBench 1.2

| Tools        | Precision | Recall | Accuracy | F1 Score | Diagnostic odds ratio |
|--------------|-----------|--------|----------|----------|-----------------------|
| HELGRIND     | 0.50      | 0.95   | 0.50     | 0.66     | 0.68                  |
| VALGRIND DRD | 0.64      | 0.95   | 0.71     | 0.77     | 15.66                 |
| TSAN-LLVM    | 0.51      | 0.97   | 0.51     | 0.67     | 1.04                  |
| ARCHER       | 0.50      | 0.95   | 0.50     | 0.66     | 0.68                  |
| SWORD        | 0.92      | 0.92   | 0.90     | 0.92     | 70.50                 |
| LLOV         | 0.91      | 0.96   | 0.92     | 0.93     | 172.80                |

## Results

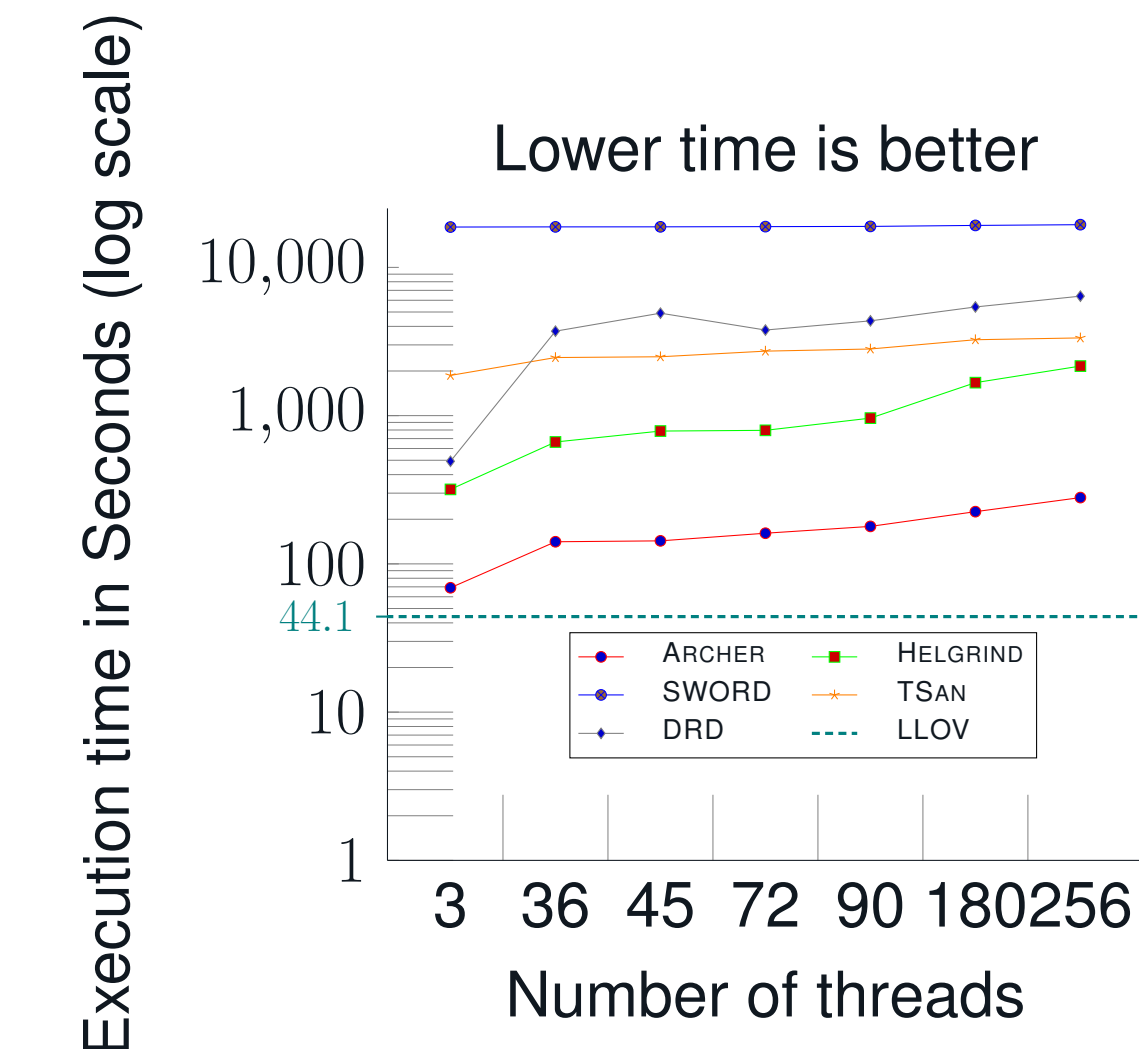


Figure 4: DataRaceBench v1.2 total execution time by different tools on logarithmic scale

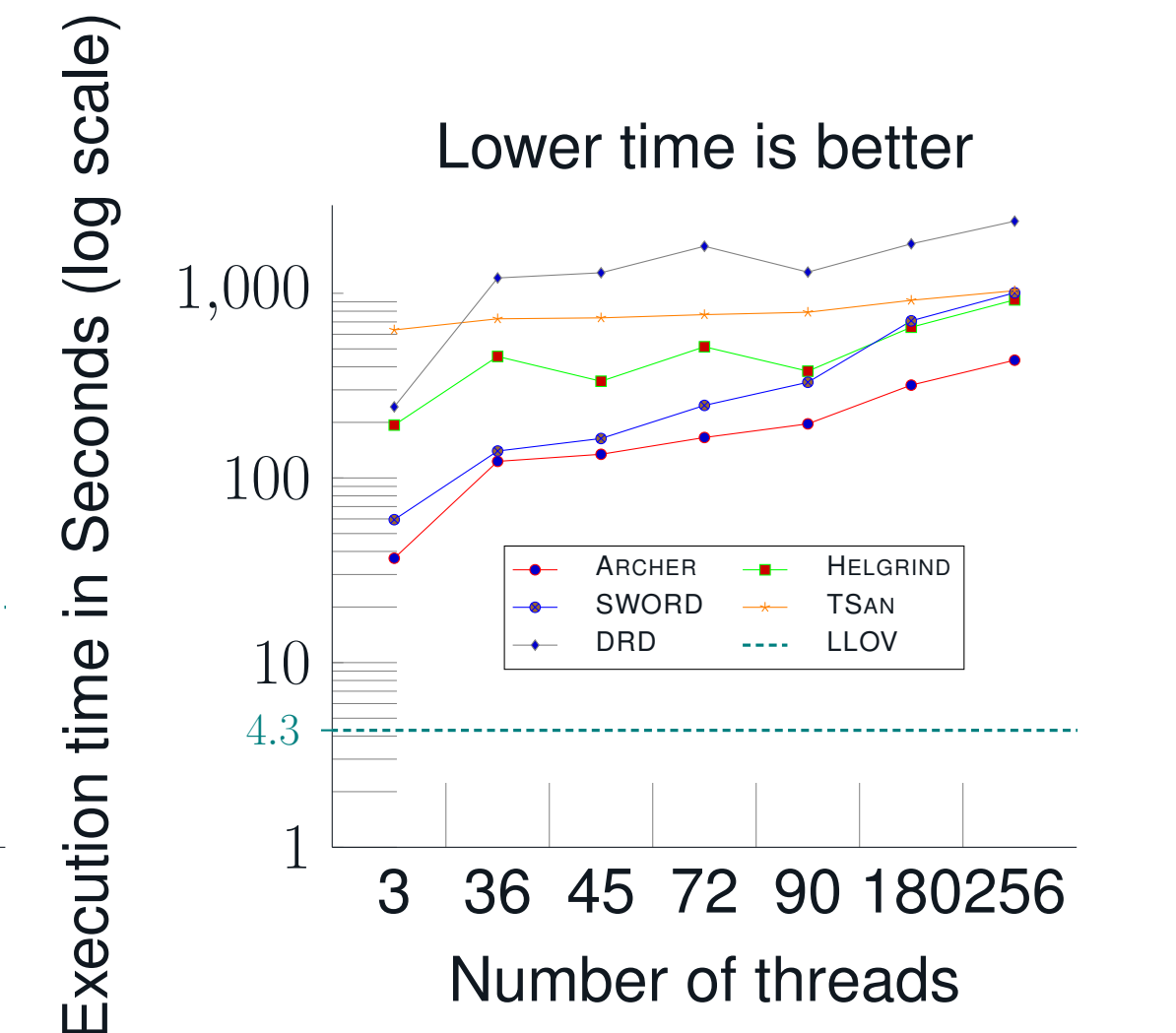


Figure 5: DataRaceBench v1.2 total time taken by different tools for common 61 kernels on logarithmic scale

## Conclusion

We propose and implemented a fast, static data-race checker for OpenMP programs. LLOV is freely available for download.

- Link: <https://github.com/utpalbora/llov>
- Blog: <https://compilers.cse.iith.ac.in/projects/llov/>
- DataRaceBench FORTRAN: [https://github.com/IITH-Compilers/dr\\_b\\_fortran](https://github.com/IITH-Compilers/dr_b_fortran)

## Acknowledgements

We thank Tobias Grosser, Johannes Doerfert, Michael Kruse for their help with the initial version of Polly as an analysis pass, which we extended for this work. We also thank Govindarajan Ramaswamy and V Krishna Nandivada for their feedback on this work.

## References

- [1] Simone Atzeni et al. "ARCHER: effectively spotting data races in large OpenMP applications". In: *Parallel and Distributed Processing Symposium, 2016 IEEE International*. IEEE, Chicago, IL, USA: IEEE, 2016, pp. 53–62.
- [2] Simone Atzeni et al. "Sword: A bounded memory-overhead detector of OpenMP data races in production runs". In: *2018 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, Vancouver, BC, Canada: IEEE, 2018, pp. 845–854.
- [3] Konstantin Serebryany and Timur Iskhodzhanov. "ThreadSanitizer: Data Race Detection in Practice". In: *Proceedings of the Workshop on Binary Instrumentation and Applications*. WBIA '09. New York, New York, USA: ACM, 2009, pp. 62–71. ISBN: 978-1-60558-793-6. DOI: 10.1145/1791194.1791203. URL: <http://doi.acm.org/10.1145/1791194.1791203>.
- [4] Valgrind-project. *DRD: a thread error detector*. <http://valgrind.org/docs/manual/drd-manual.html>. [Online; accessed 08-May-2019]. 2007.
- [5] Valgrind-project. *Helgrind: a thread error detector*. <http://valgrind.org/docs/manual/>