

Release Codee 2024.2

Verifysoft Technology GmbH, den 21. Mai 2024

Wir freuen uns, Codee 2024.2 vorzustellen, die neueste Version des statischen Code-Analyse Tools zur Verbesserung der Software-Performance. Diese Version stellt einen bedeutenden Meilenstein dar, um Fortran-, C- und C++-Entwickler dabei zu unterstützen, qualitativ hochwertige, moderne und leistungsfähige Software zu erstellen, die den Industriestandards entspricht.

In dieser Version wurde insbesondere die Unterstützung für Fortran verbessert und auf die Modernisierung bestehender Codebasen konzentriert. Codee bietet einen systematischen und vorhersehbaren Workflow, der sich nahtlos in bestehende Softwareentwicklungssysteme integriert, einschließlich Compiler und Leistungstools wie Profiler und Überwachungstools.

Hauptmerkmale und Verbesserungen:

Technical Debt Report

```
1 TECHNICAL DEBT REPORT
2
3 This report quantifies the technical debt associated with the modernization of legacy code by assessing the
4 extent of refactoring required for language constructs. The score is determined based on the number of
5 language constructs necessitating refactoring to bring the source code up to modern standards. Additionally,
6 the metric identifies the impacted source code segments, detailing affected files, functions, and loops.
7
8 Score Affected files Affected functions Affected loops
9 -----
10 26094 355 7798 28
11
12 TECHNICAL DEBT BREAKDOWN
13
14 Target Lines of code Analysis time Checkers Technical debt score
15 -----
16 /WRF/src/WRFV4.5.1/compile_commands.json 946759 13 h 32 m 17 s 19883 26094
17 -----
18 Total 946759 13 h 32 m 17 s 19883 26094
19
20 The listing of language constructs associated with legacy code found in the source code is as follows:
21 - Double precision
22 - Assumed size array
23 - COMMON blocks
24 - BACKSPACE
25 - DATA
26 - Arithmetic IF
27 - PAUSE
28 - Equivalence
29
30 488 files, 6423 functions, 15040 loops successfully analyzed and 17 non-analyzed files in 13 h 32 m 19 s
31
```

Der neue Technical Debt Report (codee technical-debt) quantifiziert die technische Schuld, die mit der Modernisierung von Legacy-Code verbunden ist. Er bewertet das Ausmaß der erforderlichen Refaktorisierung von Sprach

Konstrukten und weist basierend auf der Anzahl der zu modernisierenden Konstrukten eine Punktzahl zu. Diese Metrik identifiziert auch betroffene Quellcode-Segmente und beschreibt die betroffenen Dateien, Funktionen und Schleifen, um einen klaren Fahrplan für Modernisierungsbemühungen zu bieten.

ROI-Report

Der ROI-Report (codee roi) hebt die greifbaren Vorteile der Verwendung von Codee im Entwicklungsprozess hervor. Er zeigt signifikante Einsparungen bei Entwicklungsaufwand und Kosten und unterstreicht den Wert, den Codee für Ihre Organisation bringt.

```

1  $ pwreport --screening --config /WRF/src/WRFV4.5.1/compile_commands.json --check-id
2
3  [Fortran] target compiler: <none> (Compiler Agnostic Mode)
4  [C] target compiler: <none> (Compiler Agnostic Mode)
5
6  SCREENING REPORT
7  ---Number of files---
8  Total | C   C++ Fortran
9  -----|-----
10 505 | 122 0 383
11
12 Target      Lines of code Analysis time # checks Profiling
13 -----|-----
14 commands.json 946759          13 h 15 m 1 s 19883 n/a
15 -----|-----
16 Total        946759          13 h 15 m 1 s 19883 n/a
17
18 CHECKS PER CATEGORY AND PRIORITY LEVELS
19 Target      |-----Checks per category-----| ---Priority--- |
20 | Scalar Control Memory Vector Multi Offload Quality | L1  L2  L3  |
21 -----|-----|-----|-----|-----|-----|-----|-----|
22 commands.json | n/a  n/a  n/a  n/a  n/a  n/a  n/a  | 19883 | 8983 4858 6042 |
23 -----|-----|-----|-----|-----|-----|-----|-----|
24 Total        | n/a  n/a  n/a  n/a  n/a  n/a  n/a  | 19883 | 8983 4858 6042 |
25
26 RANKING OF CHECKERS
27 Checker Level Priority # Title
28 -----|-----|-----|-----|-----|-----|-----|-----|
29 PWR008 L1 P18 6236 Declare the intent for each procedure parameter
30 PWR003 L1 P18 2623 Explicitly declare pure functions
31 PWR063 L1 P12 124  Avoid using legacy Fortran constructs
32 PWR007 L2 P6 4858 Disable implicit declaration of variables
33 PWR001 L3 P3 5906 Declare global variables as function parameters
34 PWR002 L3 P3 27  Declare scalar variables in the smallest possible scope
35 PWR012 L3 P2 109  Pass only required fields from derived type as parameters
36
37 SUGGESTIONS
38 Focus the analysis on a specific file before proceeding with the Codee auto mode or the guided mode:
39 pwreport --screening specific/file.c --config /WRF/src/WRFV4.5.1/compile_commands.json
40 Use --target-arch to focus on the checks most relevant to your hardware type [cpu | gpu | mcu], e.g.:
41 pwreport --screening --target-arch cpu --config /WRF/src/WRFV4.5.1/compile_commands.json
42
43 488 files, 6423 functions, 15040 loops successfully analyzed and 17 non-analyzed files in 13 h 15 m 3 s

```

Compiler Efficiency Report

Der Compiler Efficiency Report (codee compiler-efficiency) liefert Einblicke in die vom Compiler für ein bestimmtes Projekt durchgeführten Optimierungen. Er bewertet die Auswirkungen der Feinabstimmung von Compiler-Flags auf die Gesamtleistung und hilft Entwicklern, die Effizienz ihres Codes zu verstehen und zu verbessern.

Verbesserte Code-Abdeckung

Codee 2024.2 bietet eine verbesserte Code-Abdeckung für sowohl C- als auch Fortran-Code, mit erweiterten Analysemöglichkeiten für Schleifen, die zuvor nicht analysierbar waren. Diese Verbesserungen in der Kerntechnologie von Codee sorgen für eine umfassendere Analyse Ihrer Codebasis.

Plattform- und Architekturunterstützung

- **Fortran-Analyse unter Windows und MacOS:**
Fortran-Analyse ist jetzt in den Windows- und MacOS-Versionen von Codee verfügbar.
- **ARM „big.LITTLE“-Architektur-Flags:**
Die Unterstützung für ARM's „big.LITTLE“-Architektur-Flags wurde hinzugefügt.

Verbesserungen für C++ und Compiler

- **Verbesserte Unterstützung für C++-Member Functions:**
Erweiterte Analyse für C++-Mitgliedsfunktionen, die genauere und umfassendere Codebewertungen ermöglicht.
- **Unterstützung des IAR iccarm-Compilers:**
Bessere Unterstützung für Projekte, die den IAR iccarm-Compiler verwenden, was eine reibungslosere Integration und Analyse ermöglicht.
- **Unterstützung für vorcompilierte Header:**
Verbesserte Handhabung von Projekten, die vorcompilierte Header verwenden, wodurch die Analysegenauigkeit und Effizienz verbessert werden.

Screening mit Ranking

Die neue Funktion Screening mit Ranking hilft, die Implementierung von Checkern zu priorisieren, die am besten für die Zielumgebung geeignet sind. Die Regeln werden von oben nach unten gerankt und leiten die Benutzer an, mit den höchstrangigen Regeln zu beginnen, um die Leistungswirkung ihres Codes zu maximieren.

Codee 2024.2 stellt einen erheblichen Fortschritt in der statischen Code-Analyse dar und bietet Entwicklern die Werkzeuge, die sie benötigen, um ihre Codebasen effektiv zu modernisieren, zu optimieren und zu warten. Durch die Reduzierung technischer Schulden, die Verbesserung des ROI und die Steigerung der Compiler-Effizienz unterstützt Codee weiterhin die Entwicklung von qualitativ hochwertiger, leistungsstarker Software.

Besuchen Sie unsere Website für weitere Informationen:

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

