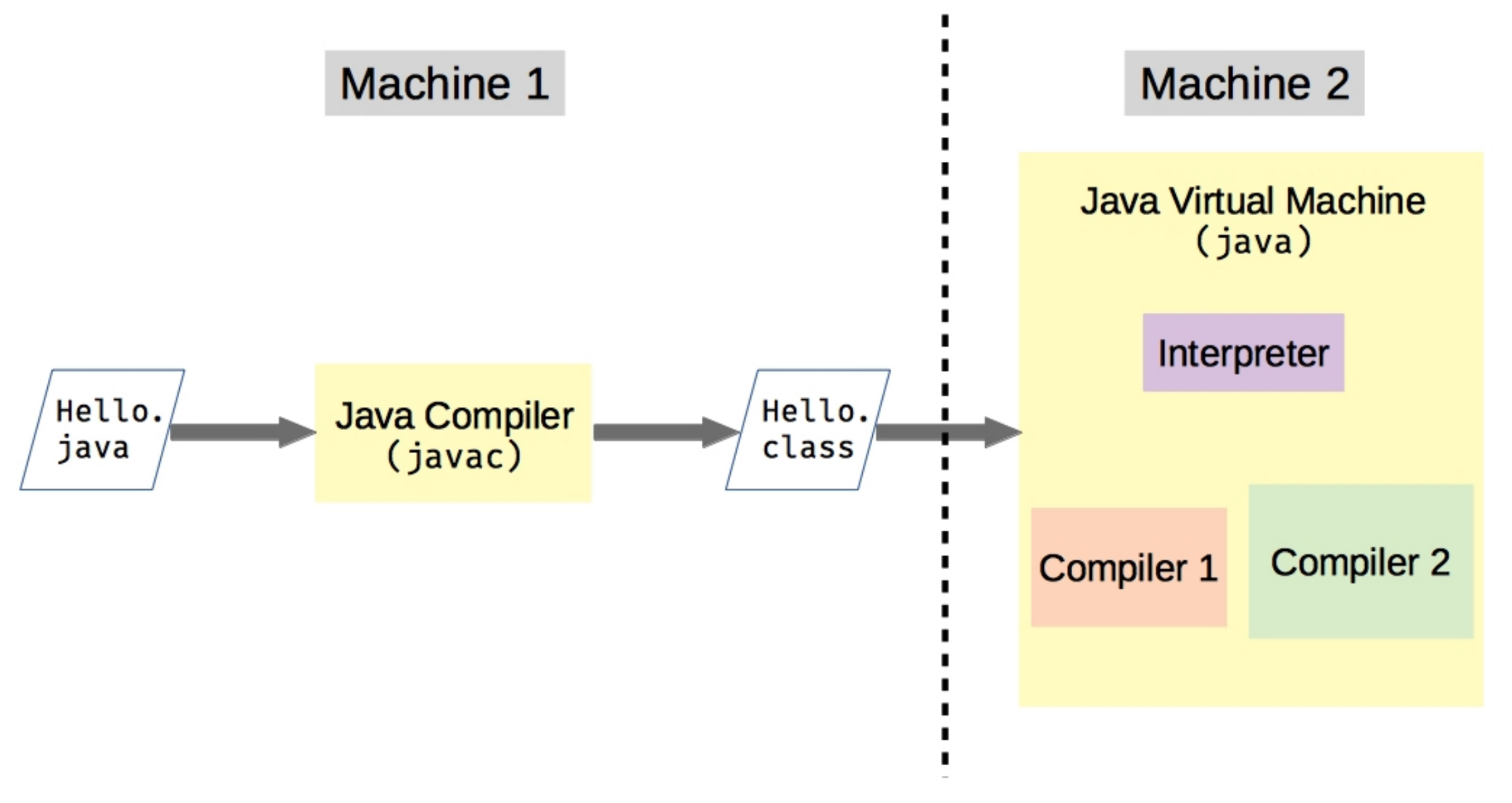
Precise, Efficient and Secure Just-In-Time Analysis of Java Programs*

Manas Thakur[†] and V. Krishna Nandivada[†]

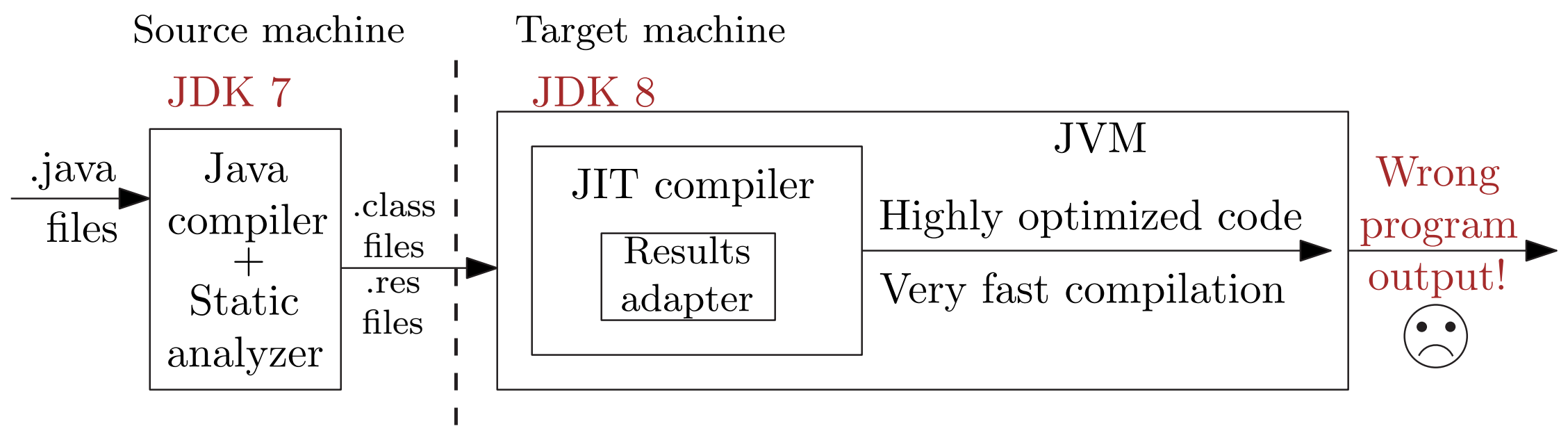
INDIAN INSTITUTE OF TECHNOLOGY MADRAS · सिद्धिर्भवति कर्मजा

Java translation model



- Java programs are compiled statically as well as just-in-time (JIT).

Static Analysis



- Must treat the library calls conservatively.

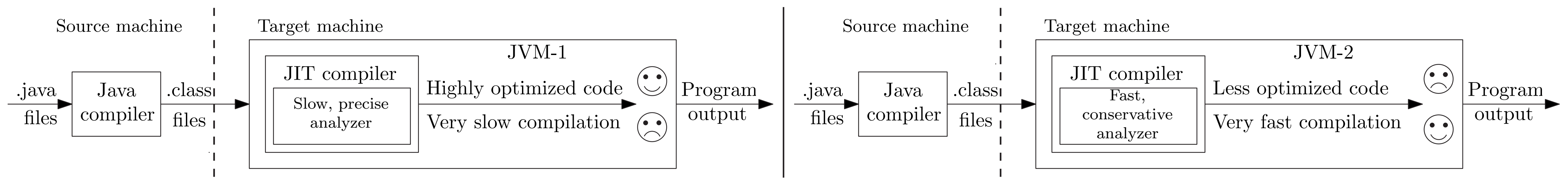
PYE: Instantiations

- Escape Analysis for Synchronization Elimination (EASE)
- Points-to Analysis for null-Check Elimination (PACE)

Comparison:

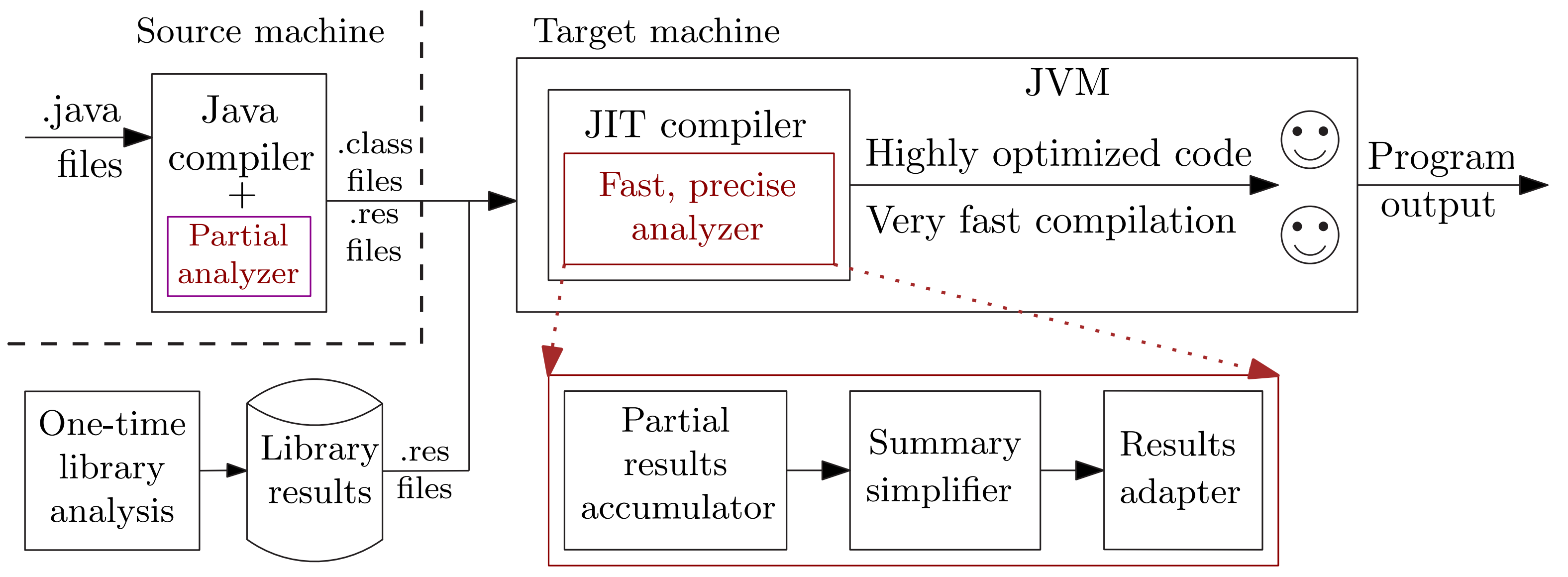
- Against the respective existing analyzers of the server compiler (C2) of the HotSpot JVM [1].

Analysis During JIT Compilation



- Analysis time during JIT compilation gets added to the execution time.
- Typical JIT compilers perform imprecise analyses and sacrifice precision.

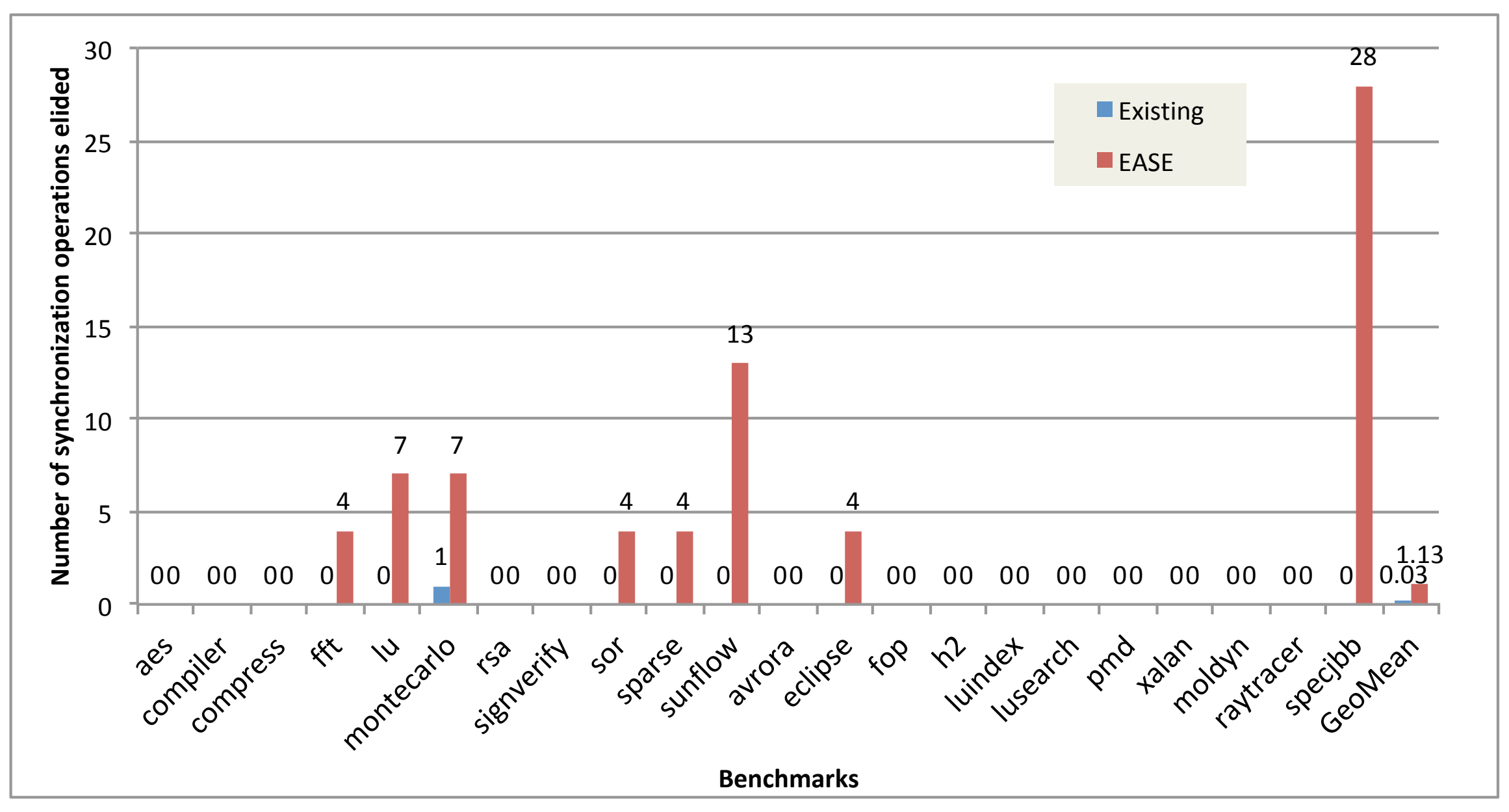
Our Solution: The PYE Framework*



1. **Partial analyzer:**
   - Analyzes Java applications independent of the libraries and generates *partial summaries*.
   - Encodes the dependence on the libraries in the form of *conditional values*.
   - Analyzes each library installation independent of the application.
   - Stores partial summaries in the form of *.res files*.
2. **Fast, precise analyzer:**
   - Reads the relevant partial summaries for the application being executed by the JVM *(partial results accumulator)*.
   - Simplifies the partial summaries by resolving the dependences between the application and the libraries *(summary simplifier)*.
   - Stores the final analysis-results in appropriate data structures to enable the relevant optimizations *(results adapter)*.

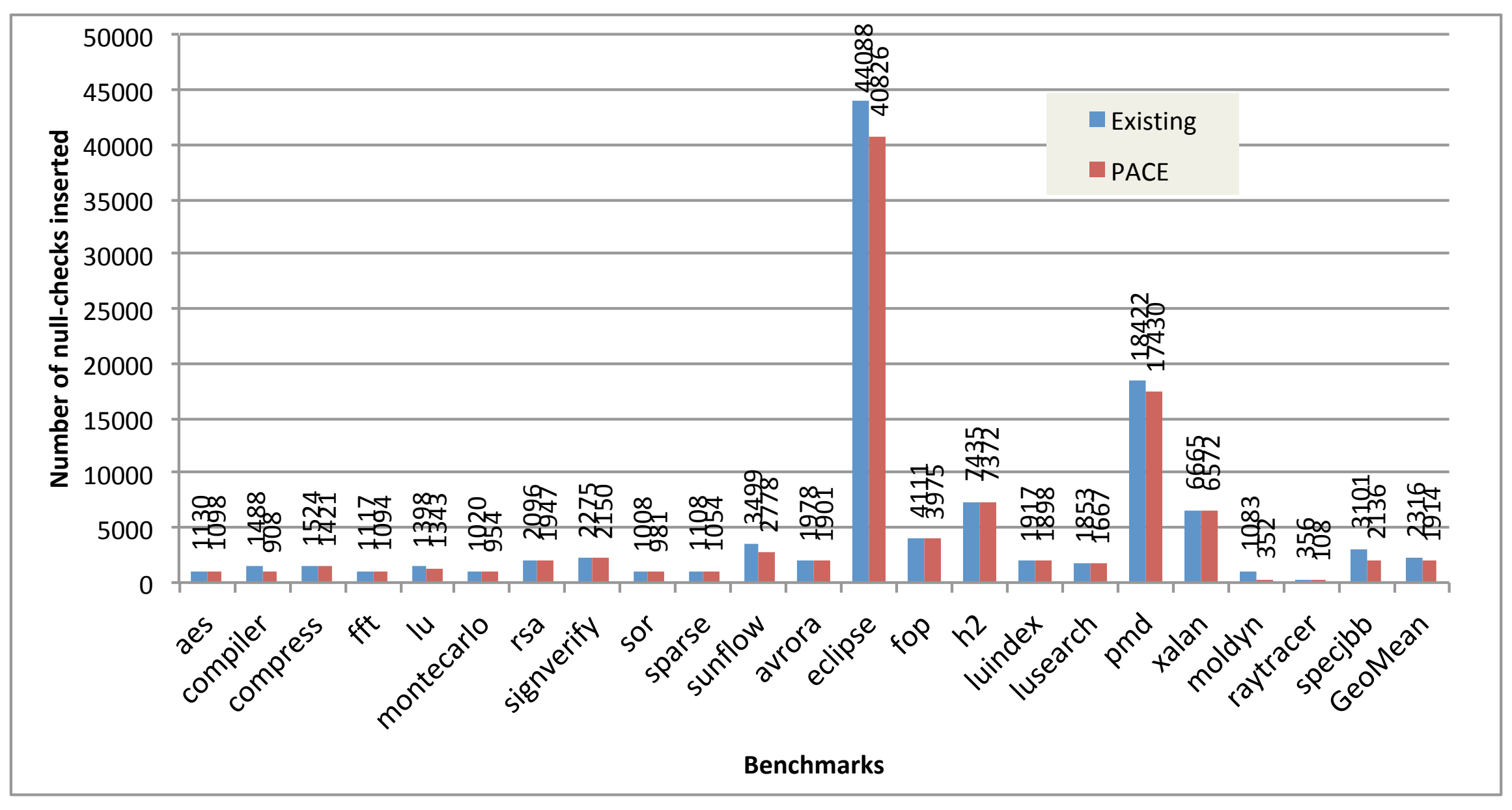
Evaluation Results

Precision: EASE



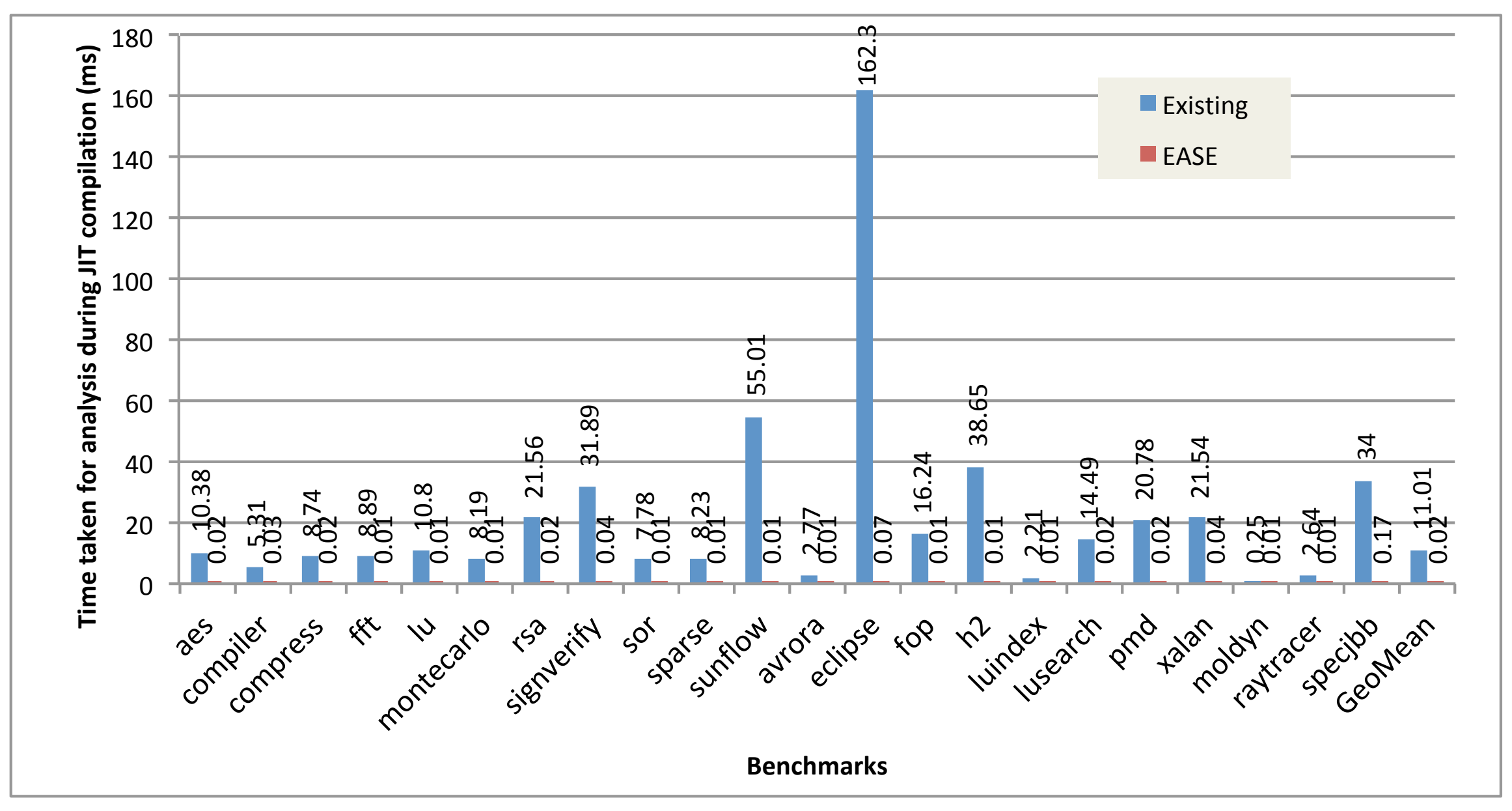
EASE elides a significantly higher number of synchronization operations.

Precision: PACE



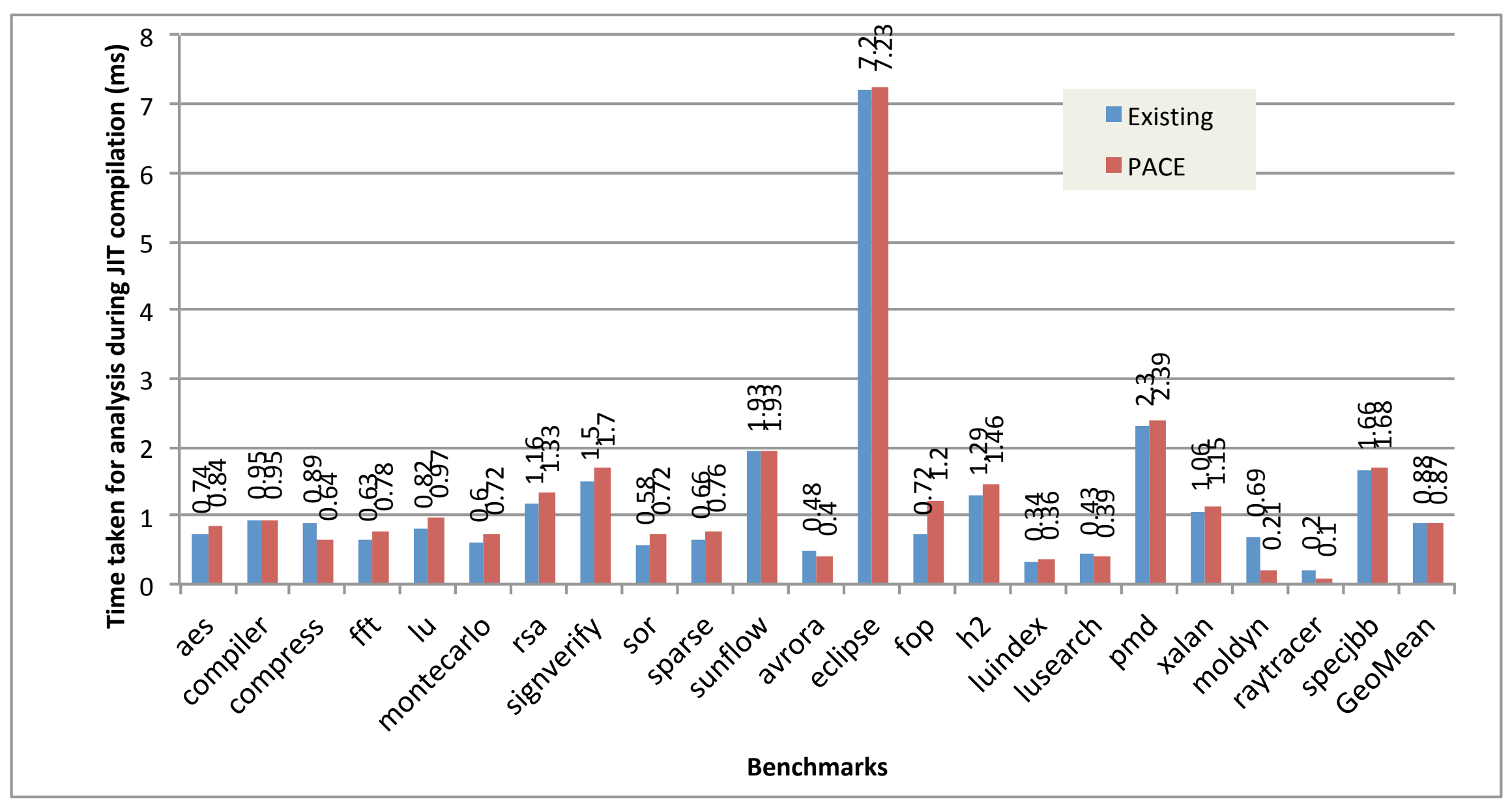
PACE inserts 23.67% lesser null-checks than the existing analyzer.

Time: EASE



EASE takes 99.78% lesser time than the existing analyzer.

Time: PACE



Times taken by both – PACE and the existing analyzer – are negligible.

Conclusion

- The proposed strategy *solves an important challenge* in modern just-in-time compilers.
- PYE effectively obtains *precise* analysis-results *efficiently* during JIT compilation.
- PACE and EASE could be *practical alternatives* for the existing analyzers of the C2 compiler.
- The techniques are *general enough* to be extended to other analyses and languages.

Future Work

- Bring PYE to production.
  Identified candidate: Eclipse OpenJ9.
- Identify more clients that could benefit using the proposed approach.
- Ensure security of the results.
- Take advantage of Java 9 modules to store and verify results in a modular manner.

References

[1] Michael Paleczny, Christopher Vick, and Cliff Click. 2001. The Java HotSpot^{TM} Server Compiler. *In Proceedings of the 2001 Symposium on JavaTM Virtual Machine Research and Technology Symposium - Volume 1 (JVM'01).*

33rd European Conference on Programming Languages (ECOOP 2019), London, UK.

* Associated talk in the International Workshop on Advances in Open Runtime Technologies and Applications (AORTA 2019), on July 18th.

[†] Author addresses: {manas, nvk}@cse.iitm.ac.in; PACE Lab, Department of CSE, IIT Madras, Chennai.

⋆ Manas Thakur and V. Krishna Nandivada. PYE: A Framework for Precise-Yet-Efficient Just-In-Time Analyses for Java Programs. *ACM Transactions on Programming Languages and Systems (TOPLAS), 41(3):16:1–16:37, July 2019.* URL: `https://doi.org/10.1145/3337794`.