Coding guides

This is a guide for code contribution to Pegasus. Before writing your code, please take a second reading this doc.

Clang-Format

We currently use clang-format-3.9 for C++ code formatting. Please install this tool and run before submitting a PR:

```
./scripts/format_files.sh

./rdsn/scripts/linux/run-clang-format.sh
```

CCache

Pegasus supports ccache to accelerate compilation. It significantly reduces the building time, by caching the intermediate results on your disk. We highly recommend this tool for Pegasus development.

```
sudo apt install ccache

ccache -M 5G # configure the cache size
```

Once you installed ccache, Pegasus's building process will automatically detect it without any manual configuration.

Continuous Integration

Pegasus runs a Github Actions workflow/TravisCI for each of the submitted pull-request. The CI procedure checks:

- If your PR title matches conventional commit
- If your code is well-formatted
- If your changes can pass all the tests. You can run `./run.sh test` on your local environment.

For more details about how we run Github Actions, please refer to this doc: Github Actions .

In addition to testing-per-PR, we also run daily workflows at https://github.com/pegasus-kv/pegasus-docker that check:

- If rdsn/pegasus can be built upon various platforms and compilers.
- If all code is sanitized (address/leak/thread/undefined).

Heap Profiling

By default, Pegasus enables gperftools and utilizes tcmalloc for memory allocation. To analyzing the memory/cpu usage of Pegasus server, you can run pprof against the target server. The Pegasus server will retrieve profiling data via gperftools library and respond via HTTP.

```
pprof --svg http://127.0.0.1:34801/pprof/heap > heap.svg
```

Thrift

Pegasus uses thrift as the serialization tool for RPC data. To modify the RPC structs in Pegasus, or to add a new RPC type, you need to update the corresponding ".thrift" protocol file, and run:

```
./rdsn/compile_thrift.py # if the thrift is in rdsn

./src/idl/recompile_thrift.sh # if the thrift is in pegasus
```

We directly manage thrift as one of our third-parties, that is, it requires no separate installation. After running `./run.sh build` it will be automatically installed under `./rdsn/thirdparty/output/bin`.