Debian RVA22 +V 移植计划

东东

目录

- RVA22 及 V 扩展介绍
- 项目介绍
- 未来扩展

RVA22 介绍

- RVA22 是 2021 底年提出的 RISC-V 架构平台规范
- 目的是简化开发流程，提高开发效率

特性	RVA20U64	RVA22U64
基本指令集	RV64I	RV64I
压缩指令（Compressed）	可选	必须
浮点扩展（Floating Point）	可选支持 F 和 D 扩展	必须支持 F 和 D 扩展
矢量扩展（Vector）	未定义	可选支持 V 扩展
密码学扩展（Crypto）	未定义	可选支持 Zbkb, Zbkc, Zbkx 扩展

向量扩展

是一种提高数据并行计算能力的机制

广泛应用于科学计算、机器学习、音视频处理和密码学加速等场景。

架构	向量扩展名称	特点
RISC-V	V (Vector)	支持动态向量长度，灵活性高，开源架构。
x86	SSE/AVX	固定长度寄存器（如 128 位、256 位、512 位）。
ARM	NEON/SVE	NEON（固定 128 位），SVE（可变向量长度）。

V 扩展

```
debian@debian-k1:~/openssl$ LD_LIBRARY_PATH=$PWD ./apps/openssl speed -evp chacha20
Doing ChaCha20 ops for 3s on 16 size blocks: 10643557 ChaCha20 ops in 2.99s
Doing ChaCha20 ops for 3s on 64 size blocks: 4352543 ChaCha20 ops in 3.00s
Doing ChaCha20 ops for 3s on 256 size blocks: 1410406 ChaCha20 ops in 3.00s
Doing ChaCha20 ops for 3s on 1024 size blocks: 726017 ChaCha20 ops in 3.00s
Doing ChaCha20 ops for 3s on 8192 size blocks: 99271 ChaCha20 ops in 3.00s
Doing ChaCha20 ops for 3s on 16384 size blocks: 51407 ChaCha20 ops in 3.00s
version: 3.5.0-dev
built on: Thu Nov 28 12:08:56 2024 UTC
options: bn(64,64)
compiler: gcc -fPIC -pthread -Wa,--noexecstack -Wall -O3 --no-doc -DOPENSSL_USE_NODELETE -DOPENSSL_PIC -DOPENSSL_BUILDING_OPENSSL -DNDEBUG
CPUINFO: OPENSSL_riscvcap=ZBA_ZBB_ZBC_ZBS_V
The 'numbers' are in 1000s of bytes per second processed.
type             16 bytes     64 bytes    256 bytes   1024 bytes   8192 bytes  16384 bytes
ChaCha20         56955.49k    92854.25k   120354.65k   247813.80k   271076.01k   280750.76k
debian@debian-k1:~/openssl$ LD_LIBRARY_PATH=$PWD OPENSSL_riscvcap=rv64gc ./apps/openssl speed -evp chacha20
Doing ChaCha20 ops for 3s on 16 size blocks: 10669633 ChaCha20 ops in 3.00s
Doing ChaCha20 ops for 3s on 64 size blocks: 4354698 ChaCha20 ops in 3.00s
Doing ChaCha20 ops for 3s on 256 size blocks: 1237762 ChaCha20 ops in 3.00s
Doing ChaCha20 ops for 3s on 1024 size blocks: 320589 ChaCha20 ops in 3.00s
Doing ChaCha20 ops for 3s on 8192 size blocks: 40523 ChaCha20 ops in 3.00s
Doing ChaCha20 ops for 3s on 16384 size blocks: 20265 ChaCha20 ops in 3.00s
version: 3.5.0-dev
built on: Thu Nov 28 12:08:56 2024 UTC
options: bn(64,64)
compiler: gcc -fPIC -pthread -Wa,--noexecstack -Wall -O3 --no-doc -DOPENSSL_USE_NODELETE -DOPENSSL_PIC -DOPENSSL_BUILDING_OPENSSL -DNDEBUG
CPUINFO: OPENSSL_riscvcap= env:rv64gc
The 'numbers' are in 1000s of bytes per second processed.
type             16 bytes     64 bytes    256 bytes   1024 bytes   8192 bytes  16384 bytes
ChaCha20         56904.71k    92900.22k   105622.36k   109427.71k   110654.81k   110673.92k
```

openssl 性能比较

V 扩展

```
Will use Hardware counter measured at 24.0 MHz emulating 3.0 GHz
Params database has 95 entries
Box64 with Dynarec v0.3.1 1ee4959c built on Nov 28 2024 00:20:52
BOX64: Didn't detect 48bits of address space, considering it's 39bits
Counted 32 Env var
BOX64 LIB PATH: BOX64 BIN PATH: ./:bin/:/usr/local/bin/:/usr/bin/:/bin/:/usr/local/games/:/usr/games/
Looking for ./dav1d
argv[1]="-i"
argv[2]="./Chimera-AV1-8bit-480x270-552kbps.ivf"
argv[3]="--muxer"
argv[4]="null"
argv[5]="--threads"
argv[6]="8"
Rename process to "dav1d"
Using emulated lib/libdav1d.so.7
Using native(wrapped) libc.so.6
Using native(wrapped) ld-linux-x86-64.so.2
Using native(wrapped) libpthread.so.0
Using native(wrapped) libdl.so.2
Using native(wrapped) libutil.so.1
Using native(wrapped) libresolv.so.2
Using native(wrapped) librt.so.1
Using native(wrapped) libbsd.so.0
dav1d 1.5.0-43-g767efec - by VideoLAN
Decoded 8929/8929 frames (100.0%) - 72.83/23.98 fps (3.04x)
debian@debian-k1:~/test$ BOX64_DYNAREC_RV64NOVEXT=vector box64 ./dav1d -i ./Chimera-AV1-8bit-480x270-552kbps.ivf --muxer null --threads 8
Dynarec for RISC-V With extension: I M A F D C Zba Zbb Zbc Zbs PageSize:4096 Running on - with 8 Cores
Will use Hardware counter measured at 24.0 MHz emulating 3.0 GHz
Params database has 95 entries
Box64 with Dynarec v0.3.1 1ee4959c built on Nov 28 2024 00:20:52
BOX64: Didn't detect 48bits of address space, considering it's 39bits
Counted 32 Env var
BOX64 LIB PATH: BOX64 BIN PATH: ./:bin/:/usr/local/bin/:/usr/bin/:/bin/:/usr/local/games/:/usr/games/
Looking for ./dav1d
argv[1]="-i"
argv[2]="./Chimera-AV1-8bit-480x270-552kbps.ivf"
argv[3]="--muxer"
argv[4]="null"
argv[5]="--threads"
argv[6]="8"
Rename process to "dav1d"
Using emulated lib/libdav1d.so.7
Using native(wrapped) libc.so.6
Using native(wrapped) ld-linux-x86-64.so.2
Using native(wrapped) libpthread.so.0
Using native(wrapped) libdl.so.2
Using native(wrapped) libutil.so.1
Using native(wrapped) libresolv.so.2
Using native(wrapped) librt.so.1
Using native(wrapped) libbsd.so.0
dav1d 1.5.0-43-g767efec - by VideoLAN
Decoded 8929/8929 frames (100.0%) - 21.58/23.98 fps (0.90x)[[C^[[C^[[B^[[B
debian@debian-k1:~/test$
```

box64 运行 dav1d 性能比较

项目介绍

- 操作系统：Debian
  - 流行发行版的基础系统
  - 用户众多
  - 社区活跃
- 编译模式
  - qemu-user + k1 同步编译

项目介绍

- 工具链介绍
  - gcc: 成功运行

gcc 运行截图

项目介绍

- 工具链介绍
  - llvm: 成功运行
  - 正在努力
    - Go
    - Rust

```
#define __clang_version__ "19.1.4 (1.1ruyisdk)"
#define __clang_wide_literal_encoding__ "UTF-32"
#define __llvm__ 1
#define __riscv 1
#define __riscv_a 2001000
#define __riscv_arch_test 1
#define __riscv_atomic 1
#define __riscv_c 2000000
#define __riscv_cmodel_medlow 1
#define __riscv_compressed 1
#define __riscv_d 2002000
#define __riscv_div 1
#define __riscv_f 2002000
#define __riscv_fdiv 1
#define __riscv_flen 64
#define __riscv_float_abi_double 1
#define __riscv_fsqrt 1
#define __riscv_i 2001000
#define __riscv_m 2000000
#define __riscv_misaligned_avoid 1
#define __riscv_mul 1
#define __riscv_muldiv 1
#define __riscv_v 1000000
#define __riscv_v_elen 64
#define __riscv_v_elen_fp 64
#define __riscv_v_intrinsic 12000
#define __riscv_v_min_vlen 128
#define __riscv_vector 1
#define __riscv_xlen 64
#define __riscv_zicsr 2000000
#define __riscv_zifencei 2000000
#define __riscv_zmmul 1000000
#define __riscv_zve32f 1000000
#define __riscv_zve32x 1000000
#define __riscv_zve64d 1000000
#define __riscv_zve64f 1000000
#define __riscv_zve64x 1000000
#define __riscv_zvl128b 1000000
#define __riscv_zvl32b 1000000
#define __riscv_zvl64b 1000000
```

llvm 扩展查看

项目介绍

- 后续目标(WIP
  - wanna-build
  - Reprepro
  - 自动化构建，失败人工干预
  - 性能测试

后续扩展和思考

- 使用不同的编译器编译同一个包，对比性能差异
- RVA23 先行者

谢谢观看

仓库晚点见

感谢 revy cyy ksco 于波等各位老师的大力支持