Homework4: SPECjvm2008 基准测试

杨浩然 10195501441

操作系统： ubuntu 20.04

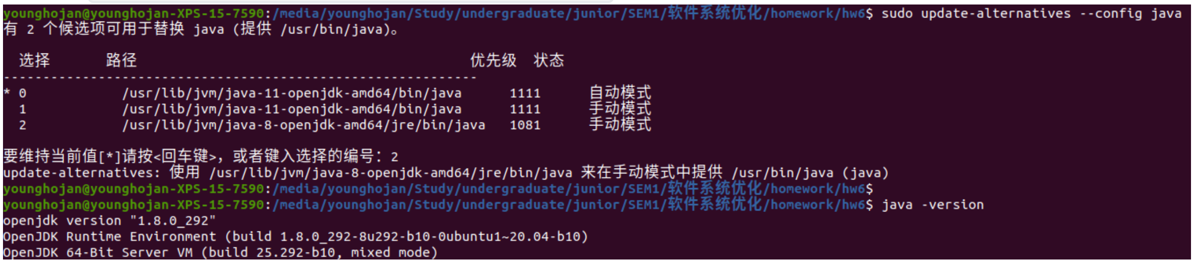
2021-11-15

访问 SPECjvm2008 官方网站，阅读 User’s Guide、Run and Report Rules、Known Issues、FAQ 四份文档，了解 SPEC 组织和一个标准的基准测试集的基本情况

下载和安装 SPECjvm2008，根据上述文档的指引，跑一次完整的基准测试（大约 2~3 小时），记录安装和测试的过程和结果，了解一个标准的基准测试集的基本执行过程

安装 JDK 8

先使用 `sudo update-alternatives –config java` 查看已安装的 JDK:



```
younghojan@younghojan-XPS-15-7590:/media/younghojan/Study/undergraduate/junior/SEM1/软件系统优化/homework/hw6$ sudo update-alternatives --config java
有 2 个候选项可用于替换 java (提供 /usr/bin/java)。

  选择       路径                                          优先级  状态
------------------------------------------------------------
* 0            /usr/lib/jvm/java-11-openjdk-amd64/bin/java      1111       自动模式
  1            /usr/lib/jvm/java-11-openjdk-amd64/bin/java      1111       手动模式
  2            /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java   1081       手动模式

要维持当前值[*]请按<回车键>，或者键入选择的编号: 2
update-alternatives: 使用 /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java 来在手动模式中提供 /usr/bin/java (java)
younghojan@younghojan-XPS-15-7590:/media/younghojan/Study/undergraduate/junior/SEM1/软件系统优化/homework/hw6$
younghojan@younghojan-XPS-15-7590:/media/younghojan/Study/undergraduate/junior/SEM1/软件系统优化/homework/hw6$ java -version
openjdk version "1.8.0_292"
OpenJDK Runtime Environment (build 1.8.0_292-8u292-b10-0ubuntu1~20.04-b10)
OpenJDK 64-Bit Server VM (build 25.292-b10, mixed mode)
```

貌似已经切换到 OpenJDK 1.9.0_292，但是与 OpenJDK 1.8.0_41 不完全一致，保险起见，还是多安装一个 OpenJDK 1.8.0_41：

- 下载 OpenJDK 1.8.0_41 RI Binaries
- 解压到 /usr/lib/jvm
- 修改环境变量

  在 ~/.bashrc 中添加：

  ```
  # set oracle jdk environment(for SPECjvm2008 use)
  export JAVA_HOME=/usr/lib/jvm/java-se-8u41-ri
  export JRE_HOME=${JAVA_HOME}/jre
  export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
  export PATH=${JAVA_HOME}/bin:$PATH
  ```

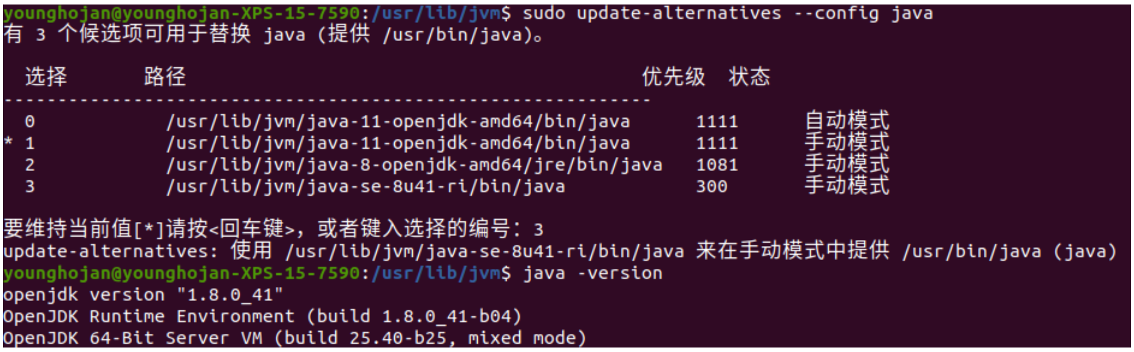
  使用 `source ~/.bashrc` 使环境变量的修改马上生效

- 系统注册此 JDK

  ```
  sudo update-alternatives --install /usr/bin/java java /usr/lib/jvm/java-se-8u41-ri/bin/java 300
  ```

- 因为安装了多个 JDK，需要切换至 OpenJDK 1.8.0_41

```
sudo update-alternatives --config java
```



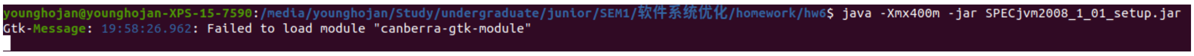
```
younghojan@younghojan-XPS-15-7590:/usr/lib/jvm$ sudo update-alternatives --config java
有 3 个候选项可用于替换 java (提供 /usr/bin/java)。

  选择       路径                                          优先级  状态
------------------------------------------------------------
  0            /usr/lib/jvm/java-11-openjdk-amd64/bin/java      1111      自动模式
* 1            /usr/lib/jvm/java-11-openjdk-amd64/bin/java      1111      手动模式
  2            /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java   1081      手动模式
  3            /usr/lib/jvm/java-se-8u41-ri/bin/java            300       手动模式

要维持当前值[*]请按<回车键>，或者键入选择的编号：3
update-alternatives: 使用 /usr/lib/jvm/java-se-8u41-ri/bin/java 来在手动模式中提供 /usr/bin/java (java)
younghojan@younghojan-XPS-15-7590:/usr/lib/jvm$ java -version
openjdk version "1.8.0_41"
OpenJDK Runtime Environment (build 1.8.0_41-b04)
OpenJDK 64-Bit Server VM (build 25.40-b25, mixed mode)
```

Running the benchmark

安装

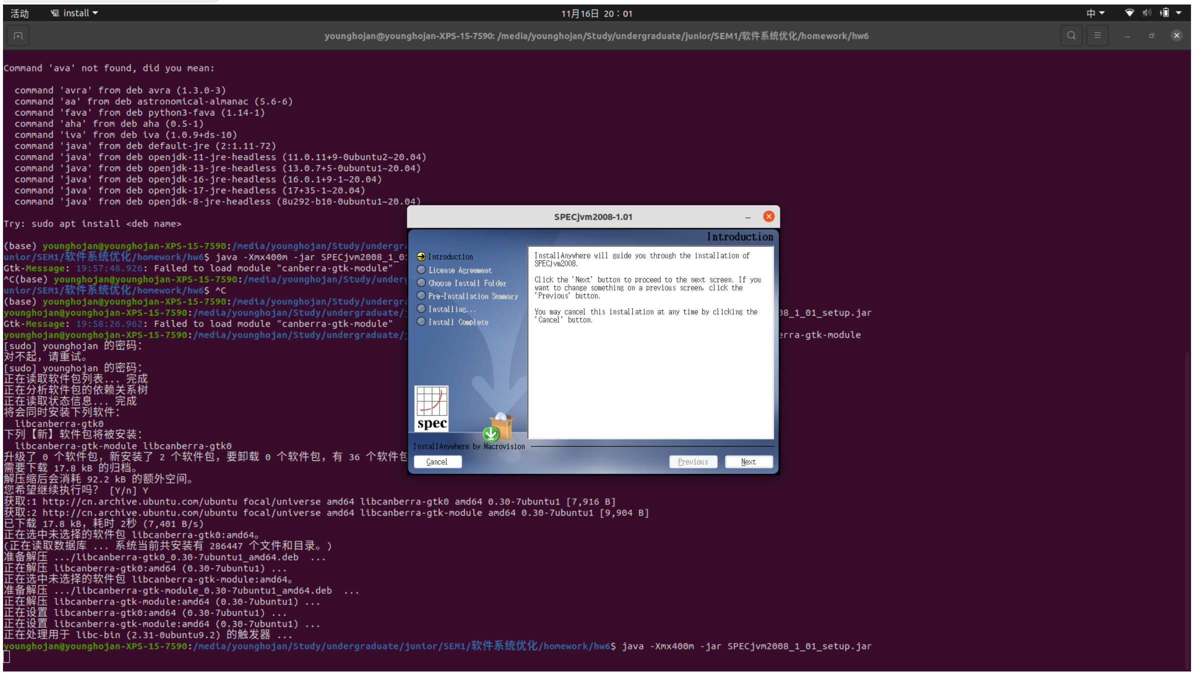
使用 `java -Xmx400m -jar SPECjvm2008.jar` 命令，这个命令的使用方法为

```
java [<jvm options>] -jar SPECjvm2008.jar [<SPECjvm2008 options>] [<benchmark name> ...]
```



```
younghojan@younghojan-XPS-15-7590:/media/younghojan/Study/undergraduate/junior/SEM1/软件系统优化/homework/hw6$ java -Xmx400m -jar SPECjvm2008_1_01_setup.jar
Gtk-Message: 19:58:26.962: Failed to load module "canberra-gtk-module"
```

好像差点东西…用 `apt-get install` 安装一下 canberra-gtk-module，然后再 `java -Xmx400m -jar SPECjvm2008.jar`：



InstallAnywhere will guide you through the installation of SPECjvm2008.

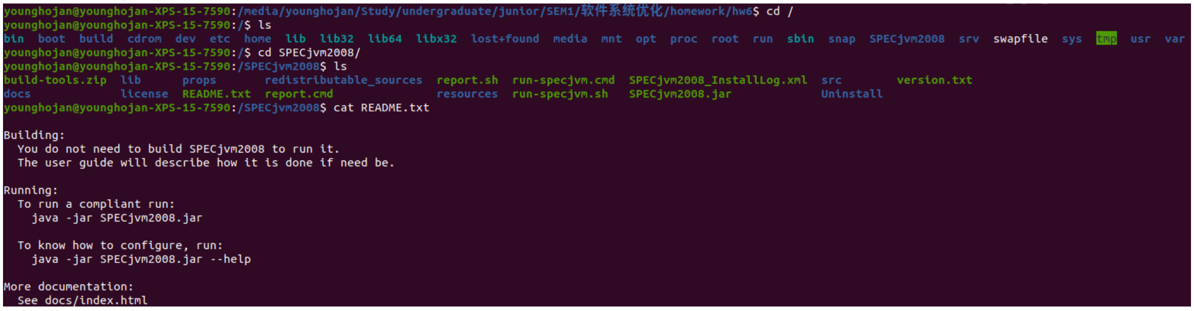
Click the 'Next' button to proceed to the next screen. If you want to change something on a previous screen, click the 'Previous' button.

You may cancel this installation at any time by clicking the 'Cancel' button.

```
Command 'ava' not found, did you mean:

  command 'avra' from deb avra (1.3.0-3)
  command 'aa' from deb astronomical-almanac (5.6-6)
  command 'fava' from deb python3-fava (1.14-1)
  command 'aha' from deb aha (0.5-1)
  command 'iva' from deb iva (1.0.9+ds-10)
  command 'java' from deb default-jre (2:1.11-72)
  command 'java' from deb openjdk-11-jre-headless (11.0.11+9-0ubuntu2~20.04)
  command 'java' from deb openjdk-13-jre-headless (13.0.7+5-0ubuntu1~20.04)
  command 'java' from deb openjdk-16-jre-headless (16.0.1+9-1~20.04)
  command 'java' from deb openjdk-17-jre-headless (17+35-1~20.04)
  command 'java' from deb openjdk-8-jre-headless (8u292-b10-0ubuntu1~20.04)

Try: sudo apt install <deb name>

(base) younghojan@younghojan-XPS-15-7590:/media/younghojan/Study/undergra
unior/SEM1/软件系统优化/homework/hw6$ java -Xmx400m -jar SPECjvm2008_1_01
Gtk-Message: 19:57:48.926: Failed to load module "canberra-gtk-module"
^C(base) younghojan@younghojan-XPS-15-7590:/media/younghojan/Study/underg
unior/SEM1/软件系统优化/homework/hw6$ ^C
(base) younghojan@younghojan-XPS-15-7590:/media/younghojan/Study/undergra
younghojan@younghojan-XPS-15-7590:/media/younghojan/Study/undergraduate/                      08_1_01_setup.jar
Gtk-Message: 19:58:26.962: Failed to load module "canberra-gtk-module"
younghojan@younghojan-XPS-15-7590:/media/younghojan/Study/undergraduate/                      erra-gtk-module
[sudo] younghojan 的密码：
对不起，请重试。
[sudo] younghojan 的密码：
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
将会同时安装下列软件：
  libcanberra-gtk0
下列【新】软件包将被安装：
  libcanberra-gtk-module libcanberra-gtk0
升级了 0 个软件包，新安装了 2 个软件包，要卸载 0 个软件包，有 36 个软件包
需要下载 17.8 kB 的归档。
解压缩后会消耗 92.2 kB 的额外空间。
您希望继续执行吗？ [Y/n] Y
获取:1 http://cn.archive.ubuntu.com/ubuntu focal/universe amd64 libcanberra-gtk0 amd64 0.30-7ubuntu1 [7,916 B]
获取:2 http://cn.archive.ubuntu.com/ubuntu focal/universe amd64 libcanberra-gtk-module amd64 0.30-7ubuntu1 [9,904 B]
已下载 17.8 kB，耗时 2秒 (7,401 B/s)
正在选中未选择的软件包 libcanberra-gtk0:amd64。
(正在读取数据库 ... 系统当前共安装有 286447 个文件和目录。)
准备解压 .../libcanberra-gtk0_0.30-7ubuntu1_amd64.deb  ...
正在解压 libcanberra-gtk0:amd64 (0.30-7ubuntu1) ...
正在选中未选择的软件包 libcanberra-gtk-module:amd64。
准备解压 .../libcanberra-gtk-module_0.30-7ubuntu1_amd64.deb  ...
正在解压 libcanberra-gtk-module:amd64 (0.30-7ubuntu1) ...
正在设置 libcanberra-gtk0:amd64 (0.30-7ubuntu1) ...
正在设置 libcanberra-gtk-module:amd64 (0.30-7ubuntu1) ...
正在处理用于 libc-bin (2.31-0ubuntu9.2) 的触发器 ...
younghojan@younghojan-XPS-15-7590:/media/younghojan/Study/undergraduate/junior/SEM1/软件系统优化/homework/hw6$ java -Xmx400m -jar SPECjvm2008_1_01_setup.jar
```

弹出来了安装的对话框，一路 next（注意要用 `sudo`，否则没有权限来选择安装路径）。

现在 SPECjvm2008 已经安装在了 / 目录下，打开 README.txt 看看：

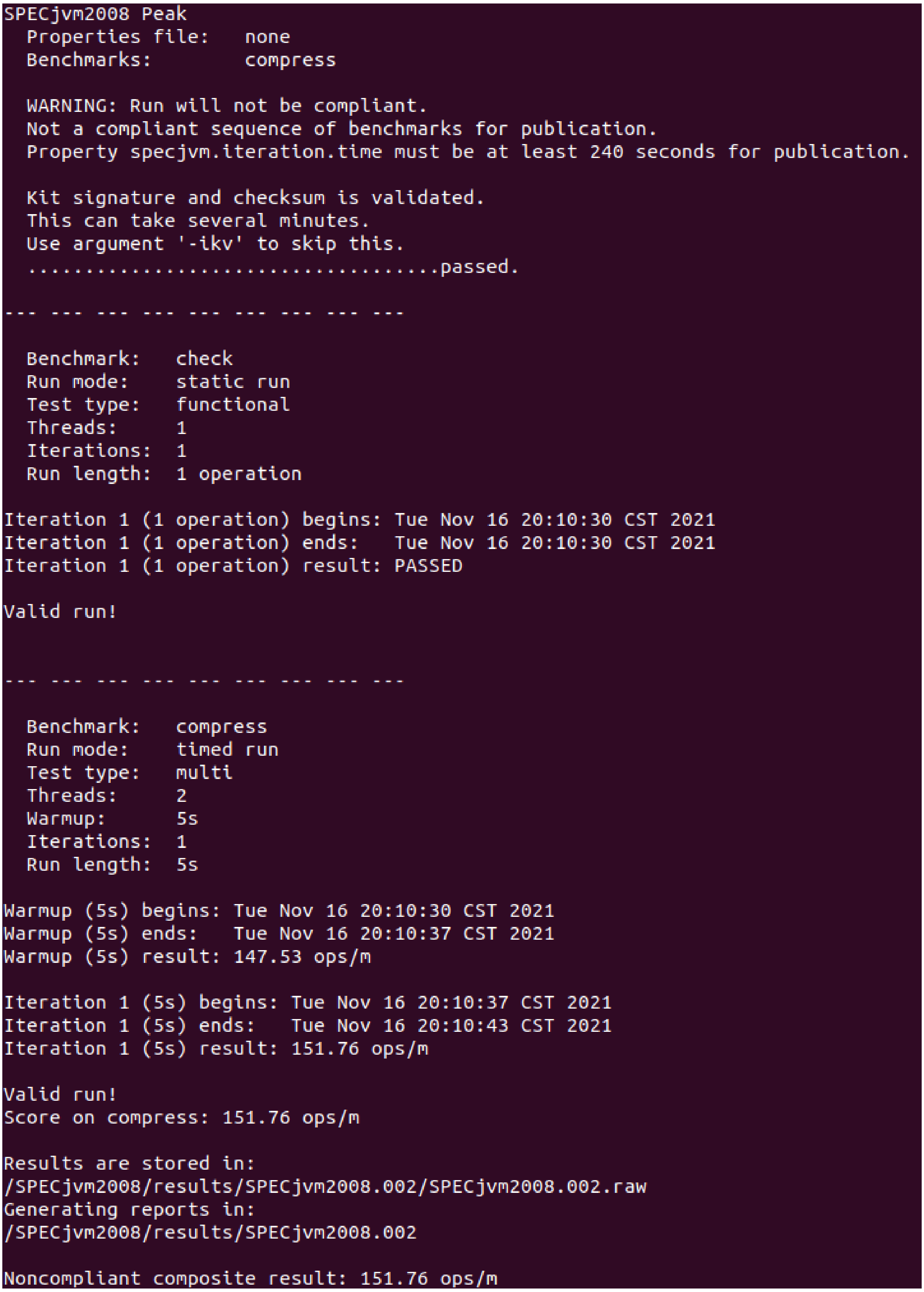


```
younghojan@younghojan-XPS-15-7590:/media/younghojan/Study/undergraduate/junior/SEM1/软件系统优化/homework/hw6$ cd /
younghojan@younghojan-XPS-15-7590:/$ ls
bin  boot  build  cdrom  dev  etc  home  lib  lib32  lib64  libx32  lost+found  media  mnt  opt  proc  root  run  sbin  snap  SPECjvm2008  srv  swapfile  sys  tmp  usr  var
younghojan@younghojan-XPS-15-7590:/$ cd SPECjvm2008/
younghojan@younghojan-XPS-15-7590:/SPECjvm2008$ ls
build-tools.zip  lib      props       redistributable_sources  report.sh  run-specjvm.cmd  SPECjvm2008_InstallLog.xml  src        version.txt
docs             license  README.txt  report.cmd               resources  run-specjvm.sh   SPECjvm2008.jar             Uninstall
younghojan@younghojan-XPS-15-7590:/SPECjvm2008$ cat README.txt

Building:
  You do not need to build SPECjvm2008 to run it.
  The user guide will describe how it is done if need be.

Running:
  To run a compliant run:
    java -jar SPECjvm2008.jar

  To know how to configure, run:
    java -jar SPECjvm2008.jar --help

More documentation:
  See docs/index.html
```

trial run

根据 userguide，先用 trial run 的命令来 check installation：

```
java -jar SPECjvm2008.jar -wt 5s -it 5s -bt 2 compress
```

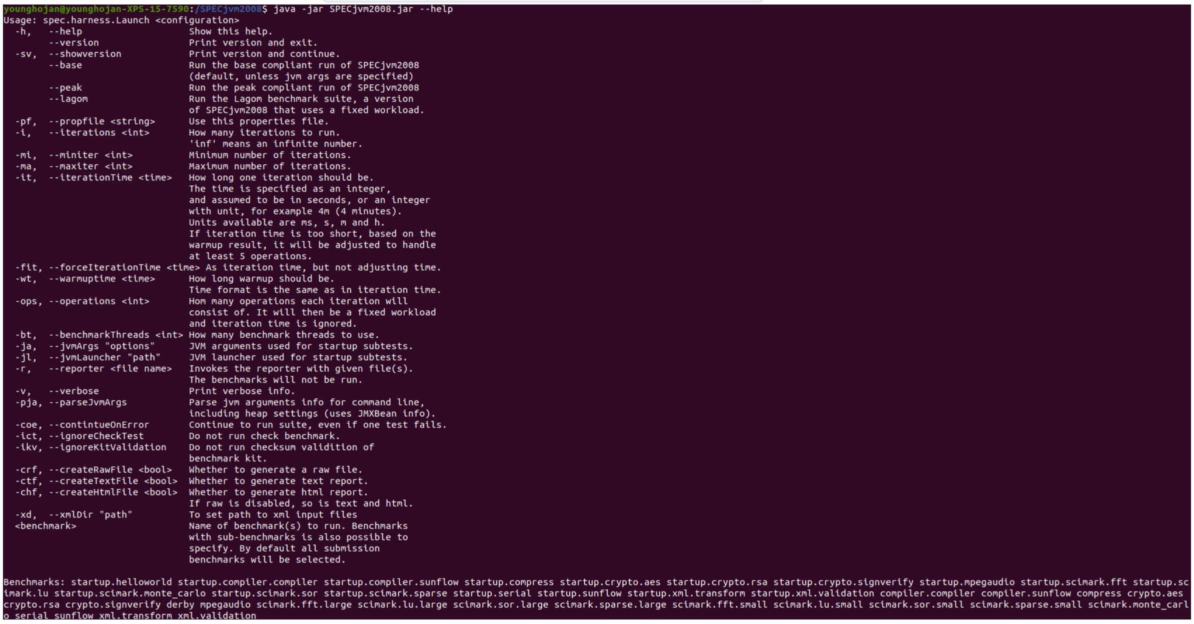


```
SPECjvm2008 Peak
  Properties file:   none
  Benchmarks:        compress

  WARNING: Run will not be compliant.
  Not a compliant sequence of benchmarks for publication.
  Property specjvm.iteration.time must be at least 240 seconds for publication.

  Kit signature and checksum is validated.
  This can take several minutes.
  Use argument '-ikv' to skip this.
  ....................................passed.

--- --- --- --- --- --- --- --- ---

  Benchmark:   check
  Run mode:    static run
  Test type:   functional
  Threads:     1
  Iterations:  1
  Run length:  1 operation

Iteration 1 (1 operation) begins: Tue Nov 16 20:10:30 CST 2021
Iteration 1 (1 operation) ends:   Tue Nov 16 20:10:30 CST 2021
Iteration 1 (1 operation) result: PASSED

Valid run!


--- --- --- --- --- --- --- --- ---

  Benchmark:   compress
  Run mode:    timed run
  Test type:   multi
  Threads:     2
  Warmup:      5s
  Iterations:  1
  Run length:  5s

Warmup (5s) begins: Tue Nov 16 20:10:30 CST 2021
Warmup (5s) ends:   Tue Nov 16 20:10:37 CST 2021
Warmup (5s) result: 147.53 ops/m

Iteration 1 (5s) begins: Tue Nov 16 20:10:37 CST 2021
Iteration 1 (5s) ends:   Tue Nov 16 20:10:43 CST 2021
Iteration 1 (5s) result: 151.76 ops/m

Valid run!
Score on compress: 151.76 ops/m

Results are stored in:
/SPECjvm2008/results/SPECjvm2008.002/SPECjvm2008.002.raw
Generating reports in:
/SPECjvm2008/results/SPECjvm2008.002

Noncompliant composite result: 151.76 ops/m
```

一切正常！下面来正式 run 一次。

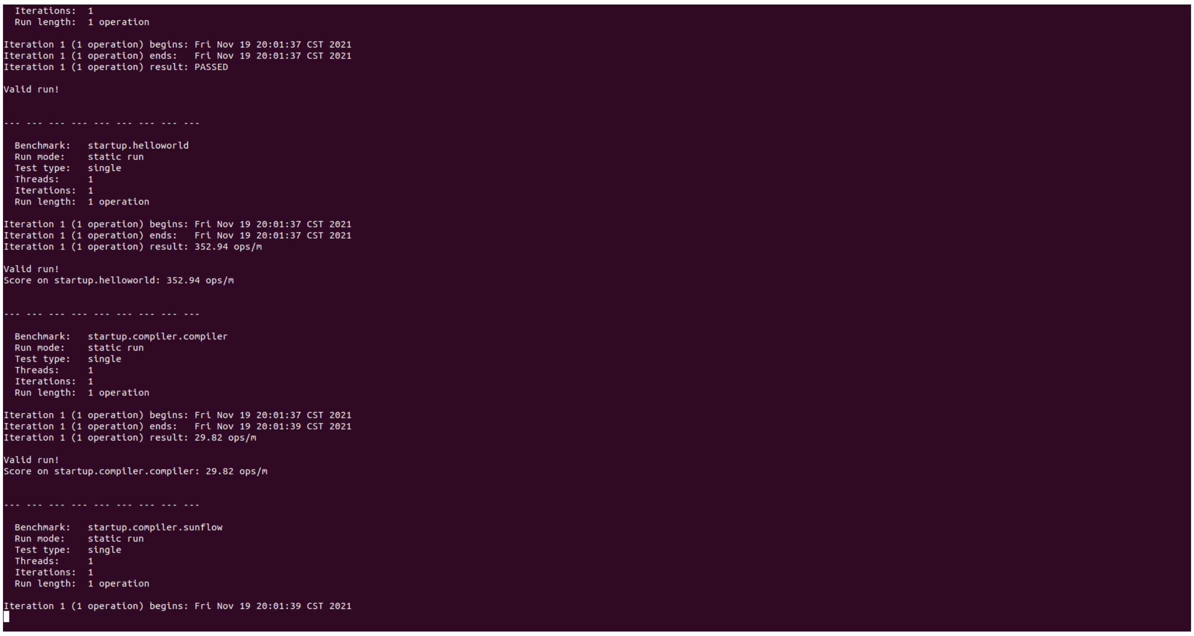
run

作业要求“只需运行 Base 类别，无需运行 Peak 类别，请注意 Base 类别运行的要求”，所以在 run 的时候需要注意参数设置。使用 `java -jar SPECjvm2008.jar --help` 来查看参数的设置规范：



```
younghojan@younghojan-XPS-15-7590:/SPECjvm2008$ java -jar SPECjvm2008.jar --help
Usage: spec.harness.Launch <configuration>
  -h,   --help                    Show this help.
        --version                 Print version and exit.
  -sv,  --showversion             Print version and continue.
        --base                    Run the base compliant run of SPECjvm2008
                                  (default, unless jvm args are specified)
        --peak                    Run the peak compliant run of SPECjvm2008
        --lagom                   Run the Lagom benchmark suite, a version
                                  of SPECjvm2008 that uses a fixed workload.
  -pf,  --propfile <string>       Use this properties file.
  -i,   --iterations <int>        How many iterations to run.
                                  'inf' means an infinite number.
  -mi,  --miniter <int>           Minimum number of iterations.
  -ma,  --maxiter <int>           Maximum number of iterations.
  -it,  --iterationTime <time>    How long one iteration should be.
                                  The time is specified as an integer,
                                  and assumed to be in seconds, or an integer
                                  with unit, for example 4m (4 minutes).
                                  Units available are ms, s, m and h.
                                  If iteration time is too short, based on the
                                  warmup result, it will be adjusted to handle
                                  at least 5 operations.
  -fit, --forceIterationTime <time> As iteration time, but not adjusting time.
  -wt,  --warmuptime <time>       How long warmup should be.
                                  Time format is the same as in iteration time.
  -ops, --operations <int>        Hom many operations each iteration will
                                  consist of. It will then be a fixed workload
                                  and iteration time is ignored.
  -bt,  --benchmarkThreads <int> How many benchmark threads to use.
  -ja,  --jvmArgs "options"       JVM arguments used for startup subtests.
  -jl,  --jvmLauncher "path"      JVM launcher used for startup subtests.
  -r,   --reporter <file name>    Invokes the reporter with given file(s).
                                  The benchmarks will not be run.
  -v,   --verbose                 Print verbose info.
  -pja, --parseJvmArgs            Parse jvm arguments info for command line,
                                  including heap settings (uses JMXBean info).
  -coe, --contintueOnError        Continue to run suite, even if one test fails.
  -ict, --ignoreCheckTest         Do not run check benchmark.
  -ikv, --ignoreKitValidation     Do not run checksum validition of
                                  benchmark kit.
  -crf, --createRawFile <bool>    Whether to generate a raw file.
  -ctf, --createTextFile <bool>   Whether to generate text report.
  -chf, --createHtmlFile <bool>   Whether to generate html report.
                                  If raw is disabled, so is text and html.
  -xd,  --xmlDir "path"           To set path to xml input files
  <benchmark>                     Name of benchmark(s) to run. Benchmarks
                                  with sub-benchmarks is also possible to
                                  specify. By default all submission
                                  benchmarks will be selected.

Benchmarks: startup.helloworld startup.compiler.compiler startup.compiler.sunflow startup.compress startup.crypto.aes startup.crypto.rsa startup.crypto.signverify startup.mpegaudio startup.scimark.fft startup.sc
imark.lu startup.scimark.monte_carlo startup.scimark.sor startup.scimark.sparse startup.serial startup.sunflow startup.xml.transform startup.xml.validation compiler.compiler compiler.sunflow compress crypto.aes
crypto.rsa crypto.signverify derby mpegaudio scimark.fft.large scimark.lu.large scimark.sor.large scimark.sparse.large scimark.fft.small scimark.lu.small scimark.sor.small scimark.sparse.small scimark.monte_carl
o serial sunflow xml.transform xml.validation
```

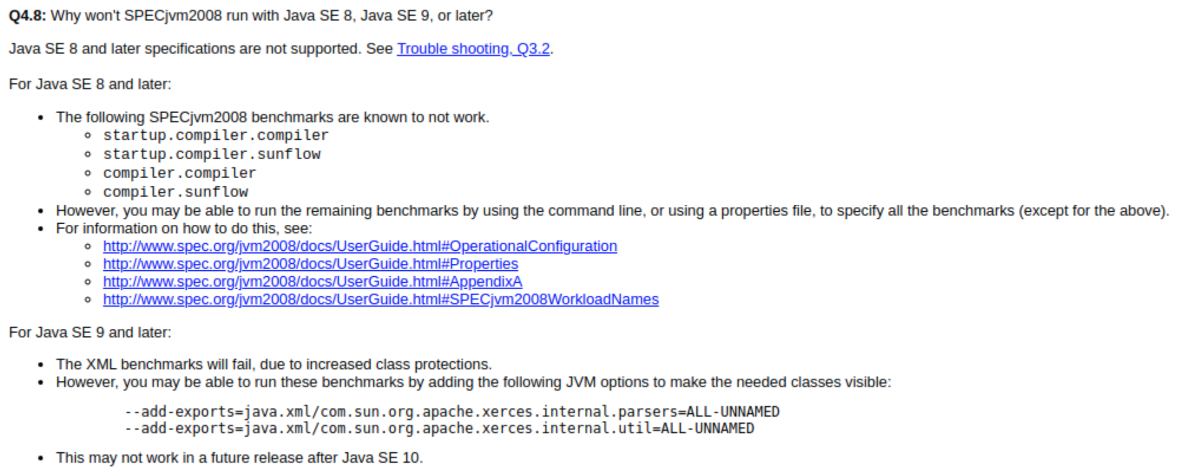
用 `sudo java -jar SPECjvm2008.jar --base` 试试。



```
  Iterations:  1
  Run length:  1 operation

Iteration 1 (1 operation) begins: Fri Nov 19 20:01:37 CST 2021
Iteration 1 (1 operation) ends:   Fri Nov 19 20:01:37 CST 2021
Iteration 1 (1 operation) result: PASSED

Valid run!


--- --- --- --- --- --- --- --- ---

  Benchmark:   startup.helloworld
  Run mode:    static run
  Test type:   single
  Threads:     1
  Iterations:  1
  Run length:  1 operation

Iteration 1 (1 operation) begins: Fri Nov 19 20:01:37 CST 2021
Iteration 1 (1 operation) ends:   Fri Nov 19 20:01:37 CST 2021
Iteration 1 (1 operation) result: 352.94 ops/m

Valid run!
Score on startup.helloworld: 352.94 ops/m


--- --- --- --- --- --- --- --- ---

  Benchmark:   startup.compiler.compiler
  Run mode:    static run
  Test type:   single
  Threads:     1
  Iterations:  1
  Run length:  1 operation

Iteration 1 (1 operation) begins: Fri Nov 19 20:01:37 CST 2021
Iteration 1 (1 operation) ends:   Fri Nov 19 20:01:39 CST 2021
Iteration 1 (1 operation) result: 29.82 ops/m

Valid run!
Score on startup.compiler.compiler: 29.82 ops/m


--- --- --- --- --- --- --- --- ---

  Benchmark:   startup.compiler.sunflow
  Run mode:    static run
  Test type:   single
  Threads:     1
  Iterations:  1
  Run length:  1 operation

Iteration 1 (1 operation) begins: Fri Nov 19 20:01:39 CST 2021
```

卡住几个小时，赶紧排查一下哪儿出错了。

在 https://www.spec.org/jvm2008/docs/FAQ.html 中找到如下说明：



Q4.8: Why won't SPECjvm2008 run with Java SE 8, Java SE 9, or later?

Java SE 8 and later specifications are not supported. See Trouble shooting, Q3.2.

For Java SE 8 and later:

- The following SPECjvm2008 benchmarks are known to not work.
  - `startup.compiler.compiler`
  - `startup.compiler.sunflow`
  - `compiler.compiler`
  - `compiler.sunflow`
- However, you may be able to run the remaining benchmarks by using the command line, or using a properties file, to specify all the benchmarks (except for the above).
- For information on how to do this, see:
  - http://www.spec.org/jvm2008/docs/UserGuide.html#OperationalConfiguration
  - http://www.spec.org/jvm2008/docs/UserGuide.html#Properties
  - http://www.spec.org/jvm2008/docs/UserGuide.html#AppendixA
  - http://www.spec.org/jvm2008/docs/UserGuide.html#SPECjvm2008WorkloadNames

For Java SE 9 and later:

- The XML benchmarks will fail, due to increased class protections.
- However, you may be able to run these benchmarks by adding the following JVM options to make the needed classes visible:

```
--add-exports=java.xml/com.sun.org.apache.xerces.internal.parsers=ALL-UNNAMED
--add-exports=java.xml/com.sun.org.apache.xerces.internal.util=ALL-UNNAMED
```

- This may not work in a future release after Java SE 10.

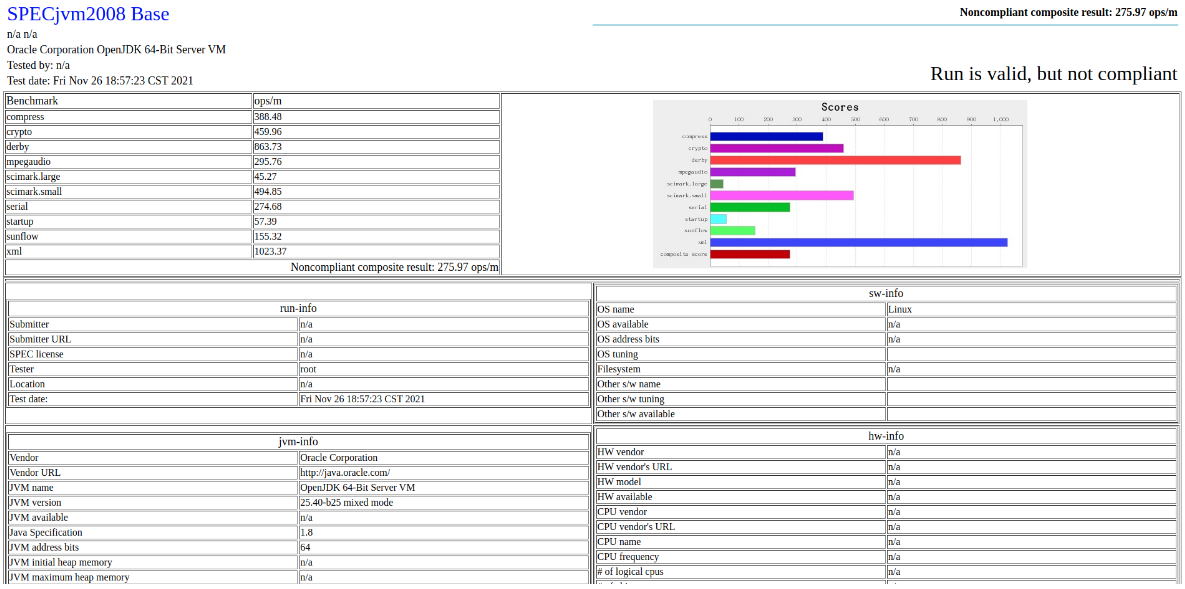
Java SE 8 及之后版本，startup.compiler.compiler，startup.compiler.sunflow，compiler.compiler，compiler.sunflow 都不再 work 了，只能 run 剩下的 benchmark。

按照配置文件示例 /SPECjvm2008/props/specjvm.properties，可以创建一个配置文件 properties，内容为

```
specjvm.benchmarks=startup.helloworld startup.compress startup.crypto.aes
startup.crypto.rsa startup.crypto.signverify startup.mpegaudio
startup.scimark.fft startup.scimark.lu startup.scimark.monte_carlo
startup.scimark.sor startup.scimark.sparse startup.serial startup.sunflow
startup.xml.transform startup.xml.validation compress crypto.aes crypto.rsa
crypto.signverify derby mpegaudio scimark.fft.large scimark.lu.large
scimark.sor.large scimark.sparse.large scimark.fft.small scimark.lu.small
scimark.sor.small scimark.sparse.small scimark.monte_carlo serial sunflow
xml.transform xml.validation
```

这样就去掉了不 work 的 benchmarks，只运行 remainings。

运行后得到报告:



SPECjvm2008 Base

n/a n/a
Oracle Corporation OpenJDK 64-Bit Server VM
Tested by: n/a
Test date: Fri Nov 26 18:57:23 CST 2021

Noncompliant composite result: 275.97 ops/m

Run is valid, but not compliant

Benchmark	ops/m
compress	388.48
crypto	459.96
derby	863.73
mpegaudio	295.76
scimark.large	45.27
scimark.small	494.85
serial	274.68
startup	57.39
sunflow	155.32
xml	1023.37
	Noncompliant composite result: 275.97 ops/m

run-info	
Submitter	n/a
Submitter URL	n/a
SPEC license	n/a
Tester	root
Location	n/a
Test date:	Fri Nov 26 18:57:23 CST 2021

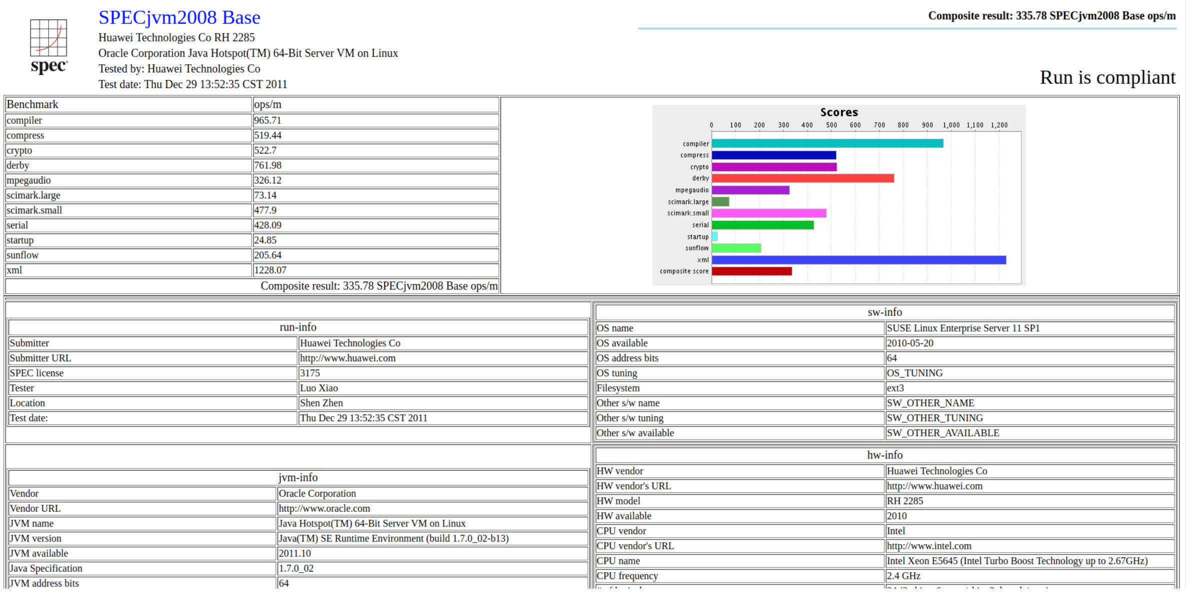
sw-info	
OS name	Linux
OS available	n/a
OS address bits	n/a
OS tuning	
Filesystem	n/a
Other s/w name	
Other s/w tuning	
Other s/w available	

jvm-info	
Vendor	Oracle Corporation
Vendor URL	http://java.oracle.com/
JVM name	OpenJDK 64-Bit Server VM
JVM version	25.40-b25 mixed mode
JVM available	n/a
Java Specification	1.8
JVM address bits	64
JVM initial heap memory	n/a
JVM maximum heap memory	n/a

hw-info	
HW vendor	n/a
HW vendor's URL	n/a
HW model	n/a
HW available	n/a
CPU vendor	n/a
CPU vendor's URL	n/a
CPU name	n/a
CPU frequency	n/a
# of logical cpus	n/a

查阅过去官方发布结果，对比最近任意一次发布结果（可从表格中任选一项）与自己执行结果的差异，并尝试解释原因

选择了华为在 2011 年的测试结果：



SPECjvm2008 Base

Huawei Technologies Co RH 2285
Oracle Corporation Java Hotspot(TM) 64-Bit Server VM on Linux
Tested by: Huawei Technologies Co
Test date: Thu Dec 29 13:52:35 CST 2011

Composite result: 335.78 SPECjvm2008 Base ops/m

Run is compliant

Benchmark	ops/m
compiler	965.71
compress	519.44
crypto	522.7
derby	761.98
mpegaudio	326.12
scimark.large	73.14
scimark.small	477.9
serial	428.09
startup	24.85
sunflow	205.64
xml	1228.07
	Composite result: 335.78 SPECjvm2008 Base ops/m

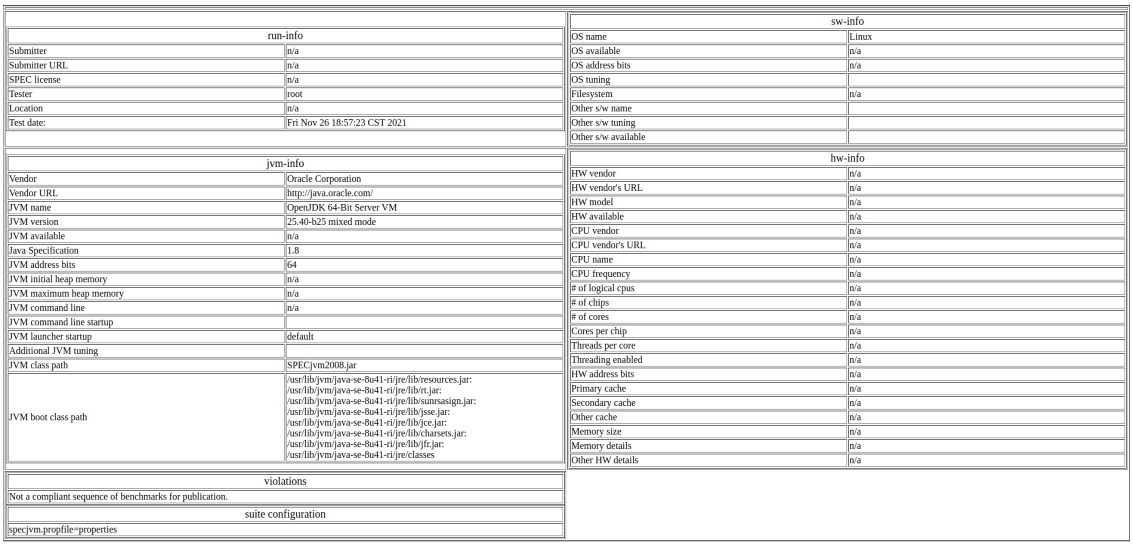
run-info	
Submitter	Huawei Technologies Co
Submitter URL	http://www.huawei.com
SPEC license	3175
Tester	Luo Xiao
Location	Shen Zhen
Test date:	Thu Dec 29 13:52:35 CST 2011

sw-info	
OS name	SUSE Linux Enterprise Server 11 SP1
OS available	2010-05-20
OS address bits	64
OS tuning	OS_TUNING
Filesystem	ext3
Other s/w name	SW_OTHER_NAME
Other s/w tuning	SW_OTHER_TUNING
Other s/w available	SW_OTHER_AVAILABLE

jvm-info	
Vendor	Oracle Corporation
Vendor URL	http://www.oracle.com
JVM name	Java Hotspot(TM) 64-Bit Server VM on Linux
JVM version	Java(TM) SE Runtime Environment (build 1.7.0_02-b13)
JVM available	2011.10
Java Specification	1.7.0_02
JVM address bits	64

hw-info	
HW vendor	Huawei Technologies Co
HW vendor's URL	http://www.huawei.com
HW model	RH 2285
HW available	2010
CPU vendor	Intel
CPU vendor's URL	http://www.intel.com
CPU name	Intel Xeon E5645 (Intel Turbo Boost Technology up to 2.67GHz)
CPU frequency	2.4 GHz

- 首先，一个明显的不同之处是，我的设备信息没有显示：



run-info	
Submitter	n/a
Submitter URL	n/a
SPEC license	n/a
Tester	root
Location	n/a
Test date:	Fri Nov 26 18:57:23 CST 2021

sw-info	
OS name	Linux
OS available	n/a
OS address bits	n/a
OS tuning	
Filesystem	n/a
Other s/w name	
Other s/w tuning	
Other s/w available	

jvm-info	
Vendor	Oracle Corporation
Vendor URL	http://java.oracle.com/
JVM name	OpenJDK 64-Bit Server VM
JVM version	25.40-b25 mixed mode
JVM available	n/a
Java Specification	1.8
JVM address bits	64
JVM initial heap memory	n/a
JVM maximum heap memory	n/a
JVM command line	n/a
JVM command line startup	
JVM launcher startup	default
Additional JVM tuning	
JVM class path	SPECjvm2008.jar
JVM boot class path	/usr/lib/jvm/java-se-8u41-ri/jre/lib/resources.jar: /usr/lib/jvm/java-se-8u41-ri/jre/lib/rt.jar: /usr/lib/jvm/java-se-8u41-ri/jre/lib/sunrsasign.jar: /usr/lib/jvm/java-se-8u41-ri/jre/lib/jsse.jar: /usr/lib/jvm/java-se-8u41-ri/jre/lib/jce.jar: /usr/lib/jvm/java-se-8u41-ri/jre/lib/charsets.jar: /usr/lib/jvm/java-se-8u41-ri/jre/lib/jfr.jar: /usr/lib/jvm/java-se-8u41-ri/jre/classes

hw-info	
HW vendor	n/a
HW vendor's URL	n/a
HW model	n/a
HW available	n/a
CPU vendor	n/a
CPU vendor's URL	n/a
CPU name	n/a
CPU frequency	n/a
# of logical cpus	n/a
# of chips	n/a
# of cores	n/a
Cores per chip	n/a
Threads per core	n/a
Threading enabled	n/a
HW address bits	n/a
Primary cache	n/a
Secondary cache	n/a
Other cache	n/a
Memory size	n/a
Memory details	n/a
Other HW details	n/a

violations
Not a compliant sequence of benchmarks for publication.

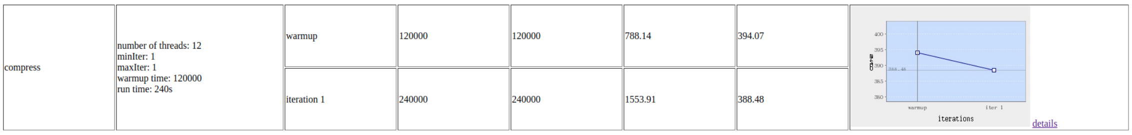
suite configuration
specjvm.propfile=properties

这是因为在配置文件 properties 没有设置这些信息，比如：

```
# # # # # # # # # # # # #
#
# Information about the hardware system
#
# # # # #  # # #

# Company which sells the hardware
spec.jvm2008.report.hw.vendor=HW_VENDOR

# Home page for company that sells the hardware
spec.jvm2008.report.hw.vendor.url=HW_VENDOR_URL

# What type of system was used when testing
spec.jvm2008.report.hw.model=HW_MODEL

# Date when the hardware product is shipping and
# generally available to the public
spec.jvm2008.report.hw.available=HW_AVAILABLE

# Name of the processor vendor
spec.jvm2008.report.hw.cpu.vendor=HW_CPU_VENDOR
```

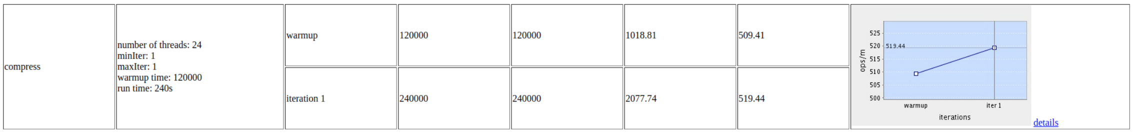
```
# URL to the processor vendor
spec.jvm2008.report.hw.cpu.vendor.url=HW_CPU_VENDOR_URL

# Product name of the processor(s)
spec.jvm2008.report.hw.cpu.name=HW_CPU_NAME

# MegaHertz rating of the chip. Usually an integer
spec.jvm2008.report.hw.cpu.speed=HW_CPU_SPEED
```

- 我的 compress 测试中，warmup 的每秒操作次数(ops/s)比正式测试的每秒操作次数更高，如下图。

compress	number of threads: 12 minIter: 1 maxIter: 1 warmup time: 120000 run time: 240s	warmup	120000	120000	788.14	394.07	details
		iteration 1	240000	240000	1553.91	388.48	

  而华为的 compress 测试则与我的相反：

compress	number of threads: 24 minIter: 1 maxIter: 1 warmup time: 120000 run time: 240s	warmup	120000	120000	1018.81	509.41	details
		iteration 1	240000	240000	2077.74	519.44	

  查了一下文档，这个测试是关于 finds common substrings and replaces them with a variable size code，并且它的数据是来自于真实文件，"it compresses real data from files instead of synthetically generated data"。

  一个猜想，不一定对：warmup 应该是读了一些数据进内存，保证正式测试的时候是热启动。但是我的内存比较小，只有华为硬件的 1/3，所以我正式测试的时候还在不停进行磁盘 IO，导致速度下降(?)

其他项也有差异，这里不详述了。

谈谈自己运行一次标准的基准测试的感想和体会

要成功运行一个标准的 benchmark 主要有两点需要注意：

1. 遵守 user guide

   按照 user guide 一步步进行，能避免很多错误。对于不理解的地方，要善于检索信息(English 要好)。

2. 善于排查自身系统的问题

   很多问题出现的原因，都是自身系统的问题。