perfmum Documentation

Release

Tristan Carel

Sep 05, 2017

Contents

This is the documentation of **perfmum**.

perfmum is a set of utilities allowing developers to manage performance benchmarks. A benchmark is made of several steps:

1. build
2. execution
3. metrics extraction and archiving
4. analytics and monitoring

perfmum is named that way because it provides everything required to manage those steps.

Note: perfmum specifications are currently being edited, available in the *Specifications* page. If you think this utility can help you, add your use-case in the dedicated section!

CHAPTER 1

Contents

Specifications

This page gathers use-cases, requirements, and technical specifications of **perfmum** utility.

Feel free to use-cases and requirements by creating a pull-request

Use-cases

Micro-benchmarks in CI

As a developer, given a C++ library I maintain, when I push Git changes, then continuous integration executes micro-benchmarks of some critical C++ sections of the library.

Micro-benchmarks CMake helpers

As a developer, given a C++ library built with CMake, **perfmum** provides dedicaded CMake modules to add new micro-benchmarks to the `test` target.

Micro-benchmarks monitoring over time

As a developer, given a C++ library I maintain with a micro-benchmark integrated in the continuous integration process, I can review the history of metrics of this benchmark in a web browser.

Launch coreneuron simulation

As a HPC developer, I can use **perfmum** to describe a coreneuron simulation from the different build configurations to the runtime parameters. **perfmum** will take care of the different phases: build, run, results extraction and archiving, analytics and monitoring.

Simulations for coreneuron UT

As a developer, I can describe the different simulations used by the coreneuron continuous integration workflow.

Coreneuron integration tests

As a HPC developer, I can describe the different simulations that must be executed once a week to validate **coreneuron**.

Alerting in case of performance regression

As a HPC developer owner of a component, if continuous integration detects a performance regression, then I am notified via email or Matrix.

Requirements

- **Can configure build according to several dimensions:**
  - toolchain: gcc+mvapich, intel+intelmpi, ...
  - architectures: cpu, gpu, knl
  - variants: profile, non-profile, optimization-types
- **Can configure the runtime environment of the simulation/test:**
  - datasets
  - input parameters
- Can configure what metrics to extract
- Push all metrics and row results to a DB (Elasticsearch, GPFS)

Technical Specifications

To be defined

License


```
The MIT License (MIT)

Copyright (c) 2017 Tristan Carel

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
```

```
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

Developers

- Tristan Carel <tristan.carel@epfl.ch>

Changelog

Version 0.1

- Feature A added
- FIX: nasty bug #1729 fixed
- add your changes here!

perfmum

perfmum package

Submodules

perfmum.skeleton module

Module contents

CHAPTER 2

Indices and tables

- genindex
- modindex
- search