

lwaio: a light-weight asynchronous input/output library

Diego Fabregat-Traver

December, 2014

Abstract

The `lwaio` library is a light-weight library for asynchronous input/output (I/O). `lwaio` replicates the basic functionality of the POSIX `aio` library with the following additional features: it only spawns one thread to perform the I/O operations, and this thread may be pinned to a specific core. The main reason why I developed this library is to overcome the performance penalty observed when using the original `aio` library to overlap I/O operations with the execution of highly optimized BLAS [?] routines. This problem was experienced while developing the code presented in [?, ?]. In the specific case of [?], the use of the `aio` library limited scalability of the code on a 40-core node to about 20x, while with our library we attained more than 36x.

This document is under construction. If you have any comment or question, feel free to contact me at

`fabregat@aices.rwth-aachen.de`

1 Overview

2 Interface

```
typedef struct lwaio_task
{
    FILE    *fp;
    off_t   offset;
    size_t  nbytes;
    void    *buffer;
} lwaio_task;

void lwaio_init( void );
void lwaio_finalize( void );

void lwaio_read( lwaio_task *task );
void lwaio_write( lwaio_task *task );
void lwaio_wait( lwaio_task *task );

LWAIO_PIN_TO="0"
```

3 Internals

4 Example of use: double buffering