Unpick performance mysteries benchmarking GraalVM native executables

Galder Zamarreño

Principal Software Engineer

Source Code Reference

```
@BenchmarkMode(Mode.AverageTime)
// ...
class CharAt
{
    String[] values;
    int charAtIndex;

    @Setup
    void setup()
    {
        values = new String[2];
        values[0] = "Latin1 string";
        values[1] = "UTF-\uFF11\uFF16 string";
        charAtIndex = 3;
    }

    @Benchmark
    char charAtLatin1()
    {
        return values[0].charAt(charAtIndex);
    }

    @Benchmark
    char charAtUtf16()
    {
        return values[1].charAt(charAtIndex);
    }
}
```

```
class String
{
    static final byte LATIN1 = 0;
    final byte[] value;
    final byte coder;

    char charAt(int index)
    {
        if (isLatin1())
        {
            return StringLatin1.charAt(value, index);
        }
        else
        {
            return StringUTF16.charAt(value, index);
        }
    }

    boolean isLatin1()
    {
        return /* ... */ && coder == LATIN1;
    }
}
```

Source Code Reference

```
class StringLatin1
{
    static char charAt(byte[] value, int index)
    {
        checkIndex(index, value.length);
        return (char)(value[index] & 0xff);
    }
}
```

```
class CharAt_charAtLatin1_jmhTest
{
    public static void charAtLatin1_avgt_jmhStub(
        InfraControl control
        , RawResults result
        , Blackhole bh
        , CharAt_jmhType l_charat0_0
    ) throws Throwable
    {
        long operations = 0;
        long realTime = 0;
        result.startTime = System.nanoTime();
        do
        {
            bh.consume(l_charat0_0.charAtLatin1());
            operations++;
        } while(!control.isDone);
        result.stopTime = System.nanoTime();
        result.realTime = realTime;
        result.measuredOps = operations;
    }
}
```

Who am I?

Engineer @ OpenJDK team

Work on GraalVM native image and HotSpot JIT compilers

Created a JMH extension to benchmark Java @ GraalVM native image

First Run

```
$ mvn package
...
[INFO] --- compiler:3.12.1:compile (default-compile) @ strings ---
[INFO] Recompiling the module because of changed source code.
[INFO] Compiling 1 source file with javac [debug release 21] to target/classes
[INFO] Write reflection configuration
[INFO] GraalVM blackhole mode resolved: enabled (JDK 21 or earlier)
...
[INFO] --- shade:3.6.0:shade (default) @ strings ---
...
[INFO] Replacing /root/1/fosdem/first-run/target/benchmarks.jar with
/root/1/fosdem/first-run/target/strings-1.0.0-SNAPSHOT-shaded.jar
...
[INFO] --- native:0.10.3:compile-no-fork (build-native) @ strings ---
[INFO] Found GraalVM installation from JAVA_HOME variable.
[INFO] Executing: /root/opt/graal-21/bin/native-image ... -o
/root/1/fosdem/first-run/target/benchmarks org.mendrugo.fibula.NativeForkedMain
====================================================================================================
GraalVM Native Image: Generating 'benchmarks' (executable)...
====================================================================================================
...
Finished generating 'benchmarks' in 27.5s.
```

First Run

```
$ java -jar target/benchmarks.jar

# JMH version: fibula:999-SNAPSHOT
# VM version: JDK 21.0.2, Substrate VM, GraalVM CE 21.0.2+13.1
# *** WARNING: This VM is not supported by JMH. The produced benchmark data can be completely wrong.
# VM invoker: target/benchmarks
...
Benchmark              Mode  Cnt  Score   Error  Units
CharAt.charAtLatin1    avgt    5  6.601 ± 0.003  ns/op
CharAt.charAtUtf16     avgt    5  5.142 ± 0.002  ns/op
```

First Profile

```
$ mvn package -Ddebug=true
-DbuildArgs=-H:-DeleteLocalSymbols,-H:+SourceLevelDebug,-H:+TrackNodeSourcePosition,\
-H:+DebugCodeInfoUseSourceMappings
...
====================================================================================================
GraalVM Native Image: Generating 'benchmarks' (executable)...
====================================================================================================
...
----------------------------------------------------------------------------------------------------
 3 experimental option(s) unlocked:
 - '-H:+DebugCodeInfoUseSourceMappings' (origin(s): command line)
 - '-H:-DeleteLocalSymbols' (origin(s): command line)
 - '-H:+SourceLevelDebug' (origin(s): command line)
----------------------------------------------------------------------------------------------------
...
Produced artifacts:
 /root/1/fosdem/first-profile/target/benchmarks (executable)
 /root/1/fosdem/first-profile/target/benchmarks.debug (debug_info)
 /root/1/fosdem/first-profile/target/sources (debug_info)
====================================================================================================
Finished generating 'benchmarks' in 45.5s.
```

First Profile

```
$ java -jar target/benchmarks.jar -prof org.mendrugo.fibula.PerfDwarfProfiler:events=cycles:pp

# JMH version: fibula:999-SNAPSHOT
# VM version: JDK 21.0.2, Substrate VM, GraalVM CE 21.0.2+13.1
# *** WARNING: This VM is not supported by JMH. The produced benchmark data can be completely wrong.
# VM invoker: target/benchmarks
...
Secondary result "org.sample.strings.CharAt.charAtLatin1:asm":
Perf binary output saved to ./org.sample.strings.CharAt.charAtLatin1-AverageTime.perfbin
...
Secondary result "org.sample.strings.CharAt.charAtUtf16:asm":
Perf binary output saved to ./org.sample.strings.CharAt.charAtUtf16-AverageTime.perfbin
```

First Profile

```
$ perf annotate -i org.sample.strings.CharAt.charAtLatin1-AverageTime.perfbin

         │    0000000000663000 <char java.lang.StringLatin1::charAt(byte[]*, int)>:
    4.89 │      sub    $0x18,%rsp
    2.34 │      cmp    0x8(%r15),%rsp           <--- stackoverflow check
         │    ↓ jbe    80
    2.48 │      cmp    %r14,%rdi                <--- null check for rdi, which is the byte[]
         │    ↓ je     8a
    1.87 │      mov    0xc(%rdi),%eax           <--- read byte[] length into eax
                                                <--- prepare arguments to call Preconditions.checkIndex
    2.34 │      movabs $0x841310,%rdx
    2.45 │      lea    (%r14,%rdx,1),%rdx       <--- obtain reference to Preconditions.SIOOBE_FORMATTER (3rd argument)
    2.19 │      mov    %rdi,0x8(%rsp)           <--- copy byte[] to 0x8(%rsp)
    2.27 │      mov    %esi,%edi                <--- prepare index (esi) parameter for call (1st argument)
    4.93 │      mov    %esi,0x14(%rsp)
    2.12 │      mov    %eax,%esi                <--- prepare byte[] length (eax) parameter for call (2nd argument)
    2.52 │      mov    %eax,0x10(%rsp)
    2.26 │      data16 xchg %ax,%ax
                                                <--- invoke with index (edi), length (esi) and formatter (rdx)
    4.28 │    → callq  _ZN31jdk.internal.util.Preconditions10checkIndexEJiiiP29java.util.function.BiFunction
...
         │80:    mov    %esi,0x14(%rsp)
         │      → callq  _ZN57com.oracle.svm.core.graal.snippets.StackOverflowCheckImpl26throwNewStackOverflowErrorEJvv
         │8a:    mov    %esi,0x14(%rsp)
         │      → callq  _ZN47com.oracle.svm.core.snippets.ImplicitExceptions28throwNewNullPointerExceptionEJvv
```

First Profile

```
$ perf annotate -i org.sample.strings.CharAt.charAtLatin1-AverageTime.perfbin

          │    0000000000663000 <char java.lang.StringLatin1::charAt(byte[]*, int)>:
...
     2.23 │      mov    0x10(%rsp),%eax         <--- copy byte[] length back to eax
     4.61 │      cmp    0x14(%rsp),%eax         <--- compare index and length for array out of bounds check
          │    ↓ jbe    94
     2.81 │      mov    0x8(%rsp),%rdi          <--- copy byte[] back to rdi
     2.84 │      mov    0x14(%rsp),%eax         <--- copy index back to eax
     1.98 │      movzbl 0x10(%rdi,%rax,1),%esi  <--- copies byte[index] to esi (byte[] rdi, index rax)
                                                <--- movzbl zero extends the char to integer to make it unsigned
                                                <---   equivalent of `& 0xFF` java code
     5.07 │      subl   $0x1,0x10(%r15)         <--- safepoint check
     2.19 │    ↓ jle    78
     2.52 │71:   mov    %esi,%eax               <--- copy esi to eax to return the char
     5.39 │      add    $0x18,%rsp              <--- method complete, reduce the stack size
     2.66 │    ← retq
          │78: → callq  _ZN36com.oracle.svm.core.thread.Safepoint27enterSlowPathSafepointCheckEJvv
          │    ↑ jmp    71
          │94:   mov    0x14(%rsp),%esi
          │      mov    %esi,%edi
          │      mov    %eax,%esi
          │    → callq  _ZN47com.oracle.svm.core.snippets.ImplicitExceptions36throwNewOutOfBoundsExceptionWithArgsEJvii
```

First Profile

```
$ perf annotate -i org.sample.strings.CharAt.charAtLatin1-AverageTime.perfbin

          │     0000000000bc6230 <void org.sample.strings.jmh_generated.CharAt_charAtLatin1_jmhTest::charAtLatin1_avgt_jmhStub(...)
...
          │     → callq      _ZN16java.lang.System8nanoTimeEJlv                    <--- get start time
...
          │     → callq      _ZN25org.sample.strings.CharAt12charAtLatin1EJ4charv <--- invoke the benchmark once
...
          │       cmpb       $0x0,0xdc(%rdi)                                       <--- is done check
          │     ↓ je         d7
          │       movq       $0x1,0x30(%rsp)
          │ 9f:   nop
          │     → callq      _ZN16java.lang.System8nanoTimeEJlv                    <--- is done so get end time
...
          │ d7:   movq       $0x1,0x30(%rsp)                                       <--- initialize number of operations to 1
```

First Profile

```
$ perf annotate -i org.sample.strings.CharAt.charAtLatin1-AverageTime.perfbin

          │      0000000000bc6230 <void org.sample.strings.jmh_generated.CharAt_charAtLatin1_jmhTest::charAtLatin1_avgt_jmhStub(...)
...
   5.67   │ f0: ┌─→nop
   6.43   │     │  mov       %rax,%rdi
   5.77   │     │→ callq     _ZN25org.sample.strings.CharAt12charAtLatin1EJ4charv
   2.84   │     │  mov       0x30(%rsp),%rax                    <--- copy operations to rax
   6.33   │     │  inc       %rax                               <--- increment operations (rax)
   7.00   │     │  mov       0x28(%rsp),%rdi
   5.67   │     │  cmpb      $0x0,0xdc(%rdi)                    <--- is done check
   6.90   │     │↓ jne       132                                <--- is done
   3.41   │     │  subl      $0x1,0x10(%r15)
   6.01   │     │↓ jle       181
   5.77   │     │  mov       %rax,0x30(%rsp)
   2.65   │     │  mov       0x10(%rsp),%rax
   7.57   │     └──jmp       f0
          │132:    mov       %rax,0x30(%rsp)                    <--- is done
          │      ↑ jmpq      9f                                 <--- jump back up to get the time
...
          │181:    mov       0x10(%rsp),%rcx
          │        mov       0x18(%rsp),%r9
          │        mov       0x20(%rsp),%rsi
          │      → callq     _ZN36com.oracle.svm.core.thread.Safepoint27enterSlowPathSafepointCheckEJvv
```

First Profile

```
$ perf annotate -i org.sample.strings.CharAt.charAtLatin1-AverageTime.perfbin

        |    00000000006496c0 <char java.lang.String::charAt(int)>:
   5.71 |      sub    $0x18,%rsp
   3.42 |      cmp    0x8(%r15),%rsp          <--- stackoverflow check
        |    ↓ jbe    c8
   6.56 |      mov    0x10(%rdi),%rdx
   3.42 |      lea    (%r14,%rdx,1),%rax      <--- get address of String.value into rax
   3.61 |      cmpb   $0x0,0x18(%rdi)         <--- is latin1 check, inlined isLatin1()
   3.86 |    ↓ je     94
...
   5.81 |94:   mov    %rax,%rcx
   2.68 |      mov    %esi,0x14(%rsp)         <--- copy index to 0x14(%rsp) for call to StringLatin1.charAt (2nd param)
   3.56 |      mov    %rcx,%rdi               <--- copy rax to rdi for call to StringLatin1.charAt (1st param)
                                              <--- StringLatin1.charAt not inlined
   9.59 |    → callq  _ZN22java.lang.StringLatin16charAtEJ4charP6byte[]i
   3.13 |      subl   $0x1,0x10(%r15)
   3.03 |    ↓ jle    b7
   6.96 |b2:   add    $0x18,%rsp
   3.95 |    ← retq
        |b7: → callq  _ZN36com.oracle.svm.core.thread.Safepoint27enterSlowPathSafepointCheckEJvv
        |    ↑ jmp    b2
        |c8:   mov    %esi,0x14(%rsp)
        |    → callq  _ZN57com.oracle.svm.core.graal.snippets.StackOverflowCheckImpl26throwNewStackOverflowErrorEJvv
```

First Profile

```
$ perf annotate -i org.sample.strings.CharAt.charAtUtf16-AverageTime.perfbin

       │    00000000006496c0 <char java.lang.String::charAt(int)>:
  3.07 │      sub    $0x18,%rsp
  1.36 │      cmp    0x8(%r15),%rsp            <--- stackoverflow check
       │    ↓ jbe    c8
  1.67 │      mov    0x10(%rdi),%rdx
  1.69 │      lea    (%r14,%rdx,1),%rax        <--- get address of String.value byte[] into rax
  1.40 │      cmpb   $0x0,0x18(%rdi)           <--- is latin1 check, inlined isLatin1()
  1.38 │    ↓ je     94
                                               <--- inlined StringUTF16.charAt
  3.16 │      test   %rdx,%rdx                 <--- null pointer check
       │    ↓ je     d2
...
       │94:   mov    %rax,%rcx
       │      mov    %esi,0x14(%rsp)
       │      mov    %rcx,%rdi
       │    → callq  _ZN22java.lang.StringLatin16charAtEJ4charP6byte[]i
       │c8:   mov    %esi,0x14(%rsp)
       │    → callq  _ZN57com.oracle.svm.core.graal.snippets.StackOverflowCheckImpl26throwNewStackOverflowErrorEJvv
       │d2:   mov    %rax,%rcx
       │    → callq  _ZN47com.oracle.svm.core.snippets.ImplicitExceptions28throwNewNullPointerExceptionEJvv
```

First Profile

```
$ perf annotate -i ./org.sample.strings.CharAt.charAtUtf16-AverageTime.perfbin
```

```
        │      00000000006496c0 <char java.lang.String::charAt(int)>:
  1.67  │        mov    %rax,0x8(%rsp)             <--- copy String.value byte[] to 0x8(%rsp)
  1.49  │        mov    0xc(%r14,%rdx,1),%edi      <--- copy byte[] length into edi
  1.51  │        mov    %edi,%edx                  <--- copy length into edx
  3.24  │        shr    %edx                       <--- byte[].length >> 1 happening in StringUTF16.length()
  1.58  │        movabs $0x841310,%rcx
  1.86  │        lea    (%r14,%rcx,1),%rcx         <--- obtain reference to Preconditions.SIOOBE_FORMATTER (3rd argument)
  3.11  │        mov    %esi,%edi                  <--- prepare index (esi) parameter for call (1st argument)
  1.36  │        mov    %esi,0x14(%rsp)
  2.08  │        mov    %edx,%esi                  <--- copy length into esi (2nd argument)
  1.73  │        mov    %rcx,%rdx                  <--- copy formatter into rdx (3rd argument)
  2.94  │      → callq  _ZN31jdk.internal.util.Preconditions10checkIndexEJiiiP29java.util.function.BiFunction
  1.78  │        mov    0x14(%rsp),%esi            <--- move index to esi
  1.91  │        shl    %esi                       <--- left shift index
  1.40  │        movslq %esi,%rax
  1.45  │        mov    0x8(%rsp),%rcx             <--- copy String.value byte[] to rcx
  3.51  │        movzwl 0x10(%rcx,%rax,1),%eax     <--- extract the UTF16 char out of the byte[]
  1.75  │        subl   $0x1,0x10(%r15)
  1.82  │      ↓ jle    bf
  2.67  │8f:     add    $0x18,%rsp
  1.47  │      ← retq
        │bf:     nop
        │      → callq  _ZN36com.oracle.svm.core.thread.Safepoint27enterSlowPathSafepointCheckEJvv
        │      ↑ jmp    8f
```

Trivial Run

```
$ mvn package -DbuildArgs=-H:MaxNodesInTrivialMethod=40
...
====================================================================================================
GraalVM Native Image: Generating 'benchmarks' (executable)...
====================================================================================================
...
----------------------------------------------------------------------------------------------------
 1 experimental option(s) unlocked:
 - '-H:MaxNodesInTrivialMethod' (origin(s): command line)
----------------------------------------------------------------------------------------------------
...
Produced artifacts:
 /root/1/fosdem/trivial-run/target/benchmarks (executable)
====================================================================================================
Finished generating 'benchmarks' in 28.3s.
```

Trivial Run

```
$ java -jar target/benchmarks.jar

# JMH version: fibula:999-SNAPSHOT
# VM version: JDK 21.0.2, Substrate VM, GraalVM CE 21.0.2+13.1
# *** WARNING: This VM is not supported by JMH. The produced benchmark data can be completely wrong.
# VM invoker: target/benchmarks
...
Benchmark              Mode  Cnt  Score   Error  Units
CharAt.charAtLatin1    avgt    5  2.410 ± 0.135  ns/op
CharAt.charAtUtf16     avgt    5  2.426 ± 0.072  ns/op
```

Trivial Profile

```
$ mvn package -Ddebug=true
-DbuildArgs=-H:-DeleteLocalSymbols,-H:+SourceLevelDebug,-H:+TrackNodeSourcePosition,\
-H:+DebugCodeInfoUseSourceMappings,-H:MaxNodesInTrivialMethod=40
...
====================================================================================================
GraalVM Native Image: Generating 'benchmarks' (executable)...
====================================================================================================
...
----------------------------------------------------------------------------------------------------
 4 experimental option(s) unlocked:
 - '-H:+DebugCodeInfoUseSourceMappings' (origin(s): command line)
 - '-H:-DeleteLocalSymbols' (origin(s): command line)
 - '-H:MaxNodesInTrivialMethod' (origin(s): command line)
 - '-H:+SourceLevelDebug' (origin(s): command line)
----------------------------------------------------------------------------------------------------
...
Produced artifacts:
 /root/1/fosdem/trivial-profile/target/benchmarks (executable)
 /root/1/fosdem/trivial-profile/target/benchmarks.debug (debug_info)
 /root/1/fosdem/trivial-profile/target/sources (debug_info)
====================================================================================================
Finished generating 'benchmarks' in 47.9s.
```

Trivial Profile

```
$ java -jar target/benchmarks.jar -prof org.mendrugo.fibula.PerfDwarfProfiler:events=cycles:pp

# JMH version: fibula:999-SNAPSHOT
# VM version: JDK 21.0.2, Substrate VM, GraalVM CE 21.0.2+13.1
# *** WARNING: This VM is not supported by JMH. The produced benchmark data can be completely wrong.
# VM invoker: target/benchmarks
...
Secondary result "org.sample.strings.CharAt.charAtLatin1:asm":
Perf binary output saved to ./org.sample.strings.CharAt.charAtLatin1-AverageTime.perfbin
...
Secondary result "org.sample.strings.CharAt.charAtUtf16:asm":
Perf binary output saved to ./org.sample.strings.CharAt.charAtUtf16-AverageTime.perfbin
```

Trivial Profile

```
$ perf annotate -i org.sample.strings.CharAt.charAtLatin1-AverageTime.perfbin

        │     000000000067cc00 <char java.lang.String::charAt(int)>:
...
   2.76 │       mov         0x10(%rdi),%rdx
   2.18 │       lea         (%r14,%rdx,1),%rax            <--- get address of String.value byte[] into rax
   2.89 │       lea         0x80(%rsi),%ecx
   3.54 │       cmpb        $0x0,0x18(%rdi)               <--- is latin1 check, inlined isLatin1()
   1.98 │     ↓ je          75
...
   2.72 │ 75:   nop
   1.98 │       test        %rdx,%rdx                     <--- null pointer check
        │     ↓ je          d6
   2.69 │       mov         0xc(%r14,%rdx,1),%edx         <--- copy byte[] length into edx
   3.06 │       cmp         %esi,%edx                     <--- index out of bounds length check?
        │     ↓ jbe         26e
   2.69 │       mov         %esi,%edx                     <--- copy index to edx
   3.21 │       movzbl      0x10(%rax,%rdx,1),%edx        <--- copies byte[index] to edx (byte[] rax, index rdx)
                                                          <--- movzbl zero extends the char to integer to make it unsigned
                                                          <---   equivalent of `& 0xFF` java code
   2.92 │ b3:   mov         %edx,%eax                     <--- copy edx to eax to return the char
   3.30 │       add         $0x38,%rsp
   2.18 │     ← retq
```

HotSpot

```
$ mvn package -Djvm.mode
...
[INFO] --- shade:3.6.0:shade (default) @ strings ---
[INFO] Including org.mendrugo.fibula:fibula-core:jar:999-SNAPSHOT in the shaded jar.
[INFO] Including org.openjdk.jmh:jmh-core:jar:1.37 in the shaded jar.
[INFO] Including net.sf.jopt-simple:jopt-simple:jar:5.0.4 in the shaded jar.
[INFO] Including org.apache.commons:commons-math3:jar:3.6.1 in the shaded jar.
[INFO] Dependency-reduced POM written at: /root/1/fosdem/hotspot/dependency-reduced-pom.xml
[INFO] Replacing /root/1/fosdem/hotspot/target/benchmarks.jar with
/root/1/fosdem/hotspot/target/strings-1.0.0-SNAPSHOT-shaded.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
```

HotSpot

```
$ java -jar target/benchmarks.jar

# JMH version: fibula:999-SNAPSHOT
# VM version: JDK 21.0.5, OpenJDK 64-Bit Server VM, 21.0.5+11-LTS
# VM invoker: /root/opt/jdk-21.0.5+11/bin/java
...
Benchmark              Mode  Cnt  Score    Error  Units
CharAt.charAtLatin1    avgt    5  1.722 ± 0.035  ns/op
CharAt.charAtUtf16     avgt    5  1.713 ± 0.001  ns/op
```

HotSpot

```
$ java -jar target/benchmarks.jar -prof perfasm
....[Hottest Region 1]............................................................................
c2, level 4, org.sample.strings.jmh_generated.CharAt_charAtLatin1_jmhTest::charAtLatin1_avgt_jmhStub, version 6, compile id 598

            0x00007fdb00304da0:   movl               0x10(%r8), %r11d    ;*getfield values {reexecute=0 rethrow=0 return_oop=0}
                                                                      ; - org.sample.strings.CharAt::charAtLatin1@1 (line 39)
                                                                      ; - org.sample.strings.jmh_generated.CharAt_charAtLatin1_jmhTest::charAtLatin1_avgt_jmhStub@17 (line 190)
            0x00007fdb00304da4:   movl               0xc(%r12, %r11, 8), %r9d; implicit exception: dispatches to 0x00007fdb00304ff0
            0x00007fdb00304da9:   testl              %r9d, %r9d
            0x00007fdb00304dac:   jbe        0x7fdb00304eac
            0x00007fdb00304db2:   movl               0x10(%r12, %r11, 8), %r10d
                                                                      ;*aaload {reexecute=0 rethrow=0 return_oop=0}
                                                                      ; - org.sample.strings.CharAt::charAtLatin1@5 (line 39)
                                                                      ; - org.sample.strings.jmh_generated.CharAt_charAtLatin1_jmhTest::charAtLatin1_avgt_jmhStub@17 (line 190)
  14.47%    0x00007fdb00304db7:   movl               0xc(%r8), %r9d      ;*getfield charAtIndex {reexecute=0 rethrow=0 return_oop=0}
                                                                      ; - org.sample.strings.CharAt::charAtLatin1@9 (line 40)
                                                                      ; - org.sample.strings.jmh_generated.CharAt_charAtLatin1_jmhTest::charAtLatin1_avgt_jmhStub@17 (line 190)
            0x00007fdb00304dbb:   movsbl             0x10(%r12, %r10, 8), %r11d
                                                                      ; implicit exception: dispatches to 0x00007fdb00305004
                                                                      ;*getfield coder {reexecute=0 rethrow=0 return_oop=0}
                                                                      ; - java.lang.String::isLatin1@7 (line 4821)
                                                                      ; - java.lang.String::charAt@1 (line 1554)
                                                                      ; - org.sample.strings.CharAt::charAtLatin1@12 (line 40)
                                                                      ; - org.sample.strings.jmh_generated.CharAt_charAtLatin1_jmhTest::charAtLatin1_avgt_jmhStub@17 (line 190)
   0.02%    0x00007fdb00304dc1:   testl              %r11d, %r11d
   4.75%    0x00007fdb00304dc4:   jne        0x7fdb00304edc      ;*ifne {reexecute=0 rethrow=0 return_oop=0}
                                                                      ; - java.lang.String::isLatin1@10 (line 4821)
                                                                      ; - java.lang.String::charAt@1 (line 1554)
                                                                      ; - org.sample.strings.CharAt::charAtLatin1@12 (line 40)
                                                                      ; - org.sample.strings.jmh_generated.CharAt_charAtLatin1_jmhTest::charAtLatin1_avgt_jmhStub@17 (line 190)
            0x00007fdb00304dca:   movl               0x14(%r12, %r10, 8), %r11d
                                                                      ;*getfield value {reexecute=0 rethrow=0 return_oop=0}
                                                                      ; - java.lang.String::charAt@8 (line 1555)
                                                                      ; - org.sample.strings.CharAt::charAtLatin1@12 (line 40)
                                                                      ; - org.sample.strings.jmh_generated.CharAt_charAtLatin1_jmhTest::charAtLatin1_avgt_jmhStub@17 (line 190)
```

HotSpot

```
$ java -jar target/benchmarks.jar -prof perfasm
....[Hottest Region 1]..............................................................................
c2, level 4, org.sample.strings.jmh_generated.CharAt_charAtLatin1_jmhTest::charAtLatin1_avgt_jmhStub, version 6, compile id 598

            0x00007fdb00304dcf:   movl              0xc(%r12, %r11, 8), %r10d; implicit exception: dispatches to 0x00007fdb0030501c
                                                                      ;*arraylength {reexecute=0 rethrow=0 return_oop=0}
                                                                      ; - java.lang.StringLatin1::charAt@2 (line 46)
                                                                      ; - java.lang.String::charAt@12 (line 1555)
                                                                      ; - org.sample.strings.CharAt::charAtLatin1@12 (line 40)
                                                                      ; - org.sample.strings.jmh_generated.CharAt_charAtLatin1_jmhTest::charAtLatin1_avgt_jmhStub@17 (line 190)
  50.97%    0x00007fdb00304dd4:   cmpl              %r10d, %r9d
            0x00007fdb00304dd7:   jae        0x7fdb00304e40      ;*invokestatic checkIndex {reexecute=0 rethrow=0 return_oop=0}
                                                                      ; - java.lang.String::checkIndex@5 (line 4832)
                                                                      ; - java.lang.StringLatin1::charAt@3 (line 46)
                                                                      ; - java.lang.String::charAt@12 (line 1555)
                                                                      ; - org.sample.strings.CharAt::charAtLatin1@12 (line 40)
                                                                      ; - org.sample.strings.jmh_generated.CharAt_charAtLatin1_jmhTest::charAtLatin1_avgt_jmhStub@17 (line 190)
            0x00007fdb00304dd9:   cmpl              %r10d, %r9d
   4.69%    0x00007fdb00304ddc:   jae        0x7fdb00304e78
            0x00007fdb00304de2:   movzbl            0x94(%rdi), %edx    ;*getfield isDone {reexecute=0 rethrow=0 return_oop=0}
                                                                      ; - org.sample.strings.jmh_generated.CharAt_charAtLatin1_jmhTest::charAtLatin1_avgt_jmhStub@30 (line 192)
            0x00007fdb00304de9:   leaq              (%r12, %r11, 8), %r10
            0x00007fdb00304ded:   movzbl            0x10(%r10, %r9), %r11d;*iand {reexecute=0 rethrow=0 return_oop=0}
                                                                      ; - java.lang.StringLatin1::charAt@12 (line 47)
                                                                      ; - java.lang.String::charAt@12 (line 1555)
                                                                      ; - org.sample.strings.CharAt::charAtLatin1@12 (line 40)
                                                                      ; - org.sample.strings.jmh_generated.CharAt_charAtLatin1_jmhTest::charAtLatin1_avgt_jmhStub@17 (line 190)
  22.93%    0x00007fdb00304df3:   addq              $1, %rbx            ;*getfield isDone {reexecute=0 rethrow=0 return_oop=0}
                                                                      ; - org.sample.strings.jmh_generated.CharAt_charAtLatin1_jmhTest::charAtLatin1_avgt_jmhStub@30 (line 192)
            0x00007fdb00304df7:   movq              0x450(%r15), %r10   ; ImmutableOopMap {r8=Oop rcx=Oop rdi=Oop r13=Oop }
                                                                      ;*ifeq {reexecute=1 rethrow=0 return_oop=0}
                                                                      ; - (reexecute) org.sample.strings.jmh_generated.CharAt_charAtLatin1_jmhTest::charAtLatin1_avgt_jmhStub@33
            0x00007fdb00304dfe:   testl             %eax, (%r10)        ;   {poll}
            0x00007fdb00304e01:   testl             %edx, %edx
```

PGO Run

```
$ mvn package -Dpgo
...
====================================================================================================
GraalVM Native Image: Generating 'benchmarks' (executable)...
====================================================================================================
[1/8] Initializing...
(2.7s @ 0.19GB)
 Java version: 21.0.5+9-LTS, vendor version: Oracle GraalVM 21.0.5+9.1
 Graal compiler: optimization level: 2, target machine: x86-64-v3, PGO: instrument
...
Finished generating 'benchmarks' in 1m 2s.

Native Image Bundles: Bundle build output written to /root/1/fosdem/pgo/target/benchmarks.output
```

PGO Run

```
$ java -jar target/benchmarks.jar charAtLatin1

# JMH version: fibula:999-SNAPSHOT
# VM version: JDK 21.0.5, Substrate VM, Oracle GraalVM 21.0.5+9.1
# *** WARNING: This VM is not supported by JMH. The produced benchmark data can be completely wrong.
# VM invoker: target/benchmarks.output/default/benchmarks
...
# Run progress: 0.00% complete, ETA 00:00:20
# PGO: Instrumented Warmup Fork
# Warmup Fork: 1 of 1
...
Iteration   5: 2.459 ns/op

# PGO: Rebuild bundle with profiling data
# PGO: Rebuild native from bundle

# Run progress: 50.00% complete, ETA 00:00:10
# Fork: 1 of 1
...
Iteration   5: 1.443 ns/op

Benchmark            Mode  Cnt  Score   Error  Units
CharAt.charAtLatin1  avgt    5  1.446 ± 0.013  ns/op
```

PGO Profile

```
$ mvn package -Dpgo.perf
...
====================================================================================================
GraalVM Native Image: Generating 'benchmarks' (executable)...
====================================================================================================
[1/8] Initializing...
(2.7s @ 0.19GB)
 Java version: 21.0.5+9-LTS, vendor version: Oracle GraalVM 21.0.5+9.1
 Graal compiler: optimization level: 2, target machine: x86-64-v3, PGO: instrument
...
----------------------------------------------------------------------------------------------------
 3 experimental option(s) unlocked:
 - '-H:+DebugCodeInfoUseSourceMappings' (origin(s): command line)
 - '-H:-DeleteLocalSymbols' (origin(s): command line)
 - '-H:+SourceLevelDebug' (origin(s): command line)
...
Finished generating 'benchmarks' in 1m 23s.

Native Image Bundles: Bundle build output written to
/root/1/fosdem/pgo-profile/target/benchmarks.output
```

PGO Profile

```
$ java -jar target/benchmarks.jar -prof org.mendrugo.fibula.PerfDwarfProfiler:events=cycles:pp
charAtLatin1
...
Secondary result "org.sample.strings.CharAt.charAtLatin1:asm":
Perf binary output saved to ./org.sample.strings.CharAt.charAtLatin1-AverageTime.perfbin
```

```
$ pahole --class_name=java.lang.String benchmarks
...
     class _z_.byte[] *          value;                   /*     4      8 */

     /* Bitfield combined with next fields */

     int                         hash;                    /*     8      4 */
     byte                        coder;                   /*    12      1 */
```

PGO Profile

```
$ perf annotate -i ./org.sample.strings.CharAt.charAtLatin1-AverageTime.perfbin

_ZN60org.sample.strings.jmh_generated.CharAt_charAtLatin1_jmhTest42charAtLatin1_AverageTime%%Hot_Method_Key_6EJP43org.openjdk.jmh.r
esults.BenchmarkTaskResultP35org.openjdk.jmh.runner.InfraControlP34org.openjdk.jmh.infra.ThreadParams
/root/1/fosdem/pgo-profile/target/benchmarks-optimized.output/default/benchmarks [
...
     0.38 │ 380:   cmpl        $0x0,0x4(%r14,%rbp,8)
     0.73 │      ↓ je          b21
...
     0.40 │ 3ed:   mov         0x4(%rcx),%ebp                   <--- copy CharAt.values String[] to ebp
                                                                <--- rcx holds CharAt object instance
     0.29 │        test        %ebp,%ebp                        <--- null check CharAt.values
          │      ↓ je          c57
     0.62 │        cmpl        $0x0,0x4(%r14,%rbp,8)            <--- out of bounds check
     0.41 │      ↓ je          c66
     0.42 │        mov         0x8(%r14,%rbp,8),%ebp            <--- copy CharAt.values[0] String into ebp
     0.48 │        test        %ebp,%ebp                        <--- null pointer check on String
          │      ↓ je          c7e
```

PGO Profile

```
$ perf annotate -i ./org.sample.strings.CharAt.charAtLatin1-AverageTime.perfbin

_ZN60org.sample.strings.jmh_generated.CharAt_charAtLatin1_jmhTest42charAtLatin1_AverageTime%%Hot_Method_Key_6EJP43org.openjdk.jmh.r
esults.BenchmarkTaskResultP35org.openjdk.jmh.runner.InfraControlP34org.openjdk.jmh.infra.ThreadParams
/root/1/fosdem/pgo-profile/target/benchmarks-optimized.output/default/benchmarks [
...
     0.42 │        mov        %rax,0x130(%rsp)
     0.78 │        mov        0x4(%r14,%rbp,8),%r8d              <--- copy String.value byte[] into r8d
     0.56 │        movsbl     0xc(%r14,%rbp,8),%ebp              <--- copy String.coder into ebp
     0.50 │        mov        0x8(%rcx),%r12d                    <--- copy CharAt.charAtIndex to r12d
                                                                 <--- rcx holds CharAt object instance
     0.43 │        lea        (%r14,%r8,8),%r13                  <--- calculate String.value byte[] address and put into r13
     0.37 │        lea        0x2(%r11),%rax                     <--- rax = r11 (operations) + 2
     0.50 │        test       %bpl,%bpl                          <--- String.coder check for latin1 or utf16
          │      ↓ jne        7fa
     0.59 │        test       %r8d,%r8d                          <--- null check String.value byte[]
          │      ↓ je         c8c
     0.54 │        mov        0x4(%r14,%r8,8),%ebp
     0.59 │        cmp        %r12d,%ebp                         <--- check index? (index r12d, ebp String.value bute[] length)
          │      ↓ jbe        c92
     1.06 │        mov        %r12d,%ebp                         <--- copy CharAt.charAtIndex to ebp
     0.50 │        movzbl     0x8(%r13,%rbp,1),%ebp              <--- copies String.value[index] to ebp (byte[] r13, index rbp)
     0.29 │        cmpb       $0x0,0xb0(%rbx)                    <--- is done check
```

Thank you

github.com/galderz/fibula

github.com/galderz/fibula-show/tree/main/2501-fosdem