codee

Codee Training: Write Accelerated Code at Expert Level

Codee: Automated Analysis of Large-Scale Fortran/C/C++ Codes

NERSC Codee Training Series

September 5-6, 2024

Schedule

Day 1 (Thursday 5th, 9:00 - 12:30 PDT)

Codee: Automated Code Inspection for Modernization and Optimization

- Lecture:
  - *Codee's command-line tool*
  - *Open Catalog of Best Practices for Fortran/C/C++ Modernization and Optimization for CPU and GPU*
- Demo using Fortran:
  - *HIMENO modernization*
  - *HIMENO optimization through GPU parallelism*
- Demo using C/C++:
  - *MATMUL optimization through CPU parallelism*
- Hands-on: PI, MATMUL, COULOMB, HIMENO

Day 2 (Friday 6th, 9:00 - 12:30 PDT)

Codee: Automated Analysis of Large-Scale Fortran/C/C++ Codes

- Lecture:
  - *Codee's command-line tool using compilation databases*
  - *Automated testing of large codes using Codee on Perlmutter*
  - *Use case: Optimizing the Weather Research and Forecasting Model with OpenMP Offload and Codee*
- Demo using Fortran:
  - *Putting it all together with HYCOM*
- Demo using C/C++:
  - *Putting it all together with MBedTLS*
- Hands-on: HYCOM, NUCCOR, ATMUX, LULESHmk, MBedTLS
- Bring your own applications!

Options for CI/CD pipelines

Common options:

`--csv`
Output results in CSV format

`--json`
Output results in JSON format

`--html`[1]
Output results in HTML format

`--accept-eula`
Confirm the acceptance of the EULA

Environment variables:

`CODEE_LICENSE_PATH`
Full path (including the file name) to the Codee license file

Example:

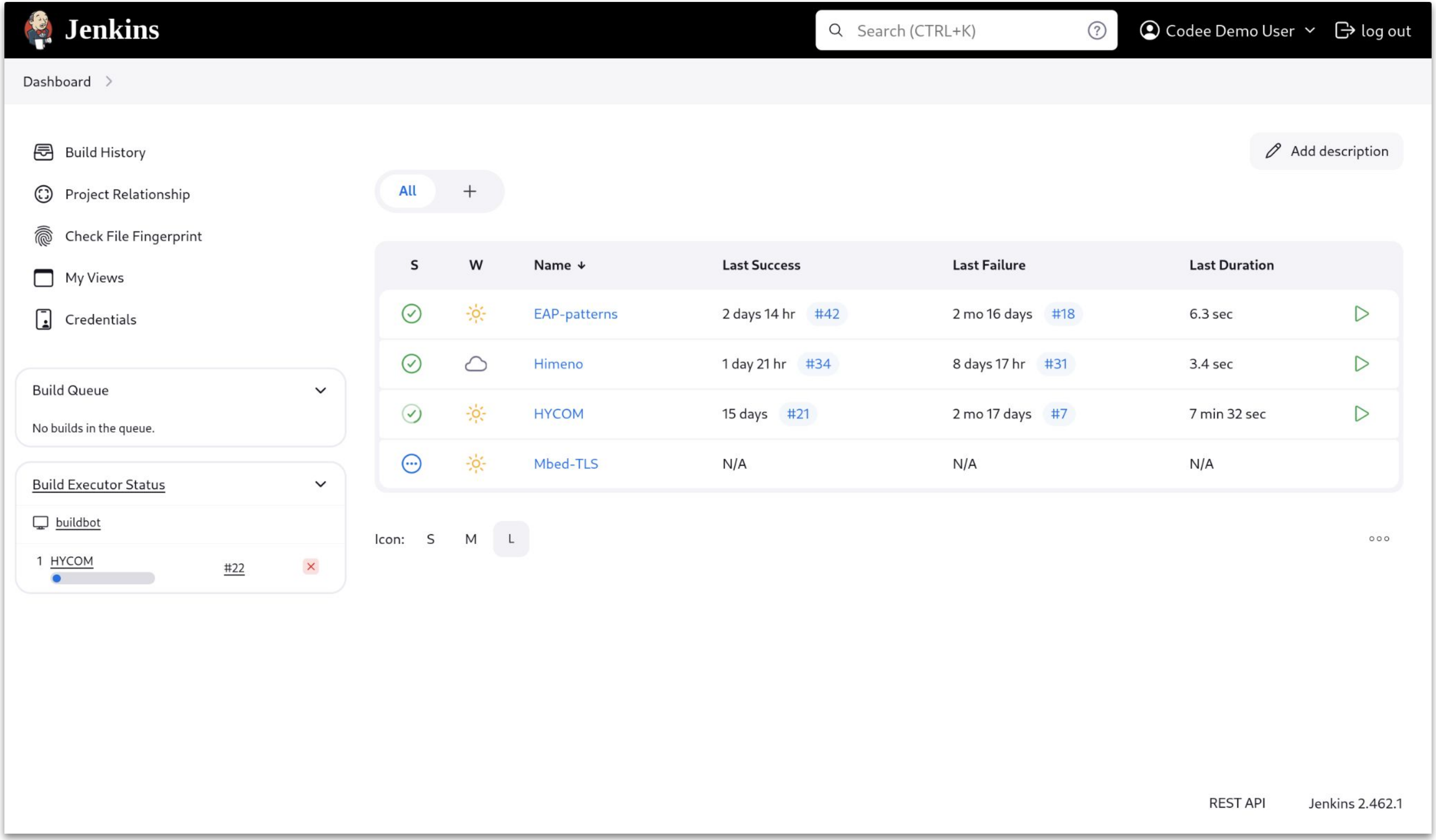
```
$ codee checks -p compile_commands.json --json > checks-report.json
```

codee

[1] Use "`codee report`" on Codee 2024.3.0. The flag will be available at a later release.

Demo

Codee Invocation in a CI/CD pipeline (I)



Jenkins

Search (CTRL+K)

Codee Demo User

log out

Dashboard >

- Build History
- Project Relationship
- Check File Fingerprint
- My Views
- Credentials

Build Queue

No builds in the queue.

Build Executor Status

buildbot

1 HYCOM #22 ✕

Add description

All +

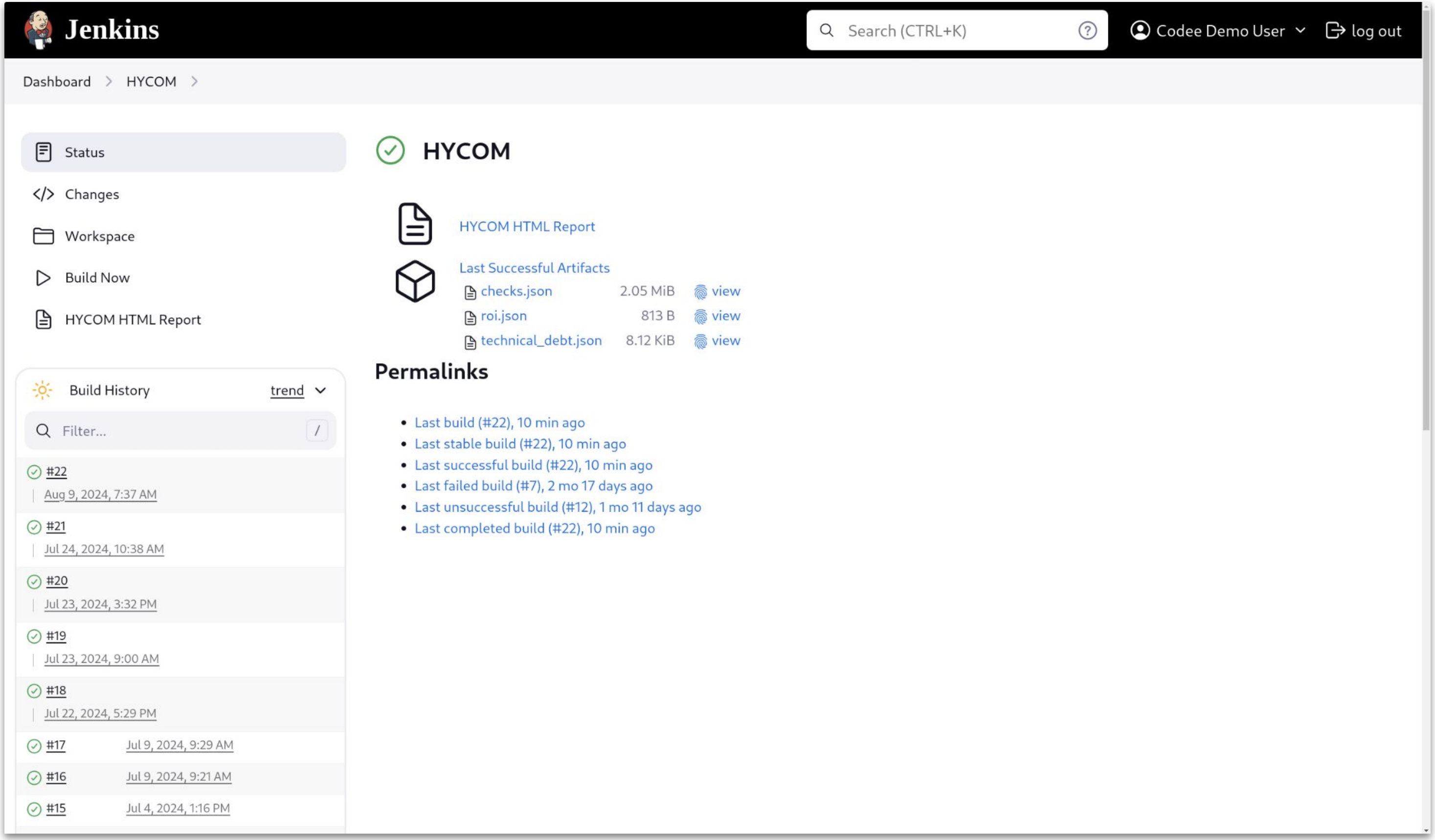
S	W	Name ↓	Last Success	Last Failure	Last Duration	
✓	☀	EAP-patterns	2 days 14 hr #42	2 mo 16 days #18	6.3 sec	▷
✓	☁	Himeno	1 day 21 hr #34	8 days 17 hr #31	3.4 sec	▷
✓	☀	HYCOM	15 days #21	2 mo 17 days #7	7 min 32 sec	▷
⋯	☀	Mbed-TLS	N/A	N/A	N/A	

Icon: S M L

REST API Jenkins 2.462.1

Demo

Codee Invocation in a CI/CD pipeline (II)



HYCOM

HYCOM HTML Report

Last Successful Artifacts

checks.json	2.05 MiB	view
roi.json	813 B	view
technical_debt.json	8.12 KiB	view

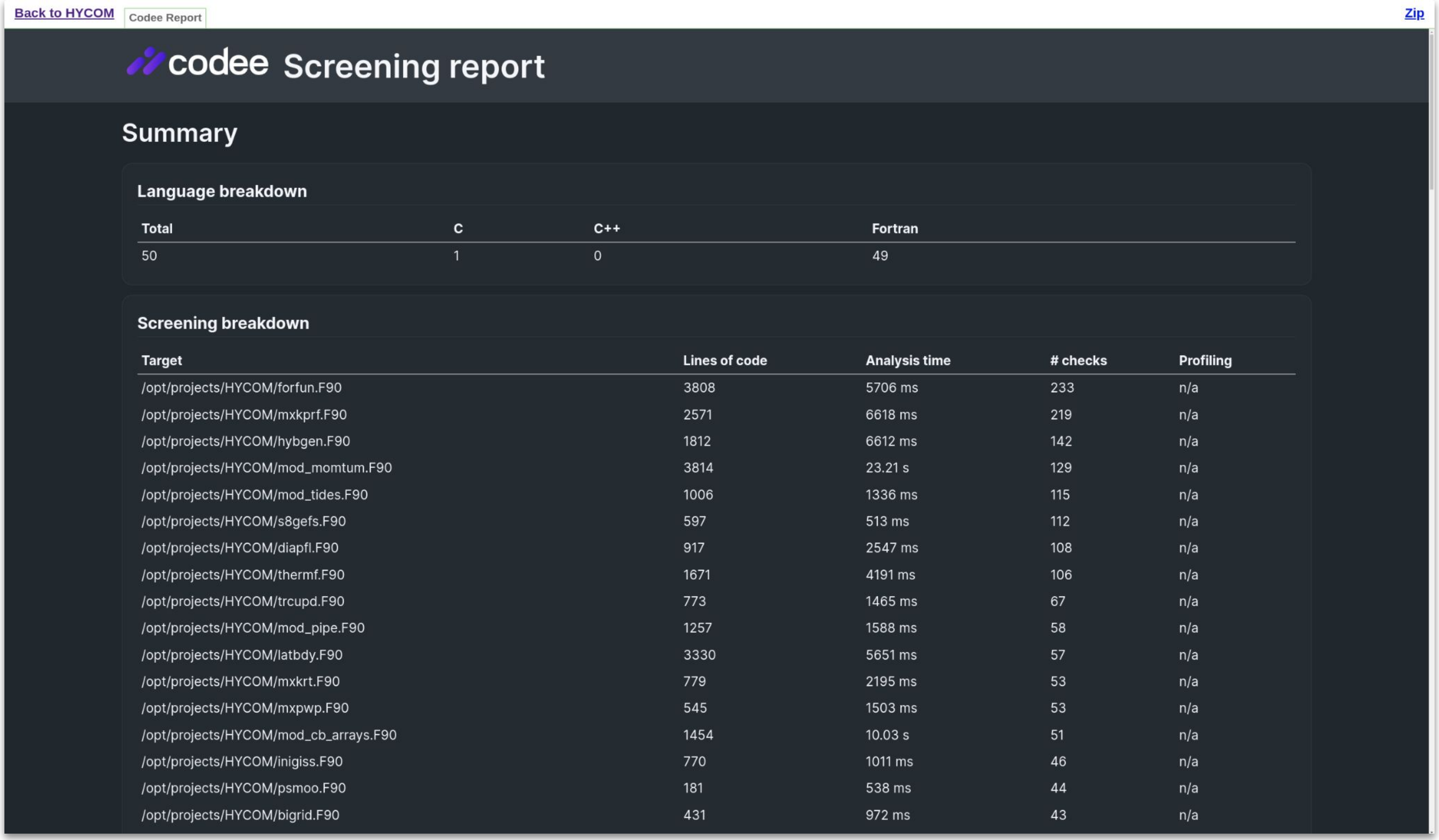
Permalinks

- Last build (#22), 10 min ago
- Last stable build (#22), 10 min ago
- Last successful build (#22), 10 min ago
- Last failed build (#7), 2 mo 17 days ago
- Last unsuccessful build (#12), 1 mo 11 days ago
- Last completed build (#22), 10 min ago

#22	Aug 9, 2024, 7:37 AM
#21	Jul 24, 2024, 10:38 AM
#20	Jul 23, 2024, 3:32 PM
#19	Jul 23, 2024, 9:00 AM
#18	Jul 22, 2024, 5:29 PM
#17	Jul 9, 2024, 9:29 AM
#16	Jul 9, 2024, 9:21 AM
#15	Jul 4, 2024, 1:16 PM

Demo

Codee Invocation in a CI/CD pipeline (III)



codee Screening report

Summary

Language breakdown

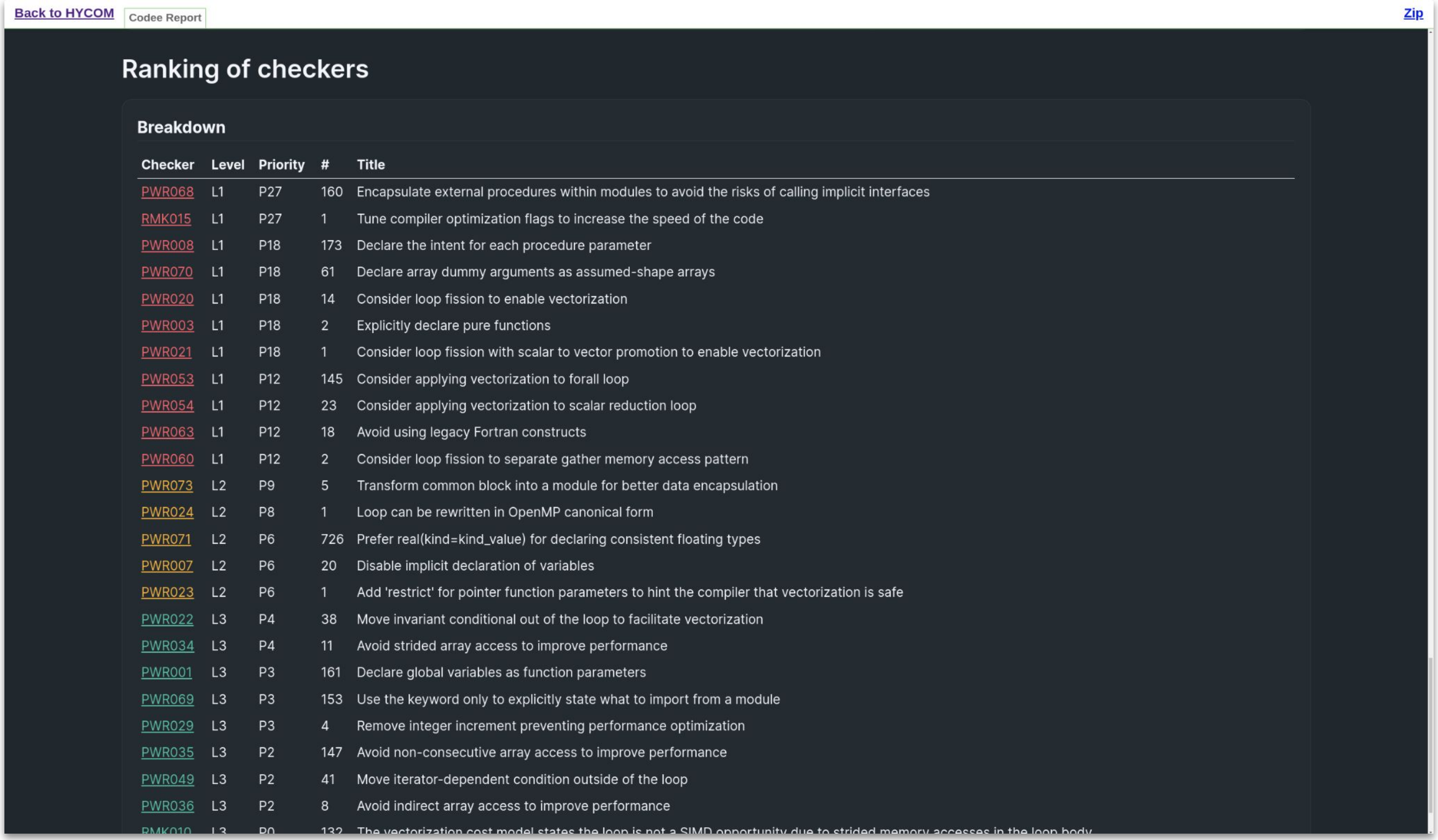
Total	C	C++	Fortran
50	1	0	49

Screening breakdown

Target	Lines of code	Analysis time	# checks	Profiling
/opt/projects/HYCOM/forfun.F90	3808	5706 ms	233	n/a
/opt/projects/HYCOM/mxkprf.F90	2571	6618 ms	219	n/a
/opt/projects/HYCOM/hybgen.F90	1812	6612 ms	142	n/a
/opt/projects/HYCOM/mod_momtum.F90	3814	23.21 s	129	n/a
/opt/projects/HYCOM/mod_tides.F90	1006	1336 ms	115	n/a
/opt/projects/HYCOM/s8gefs.F90	597	513 ms	112	n/a
/opt/projects/HYCOM/diapfl.F90	917	2547 ms	108	n/a
/opt/projects/HYCOM/thermf.F90	1671	4191 ms	106	n/a
/opt/projects/HYCOM/trcupd.F90	773	1465 ms	67	n/a
/opt/projects/HYCOM/mod_pipe.F90	1257	1588 ms	58	n/a
/opt/projects/HYCOM/latbdy.F90	3330	5651 ms	57	n/a
/opt/projects/HYCOM/mxkrt.F90	779	2195 ms	53	n/a
/opt/projects/HYCOM/mxpwp.F90	545	1503 ms	53	n/a
/opt/projects/HYCOM/mod_cb_arrays.F90	1454	10.03 s	51	n/a
/opt/projects/HYCOM/inigiss.F90	770	1011 ms	46	n/a
/opt/projects/HYCOM/psmoo.F90	181	538 ms	44	n/a
/opt/projects/HYCOM/bigrid.F90	431	972 ms	43	n/a

Demo

Codee Invocation in a CI/CD pipeline (IV)



Ranking of checkers

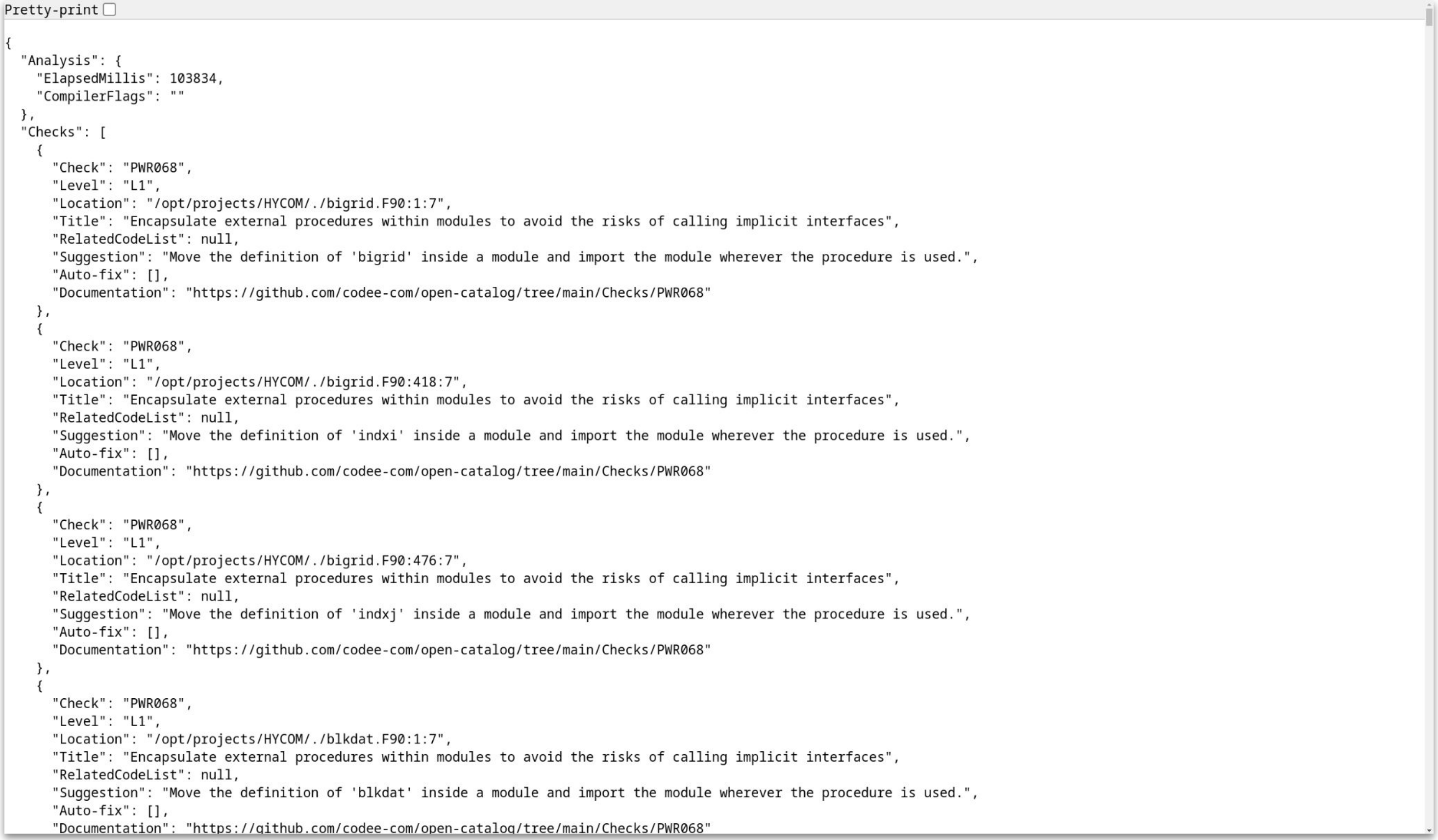
Breakdown

Checker	Level	Priority	#	Title
PWR068	L1	P27	160	Encapsulate external procedures within modules to avoid the risks of calling implicit interfaces
RMK015	L1	P27	1	Tune compiler optimization flags to increase the speed of the code
PWR008	L1	P18	173	Declare the intent for each procedure parameter
PWR070	L1	P18	61	Declare array dummy arguments as assumed-shape arrays
PWR020	L1	P18	14	Consider loop fission to enable vectorization
PWR003	L1	P18	2	Explicitly declare pure functions
PWR021	L1	P18	1	Consider loop fission with scalar to vector promotion to enable vectorization
PWR053	L1	P12	145	Consider applying vectorization to forall loop
PWR054	L1	P12	23	Consider applying vectorization to scalar reduction loop
PWR063	L1	P12	18	Avoid using legacy Fortran constructs
PWR060	L1	P12	2	Consider loop fission to separate gather memory access pattern
PWR073	L2	P9	5	Transform common block into a module for better data encapsulation
PWR024	L2	P8	1	Loop can be rewritten in OpenMP canonical form
PWR071	L2	P6	726	Prefer real(kind=kind_value) for declaring consistent floating types
PWR007	L2	P6	20	Disable implicit declaration of variables
PWR023	L2	P6	1	Add 'restrict' for pointer function parameters to hint the compiler that vectorization is safe
PWR022	L3	P4	38	Move invariant conditional out of the loop to facilitate vectorization
PWR034	L3	P4	11	Avoid strided array access to improve performance
PWR001	L3	P3	161	Declare global variables as function parameters
PWR069	L3	P3	153	Use the keyword only to explicitly state what to import from a module
PWR029	L3	P3	4	Remove integer increment preventing performance optimization
PWR035	L3	P2	147	Avoid non-consecutive array access to improve performance
PWR049	L3	P2	41	Move iterator-dependent condition outside of the loop
PWR036	L3	P2	8	Avoid indirect array access to improve performance
RMK010	L3	P0	132	The vectorization cost model states the loop is not a SIMD opportunity due to strided memory accesses in the loop body

codee

Demo

Codee Invocation in a CI/CD pipeline (V)

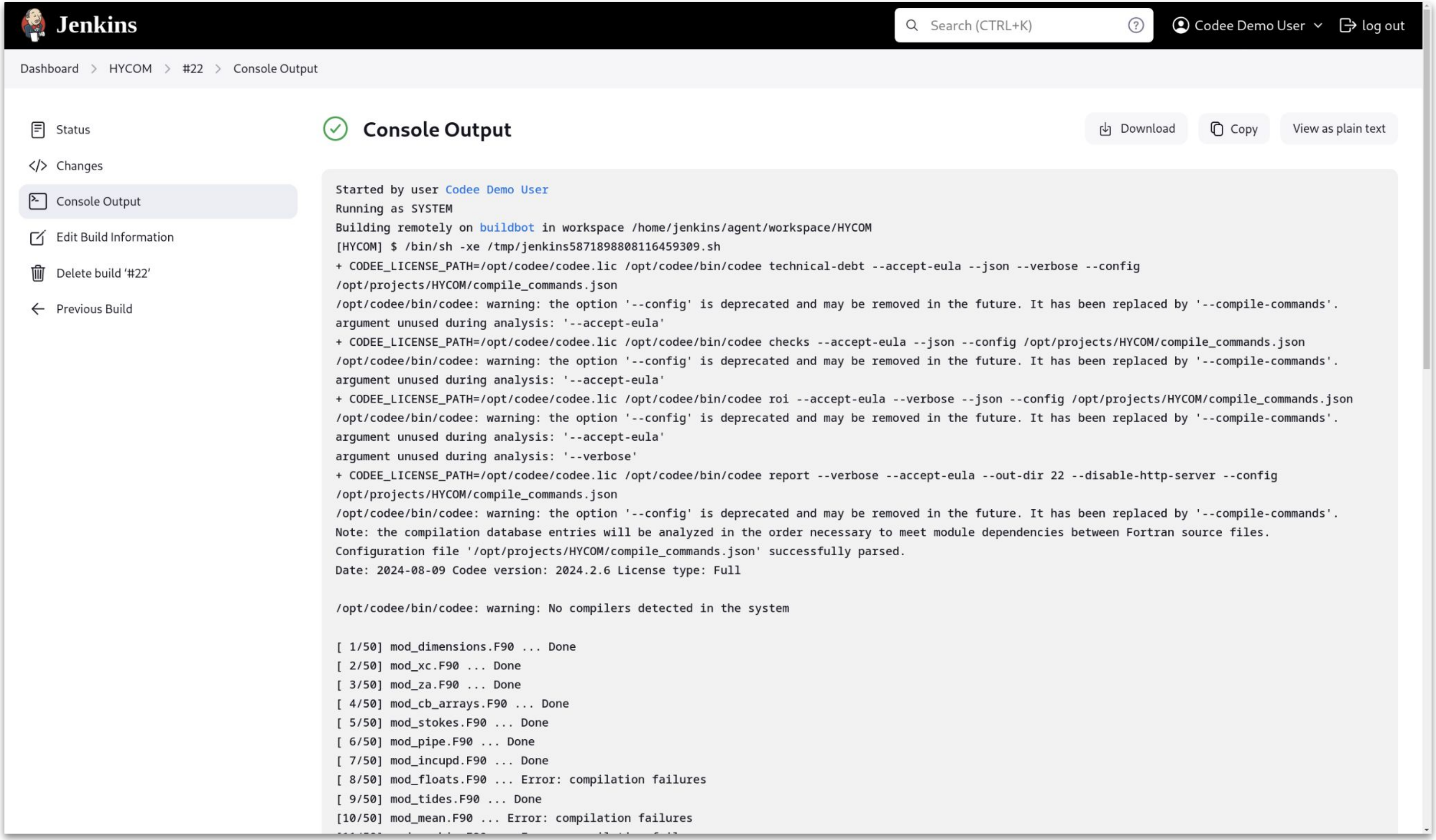


Pretty-print ☐

```
{
  "Analysis": {
    "ElapsedMillis": 103834,
    "CompilerFlags": ""
  },
  "Checks": [
    {
      "Check": "PWR068",
      "Level": "L1",
      "Location": "/opt/projects/HYCOM/./bigrid.F90:1:7",
      "Title": "Encapsulate external procedures within modules to avoid the risks of calling implicit interfaces",
      "RelatedCodeList": null,
      "Suggestion": "Move the definition of 'bigrid' inside a module and import the module wherever the procedure is used.",
      "Auto-fix": [],
      "Documentation": "https://github.com/codee-com/open-catalog/tree/main/Checks/PWR068"
    },
    {
      "Check": "PWR068",
      "Level": "L1",
      "Location": "/opt/projects/HYCOM/./bigrid.F90:418:7",
      "Title": "Encapsulate external procedures within modules to avoid the risks of calling implicit interfaces",
      "RelatedCodeList": null,
      "Suggestion": "Move the definition of 'indxi' inside a module and import the module wherever the procedure is used.",
      "Auto-fix": [],
      "Documentation": "https://github.com/codee-com/open-catalog/tree/main/Checks/PWR068"
    },
    {
      "Check": "PWR068",
      "Level": "L1",
      "Location": "/opt/projects/HYCOM/./bigrid.F90:476:7",
      "Title": "Encapsulate external procedures within modules to avoid the risks of calling implicit interfaces",
      "RelatedCodeList": null,
      "Suggestion": "Move the definition of 'indxj' inside a module and import the module wherever the procedure is used.",
      "Auto-fix": [],
      "Documentation": "https://github.com/codee-com/open-catalog/tree/main/Checks/PWR068"
    },
    {
      "Check": "PWR068",
      "Level": "L1",
      "Location": "/opt/projects/HYCOM/./blkdat.F90:1:7",
      "Title": "Encapsulate external procedures within modules to avoid the risks of calling implicit interfaces",
      "RelatedCodeList": null,
      "Suggestion": "Move the definition of 'blkdat' inside a module and import the module wherever the procedure is used.",
      "Auto-fix": [],
      "Documentation": "https://github.com/codee-com/open-catalog/tree/main/Checks/PWR068"
```

Demo

Codee Invocation in a CI/CD pipeline (VI)



```
Started by user Codee Demo User
Running as SYSTEM
Building remotely on buildbot in workspace /home/jenkins/agent/workspace/HYCOM
[HYCOM] $ /bin/sh -xe /tmp/jenkins5871898808116459309.sh
+ CODEE_LICENSE_PATH=/opt/codee/codee.lic /opt/codee/bin/codee technical-debt --accept-eula --json --verbose --config
/opt/projects/HYCOM/compile_commands.json
/opt/codee/bin/codee: warning: the option '--config' is deprecated and may be removed in the future. It has been replaced by '--compile-commands'.
argument unused during analysis: '--accept-eula'
+ CODEE_LICENSE_PATH=/opt/codee/codee.lic /opt/codee/bin/codee checks --accept-eula --json --config /opt/projects/HYCOM/compile_commands.json
/opt/codee/bin/codee: warning: the option '--config' is deprecated and may be removed in the future. It has been replaced by '--compile-commands'.
argument unused during analysis: '--accept-eula'
+ CODEE_LICENSE_PATH=/opt/codee/codee.lic /opt/codee/bin/codee roi --accept-eula --verbose --json --config /opt/projects/HYCOM/compile_commands.json
/opt/codee/bin/codee: warning: the option '--config' is deprecated and may be removed in the future. It has been replaced by '--compile-commands'.
argument unused during analysis: '--accept-eula'
argument unused during analysis: '--verbose'
+ CODEE_LICENSE_PATH=/opt/codee/codee.lic /opt/codee/bin/codee report --verbose --accept-eula --out-dir 22 --disable-http-server --config
/opt/projects/HYCOM/compile_commands.json
/opt/codee/bin/codee: warning: the option '--config' is deprecated and may be removed in the future. It has been replaced by '--compile-commands'.
Note: the compilation database entries will be analyzed in the order necessary to meet module dependencies between Fortran source files.
Configuration file '/opt/projects/HYCOM/compile_commands.json' successfully parsed.
Date: 2024-08-09 Codee version: 2024.2.6 License type: Full

/opt/codee/bin/codee: warning: No compilers detected in the system

[ 1/50] mod_dimensions.F90 ... Done
[ 2/50] mod_xc.F90 ... Done
[ 3/50] mod_za.F90 ... Done
[ 4/50] mod_cb_arrays.F90 ... Done
[ 5/50] mod_stokes.F90 ... Done
[ 6/50] mod_pipe.F90 ... Done
[ 7/50] mod_incupd.F90 ... Done
[ 8/50] mod_floats.F90 ... Error: compilation failures
[ 9/50] mod_tides.F90 ... Done
[10/50] mod_mean.F90 ... Error: compilation failures
```

codee

Codee for Automated Testing on Perlmutter

- Supercomputers provide advanced programming environments, usually including SLURM.
- Use **scrontab** as a simpler alternative to a CI/CD pipeline:
  - Allows to **schedule periodic jobs**.
  - Obtain **automatic Codee reports of any code** of interest.

Codee with Scrontab: Codee Analysis Script

```
$ cat himeno-checks.sh
#!/bin/bash

module load codee/2024.3.0

# Go to the target project
cd ~/codee-demos/Fortran/Himeno

# Analysis to run
codee checks --json --accept-eula -- gfortran himeno.f90
```

Codee with Scrontab: Scrontab Configuration

```
$ scrontab -e
#SCRON -q cron
#SCRON -C cron

#SCRON -A ntrain6
#SCRON -t 00:05:00

#SCRON -o output-%j.out
#SCRON --open-mode=append

# Run every day on Mon-Fri
# Store the result in a separate file with timestamp
# <minute> <hour> <day of month> <month> <day of week> <script>
       0       0              *       *            1-5 ~/himeno-checks.sh > ~/himeno-checks-$(date +%Y%m%d%H%M).out
```

Codee with Scrontab: Notify by Email

```
$ scrontab -e
#SCRON -q cron
#SCRON -C cron

#SCRON -A ntrain6
#SCRON -t 00:05:00

#SCRON -o output-%j.out
#SCRON --open-mode=append

#SCRON --mail-user=<mail>
#SCRON --mail-type=end

# Run every day on Mon-Fri
# Store the result in a separate file with timestamp
# <minute> <hour> <day of month> <month> <day of week> <script>
       0       0              *       *            1-5 ~/himeno-checks.sh > ~/himeno-checks-$(date +%Y%m%d%H%M).out
```

Main Takeaways

- **Codee is a production-ready tool** for automated testing of Fortran/C/C++ code, designed for **integration with CI/CD**, IDEs and programming environments.

- **Automate** the execution of **Codee** to **track** your **code's health** over time:
    - **Inside CI/CD frameworks** (e.g., Jenkins, GitLab).
    - **In SLURM infrastructures:** `scrontab`
    - **Inside containers** (e.g., Docker): `CODEE_LICENSE_PATH=<path> codee --accept-eula`

- **Post-process Codee's output** to extract metrics using **data-exchange formats** (e.g., JSON, CSV):
    - `codee --json`

Hands-on Labs on Perlmutter @NERSC

Step-by-step guides available at docs.codee.com:

- HYCOM modernization (Fortran)
- NUCCOR offloading to GPU at Perlmutter (Fortran)
- ATMUX parallelization on CPU at Perlmutter (C/C++)
- LULESHmk offloading to GPU at Perlmutter (C/C++)
- MBedTLS optimization through vectorization (C/C++)
- Bring your own applications!

codee

Automated Code Inspection for
Modernization and Optimization

www.codee.com

info@codee.com

Subscribe: codee.com/newsletter/

Spain

codee_com

in /codee-com/