Benchmark Module

Bastian Friedrich
Collax GmbH

Edited by
Bastian Friedrich

Benchmark Module

Edited by Bastian Friedrich and Bastian Friedrich

Revision History

Revision $Revision: 1.1.1.1 $ $Date: 2005/06/13 16:47:40 $

Table of Contents

List of Examples

Chapter 1. User's Guide

1.1. Overview

This module helps developers to benchmark their module functions. By adding this module's functions via the configuration file or through its API, OpenSER can log profiling information for every function.

The duration between calls to start_timer and log_timer is stored and logged via OpenSER's logging facility. Please note that all durations are given as microseconds (don't confuse with milliseconds!).

1.2. Dependencies

1.2.1. OpenSER Modules

The following modules must be loaded before this module:

- *No dependencies on other OpenSER modules*.

1.2.2. External Libraries or Applications

The following libraries or applications must be installed before running OpenSER with this module loaded:

- *None*.

1.3. Exported Parameters

1.3.1. `enable` (int)

Even when the module is loaded, benchmarking is not enabled per default. This variable may have three different values:

- -1 - Globally disable benchmarking

- 0 - Enable per-timer enabling. Single timers are inactive by default and can be activated through the MI interface as soon as that feature is implemented.
- 1 - Globally enable benchmarking

Default value is "0".

Example 1-1. Set `enable` parameter

```
...
modparam("benchmark", "enable", "1")
...
```

1.3.2. `granularity` (int)

Logging normally is not done for every reference to the log_timer() function, but only every n'th call. n is defined through this variable. A sensible granularity seems to be 100.

Default value is "100".

Example 1-2. Set `granularity` parameter

```
...
modparam("benchmark", "granularity", "500")
...
```

1.3.3. `loglevel` (int)

Set the log level for the benchmark logs. These levels should be used:

- -3 - L_ALERT
- -2 - L_CRIT
- -1 - L_ERR
- 1 - L_WARN
- 2 - L_NOTICE
- 3 - L_INFO
- 4 - L_DBG

Default value is "3" (L_INFO).

Example 1-3. Set `loglevel` parameter

```
...
modparam("benchmark", "loglevel", "4")
...
```

This will set the logging level to L_DBG.

1.4. Exported Functions

1.4.1. `start_timer(timer_id)`

Start timer id "timer_id". A later call to "log_timer()" logs this timer. 32 timer IDs (0 through 31) are available.

Example 1-4. `start_timer` usage

```
...
start_timer("1");
...
```

1.4.2. `log_timer(timer_id)`

This function logs the timer with the given ID. The following data are logged:

- *Last msgs* is the number of calls in the last logging interval. This equals the granularity variable.
- *Last sum* is the accumulated duration in the current logging interval (i.e. for the last "granularity" calls).
- *Last min* is the minimum duration between start/log_timer calls during the last interval.
- *Last max* - maximum duration.
- *Last average* is the average duration between start_timer() and log_timer() since the last logging.
- *Global msgs* number of calls to log_timer.
- *Global sum* total duration in microseconds.

- *Global min*... You get the point. :)
- *Global max* also obvious.
- *Global avg* possibly the most interesting value.

Example 1-5. `log_timer` usage

```
...
log_timer("1");
...
```

Chapter 2. Developer's Guide

The benchmark module provides an internal API to be used by other OpenSER modules. The available functions are identical to the user exported functions.

Please note that this module is intended for developers. It should not be used in production code. If you do that anyways, consider using a module parameter for the timer id to use to prevent collisions of IDs.

2.1. Available Functions

2.1.1. `bm_start(timer_id)`

This function equals the user-exported function start_timer. The timer_id is passed as an integer, though.

2.1.2. `bm_log(timer_id)`

This function equals the user-exported function log_timer. The timer_id is passed as an integer, though.

2.1.3. Examples

Example 2-1. Using the benchmark module's API from another module

```
...
#include "../benchmark/benchmark.h"
...
struct bm_binds bmb;
...
...
/* load the benchmarking API */
if (load_bm_api( &bmb )!=0) {
    LOG(L_ERR, "ERROR: can't load benchmark API\n");
    goto error;
}
...
...
/* Start/log timers during a (usually user-exported) module function */
bmb.bm_start(1);
do_something();
bmb.bm_log(1);
...
```

Chapter 3. Frequently Asked Questions

1. Where can I find more about OpenSER?

Take a look at http://openser.org/.

2. Where can I post a question about this module?

First at all check if your question was already answered on one of our mailing lists:

- User Mailing List - http://openser.org/cgi-bin/mailman/listinfo/users
- Developer Mailing List - http://openser.org/cgi-bin/mailman/listinfo/devel

E-mails regarding any stable OpenSER release should be sent to <`users@openser.org`> and e-mails regarding development versions should be sent to <`devel@openser.org`>.

If you want to keep the mail private, send it to <`team@openser.org`>.

3. How can I report a bug?

Please follow the guidelines provided at: http://sourceforge.net/tracker/?group_id=139143.