

Java on RISCV简介与毕昇JDK在HiFive Unleashed测试

PLCT-张定立

dingli@iscas.ac.cn

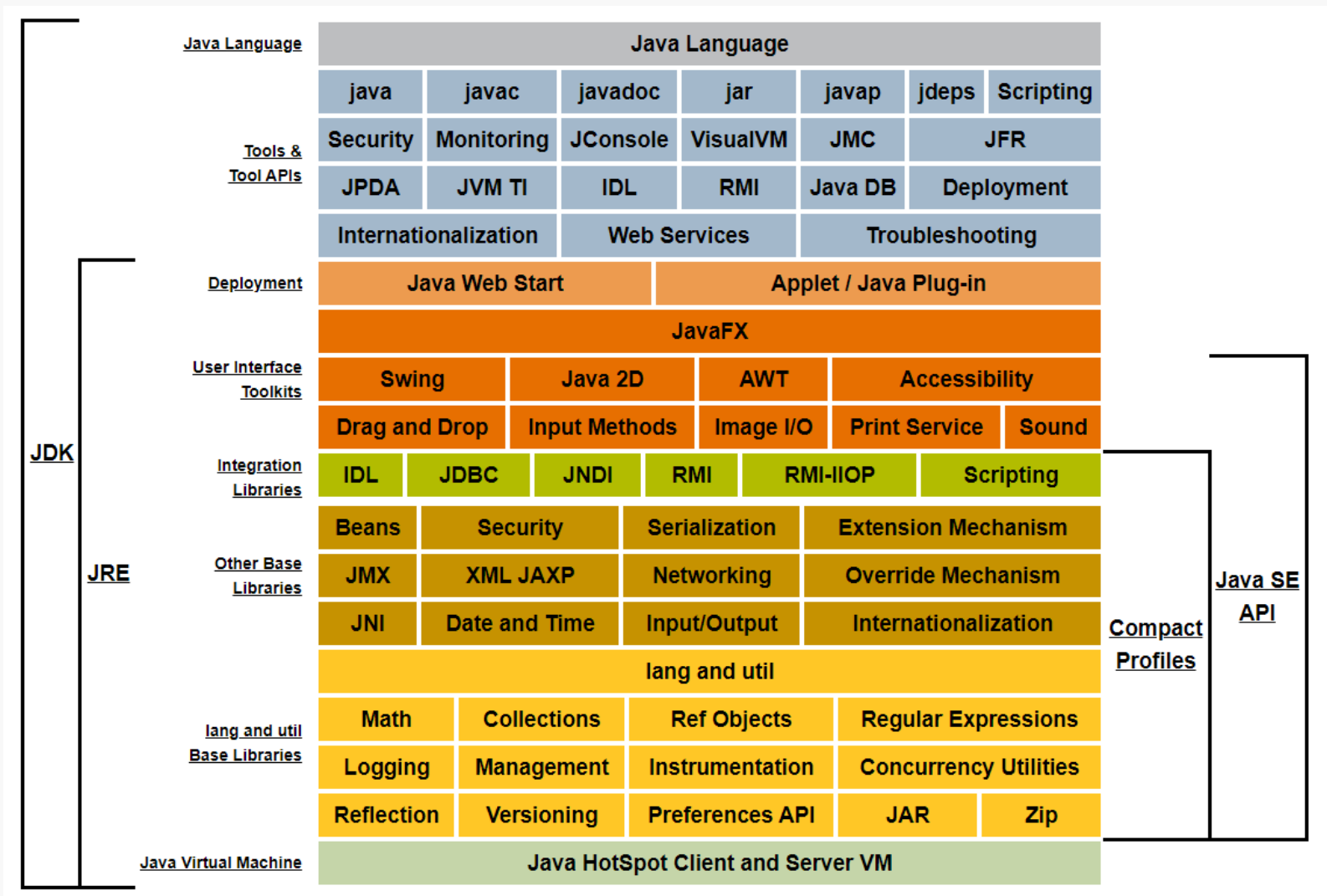
01 Java技术体系概述及RISC-V支持现状

02 OpenJDK移植进展

03 毕昇JDK for RV64G简介

04 HiFive Unleashed及毕昇JDK for RV64G调试

01 Java技术体系概述及RISC-V支持现状



Java技术体系

参考资料:

[1] <https://docs.oracle.com/javase/8/docs/>

01 Java技术体系概述及RISC-V支持现状

Name	Links	License	Maintainers
Maxine VM (Java Virtual Machine)	Upstream	GPLv2	Maxine team
Jikes RVM (Java Virtual Machine)	Upstream	Eclipse Public License (EPL)	Martin Maas (University of California, Berkeley)
OpenJDK/HotSpot (Java Virtual Machine)	?	?	Alexey Baturo, Michael Knysnek, Martin Maas
OpenJDK/OpenJ9 (Java Virtual Machine)	Upstream	Eclipse Public License 2.0 (EPLv2) with ClassPath Exception & Apache 2.0	Cheng Jin

RISC-V官方Github仓库中列出的Java software列表

参考资料:
[1] <https://github.com/riscv/riscv-software-list>

OpenJ9 移植进度和现状

- 移植基于OpenJDK11+OpenJ9+OMR, OpenJ9用以替代Hotspot
- 目前OpenJ9暂不支持针对RISCV64的JIT编译器, 在这种情况下, 无论是否在命令行上指定了-Xint选项, JDK默认都会以相同的输出结束。

```
[root@fedora-riscv linux]# /root/jdk/bin/java -version
openjdk version "11.0.9-internal" 2020-10-20
OpenJDK Runtime Environment (build 11.0.9-internal+0-adhoc.linux.openj9-openjdk-jdk11)
Eclipse OpenJ9 VM (build master-6648d4d, JRE 11 Linux riscv-64-Bit Compressed References 2
0200914_000000 (JIT disabled, AOT disabled)
OpenJ9      - 6648d4d
OMR         - 58b0c18
JCL         - based on jdk-11.0.9+5)
```



janvrany commented 4 days ago • edited

Author 😊 ...

Last week I managed to compile openj9-openjdk-11 with openj9 JIT component (`libj9jit29.so`) using CMake. It is just a skeleton, but it does compile and the VM starts (and segfaults in JIT thread because of lack of private linkage).

It is not much, but at least gives us a defined starting point that compiles.

Some changes to OMR were required, they're reasonably OK so I'll start upstreaming them this and next week.

I'd like to thank to @knn-k , @ChengJin01 and @dnakamura who helped me with this first little step.



1

参考资料:

[1] <https://github.com/eclipse/openj9/issues/11136>

02 OpenJDK移植进展

OpenJDK / jdk / jdk changeset 58699:6d2586f260a0

8199138: Add RISC-V support to Zero
Reviewed-by: erikj, stuefe

author glaubit
date Tue, 07 Apr 2020 08:45:01 +0200 (7 months ago)
parents 3f8d03880bf5
children 4fd2b4355e36
files make/autoconf/platform.m4 src/hotspot/os/linux/os_linux.cpp
diffstat 2 files changed, 15 insertions(+), 1 deletions(-) [+]

line diff

```
1.1 --- a/make/autoconf/platform.m4 Tue Apr 07 03:25:11 2020 +0000
1.2 +++ b/make/autoconf/platform.m4 Tue Apr 07 08:45:01 2020 +0200
1.3 @@ -120,6 +120,12 @@
1.4     VAR_CPU_BITS=64
1.5     VAR_CPU_ENDIAN=little
1.6     ;;
1.7 + riscv64)
1.8 +     VAR_CPU=riscv64
1.9 +     VAR_CPU_ARCH=riscv
1.10 +     VAR_CPU_BITS=64
1.11 +     VAR_CPU_ENDIAN=little
1.12 +     ;;
1.13     s390)
1.14         VAR_CPU=s390
1.15         VAR_CPU_ARCH=s390
1.16 @@ -485,6 +491,8 @@
1.17     HOTSPOT_${1_CPU_DEFINE}=S390
1.18     elif test "x$OPENJDK_${1_CPU}" = xs390x; then
1.19         HOTSPOT_${1_CPU_DEFINE}=S390
1.20 +     elif test "x$OPENJDK_${1_CPU}" = xriscv64; then
1.21 +         HOTSPOT_${1_CPU_DEFINE}=RISCV
1.22     elif test "x$OPENJDK_${1_CPU}" != x; then
1.23         HOTSPOT_${1_CPU_DEFINE}=$(echo $OPENJDK_${1_CPU} | tr a-z A-Z)
1.24     fi
```

bash configure \

--with-boot-jdk=/home/linux/opt/jdk-14 \

--with-jvm-variants=zero \

--openjdk-target=riscv64-unknown-linux-gnu \

--with-toolchain-path=riscv64-unknown-linux-gnu \

--with-sysroot=/home/linux/opt/fedora_mount \

--disable-warnings-as-errors

```
[root@fedora-riscv ~]# dnf search jdk
Last metadata expiration check: 1 day, 4:06:29 ago on Tue 03 Nov 2020 09:26:54 PM EST.
===== Name & Summary Matched: jdk =====
slf4j-jdk14.noarch : SLF4J JDK14 Binding
ldapjdk-javadoc.noarch : Javadoc for ldapjdk
java-11-openjdk-demo.riscv64 : OpenJDK Demos 11
java-1.8.0-openjdk-demo.riscv64 : OpenJDK Demos 8
java-11-openjdk-jmods.riscv64 : JMods for OpenJDK 11
java-11-openjdk-src.riscv64 : OpenJDK Source Bundle 11
java-11-openjdk.riscv64 : OpenJDK Runtime Environment 11
java-1.8.0-openjdk-src.riscv64 : OpenJDK Source Bundle 8
copy-jdk-configs.noarch : JDKs configuration files copier
java-1.8.0-openjdk.riscv64 : OpenJDK Runtime Environment 8
java-11-openjdk-javadoc.riscv64 : OpenJDK 11 API documentation
java-1.8.0-openjdk-javadoc.noarch : OpenJDK 8 API documentation
java-11-openjdk-devel.riscv64 : OpenJDK Development Environment 11
java-1.8.0-openjdk-devel.riscv64 : OpenJDK Development Environment 8
java-11-openjdk-headless.riscv64 : OpenJDK Headless Runtime Environment 11
java-1.8.0-openjdk-accessibility.riscv64 : OpenJDK 8 accessibility connector
java-1.8.0-openjdk-headless.riscv64 : OpenJDK Headless Runtime Environment 8
java-11-openjdk-javadoc-zip.riscv64 : OpenJDK 11 API documentation compressed in single archive
java-1.8.0-openjdk-javadoc-zip.noarch : OpenJDK 8 API documentation compressed in single archive
```

参考资料:

[1] <https://bugs.openjdk.java.net/browse/JDK-8199138>

03 毕昇JDK for RV64G构建

毕昇JDK简介

- BishengJDK 是基于 OpenJDK 的Arm64优化定制项目，由华为工程师维护，目前开源托管在 openEuler 项目下。
- 毕昇JDK RISC-V版本目前已在Gitee上开源：
<https://gitee.com/openeuler/bishengjdk-11>
- github上的镜像：
<https://github.com/isrc-cas/bishengjdk-11-mirror>

编译毕昇JDK

- 编译交叉工具链（gcc 推荐使用9.2版本）
- 安装额外的External Libraries：
 - Fedora for RISCV镜像挂载
 - 直接在进行编译的主机上编译安装riscv版本的库
- 获取源码及bootJDK并安装所需的基础软件
- 编译并构建：

```
$ bash configure openjdk-target=riscv64-unknown-linux-gnu \  
--disable-warnings-as-errors \  
--with-sysroot=/riscv/toolchain/sysroot \  
--x-includes=/riscv/toolchain/sysroot/usr/include \  
--x-libraries=/riscv/toolchain/sysroot/usr/lib \  
--with-boot-jdk=/path/to/boot/jdk \  
--with-debug-level=fastdebug \  
--with-native-debug-symbols=internal  
$ make images
```

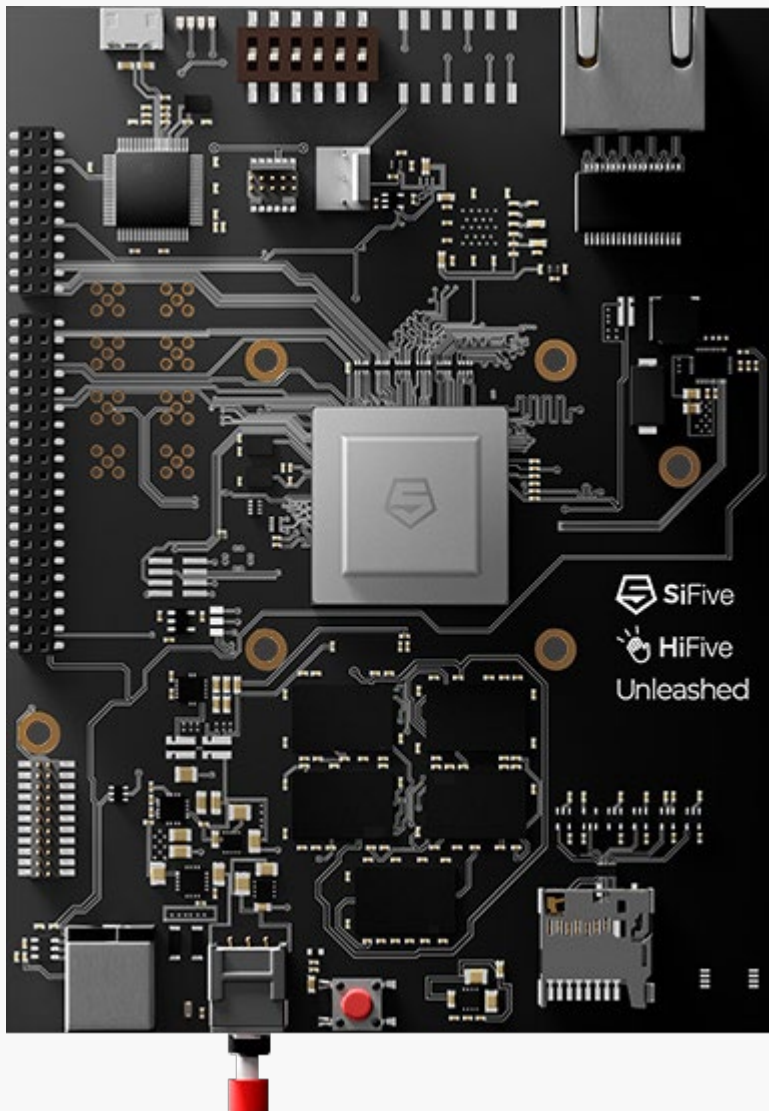
参考资料：

[1] https://gitee.com/openeuler/bishengjdk-11/blob/risc-v/DEPENDENCY_BUILD.md

[2] <https://github.com/azul-research/jdk-riscv/blob/riscv/dev-riscv/toolchain/Dockerfile>

04 HiFive Unleashed及毕昇JDK调试

开发板简介



```
[root@testing ~]# uname -a
Linux testing.example.com 5.5.0-0.rc5.git0.1.1.riscv64.fc32.riscv64 #1 SMP Mon Jan 6 17:31:22 UTC 2020
riscv64 riscv64 riscv64 GNU/Linux
[root@testing ~]# lscpu
Architecture:      riscv64
Byte Order:        Little Endian
CPU(s):            4
On-line CPU(s) list: 0-3
Thread(s) per core: 4
Core(s) per socket: 1
Socket(s):         1
CPU max MHz:       1400.0000
CPU min MHz:       350.0000
L1d cache:         32 KiB
L1i cache:         32 KiB
L2 cache:          2 MiB
[root@testing ~]# free -h
```

	total	used	free	shared	buff/cache	available
Mem:	7.8Gi	154Mi	7.3Gi	0.0Ki	342Mi	7.5Gi
Swap:	0B	0B	0B			

刷入系统

Freedom-u-sdk

```
xzcat demo-coreip-cli-freedom-u540.wic.xz |  
sudo dd of=/dev/sdX bs=512K iflag=fullblock  
oflag=direct conv=fsync status=progress
```

- 其中xzcat对应的是.xz后缀的镜像，如果采用的是.gz后缀的镜像则需要替换为zcat
- /dev/sdX为microSD卡的路径，请注意不能填写类似/dev/sdb1的子分区
- 镜像名称中带有xfce4包含了桌面环境，而带有cli则是不含桌面环境的镜像

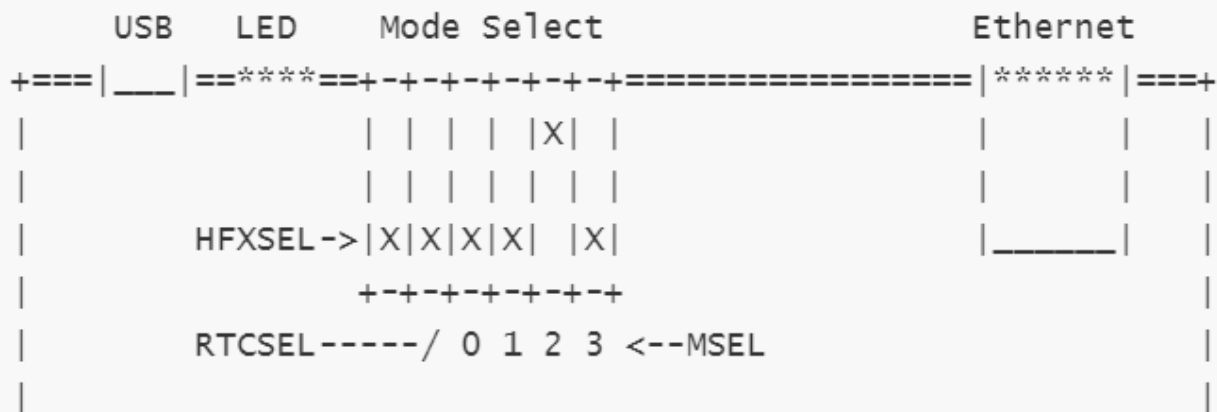
Fedora

```
sudo virt-builder \  
--source  
https://mirror.iscas.ac.cn/fedora/alt/risc-  
v/repo/virt-builder-images/images/index \  
--no-check-signature \  
--arch riscv64 \  
--format raw \  
--hostname testing.example.com \  
--output /dev/sdXXX \  
--root-password password:riscv \  
fedora-rawhide-developer-20200108.n.0
```

参考资料:

[1] <https://github.com/sifive/freedom-u-sdk#running-on-hardware>

[2] https://fedoraproject.org/wiki/Architectures/RISC-V/Installing#Install_on_the_HiFive_Unleashed_SD_card



- 启动之前需要修改MSEL以从SD卡使用FSBL和OpenSBI + U-Boot bootloaders来替代SPI-NOR闪存引导
- Fedora的镜像默认的OpenSSH守护进程配置文件遵循上游禁止root登录密码的默认设置，需要使用串口配置公钥或者修改/etc/ssh/sshd_config

参考资料:

[1] <https://github.com/sifive/freedom-u-sdk#running-on-hardware>

04 HiFive Unleashed及毕昇JDK调试

在HiFive Unleashed上使用gdb调试毕昇JDK11

- 获取源码,在开发板上执行:
\$ git clone -b risc-v <https://gitee.com/openeuler/bishengjdk-11.git>
- 之后进入到编译好的jdk的bin目录下, 执行:
[root@testing bin]# gdb --args java -version
- 执行如下命令来替换源码的路径:
(gdb) set substitute-path /path/to/host/bishengjdk-11/src
/path/to/hifive/bishengjdk-11/src
(gdb) b main
(gdb) r

04 HiFive Unleashed及毕昇JDK调试

毕昇JDK开源后从qemu到HiFive Unleashed的crash问题

```
[root@testing bin]# ./java -version
# To suppress the following error report, specify this argument
# after -XX: or in .hotspotrc: SuppressErrorAt=/constantPool.hpp:242
#
# A fatal error has been detected by the Java Runtime Environment:
#
# Internal Error (/home/zhangdingli/old_bisheng/bishengjdk-11/src/hotspot/share/oops/constantPool.hpp
:242), pid=992, tid=994
# Error: assert(is_invokedynamic_index(i)) failed
#
# JRE version: (11.0.8) (fastdebug build )
# Java VM: OpenJDK 64-Bit Server VM (fastdebug 11.0.8-internal+0-adhoc.zhangdingli.bishengjdk-11, mixe
d mode, tiered, compressed oops, gl gc, linux-riscv64)
# Core dump will be written. Default location: Core dumps may be processed with "/usr/lib/systemd/syst
emd-coredump %P %u %g %s %t %c %h" (or dumping to /root/jdk_old/jdk/bin/core.992)
#
# An error report file with more information is saved as:
# /root/jdk_old/jdk/bin/hs_err_pid992.log
#
#
Current thread is 994
Dumping core ...
Aborted (core dumped)
```

```
237 // Invokedynamic indexes.
238 // They must look completely different from normal indexes.
239 // The main reason is that byte swapping is sometimes done on normal indexes.
240 // Finally, it is helpful for debugging to tell the two apart.
241 static bool is_invokedynamic_index(int i) { return (i < 0); }
242 static int decode_invokedynamic_index(int i) { assert(is_invokedynamic_index(i), ""); return ~i; }
243 static int encode_invokedynamic_index(int i) { assert(!is_invokedynamic_index(i), ""); return ~i; }
```

04 HiFive Unleashed及毕昇JDK调试

毕昇JDK开源后从qemu到HiFive Unleashed的crash问题

```
(gdb) x/20i $pc -40
0x3ff78bc890 <Rewriter::patch_invokedynamic
0x3ff78bc894 <Rewriter::patch_invokedynamic
0x3ff78bc896 <Rewriter::patch_invokedynamic
0x3ff78bc89a <Rewriter::patch_invokedynamic
0x3ff78bc89e <Rewriter::patch_invokedynamic
0x3ff78bc8a2 <Rewriter::patch_invokedynamic
0x3ff78bc8a6 <Rewriter::patch_invokedynamic
0x3ff78bc8aa <Rewriter::patch_invokedynamic
0x3ff78bc8ae <Rewriter::patch_invokedynamic
0x3ff78bc8b0 <Rewriter::patch_invokedynamic
0x3ff78bc8b4 <Rewriter::patch_invokedynamic
=> 0x3ff78bc8b8 <Rewriter::patch_invokedynamic=>
    bgez      s4,0x3ff78bca68 <Rewriter::pat
0x3ff78bc8bc <Rewriter::patch_invokedynamic
0x3ff78bc8c0 <Rewriter::patch_invokedynamic
0x3ff78bc8c4 <Rewriter::patch_invokedynamic
0x3ff78bc8c8 <Rewriter::patch_invokedynamic
0x3ff78bc8cc <Rewriter::patch_invokedynamic
0x3ff78bc8d0 <Rewriter::patch_invokedynamic
0x3ff78bc8d4 <Rewriter::patch_invokedynamic
    bltz      a5,0x3ff78bcac2 <Rewriter::pat
0x3ff78bc8d8 <Rewriter::patch_invokedynamic

(gdb) i r s3 s4
s3      0x3fd8bd57bb      274219227067
s4      0xfffffee8       4294967016
(gdb) p/a *($s3)
$17 = 0xfffffffffffee8

(gdb) x/20i $pc -40
0x3ff78bc890 <Rewriter::patch_invokedynamic_bytecodes()+162>:
0x3ff78bc894 <Rewriter::patch_invokedynamic_bytecodes()+166>:
0x3ff78bc896 <Rewriter::patch_invokedynamic_bytecodes()+168>:
0x3ff78bc89a <Rewriter::patch_invokedynamic_bytecodes()+172>:
0x3ff78bc89e <Rewriter::patch_invokedynamic_bytecodes()+176>:
0x3ff78bc8a2 <Rewriter::patch_invokedynamic_bytecodes()+180>:
0x3ff78bc8a6 <Rewriter::patch_invokedynamic_bytecodes()+184>:
0x3ff78bc8aa <Rewriter::patch_invokedynamic_bytecodes()+188>:
0x3ff78bc8ae <Rewriter::patch_invokedynamic_bytecodes()+192>:
0x3ff78bc8b0 <Rewriter::patch_invokedynamic_bytecodes()+194>:
0x3ff78bc8b4 <Rewriter::patch_invokedynamic_bytecodes()+198>:
0x3ff78bc8b8 <Rewriter::patch_invokedynamic_bytecodes()+202>:
0x3ff78bc8bc <Rewriter::patch_invokedynamic_bytecodes()+206>:
0x3ff78bc8c0 <Rewriter::patch_invokedynamic_bytecodes()+210>:
0x3ff78bc8c4 <Rewriter::patch_invokedynamic_bytecodes()+214>:
0x3ff78bc8c8 <Rewriter::patch_invokedynamic_bytecodes()+218>:
0x3ff78bc8cc <Rewriter::patch_invokedynamic_bytecodes()+222>:
0x3ff78bc8d0 <Rewriter::patch_invokedynamic_bytecodes()+226>:
0x3ff78bc8d4 <Rewriter::patch_invokedynamic_bytecodes()+230>:
0x3ff78bc8d8 <Rewriter::patch_invokedynamic_bytecodes()+234>:

(gdb) i r s3 s4
s3      0x3fd84ce7bb      274211858363
s4      0xfffffffffffee8      -280
(gdb) p/a *($s3)
$17 = 0xfffffffffffee8
```

```
li      a1,244
mv      a0,s10
auipc   ra,0xff736
jalr    592(ra)
auipc   ra,0xffec6
jalr    2008(ra)
ld      a5,48(s3)
slli    a4,s9,0x3
add      a5,a5,a4
ld      s3,0(a5)
lw      s4,0(s3)

li      a5,-144(s0)
not      s4,s4
addw     s5,a5,s4
sext.w   a5,s5
sd      a5,-120(s0)
slli     a5,s5,0x20
not      s5,s5
```


04 HiFive Unleashed及毕昇JDK调试

地址对齐

[src/hotspot/cpu/riscv64/bytes_aarch64.hpp](#)

```
26 #ifndef CPU_AARCH64_VM_BYTES_AARCH64_HPP
27 #define CPU_AARCH64_VM_BYTES_AARCH64_HPP
28
29 #include "memory/allocation.hpp"
30
31 class Bytes: AllStatic {
32 public:
33     // Efficient reading and writing of unaligned unsigned data in platform-specific byte ordering
34     // (no special code is needed since x86 CPUs can access unaligned data)
35     static inline u2  get_native_u2(address p)      { return *(u2*)p; }
36     static inline u4  get_native_u4(address p)      { return *(u4*)p; }
37     static inline u8  get_native_u8(address p)      { return *(u8*)p; }
38
39     static inline void put_native_u2(address p, u2 x) { *(u2*)p = x; }
40     static inline void put_native_u4(address p, u4 x) { *(u4*)p = x; }
41     static inline void put_native_u8(address p, u8 x) { *(u8*)p = x; }
```

[src/hotspot/cpu/riscv64/bytes_riscv64.hpp](#)

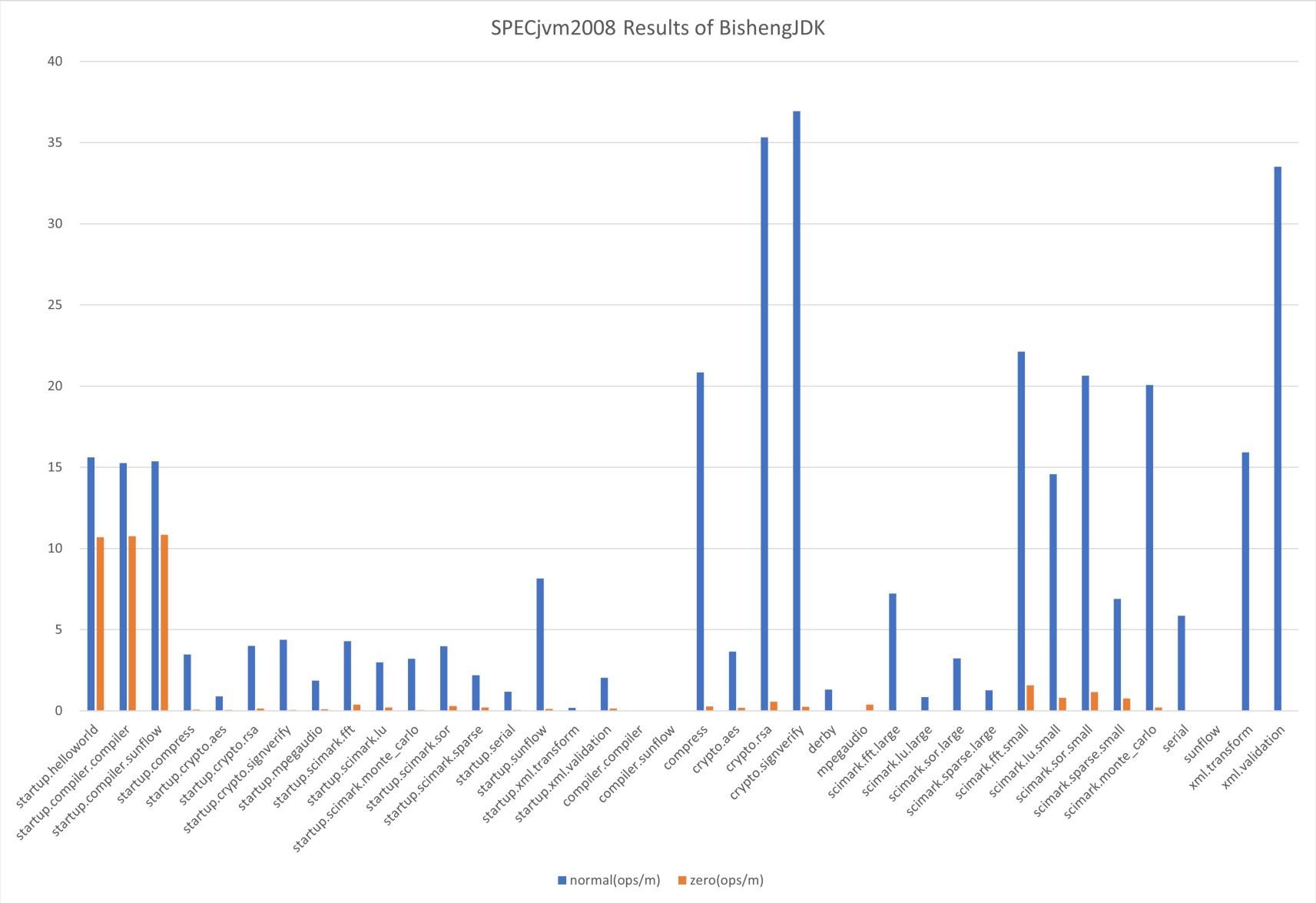
```
41 + static inline u2 get_native_u2(address p) {
42 +     if ((intptr_t(p) & 1) == 0) {
43 +         return *(u2*)p;
44 +     } else {
45 +         return ((u2)(p[1]) << 8) |
46 +             ((u2)(p[0]));
47 +     }
48 + }
49
89 + static inline void put_native_u2(address p, u2 x) {
90 +     if ((intptr_t(p) & 1) == 0) {
91 +         *(u2*)p = x;
92 +     } else {
93 +         p[1] = x >> 8;
94 +         p[0] = x;
95 +     }
96 + }
97 +
```

参考资料:

[1] <https://gitee.com/openeuler/bishengjdk-11/pulls/15/files.patch>

04 HiFive Unleashed及毕昇JDK调试

benchmark



谢 谢

欢迎关注知乎专栏

Java on RISC-V : 让RISC-V生态可以用上工业级的Java应用
https://www.zhihu.com/column/c_1287750038518161408