

JDK 12

released 2019-03-19



Java User Group Ingolstadt e.V.

189: SHENANDOAH: A LOW-PAUSE-TIME GARBAGE COLLECTOR (EXPERIMENTAL)

Add a new garbage collection (GC) algorithm named Shenandoah which reduces GC pause times by doing evacuation work concurrently with the running Java threads. Pause times with Shenandoah are independent of heap size, meaning you will have the same consistent pause times whether your heap is 200 MB or 200 GB.



NOTES

- Created by Red Hat
- Minimizes 'Stop-The-World' pauses
- Handle terabytes of heap
- Nearly full concurrent execution
- Not contained in Oracle builds (but in others like AdoptOpenJDK)



FUNCTIONING

- Concurrent Marking
- Concurrent Cleanup
- Concurrent Evacuation
- Concurrent Update References

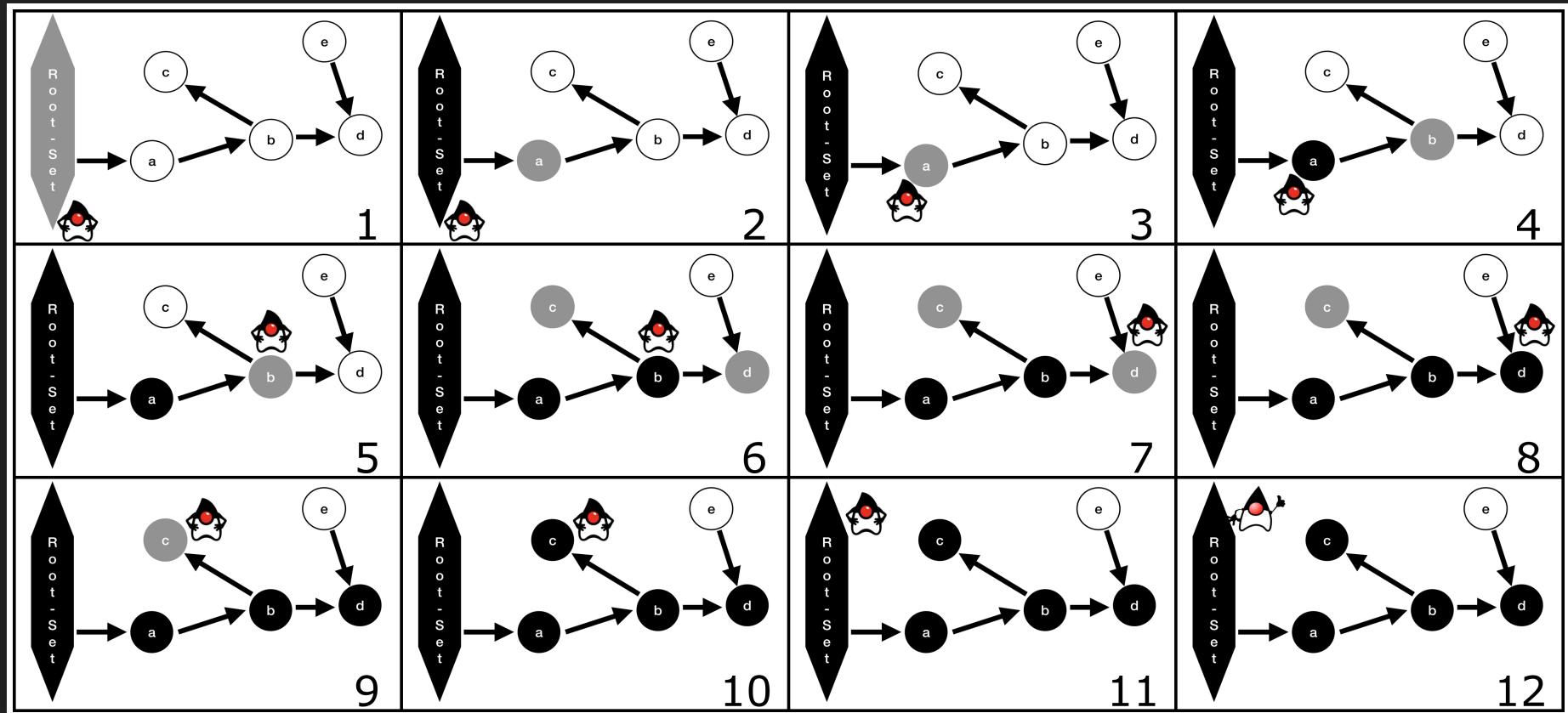


CONCURRENT MARKING

- Init Mark: Stop the World to identify the Root-Set
- Traverse Root set, perform Tri-Color-Algorithm:
 - 'Black': Alive, references scanned
 - 'Grey': Alive, references not scanned yet
 - 'White': Unreachable or not alive ⇒ Garbage
- Intercept Reference creation/deletion
 - Mark as 'Grey'



TRI-COLOR-ALGORITHM

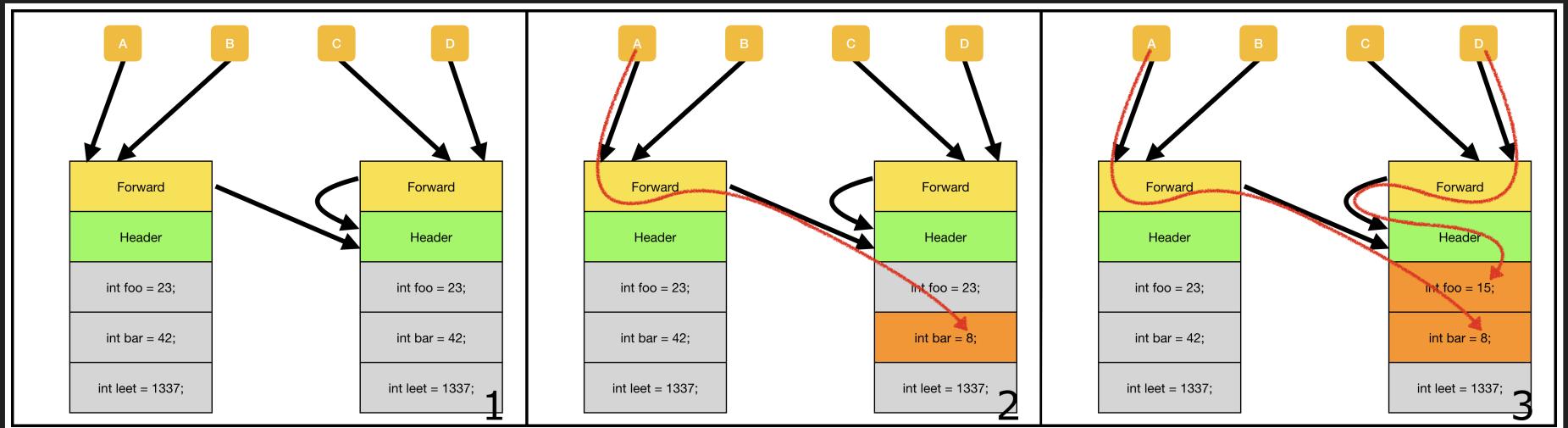
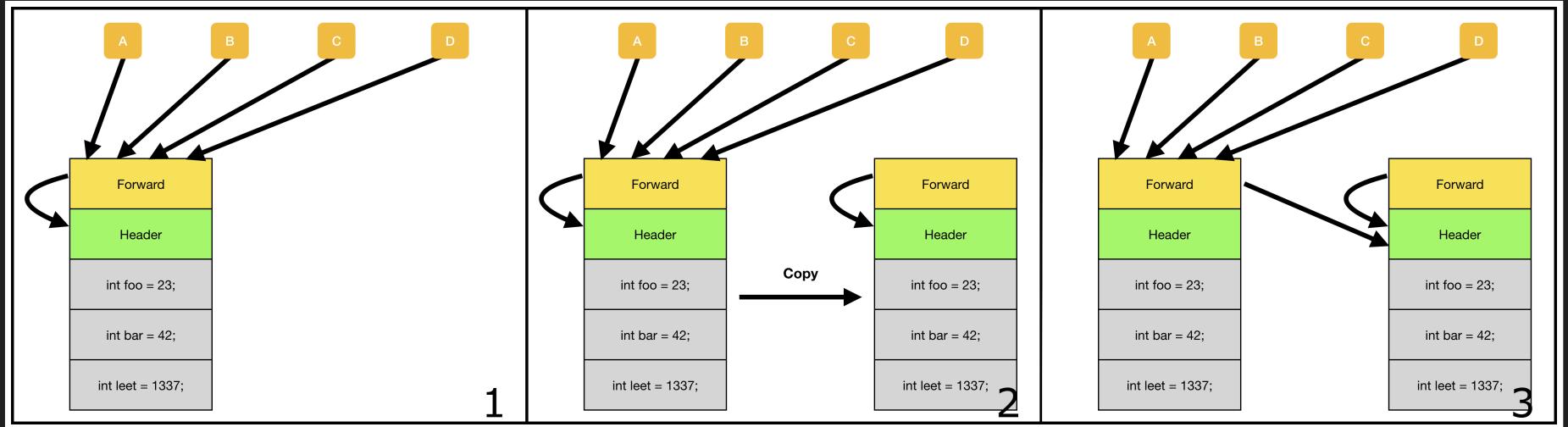


CONCURRENT EVACUATION

- Clear garbage only regions
- Evacuate Regions with few living objects
- Forward/Brooks pointer



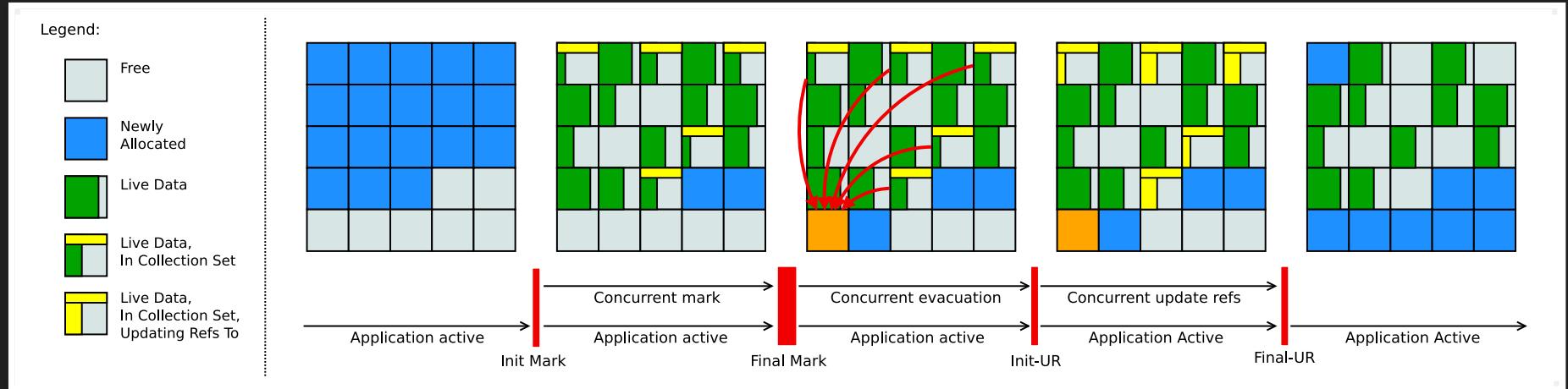
BROOKS BARRIER



CONFIGURATION

```
-XX:+UnlockExperimentalVMOptions  
-XX:+UseShenandoahGC
```

LIVECYCLE



CONCLUSION

- ParallelGC:
 - Handle STW pauses
- Shenandoah:
 - Scale Hardware



230: MICROBENCHMARK SUITE

Add a basic suite of microbenchmarks to the JDK source code, and make it easy for developers to run existing microbenchmarks and create new ones.



325: SWITCH EXPRESSIONS (PREVIEW)

Extend the switch statement so that it can be used as either a statement or an expression, and that both forms can use either a "traditional" or "simplified" scoping and control flow behavior. These changes will simplify everyday coding, and also prepare the way for the use of pattern matching (JEP 305) in switch. This will be a preview language feature.



SYNTAX (INLINE)

Switch expressions are able to return values

```
var result = switch(value) {  
    case VALID -> "It's ok dude";  
    case INVALID, UNKNOWN -> "Better think about it";  
};
```



SYNTAX (BLOCKS)

```
var result = switch(integer) {  
    case 0,1 -> integer + " is a bit too low."  
    case 2,3,5,7,11,13,17,19 -> integer + " is really prime!"  
    case 4,6,8,9,10,12,14,15,16,20 -> integer + " isn't prime!  
    default -> {  
        if (IntStream.range(21, integer).anyMatch(nut -> integ  
            break integer + " isn't that nice, pardon."  
        } else {  
            break integer + " is a really high prime";  
        }  
    }  
};
```



TYPE INFERENCE

```
Number result = switch(integer) {  
    case 2,3 -> integer *2;  
    case 4,5 -> BigDecimal.valueOf(integer);  
    case 6,7 -> integer / 2d;  
    default -> (long) integer;  
};
```

```
Serializable result = switch(integer) {  
    case 0,1 -> "A bit";  
    case 2,3 -> integer *2;  
    case 4,5 -> BigDecimal.valueOf(integer);  
    default -> (long) integer;  
};
```

```
ConstantDesc result = switch(integer) {  
    case 0,1 -> "A bit";  
    case 2,3 -> integer *2;  
    default -> (long) integer;  
};
```



334: JVM CONSTANTS API

Introduce an API to model nominal descriptions of key class-file and run-time artifacts, in particular constants that are loadable from the constant pool.

`java.lang.constant`



`java.lang.constant.Constable`

Represents a type which is constable. A constable type is one whose values are constants that can be represented in the constant pool of a Java classfile as described in JVMS 4.4, and whose instances can describe themselves nominally as a ConstantDesc.

```
public interface Constable
```

Some constable types have a native representation in the constant pool:

String, Integer, Long, Double



java.lang.constant.ConstantDesc

Represents a type which is constable. A constable type is one whose values are constants that can be represented in the constant pool of a Java classfile as described in JVMS 4.4, and whose instances can describe themselves nominally as a ConstantDesc.

```
public interface Constable
```



340: ONE AARCH64 PORT, NOT TWO

Remove all of the sources related to the arm64 port while retaining the 32-bit ARM port and the 64-bit aarch64 port.



341: DEFAULT CDS ARCHIVES

Enhance the JDK build process to generate a class data-sharing (CDS) archive, using the default class list, on 64-bit platforms.



344: ABORTABLE MIXED COLLECTIONS FOR G1

Make G1 mixed collections abortable if they might exceed the pause target.



Java User Group Ingolstadt e.V.

346: PROMPTLY RETURN UNUSED COMMITTED MEMORY FROM G1

Enhance the G1 garbage collector to automatically return Java heap memory to the operating system when idle.



STANDARD LIBRARY API CHANGES (EXCERPT)

	Changed	Added	Removed	Total
Types	343	21	0	4152
Members	486	286	9	49882



java.lang.Character

Constant for the "Chess Symbols" Unicode character block.

```
public static final Character.UnicodeBlock CHESS_SYMBOLS
```



java.lang.Class

Returns a Class for an array type whose component type is described by this Class.

```
public Class<?> arrayType()
```

Returns a nominal descriptor for this instance, if one can be constructed, or an empty Optional if one cannot be.

```
public Optional<ClassDesc> describeConstable()
```



java.lang.String

Adjusts the indentation of each line of this string based on the value of n, and normalizes line termination characters.

```
public String indent(int n)
```

This method allows the application of a function to this string. The function should expect a single String argument and produce an R result.

```
public <R> R transform(Function<? super String, ? extends R> f)
```



java.nio.file.Files

Finds and returns the position of the first mismatched byte in the content of two files, or -1L if there is no mismatch. The position will be in the inclusive range of 0L up to the size (in bytes) of the smaller file.

```
public static long mismatch(Path path, Path path2)
```



java.text.CompactNumberFormat

A concrete subclass of NumberFormat that formats a decimal number in its compact form. The compact number formatting is designed for the environment where the space is limited, and the formatted string can be displayed in that limited space.

```
public final class CompactNumberFormat extends NumberFormat
```



java.util.concurrent.CompletableFuture

Returns a new [CompletableFuture|CompletionStage] that, when this stage completes exceptionally, is executed with this stage's exception as the argument to the supplied function

```
default CompletionStage<T> exceptionallyAsync(  
    Function<Throwable, ? extends T> fn)
```



java.util.concurrent.CompletionStage

Returns a new [CompletableFuture|CompletionStage] that, when this stage completes exceptionally, is composed using the results of the supplied function applied to this stage's exception.

```
default CompletionStage<T> exceptionallyComposeAsync(  
    Function<Throwable, ? extends CompletionStage<T>> fn)
```



java.util.stream.Collectors

Returns a Collector that is a composite of two downstream collectors. Every element passed to the resulting collector is processed by both downstream collectors, then their results are merged using the specified merge function into the final result.

```
public static <T,R1,R2,R> Collector<T,?,R> teeing(  
    Collector<? super T, ?, R1> downstream1,  
    Collector<? super T, ?, R2> downstream2,  
    BiFunction<? super R1, ? super R2, R> merger)
```



java.net.ssl.HttpsURLConnection

Returns an Optional containing the SSLSession in use on this connection. Returns an empty Optional if the underlying implementation does not support this method.

```
public Optional<SSLSession> getSSLSession()
```



JDK 13

2019-06-13 Rampdown Phase One (fork)

2019-07-18 Rampdown Phase Two

2019-08-08 Initial Release Candidate

2019-08-22 Final Release Candidate

2019-09-17 General Availability



Java User Group Ingolstadt e.V.

RESOURCES

- <https://jaxenter.de/shenandoah-garbage-collector-76213>
- <https://wiki.openjdk.java.net/display/shenandoah/Mail>

