Release Codee 2024.2

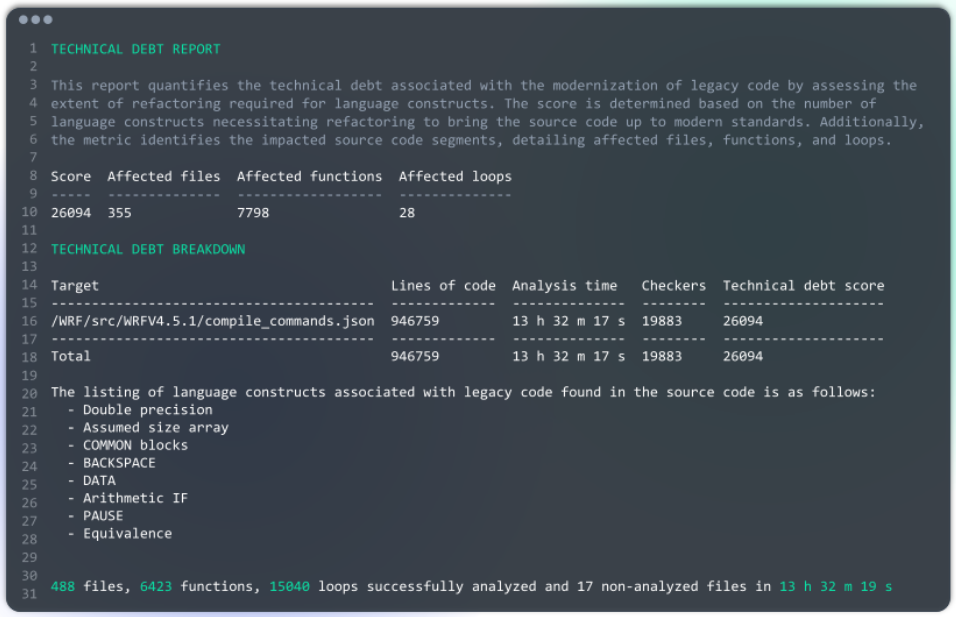
Verifysoft Technology GmbH, May 21, 2024

We are excited to introduce Codee 2024.2, the latest version of the advanced static code analyzer for performance analysis of Fortran, C, and C++ projects. The new version enables developers to produce high-quality, modern, and performant software that adheres to industry standards.

In this iteration, there is placed a strong emphasis on enhancing support for Fortran, particularly focusing on the modernization of existing codebases. Codee provides a systematic and predictable workflow that integrates seamlessly with existing software development ecosystems, including compilers and performance tools like profilers and monitoring tools.

Key Features and Improvements:

Technical Debt Report



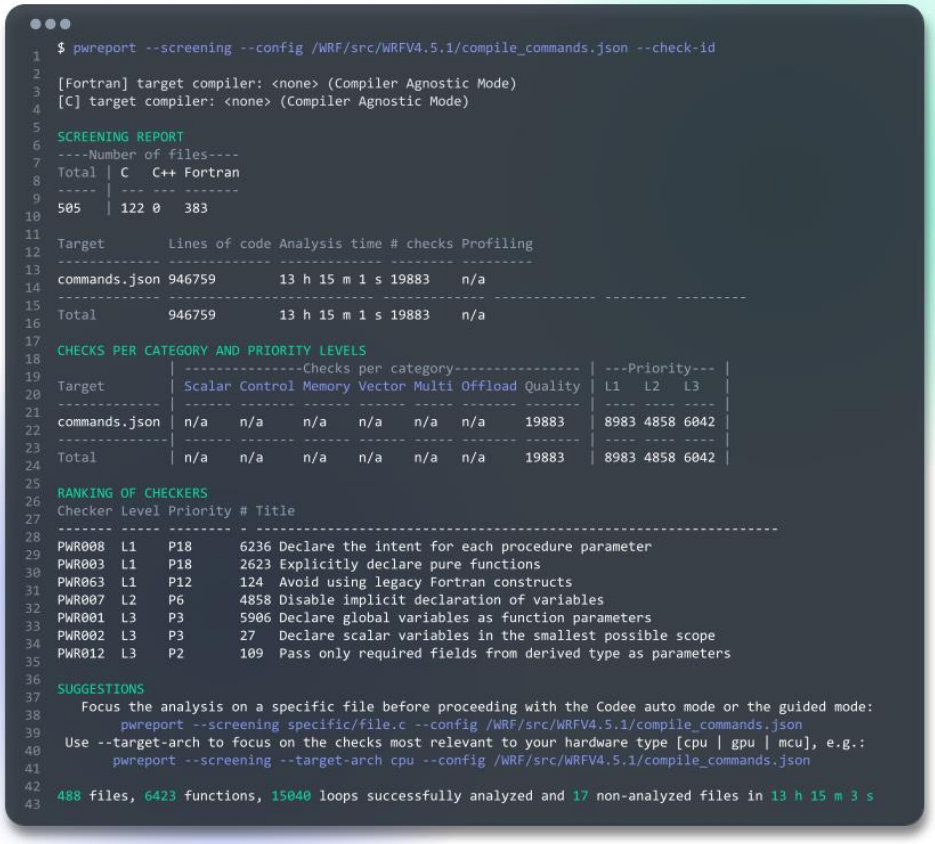
```
TECHNICAL DEBT REPORT

This report quantifies the technical debt associated with the modernization of legacy code by assessing the
extent of refactoring required for language constructs. The score is determined based on the number of
language constructs necessitating refactoring to bring the source code up to modern standards. Additionally,
the metric identifies the impacted source code segments, detailing affected files, functions, and loops.

Score  Affected files  Affected functions  Affected loops
-----  --------------  ------------------  --------------
26094  355             7798                28

TECHNICAL DEBT BREAKDOWN

Target                                  Lines of code  Analysis time   Checkers  Technical debt score
--------------------------------------  -------------  --------------  --------  --------------------
/WRF/src/WRFV4.5.1/compile_commands.json  946759         13 h 32 m 17 s  19883     26094
--------------------------------------  -------------  --------------  --------  --------------------
Total                                   946759         13 h 32 m 17 s  19883     26094

The listing of language constructs associated with legacy code found in the source code is as follows:
  - Double precision
  - Assumed size array
  - COMMON blocks
  - BACKSPACE
  - DATA
  - Arithmetic IF
  - PAUSE
  - Equivalence


488 files, 6423 functions, 15040 loops successfully analyzed and 17 non-analyzed files in 13 h 32 m 19 s
```

The new Technical Debt Report (codee technical debt) quantifies the technical debt associated with modernizing legacy code. It assesses the extent of refactoring required for language constructs and assigns a score based on the number of constructs needing modernization. This metric also identifies

affected source code segments, detailing the impacted files, functions, and loops, offering a clear roadmap for modernization efforts.

ROI Report

The ROI Report (codee roi) highlights the tangible benefits of using Codee in the development process. It demonstrates significant savings in development effort and cost efficiencies, underscoring the value that Codee brings to your organization.



```
$ pwreport --screening --config /WRF/src/WRFV4.5.1/compile_commands.json --check-id

[Fortran] target compiler: <none> (Compiler Agnostic Mode)
[C] target compiler: <none> (Compiler Agnostic Mode)

SCREENING REPORT
----Number of files----
Total | C   C++ Fortran
----- | --- --- -------
505   | 122 0   383

Target        Lines of code Analysis time # checks Profiling
------------- ------------- ------------- -------- ---------
commands.json 946759        13 h 15 m 1 s 19883    n/a
------------- ------------- ------------- -------- ---------
Total         946759        13 h 15 m 1 s 19883    n/a

CHECKS PER CATEGORY AND PRIORITY LEVELS
              | ---------------Checks per category---------------- | ---Priority--- |
Target        | Scalar Control Memory Vector Multi Offload Quality | L1   L2   L3   |
------------- | ------ ------- ------ ------ ----- ------- ------- | ---- ---- ---- |
commands.json | n/a    n/a     n/a    n/a    n/a   n/a     19883   | 8983 4858 6042 |
------------- | ------ ------- ------ ------ ----- ------- ------- | ---- ---- ---- |
Total         | n/a    n/a     n/a    n/a    n/a   n/a     19883   | 8983 4858 6042 |

RANKING OF CHECKERS
Checker Level Priority # Title
------- ----- -------- - ------------------------------------------------------------------------
PWR008  L1    P18      6236 Declare the intent for each procedure parameter
PWR003  L1    P18      2623 Explicitly declare pure functions
PWR063  L1    P12      124  Avoid using legacy Fortran constructs
PWR007  L2    P6       4858 Disable implicit declaration of variables
PWR001  L3    P3       5906 Declare global variables as function parameters
PWR002  L3    P3       27   Declare scalar variables in the smallest possible scope
PWR012  L3    P2       109  Pass only required fields from derived type as parameters

SUGGESTIONS
   Focus the analysis on a specific file before proceeding with the Codee auto mode or the guided mode:
        pwreport --screening specific/file.c --config /WRF/src/WRFV4.5.1/compile_commands.json
 Use --target-arch to focus on the checks most relevant to your hardware type [cpu | gpu | mcu], e.g.:
       pwreport --screening --target-arch cpu --config /WRF/src/WRFV4.5.1/compile_commands.json

488 files, 6423 functions, 15040 loops successfully analyzed and 17 non-analyzed files in 13 h 15 m 3 s
```

Compiler Efficiency Report

The Compiler Efficiency Report (codee compiler-efficiency) provides insights into the optimizations performed by the compiler for a given project. It evaluates the impact of fine-tuning compiler flags on overall performance, helping developers understand and enhance the efficiency of their code.

Enhanced Code Coverage

Codee 2024.2 introduces improved code coverage for both C and Fortran, with enhanced analysis capabilities for loops that were previously unanalyzable. These improvements in Codee's core technology ensure a more comprehensive analysis of your codebase.

Platform and Architecture Support

- **Fortran Analysis on Windows and MacOS:**
  Fortran analysis is now available on Windows and MacOS versions of Codee.

- **ARM "big.LITTLE" Architecture Flags**
  Support for ARM's "big. LITTLE" architecture flags have been added.

C++ and Compiler Enhancements

- **Improved C++ Member Function Support:**
  Enhanced analysis for C++ member functions, ensuring more accurate and comprehensive code assessments.

- **IAR iccarm Compiler Support:**
  Better support for projects using the IAR iccarm compiler, facilitating smoother integration and analysis.

- **Precompiled Headers Support:**
  Enhanced handling of projects utilizing precompiled headers, improving analysis accuracy and efficiency.

Screening with Ranking

The new Screening with Ranking feature helps prioritize the implementation of checkers most suited for the target environment. The rules are ranked top-down, guiding users to begin with the highest-ranked ones to maximize the performance impact of their code.

Codee 2024.2 represents a substantial advancement in static code analysis, providing developers with the tools they need to modernize, optimize, and maintain their codebases effectively. By addressing technical debt, enhancing ROI, and improving compiler efficiency, Codee continues to support the development of high-quality, performant software.

Visit our website for more information:

https://www.verifysoft.com/en_codee.html