The host dex2oat64 abort when compiling boot image in android 11

DESCRIPTION li...@xiaomi.com created issue #1

1. What happened

In android 11, the host dex2oatd64 gets aborted when compiling boot image

```
dex2oatd64 E 12-30 12:34:58 46752 46752 image_writer.cc:2172] Image object without assigned bin slot: java.lang.reflect.InvocationTargetException 0x15580000
dex2oatd64 E 12-30 12:34:58 46752 46752 image_writer.cc:2172] Image object without assigned bin slot: java.lang.Object[] 0x15580020 Type=RootJNILocal thread_
dex2oatd64 F 12-30 12:34:58 46752 46752 image_writer.cc:2176] Found 2 objects without assigned bin slots.
Runtime aborting...

All threads:
DALVIK THREADS (1):
"main" prio=5 tid=1 Runnable (still starting up)
  | group="" sCount=0 dsCount=0 flags=0 obj=(nil) self=0x56185db201e0
  | sysTid=46752 nice=0 cgrp=user.slice sched=0/0 handle=0x7ffac94a1740
  | state=R schedstat=( 20154329193 2124965744 957 ) utm=1871 stm=143 core=4 HZ=100
  | stack=0x7ffd41588000-0x7ffd4158a000 stackSize=8176KB
  | held mutexes= "abort lock" "mutator lock"(shared held)
  native: #00 pc 000000000093fef6  out/soong/host/linux-x86/lib64/libartd.so (art::DumpNativeStack(std::__1::basic_ostream<char, std::__1::char_traits<char>
  native:   art::DumpNativeStack(std::__1::basic_ostream<char, std::__1::char_traits<char> >&, int, BacktraceMap*, char const*, art::ArtMethod*, void*, bool)
  native:    art/runtime/native_stack_dump.cc:337
  native: #01 pc 0000000000a96c0c  out/soong/host/linux-x86/lib64/libartd.so (art::Thread::DumpStack(std::__1::basic_ostream<char, std::__1::char_traits<char
  native:   art::Thread::DumpStack(std::__1::basic_ostream<char, std::__1::char_traits<char> >&, bool, BacktraceMap*, bool) const
  native:    art/runtime/thread.cc:2189
  native: #02 pc 0000000000abbc47  out/soong/host/linux-x86/lib64/libartd.so (art::DumpCheckpoint::Run(art::Thread*)+583)
  native:   art::DumpCheckpoint::Run(art::Thread*)
  native:    art/runtime/thread_list.cc:222
  native: #03 pc 0000000000ab4b18  out/soong/host/linux-x86/lib64/libartd.so (art::ThreadList::RunCheckpoint(art::Closure*, art::Closure*)+1000)
  native:   art::ThreadList::RunCheckpoint(art::Closure*, art::Closure*)
  native:    art/runtime/thread_list.cc:383
  native: #04 pc 0000000000ab4470  out/soong/host/linux-x86/lib64/libartd.so (art::ThreadList::Dump(std::__1::basic_ostream<char, std::__1::char_traits<char>
  native:   art::ThreadList::Dump(std::__1::basic_ostream<char, std::__1::char_traits<char> >&, bool)
  native:    art/runtime/thread_list.cc:266
  native: #05 pc 0000000000a5f19a  out/soong/host/linux-x86/lib64/libartd.so (art::AbortState::Dump(std::__1::basic_ostream<char, std::__1::char_traits<char>
  native:   art::AbortState::Dump(std::__1::basic_ostream<char, std::__1::char_traits<char> >&) const
  native:    art/runtime/runtime.cc:534
  native: #06 pc 0000000000a4a417  out/soong/host/linux-x86/lib64/libartd.so (art::Runtime::Abort(char const*)+567)
  native:   art::Dumpable<art::AbortState>::Dump(std::__1::basic_ostream<char, std::__1::char_traits<char> >&) const
  native:    art/libartbase/base/dumpable.h:38
  native:   std::__1::basic_ostream<char, std::__1::char_traits<char> >& art::operator<< <art::AbortState>(std::__1::basic_ostream<char, std::__1::char_trait
  native:    art/libartbase/base/dumpable.h:49
  native:   art::Runtime::Abort(char const*)
  native:    art/runtime/runtime.cc:657
  native: #07 pc 000000000002f83c  out/soong/host/linux-x86/lib64/libbase.so (android::base::SetAborter(std::__1::function<void (char const*)>&&)::$_3::__inv
  native:   std::__1::__function::__value_func<void (char const*)>::operator()(char const*&&) const
  native:    external/libcxx/include/functional:1799
  native:   std::__1::function<void (char const*)>::operator()(char const*) const
  native:    external/libcxx/include/functional:2347
  native:   operator()
  native:    system/core/base/logging.cpp:421
  native:   __invoke
  native:    system/core/base/logging.cpp:421
  native: #08 pc 000000000002f069  out/soong/host/linux-x86/lib64/libbase.so (android::base::LogMessage::~LogMessage()+329)
  native:   ~LogMessage
  native:    system/core/base/logging.cpp:509
  native: #09 pc 000000000014c015  out/soong/host/linux-x86/bin/dex2oatd64 (art::linker::ImageWriter::LayoutHelper::VerifyImageBinSlotsAssigned()+6469)
  native:   art::linker::ImageWriter::LayoutHelper::VerifyImageBinSlotsAssigned()
  native:    art/dex2oat/linker/image_writer.cc:2176
  native: #10 pc 0000000000131002  out/soong/host/linux-x86/bin/dex2oatd64 (art::linker::ImageWriter::CalculateNewObjectOffsets()+4866)
  native:   art::linker::ImageWriter::CalculateNewObjectOffsets()
  native:    art/dex2oat/linker/image_writer.cc:2457
  native: #11 pc 000000000012c3d1  out/soong/host/linux-x86/bin/dex2oatd64 (art::linker::ImageWriter::PrepareImageAddressSpace(bool, art::TimingLogger*)+3457
  native:   art::linker::ImageWriter::PrepareImageAddressSpace(bool, art::TimingLogger*)
  native:    art/dex2oat/linker/image_writer.cc:297
  native: #12 pc 00000000000be09f  out/soong/host/linux-x86/bin/dex2oatd64 (art::Dex2Oat::WriteOutputFiles(_jobject*)+367)
  native:   art::Dex2Oat::WriteOutputFiles(_jobject*)
  native:    art/dex2oat/dex2oat.cc:2274
  native: #13 pc 00000000000a6c15  out/soong/host/linux-x86/bin/dex2oatd64 (main+3189)
  native:   art::CompileImage(art::Dex2Oat&)
  native:    art/dex2oat/dex2oat.cc:3158
  native:   art::Dex2oat(int, char**)
  native:    art/dex2oat/dex2oat.cc:3378
  native:   main
  native:    art/dex2oat/dex2oat.cc:3388
  native: #14 pc 000000000002083f  /lib/x86_64-linux-gnu/libc-2.23.so (__libc_start_main+239)
  native:   __libc_start_main
  native:    ??:?
  native: #15 pc 000000000005e538  out/soong/host/linux-x86/bin/dex2oatd64 (???)
  (no managed stack frames)

Cmdline: out/soong/host/linux-x86/bin/dex2oatd64 --avoid-storing-invocation --write-invocation-to=out/soong/qssi/dex_bootjars/system/framework/arm/boot.invoc
```

2. How to reproduce

1. find a framework image class, like "frameworks/base/core/java/android/util/apk/ApkSignatureVerifier.java"
2. add a reflection call which implemented by jni in static code block

```
--- a/core/java/android/util/apk/ApkSignatureVerifier.java
+++ b/core/java/android/util/apk/ApkSignatureVerifier.java
@@ -68,8 +68,23 @@ public class ApkSignatureVerifier {
+
+    static {
+        try {
+            Class clazz = Class.forName("android.os.SystemProperties");
+            java.lang.reflect.Method getMethod = clazz.getMethod("get", String.class);
+
+            String value = (String)getMethod.invoke(null, "ro.build.type");
+        } catch (Exception e) {
+            // ignore
+        }
+    }
```

3. make framework-minus-apex -j8

3. How to fix

In unstarted runtime, reflection methods may cause TransactionAbortError which is the common case in the compiler. For example, most JNI methods are not allowed in unstarted runtime, and a

https://android-review.googlesource.com/c/platform/art/+/1535797

COMMENTS

am...@google.com <am...@google.com> #2

Assigned to am...@google.com.

Thank you for reporting this issue. We've shared this with our product and engineering teams and will continue to provide updates as more information becomes available.

vm...@google.com <vm...@google.com> #3

Marked as fixed, reassigned to vm...@google.com.

The fix has been submitted.