

页 数 _____

名称 无线时序数据有损压缩技术
软件用户手册 v1.0

批 准 _____

会

修订记录

日期	修订版本	修改描述	作者
2019.04.12	V1.0	试用版软件用户手册1.0	李旭贤

目录

1	简介	1
2	安装编译.....	1
2.1	目录结构.....	1
2.2	编译安装.....	3
3	使用方式.....	4
3.1	压缩/解压.....	4
3.2	批量测试.....	5
4	误差测算.....	7

1 简介

hwzip 一种无线性能数据有损压缩软件，由武汉大学计算机学院张健教授团队开发，采用 C++ 语言编写。hwzip 是一种可以控制误差的有损压缩算法，目前提供针对数据的相对误差和平均误差的控制。该软件根据用户指定的参数，对输入的 csv 文件进行有损压缩处理，生成中间文件，再对该中间文件进行进一步的无损压缩和编码，以获得较高的压缩率。

hwzip 只适用与特定格式的 csv 文件，文件结构要求如表 1.1 所示。其中前三列为时间、名称、对象信息，其余列为具体的性能指标数据，指标数据包括浮点数和整数。

表 1.1 输入 CSV 文件格式

TIME	NENAME	OBJECT	VS_AMR_DL_RateUp	VS_AMR_DL_RateDown	...
2019-03-04 09:00:00	1	11	22.87055587768555	2.1433098316192627	...
2019-03-04 09:00:00	2	12	44.241111755371094	2.8573987483978267	...
...

2 安装编译

2.1 目录结构

hwzip 软件的目录结构如图 2.1 所示。

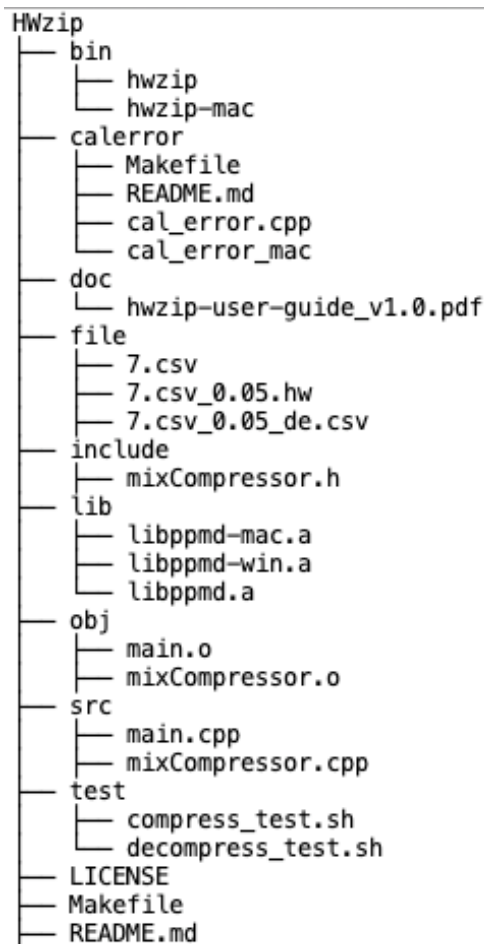


图 2.1 hwzip 软件目录结构

其中，每个目录或文件的内容如下：

bin/：该目录存放已经编译好的压缩软件可执行程序，windows 平台的软件名为 hwzip.exe，Linux/MacOS 平台的软件名为 hwzip。

calerror/：该目录存放误差测算程序的代码及可执行程序。

doc/：该目录存放软件用户手册。

file/：该目录存放 csv 文件，如名为“2.csv”的文件表示原始数据文件。后缀名为“.hw”的表示压缩文件，如“2.csv_0.01.hw”表示 2.csv 在最大平均误差为 0.01 时的压缩文。后缀为“_de.csv”表示解压出的文件，如“2.csv_0.01_de.csv”表示根据文件 2.csv_0.01.hw 解压出的 csv 文件。

include/：该目录存放 C/C++头文件。

lib/: 该目录存放程序中使用到的静态库文件。在 **linux** 平台，其后缀名为“**libxx-win.a**”表示 windows 平台用到的 **xx** 静态库文件，“**libxx-mac.a**”表示 macOS 平台用到的 **xx** 静态库文件，Linux 平台的静态库文件名为“**libxx.a**”。

obj/: 该目录存放编译出的.o 目标文件。

src/: 该目录存放 C/C++源代码文件。

test/: 该目录存放测试脚本，包括 **compress_test.sh** 和 **decompress_test.sh** 两个 shell 脚本文件。对存放在 **file** 目录中的文件进行压缩和解压测试。

LICENSE: 该文件是本开源软件采用的 GPL v2.0 的 LICENSE。

Makefile: 该文件是编译可执行程序时用到的 makefile 文件，在项目根目录下运行 **make** 命令即可，生成的可执行文件在 **bin** 目录中。

README.md: 该文件是项目说明文件，采用 Markdown 语法编写。

2.2 编译安装

本项目的 **bin** 目录中已经包含了 Linux 和 macOS 平台的可执行程序，可直接使用。其中，macOS 平台的可执行程序名为“**hwzip-mac**”，Linux 平台可执行程序为“**hwzip**”，暂不支持 windows 平台。如需重新编译，请进入项目根目录并执行“**make clean**”和“**make**”命令。

本软件目前支持 Linux/macOS 平台。编译时需进入项目根目录，运行 **make** 命令进行安装。在 macOS 和 Linux 已经自带了编译器，无需安装，在 macOS 中，使用的是自带的 **clang** 编译器，在 Linux 中使用的是 **gcc** 编译器。

如需直接使用“**hwzip**”或“**hwzip-mac**”命令，请执行“**make install**”命令；如需卸载，请执行“**make uninstall**”命令。

编译安装的流程如图 2.2 所示，安装完成后可直接执行 hwzip 命令。

```
[root@izwz96ybmtj4h7swlm6iguz HWzip]# pwd
/usr/lxx/HWzip
[root@izwz96ybmtj4h7swlm6iguz HWzip]# make
mkdir -p obj/
[root@izwz96ybmtj4h7swlm6iguz HWzip]# make install
cp ./bin/hwzip /usr/local/bin
[root@izwz96ybmtj4h7swlm6iguz HWzip]# hwzip
hwzip v1.0
How to use hwzip:
    compress:          hwzip c [input] [max_pw_rel_err] [max_avg_err]
    decompress:        hwzip d [input]
```

图 2.2 hwzip 安装流程

3 使用方式

3.1 压缩/解压

hwzip 提供命令行接口，压缩文件时，其命令格式为如图 3.1 所示。

```
How to use hwzip:
    compress:          hwzip c [input] [max_pw_rel_err] [max_avg_err]
    decompress:        hwzip d [input]
```

图 3.1 hwzip 命令格式

其中“hwzip”表示可执行程序路径，若当前路径为项目根目录，则需要使用“bin/hwzip”；“c”表示压缩命令，“d”表示解压命令；input 为输入文件路径，max_pw_rel_err 表示数值的最大相对误差；max_avg_err 表示性能指标数据的最大平均误差。例如，若输入文件为 file/2.csv，要求的最大相对误差为 0.10，最大平均误差为 0.05，则压缩命令为“hwzip c file/2.csv 0.10 0.05”。若要对文件 file/2.csv_0.05.hw 进行解压，则解压命令为“hwzip d file/2.csv_0.05.hw”。

压缩示例如图 3.2 所示，解压示例如图 3.3 所示。

```
[root@izwz96ybmtj4h7swlm6iguz HWzip]# hwzip c file/example.csv 0.1 0.05
columnSize = 194
lossy processing... 66.666667 %
lossy process 2 blocks

lossless encoding, PPMD algorithm...
3623612 > 792548, 1.75 bpb, 49.9M RAM, 1863Kb/s
Processing time: 1.911s
finish compressFile_ppmd
remove tmp file
[root@izwz96ybmtj4h7swlm6iguz HWzip]#
```

图 3.2 压缩命令及结果示例

```
[root@izwz96ybmtj4h7swlm6iguz HWzip]# hwzip d file/example.csv_0.05.hw
tempFilePath = file/example.csv_0.05.tmp
outputFilePath = file/example.csv_0.05_de.csv
lossless decoding, PPMD algorithm
792548 > 3623612, 1.75 bpb, 49.9M RAM, 1883Kb/s
Processing time: 1.891s
decompress file to file/example.csv_0.05.tmp
blockSize = 3000
columnSize = 194
fileLines = 6001
blocks = 2
lossy processing... 100.000000 %remove tmp file

finish decompress all 2 blocks
decompress file file/example.csv_0.05.hw to file file/example.csv_0.05_de.csv
[root@izwz96ybmtj4h7swlm6iguz HWzip]#
```

图 3.3 解压命令及结果示例

3.2 批量测试

使用脚本进行批量测试前，请将待测试文件放入项目的 file 文件夹中，脚本仅对 file 文件夹中后缀名为“.csv”的文件进行压缩，仅对后缀名为“.hw”的文件进行解压。

(1) 批量压缩

在项目的 test 文件夹中，包含了两个 shell 脚本，compress_test.sh 和 decompress_test.sh，分别对文件进行批量压缩测试和解压测试。脚本

使用方式步骤为：

(1) # cd test/

(2) # ./compress_test.sh [max_pw_rel_err] [max_avg_err]

步骤 1 进入 test 文件夹，即脚本所在目录，步骤 2 执行脚本，其中 max_pw_rel_err 为最大相对误差，max_avg_err 为最大平均误差。如，“./compress_test.sh 0.10 0.05”。具体步骤及结果如图 3.4 所示。

```
[root@izwz96ybmtj4h7swlm6iguz HWzip]# pwd
/usr/lxx/HWzip
[root@izwz96ybmtj4h7swlm6iguz HWzip]# cd test/
[root@izwz96ybmtj4h7swlm6iguz test]# pwd
/usr/lxx/HWzip/test
[root@izwz96ybmtj4h7swlm6iguz test]# ./compress_test.sh 0.1 0.05
/usr/lxx/HWzip/file/2.csv
file size = 473525070 bytes
cmd = ../bin/hwzip c /usr/lxx/HWzip/file/2.csv 0.1 0.05
^C
[root@izwz96ybmtj4h7swlm6iguz test]#
```

图 3.4 批量压缩脚本示例

(2) 批量解压

批量解压脚本使用方式与压缩类似，在 test 目录下执行 decompress_test.sh 脚本即可。如图 3.5 所示。

```
[root@izwz96ybmtj4h7swlm6iguz HWzip]# pwd
/usr/lxx/HWzip
[root@izwz96ybmtj4h7swlm6iguz HWzip]# cd test/
[root@izwz96ybmtj4h7swlm6iguz test]# ls *.sh
compress_test.sh decompress_test.sh
[root@izwz96ybmtj4h7swlm6iguz test]# ./decompress_test.sh
dir = /usr/lxx/HWzip/file
file size = 22209779 bytes
cmd = ../bin/hwzip d /usr/lxx/HWzip/file/2.csv_0.05.hw
^C
[root@izwz96ybmtj4h7swlm6iguz test]#
```

图 3.5 批量解压脚本示例

注：该脚本仅支持 Linux 和 MacOS 平台，且需要给 compress_test.sh 和 decompress_test.sh 添加执行权限。

4 误差测算

目录 `calerror` 中为误差测算程序,若要使用,请进入 `calerror` 目录,运行“`make`”命令进行编译。在 Linux 中编译出的可执行程序为“`cal_error`”,在 macOS 中为“`cal_error_mac`”。程序执行命令为“`./cal_error compressed_file decompressed_file max_rel_err max_avg_err`”,其中“`compress_file`”为原 csv 文件,“`decompressed_file`”为解压文件,“`max_rel_err`”为最大相对误差,“`max_avg_err`”最大平均误差(按每列指标进行计算)。

误差测算程序的使用和输出如图 4.1 所示。

```
[root@izwz96ybmtj4h7swlm6iguz calerror]# pwd
/usr/lxx/HWzip/calerror
[root@izwz96ybmtj4h7swlm6iguz calerror]# make
g++ -O2 -g -o cal_error cal_error.cpp
[root@izwz96ybmtj4h7swlm6iguz calerror]# ls
cal_error      cal_error_mac  Makefile
cal_error.cpp  cal_error_mac.dSYM  README.md
[root@izwz96ybmtj4h7swlm6iguz calerror]# ./cal_error ../file/example.csv ../file/example.csv_0.05_de.csv 0.1 0.05
col_num  avg_err  max_err
      0  0.0311565  0.04996
      1  0.0297926  0.0499954
      2   0.010674  0.047619
      3   0.030962  0.049969
      4  0.00855865  0.047619
      5   0.0303615  0.0499919
      6  0.00616328  0.047619
      7  0.00185699  0.0455991
      8  0.00386079  0.0498129
      9  0.00225988  0.0422359
     10  0.00225988  0.0422359
     11  0.00328983  0.0470588
```

图 4.1 误差测算程序使用和输出示例

程序将输出每列的列号、平均误差、最大相对误差。(注:前三列不输出,因为前三列为时间、名字、ID 等信息,未进行有损压缩,无误差)。