第 3 章

鲲鹏加速库

3.1 鲲鹏加速库简介

为了充分发挥鲲鹏 CPU 硬件设计的优良性能及 ARM 指令本身的优势，华为推出了一系列基于硬件加速和软件指令加速的鲲鹏加速库，这些加速库以基础库的形式提供，兼容开放的接口，在保证上层应用基本不需要更改代码的前提下，为鲲鹏平台的应用提供更强的能力。

鲲鹏加速库目前可以分为 7 个大类 24 个加速库，下面按照类别对这些加速库做简要说明。

3.1.1 系统库

1. Glibc-patch

- 加速库类别：指令加速；
- 开发语言：汇编；
- 下载网址：http://ftp.jaist.ac.jp/pub/GNU/libc/；
- 加速库介绍：Glibc-patch 主要对内存、字符串、锁等接口基于华为鲲鹏 920 处理器微架构的特点进行了加速优化，memcmp/memset/memcpy/memrchr/strcpy/strlen/strnlen 已合入 GNU 社区，随 Glibc 2.31 主干版本发布，同步推送 openEuler 社区，已随 openEuler 1.0 发布。

2. HyperScan

- 加速库类别：指令加速；
- 开发语言：C；
- 下载网址：https://github.com/kunpengcompute/hyperscan；
- 加速库介绍：HyperScan 是一款高性能的正则表达式匹配库，它遵循 libpcre 库通用的正则表达式语法，拥有独立的 C 语言接口。在 HyperScan 正式发布的 5.2.1 版本的基础上，参考华为鲲鹏微架构特征，重新设计核心接口的实现机制，并完成了开发和性能优化，推出了适合鲲鹏计算平台的软件包。

3. AVX2Neon

- 加速库类别：异构生态迁移；
- 开发语言：C；
- 下载网址：https://github.com/kunpengcompute/AvxToNeon；
- 加速库介绍：AVX2Neon是一款接口集合库。当使用Intrinsic类接口的应用程序从传统平台迁移到鲲鹏计算平台时，由于各个平台的Intrinsic函数的定义不同，需要逐一对Intrinsic函数重新进行适配开发。针对该问题，华为提供了AVX2Neon模块，将传统平台的Intrinsic接口集合使用鲲鹏指令重新实现，并封装为独立的接口模块(C语言头文件方式)，以减少大量迁移项目重复开发的工作量。用户通过将头文件导入应用程序即可继续使用传统平台的Intrinsic函数。

3.1.2 压缩库

1. Gzip

- 加速库类别：指令加速；
- 开发语言：C；
- 下载网址：https://github.com/kunpengcompute/gzip；
- 加速库介绍：Gzip(GNU zip)是一款发布较早并已广泛应用的压缩软件。其优化版本在官网发布的Gzip-1.10 Release版本基础上，通过数据预取、循环展开、CRC指令替换等方法，来提升其在鲲鹏计算平台上的压缩和解压缩速率，尤其对文本类型文件的压缩及解压具有更明显的性能优势。

2. ZSTD

- 加速库类别：指令加速；
- 开发语言：C；
- 下载网址：https://github.com/kunpengcompute/zstd；
- 加速库介绍：Zstandard，即ZSTD压缩库，是2016年开源的一款快速无损压缩算法，基于C语言开发，旨在提供zlib库对应级别的压缩解压速度和更高的压缩比。其补丁版本在官网发布的zstd-1.4.4 Release版本上，通过NEON指令、内联汇编、代码结构调整、内存预取、指令流水线排布优化等方法，实现ZSTD在鲲鹏计算平台上压缩和解压性能的提升。

3. Snappy

- 加速库类别：异构生态迁移；
- 开发语言：C/C++；
- 下载网址：https://github.com/kunpengcompute/snappy；
- 加速库介绍：Snappy是一款基于C++语言开发的压缩算法，旨在提供较高的压缩/解压速率和相对合理的压缩比，其优化版本在官网发布的Snappy-1.1.7 Release版本上，利用内联汇编、宽位指令、优化CPU流水线、内存预取等方法，实现Snappy在鲲鹏计算平台上的压缩和解压速率提升。

4. KAEzip

- 加速库类别：压缩硬加速；
- 开发语言：C；
- 下载网址：https://github.com/kunpengcompute/KAEzip；
- 加速库介绍：KAEzip 是鲲鹏加速引擎的压缩模块，使用鲲鹏硬加速模块实现 deflate 算法，结合无损用户态驱动框架，提供高性能 Gzip/zlib 格式压缩接口。

3.1.3 加解密库

KAE 加解密

- 加速库类别：加解密硬加速；
- 开发语言：C；
- 下载网址：https://github.com/kunpengcompute/KAE；
- 加速库介绍：KAE 加解密是鲲鹏加速引擎的加解密模块，使用鲲鹏硬加速模块实现 RSA/SM3/SM4/DH/MD5/AES 算法，结合无损用户态驱动框架，提供高性能对称加解密、非对称加解密算法能力，兼容 OpenSSL 1.1.1a 及其之后版本，支持同步 & 异步机制。

3.1.4 媒体库

1. HMPP

- 加速库类别：媒体信号库；
- 开发语言：C、汇编；
- 下载网址：https://support.huawei.com/enterprise/zh/kunpeng-computing/kunpeng-computing-dc-solutios-pid-251181670/software/252791773；
- 加速库介绍：鲲鹏超媒体性能库 HMPP(Hyper Media Performance Primitives)包括向量缓冲区的分配与释放、向量初始化、向量数学运算与统计学运算、向量采样与向量变换、滤波函数、变换函数(快速傅里叶变换)，支持 IEEE 754 浮点数运算标准，支持鲲鹏平台下使用。

2. HW265

- 加速库类别：视频优化库；
- 开发语言：C、汇编；
- 下载网址：向华为申请；
- 加速库介绍：HW265 视频编码器是符合 H.265/HEVC 视频编码标准、基于鲲鹏处理器 NEON 指令加速的华为自研 H.265 视频编码器。HW265 支持 4 个预设编码挡位可选，对应不同编码速度的应用场景，码率控制支持平均比特率模式(ABR)和恒定 QP 模式(CQP)，功能涵盖直播、点播等各个场景，整体性能优于目前的主流开源软件。

3. X265

- 加速库类别：视频转码库；
- 开发语言：C、汇编；
- 下载网址：https://bitbucket.org/multicoreware/x265_git/downloads/? tab=tags；
- 加速库介绍：针对 FFmpeg 视频转码场景，对 X265 的转码底层算子使用鲲鹏向量指令进行加速优化，提高整体性能。补丁已经回馈 X265 官网社区，已在 X265 3.4 版本正式发布。

4. X264

- 加速库类别：视频编解码库；
- 下载网址：https://www.videolan.org/developers/x264.html；
- 加速库介绍：X264 是采用 GPL 授权的视频编码免费软件，主要功能实现了 H.264/MPEG-4 AVC 的视频编码。

3.1.5 数学库

1. KML_FFT

- 加速库类别：傅里叶变换库；
- 开发语言：C；
- 下载网址：https://support.huawei.com/enterprise/zh/kunpeng-computing/kunpeng-boostkit-pid-253662225/software/253678165；
- 加速库介绍：KML_FFT 是快速傅里叶变换数学库，快速傅里叶变换(fast Fourier transform，FFT)，是快速计算序列的离散傅里叶变换(DFT)或其逆变换的方法，广泛地应用于工程、科学和数学领域，将傅里叶变换计算需要的复杂度从 $O(n^2)$ 降到了 $O(n\log n)$，被 IEEE 科学与工程计算期刊列入 20 世纪十大算法。KML_FFT 基于鲲鹏架构，通过向量化、算法改进，对快速离散傅里叶变换进行了深度优化，使快速傅里叶变换接口函数的性能有大幅度提升。

2. KML_BLAS

- 加速库类别：基础线性代数库；
- 开发语言：C；
- 下载网址：https://support.huawei.com/enterprise/zh/kunpeng-computing/kunpeng-boostkit-pid-253662225/software/253678165；
- 加速库介绍：KML_BLAS 是一个基础线性代数运算数学库，基于鲲鹏架构提供了 3 个层级的高性能向量运算：向量-向量运算、向量-矩阵运算和矩阵-矩阵运算，是计算机数值计算的基石，在制造、机器学习、大数据等领域应用广泛。KML_BLAS 基于鲲鹏架构，通过向量化、数据预取、编译优化、数据重排等手段，对 BLAS 的计算效率进行了深度挖掘，使 BLAS 接口函数的性能逼近理论峰值。

3. KML_SPBLAS

- 加速库类别：稀疏基础线性代数库；

- 开发语言：C；
- 下载网址：https://support.huawei.com/enterprise/zh/kunpeng-computing/kunpeng-boostkit-pid-253662225/software/253678165；
- 加速库介绍：KML_SPBLAS是稀疏矩阵的基础线性代数运算库，基于鲲鹏架构为压缩格式的稀疏矩阵提供了高性能向量、矩阵运算。KML_SPBLAS基于鲲鹏架构，充分利用鲲鹏的指令集和架构特点，开发了高性能稀疏矩阵运算库，提升了HPC和大数据解决方案的业务性能。

4. KML_MATH

- 加速库类别：基础数学函数库；
- 开发语言：C；
- 下载网址：https://support.huawei.com/enterprise/zh/kunpeng-computing/kunpeng-boostkit-pid-253662225/software/253678165；
- 加速库介绍：KML_MATH是数学计算的基础库，主要实现了基本的数学运算、三角函数、双曲函数、指数函数、对数函数等，广泛应用于科学计算，如气象、制造、化学等行业。KML_MATH通过周期函数规约、算法改进等手段，提供了基于鲲鹏处理器性能提升较大的函数实现。

5. KML_VML

- 加速库类别：向量运算库；
- 开发语言：C；
- 下载网址：https://support.huawei.com/enterprise/zh/kunpeng-computing/kunpeng-boostkit-pid-253662225/software/253678165；
- 加速库介绍：KML_VML是向量运算数学库，主要实现基本的数学运算、三角函数、双曲函数、指数函数、对数函数等数学接口的向量化实现。KML_VML通过NEON指令优化、内联汇编等方法，对输入数据进行向量化处理，充分利用了鲲鹏架构下的寄存器特点，实现了在鲲鹏处理器上的性能提升。

6. KML_LAPACK

- 加速库类别：线性代数运算库；
- 开发语言：C；
- 下载网址：https://support.huawei.com/enterprise/zh/kunpeng-computing/kunpeng-boostkit-pid-253662225/software/253678165；
- 加速库介绍：KML_LAPACK是线性代数运算库，提供了线性方程组运算，包括方程组求解、特征值和奇异值问题求解等。KML_LAPACK通过分块、求解算法组合、多线程、BLAS接口优化等手段，基于鲲鹏架构对LAPACK的计算效率进行了优化，实现了在鲲鹏处理器上的性能提升。

7. KML_SVML

- 加速库类别：短向量运算库；

- 开发语言：C；
- 下载网址：https://support.huawei.com/enterprise/zh/kunpeng-computing/kunpeng-boostkit-pid-253662285/software/253678165；
- 加速库介绍：KML_SVML 是短向量的数学运算，包括幂函数、三角函数、指数函数、双曲函数、对数函数等。KML_SVML 通过 NEON 指令优化、内联汇编等方法，对输入向量进行批量处理，充分利用了鲲鹏架构下的寄存器特点，实现了在鲲鹏服务器上的性能提升。

8. KML_SOLVER

- 加速库类别：稀疏迭代求解库；
- 开发语言：C；
- 下载网址：https://support.huawei.com/enterprise/zh/kunpeng-computing/kunpeng-boostkit-pid-253662285/software/253678165；
- 加速库介绍：KML_ SOLVER 是稀疏迭代求解库(Iterative Sparse Solvers)，包含了预条件共轭梯度法(PCG)和广义共轭残差法(GCR)。当前 KML_SOLVER 为单节点多线程版本。

3.1.6 存储库

1. Smart Prefetch

- 加速库类别：智能预取库；
- 开发语言：C、C++；
- 下载网址：https://support.huawei.com/enterprise/zh/kunpeng-computing/kunpeng-computing-dc-solution-pid-251181670/software/252325103；
- 加速库介绍：针对分布式存储、大数据的 Spark/HBase 等解决方案中的存储 I/O 密集型场景，访问 I/O 存储器(硬盘、SSD)的性能(带宽、延迟、单位时间操作数)，对业务整体性能影响明显。同时，在这些场景里，用户对存储器的单位容量成本也很敏感。在现在及未来的很长一段时间里，存储器容量大小与 I/O 性能不可能兼具。同时，利用小容量的高速存储介质作为缓存盘，把预测可能被访问的 I/O 数据提前放入缓存盘中，下次直接从高速缓存中获取数据，可以显著地改善系统整体的存储 I/O 性能，这里小容量高速存储介质可以是内存为介质的 Ramdisk，也可以是 NVMe SSD。Smart Prefetch(智能预取)，创新性地采用高速缓存盘配合高效的预取算法，提升了系统存储 I/O 性能，进而提升了上述解决方案中存储 I/O 密集型场景的整体性能。

2. SPDK

- 加速库类别：SSD 用户态驱动；
- 下载网址：https://github.com/spdk/spdk；
- 加速库介绍：SPDK 的全称为 Storage Performance Development Kit(高性能存储开发包)，SPDK 的目标是通过网络技术、处理技术和存储技术来提升效率和性能。通

过运行为硬件设计的软件，SPDK 已经证明很容易达到每秒数百万次 I/O 读取，通过许多处理器核心和许多 NVMe 驱动去存储，而不需要额外卸载硬件。

3. ISA-L

- 加速库类别：存储加速库；
- 下载网址：https://github.com/intel/isa-l；
- 加速库介绍：ISA-L 的全称为 Intelligent Storage Acceleration Library，是提供 RAID、纠删码、循环冗余检查、密码散列和压缩的高度优化的函数。

3.1.7 网络库

1. XPF

- 加速库类别：OVS 流表加速库；
- 开发语言：C；
- 下载网址：https://support.huawei.com/enterprise/zh/kunpeng-computing/kunpeng-computing-dc-solution-pid-251181670/software/252325103；
- 加速库介绍：XPF(Extensible Packet Framework)加速库是鲲鹏自研加速库，XPF 自研功能模块，在 OVS(Open vSwitch)软件内部实现了一个智能卸载引擎模块，该模块用于跟踪数据报文在 OVS 软件中所经历的所有流表和 CT 表，将执行的 CT 行为和所有流表行为项进行综合编排成一条综合行为项并结合统一匹配项生成一条集成流表项。后续的数据报文在进入 OVS 后，若匹配命中该集成流表，则直接执行综合行为，相比开源的处理流程，查询次数将减少，性能将大幅度提升。

2. DPDK

- 加速库类别：用户态网络驱动库；
- 开发语言：C；
- 下载网址：https://github.com/DPDK/dpdk；
- 加速库介绍：DPDK 的全称为 Data Plane Development Kit，为用户空间高效的数据包处理提供数据平面开发工具集，包括库函数和驱动。

3.2 鲲鹏加速库插件

在鲲鹏平台的应用开发中，开发者可以直接使用鲲鹏加速库来优化应用的性能，不过，因为加速库数量众多，实现的函数和匹配的汇编指令更是数以千计，人工识别并应用确实有一定的难度，为了解决这个问题，华为推出了鲲鹏加速库插件，自动扫描代码文件中可使用鲲鹏加速库优化后的函数或汇编指令，生成可视化报告；编码时能够自动匹配鲲鹏加速库函数字典，智能提示、高亮、联想字典中可以替换的库和函数。

鲲鹏加速库插件支持 Visual Studio Code 和 IntelliJ IDEA，为简单起见，本书只介绍 Visual Studio Code 中该插件的安装、使用和卸载。

3.2.1 鲲鹏加速库插件的安装与卸载

鲲鹏加速库插件的名称为 Kunpeng Library Plugin，安装步骤可参考 2.4.1 节“鲲鹏代码迁移插件的安装”；卸载步骤可参考 2.4.3 节“鲲鹏代码迁移插件的卸载”。

3.2.2 鲲鹏加速库插件的使用

为了演示插件功能，需要先创建一个代码文件，这里使用一段对给定字符串压缩，然后计算压缩率，最后解压、输出字符串的代码，命名为 zlibtest.c，代码如下：

```
//Chapter3/zlibtest.c

#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include "zlib.h"

int main()
{
  signed char * content = "This is the string to be compressed. This is the string to be
compressed. This is the string to be compressed.";
  /* 要压缩的字符串长度, +1 是为了算上最后的结束符 */
  uLong contentLen = strlen(content) + 1;

  /* 压缩后数据长度的上限 */
  uLong compressBufLen = compressBound(contentLen);

  /* 分配压缩缓冲区,缓冲区大小不超过 compressBufLen */
  unsigned char* compressBufStream = (unsigned char*)malloc(compressBufLen);

  /* 执行压缩 */
  int compressResult = compress(compressBufStream, &compressBufLen, (const unsigned char *)
content, contentLen);

  /* 缓冲区不够大 */
  if(compressResult == Z_BUF_ERROR){
    printf("Buffer is too small for compression!\n");
    return 1;
  }

  /* 内存不够用 */
  if(compressResult == Z_MEM_ERROR){
    printf("Not enough memory for compression!\n");
    return 2;
  }

  /*压缩率*/
```

```
  float ratio = (float)compressBufLen/contentLen;
  printf("Source length is %d ,target length is %d,the compression ratio is %.2f!\n",
contentLen,compressBufLen,ratio);

  unsigned char *compressedStream = compressBufStream;
  signed char* decompressBufStream = (signed char *)malloc(contentLen);

  int decompressResult = uncompress((unsigned char *)decompressBufStream, &contentLen,
compressedStream, compressBufLen);

  /* 缓冲区不够大 */
  if(decompressResult == Z_BUF_ERROR){
    printf("Buffer is too small for decompression!\n");
    return 1;
  }

  /* 内存不够用 */
  if(decompressResult == Z_MEM_ERROR){
    printf("Not enough memory for decompression!\n");
    return 2;
  }

  printf("%s\n", decompressBufStream);
  return 0;
}
```

1. 语法高亮

插件可以高亮标识出鲲鹏加速优化后的函数，打开代码文件，可以看到一些函数是绿色高亮显示的，当把鼠标放到该函数上时，会显示该函数的优化描述，包括优化点、下载网址等，如图 3-1 所示。

图 3-1 语法高亮

2. 智能联想

在编码时，插件可以自动联想出鲲鹏加速后的函数，例如，输入字符串 com，后面就会自动联想出匹配的函数来，并且会提示该函数的参数信息及所在的文件，如图 3-2 所示。

图 3-2　智能联想

3. 定义跳转

对于鲲鹏加速的函数，按着 Ctrl 键，把鼠标放在函数名称上，可以出现超链接的状态，如图 3-3 所示。

图 3-3　函数超链接

此时，单击该函数名称，会自动跳转到该函数的定义位置，如图 3-4 所示。

4. 函数搜索

单击代码编辑界面右上角的 图标，此时会弹出函数搜索窗口，提供对 function 函数的名称、优化点、描述搜索，以及 Intrinsic 函数的名称、函数详细定义、Intel 对应的 Intrinsic 功能函数名称、Intel 对应的汇编指令名称、ARM 对应的汇编指令名称的搜索，如图 3-5 所示。

5. 加速分析

在资源管理器下的文件或者文件夹上右击，此时弹出的菜单会出现“鲲鹏加速分析”“查看加速分析报告”“清除加速分析报告”菜单项，如图 3-6 所示。

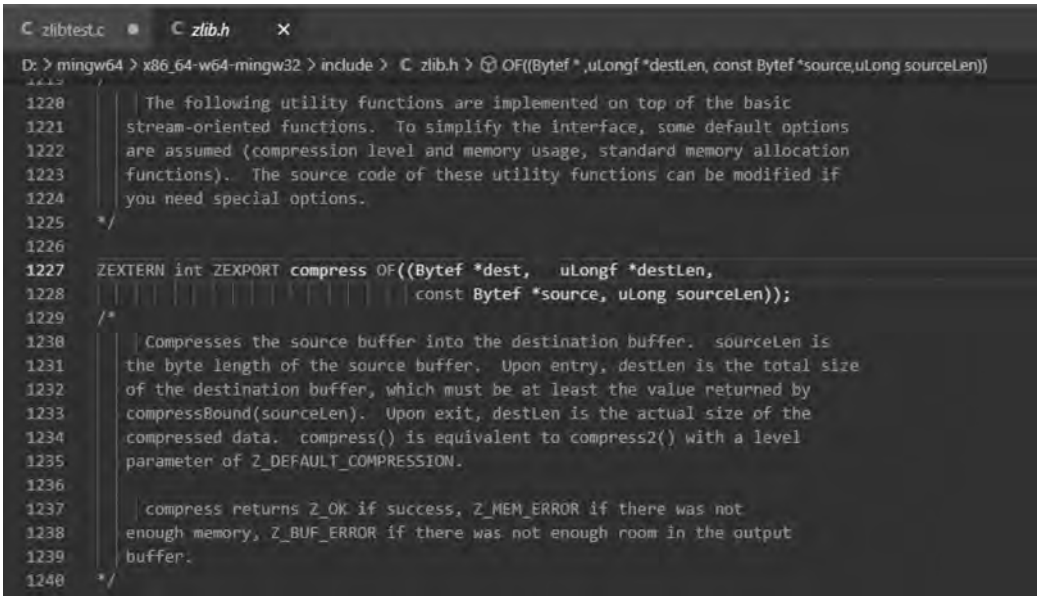

图 3-4 定义跳转

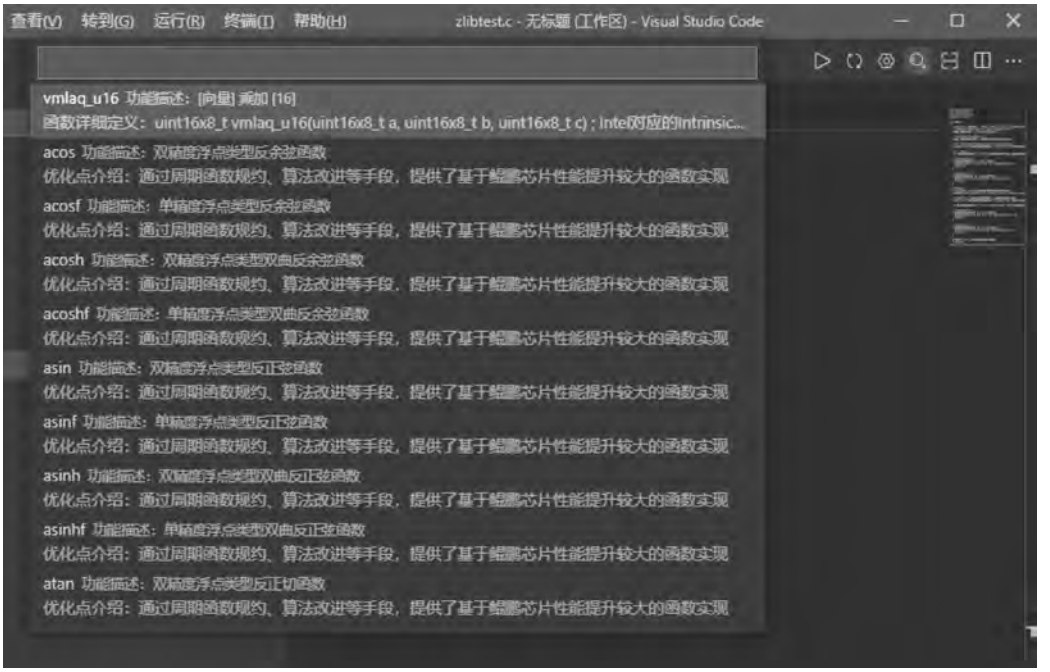

图 3-5 搜索对话框

图 3-6　加速分析菜单

1）鲲鹏加速分析

单击“鲲鹏加速分析”菜单项，此时会出现“选择加速分析类型”对话框，如图 3-7 所示。

图 3-7　选择加速分析类型

选择合适的加速库后，单击“确认分析”按钮，即可执行分析，分析结果如图 3-8 所示。

在源码上，使用波浪线并且绿色高亮显示可以通过加速库优化的函数，在源码下的“问题”选项卡，列出了所有该类函数，并且给出了函数名称、描述、优化点和下载网址，单击该问题，可以跳转到源码对应的函数。

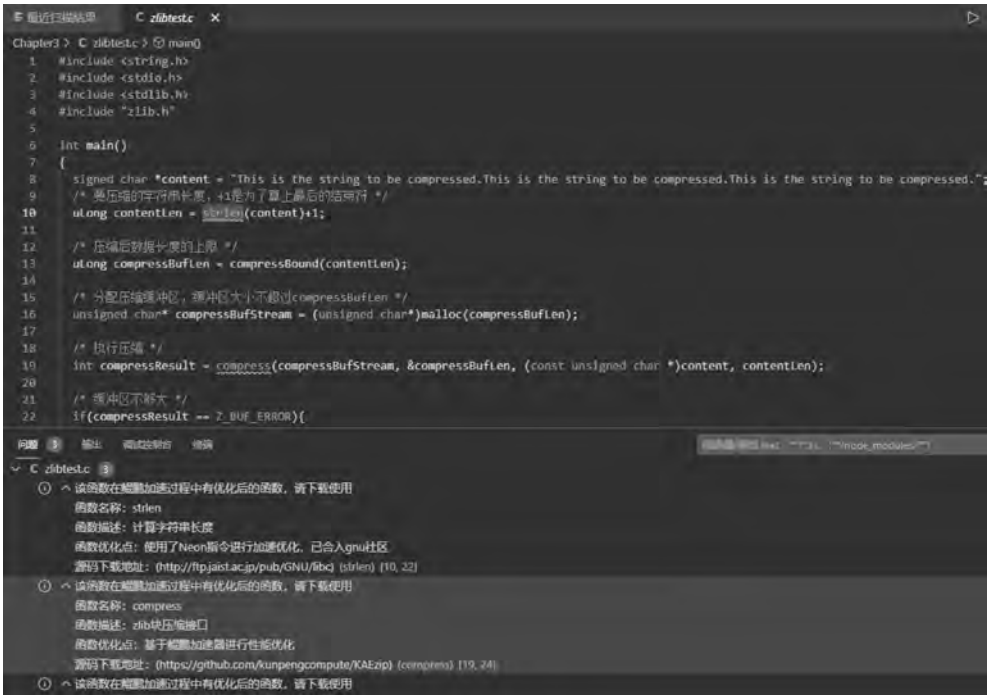

图 3-8　分析结果

2）查看加速分析报告

单击“查看加速分析报告”菜单项，此时会出现“最近扫描结果”对话框，如图 3-9 所示。

图 3-9　最近扫描结果

在推荐加速库的表格里，列出了所有可以应用鲲鹏加速库的函数信息，单击“操作”列的“查看”超链接，可以跳转到源码对应的函数；单击“下载”超链接则可以打开加速库下载网址。

3）清除加速分析报告

单击“清除加速分析报告”菜单项，可以清除最近的扫描结果。