

Instrumentation (Python and PerfStubs in general)

<https://tinyurl.com/stw-pstubs>

<TL/DR> (outbrief)

PerfStubs is a macro-based effort to add simple plugin interface to any application/library (C/C++/Fortran) to integrate a tool at runtime (similar to OMPT, Kokkos), integrated into ADIOS2, PETSc, Ginkgo, CAMTIMERS, ZeroSum

Prototyped effort to add Python support for both legacy and 3.12+ profiling/tracing for any/all tools to use - function entry/exit callbacks, selective measurement support, thread safety, Spark/TensorFlow/PyTorch support

PerfStubs team (UO) happy to work with implementers and address any problems that arise

<https://github.com/UO-OACISS/perfstubs>

PerfStubs

Copyright (c) 2019-2022 University of Oregon Distributed under the BSD Software License (See accompanying file LICENSE.txt)

Profiling API for adding external tool instrumentation support to any project.

This was motivated by the need to quickly add instrumentation to the [ADIOS2](#) library without adding a build dependency, or tying to a specific measurement tool.

The initial prototype implementation was tied to TAU, but evolved to this more generic version, which was extracted as a separate repository for testing and demonstration purposes.

For a more extensive readme, see [perfstubs_api/README.md](#).

<https://github.com/UO-OACISS/perfstubs/tree/python-3.12>

python-3.12 branch!

UO-OACISS / perfstubs

Code Issues 1 Pull requests 1 Actions Projects Wiki Security Insights Settings

Files

python-3.12

Go to file

- deprecated
- etc
- examples
- makefile_fortran_example
- perfstubs_api
 - README.md
 - README_python.md
 - config.h.default
 - config.h.in
 - pstubs.py
 - pstubs_common.py
 - pstubs_sys_monitoring.py
 - pstubs_sys_settrace.py
 - python.cpp
 - timer.c
 - timer.h
 - timer_f.h

perfstubs / perfstubs_api / README_python.md

khuck Update README_python.md 2e7686f · last month History

Preview Code Blame 196 lines (179 loc) · 11.2 KB

Raw Download Edit

Python support is now included.

Building the perfstubs library will now also build `perfstubs.so`, which is a Python C extension that uses the new low-overhead profiling support in Python 3.12+, see <https://docs.python.org/3/library/sys.monitoring.html>. There is a python module, `pstubs.py` in the perfstubs `<install-prefix>/lib` directory.

To test the Python support, do the following:

```
git clone --branch python-3.12 https://github.com/UO-OACISS/perfstubs.git
cd perfstubs
cmake -B build -DCMAKE_BUILD_TYPE=RelWithDebInfo -DCMAKE_INSTALL_PREFIX='pwd' /install
cmake --build build --parallel --target install
export PYTHONPATH='pwd' /install /lib:$PYTHONPATH
export PYTHONPATH='pwd' /install /lib64:$PYTHONPATH # might be necessary, too
```

To try it with a working TAU installation:

```
tau_exec -T pthread,serial python3 -m pstubs ./examples/firstprime_3.12.py
pprof -a
```

shoud give:

<https://docs.python.org/3/library/sys.monitoring.html>

Python 3.12 introduced `sys.monitoring` to replace `sys.settrace` / `sys.setprofile`

<https://docs.python.org/3/library/sys.html#sys.setprofile> for reference

Lower overhead, more control for selective measurement, multi-thread support

More info: <https://peps.python.org/pep-0669/>,

<https://blog.jetbrains.com/pycharm/2024/01/new-low-impact-monitoring-api-in-python-3-12/>

Goal: provide common measurement plugin for any tool, using `perfstubs`

Include backwards compatibility with older Python for completeness, benchmarking

Prototyped, works but can be improved.