Garbage Collection

Agenda

- Brief history
- Heap and pointers
- Terminology
- Garbage collection algorithms
- Two regions collectors
- G1

Brief history

- Lisp 1960

Heap and pointers

- GC purpose

Heap and pointers

- GC purpose
  - allocating memory
  - ensuring that any referenced objects remain in memory, and
  - recovering memory used by objects that are no longer reachable from references in executing code

Heap and pointers

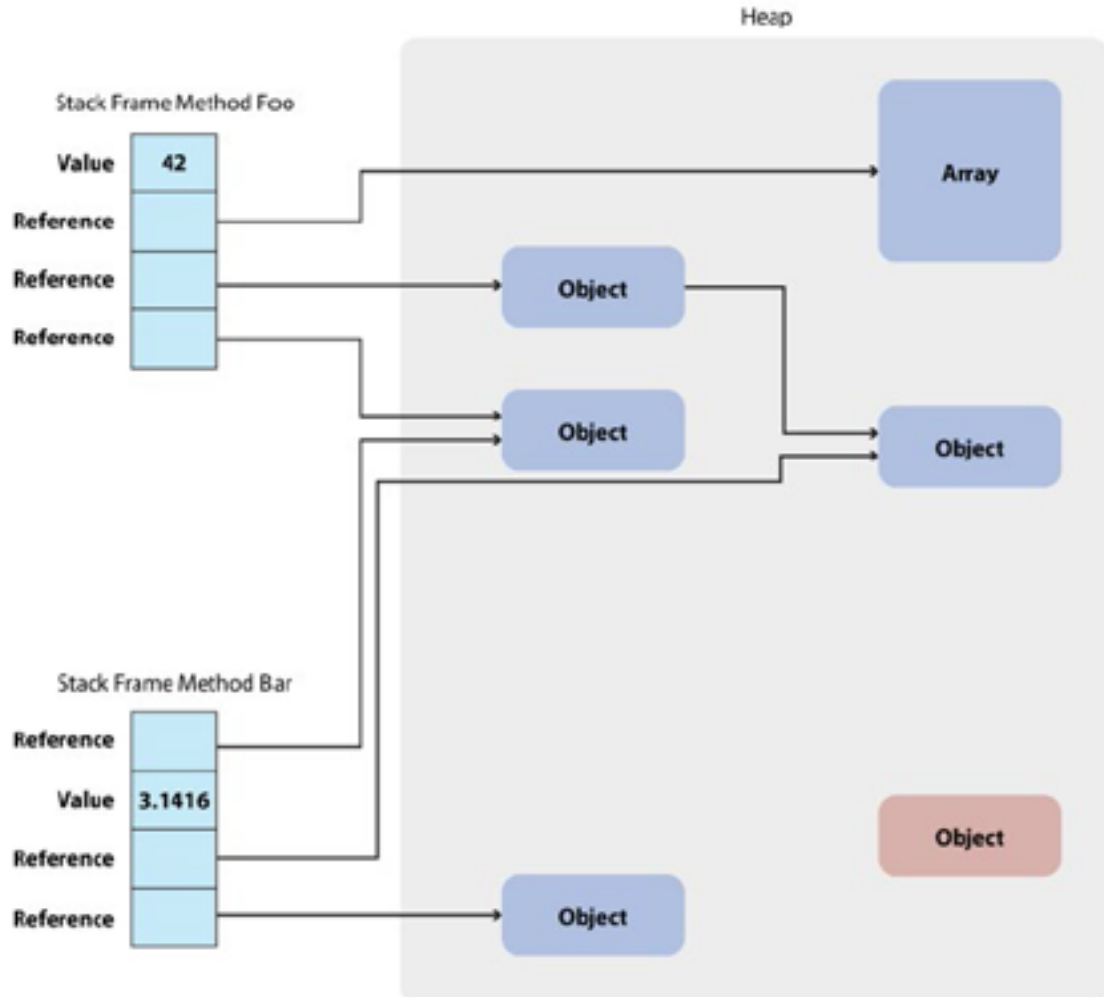
- GC purpose
- Heap

Heap and pointers

Heap and pointers

A "CollectedHeap" is an implementation of a java heap for HotSpot.
This is an abstract class: there may be many different kinds of heaps.

- SharedHeap
- GenCollectedHeap
- G1CollectedHeap
- ParallelScavengeHeap


Heap
Stack Frame Method Foo
Value
42
Reference
Reference
Reference
Array
Object
Object
Object
Stack Frame Method Bar
Reference
Value
3.1416
Reference
Reference
Object
Object

Terminology

- Single threaded

Terminology

- Single threaded
- Parallel

Terminology

- Single threaded
- Parallel
- Concurrent

Terminology

- Single threaded
- Parallel
- Concurrent
- Stop-the-world (STW)

Terminology

- Single threaded
- Parallel
- Concurrent
- Stop-the-world (STW)
- Monolithic

Terminology

- Single threaded
- Parallel
- Concurrent
- Stop-the-world (STW)
- Monolithic
- Incremental

Terminology

- Single threaded
- Parallel
- Concurrent
- Stop-the-world (STW)
- Monolithic
- Incremental
- Precise
- Conservative

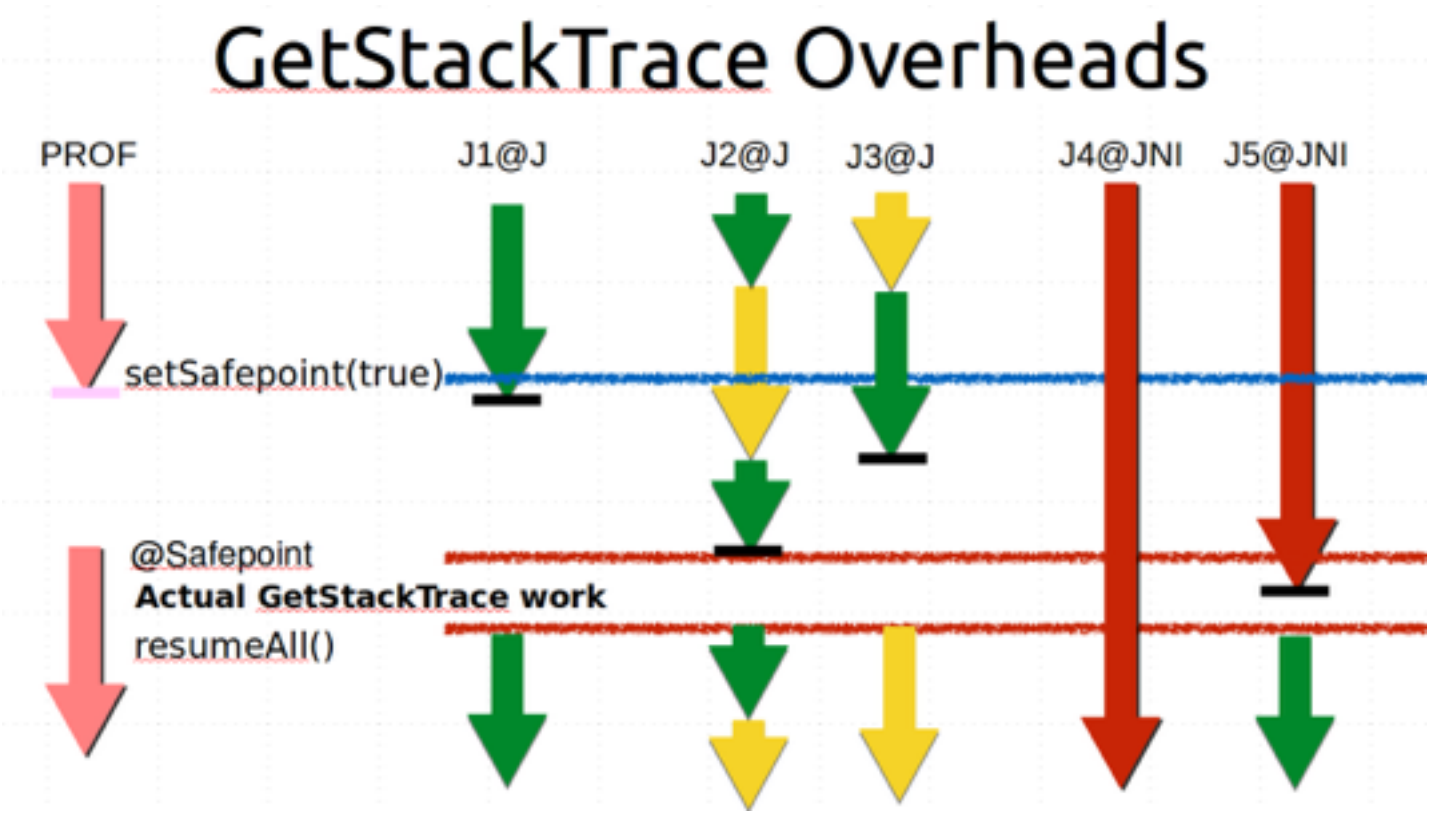
Terminology

- Safe points

Terminology

- Safe points

Terminology



Terminology

- countable loops

```
while (true) {…}
for (int k = 0; k < Integer.MAX_VALUE; k++) {…}

for (long k = 0; k < 10; k++) {…}
```

Terminology

- Safe points
- Generational collection

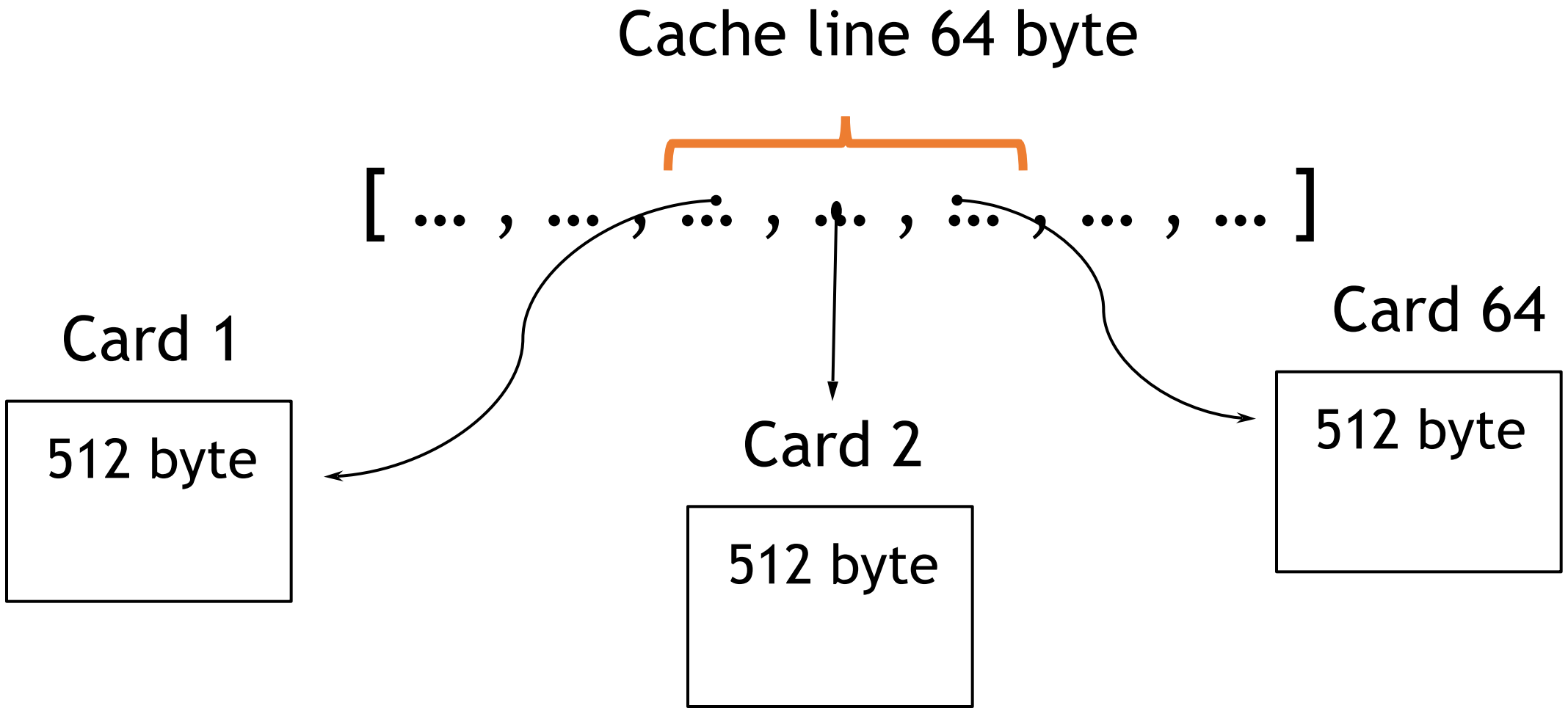
Terminology

- Safe points
- Generational collection
- Memory allocation

Terminology

- Safe points
- Generational collection
- Memory allocation
- Remembered sets and write barriers

Terminology



Terminology

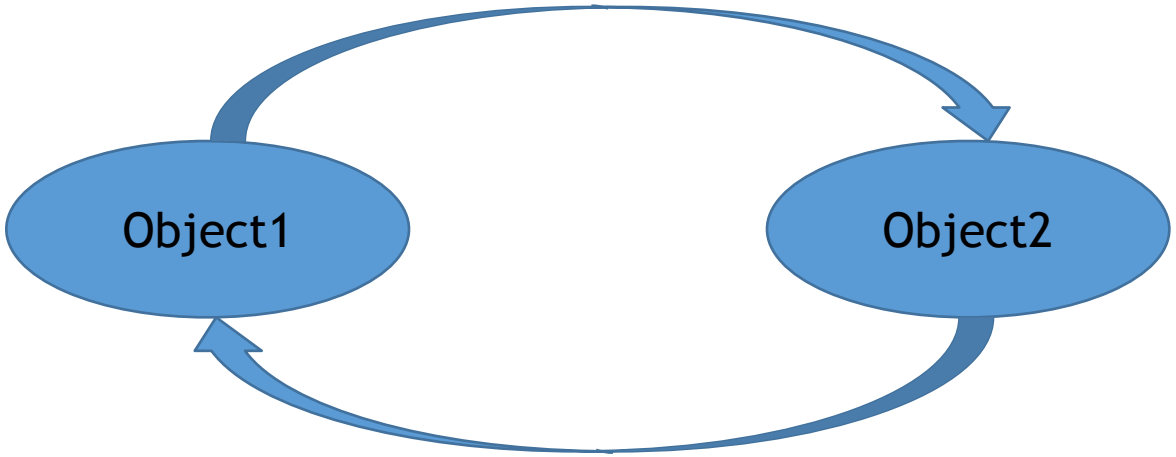
-XX:+UseCondCardMark

GC algorithms

- Reference counting

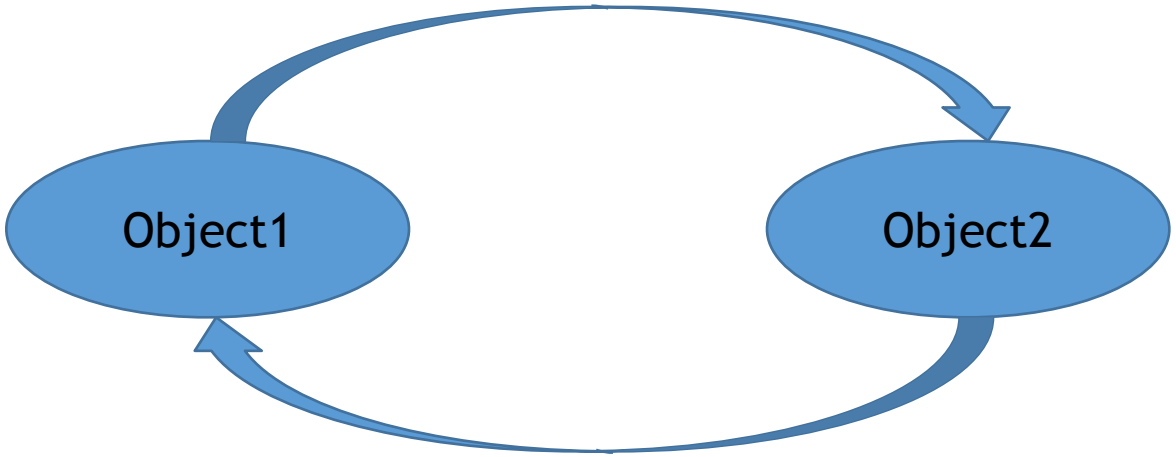
GC algorithms

- Reference counting



GC algorithms

- Reference counting



Deutsch-Bobrow algorithm

GC algorithms

- Reference counting
- Mark (trace)

GC algorithms

- Reference counting
- Mark (trace)
- Sweep

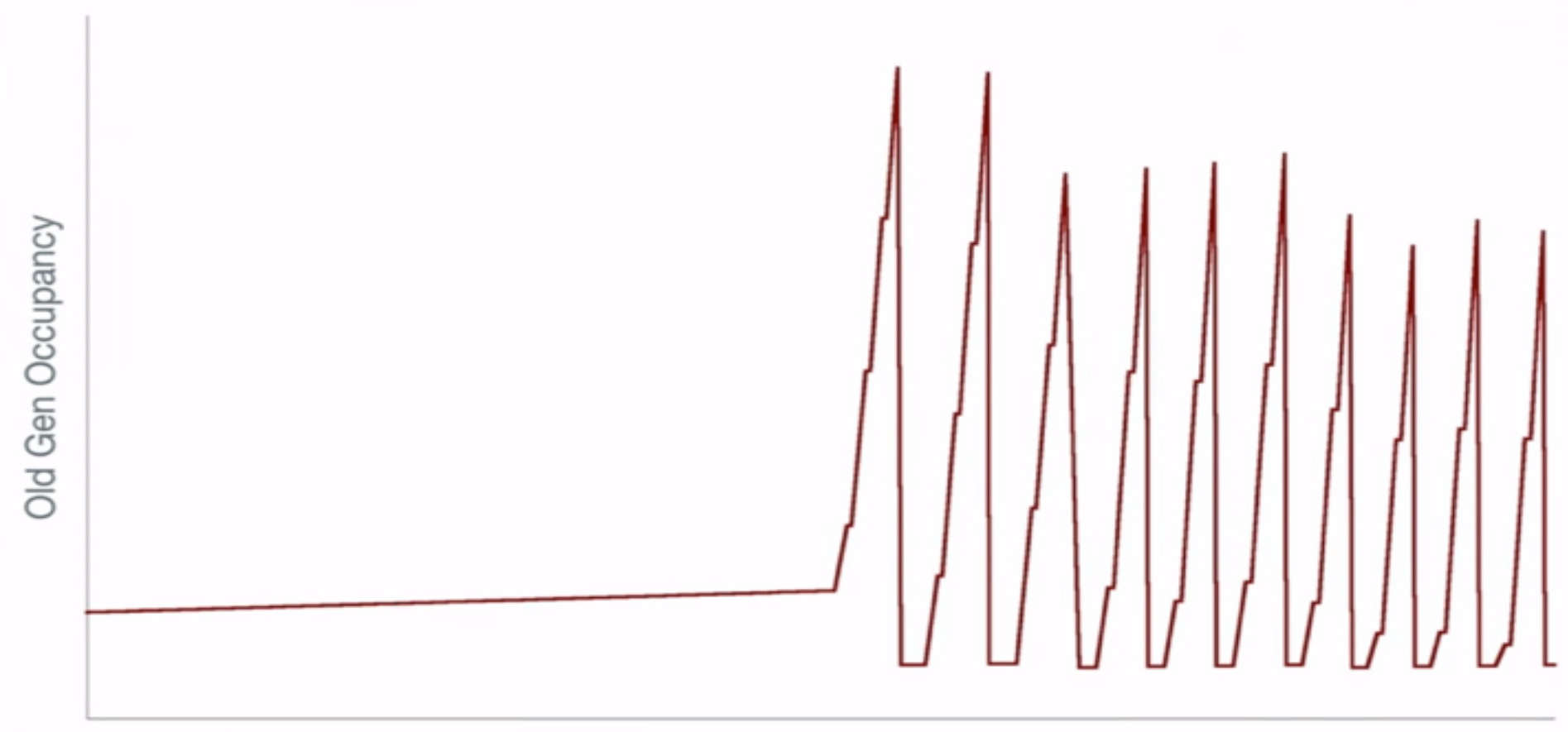
GC algorithms

- Reference counting
- Mark (trace)
- Sweep
- Compact/Relocate

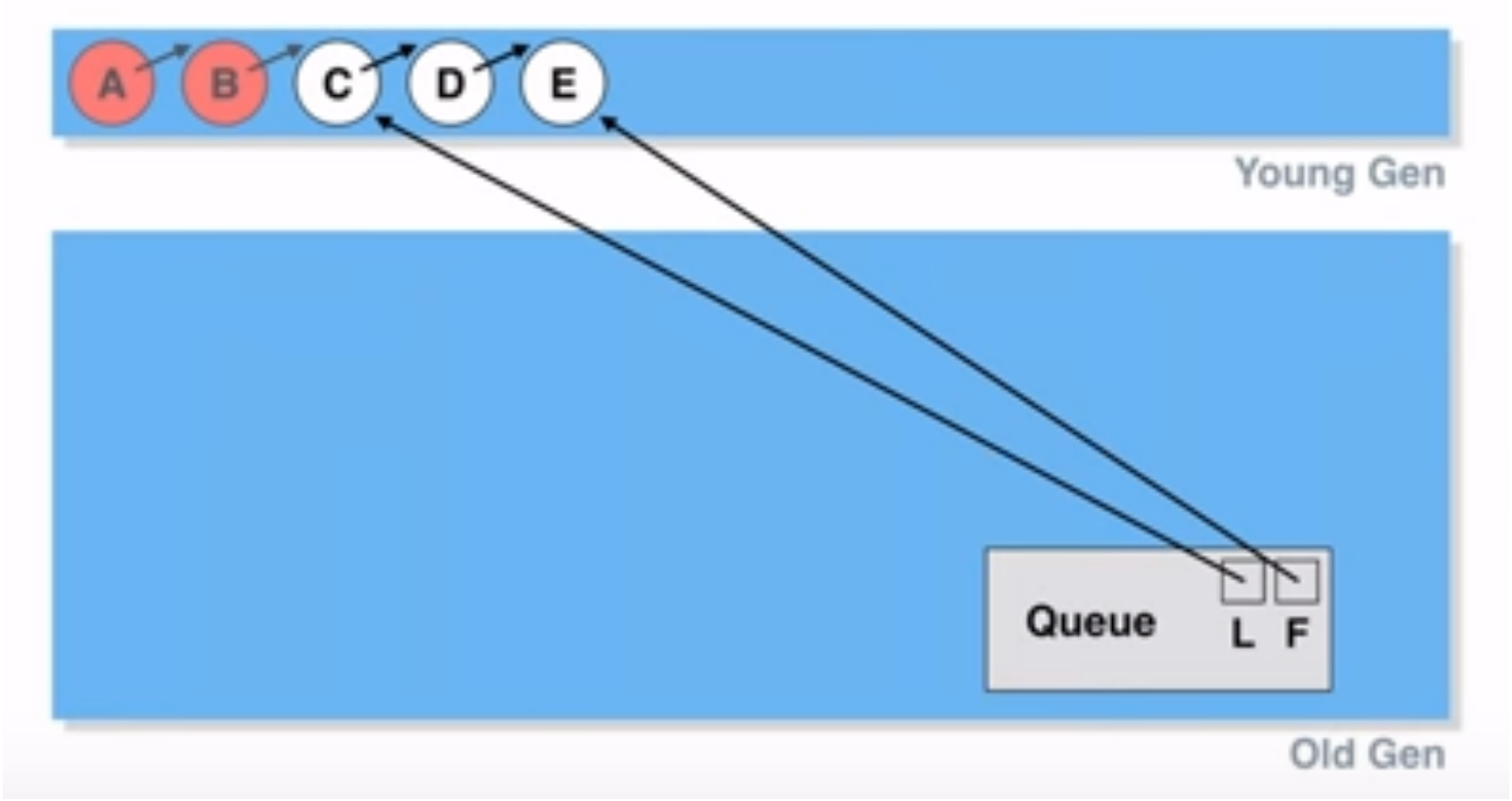
GC algorithms

- Reference counting
- Mark (trace)
- Sweep
- Compact/Relocate
- Copy

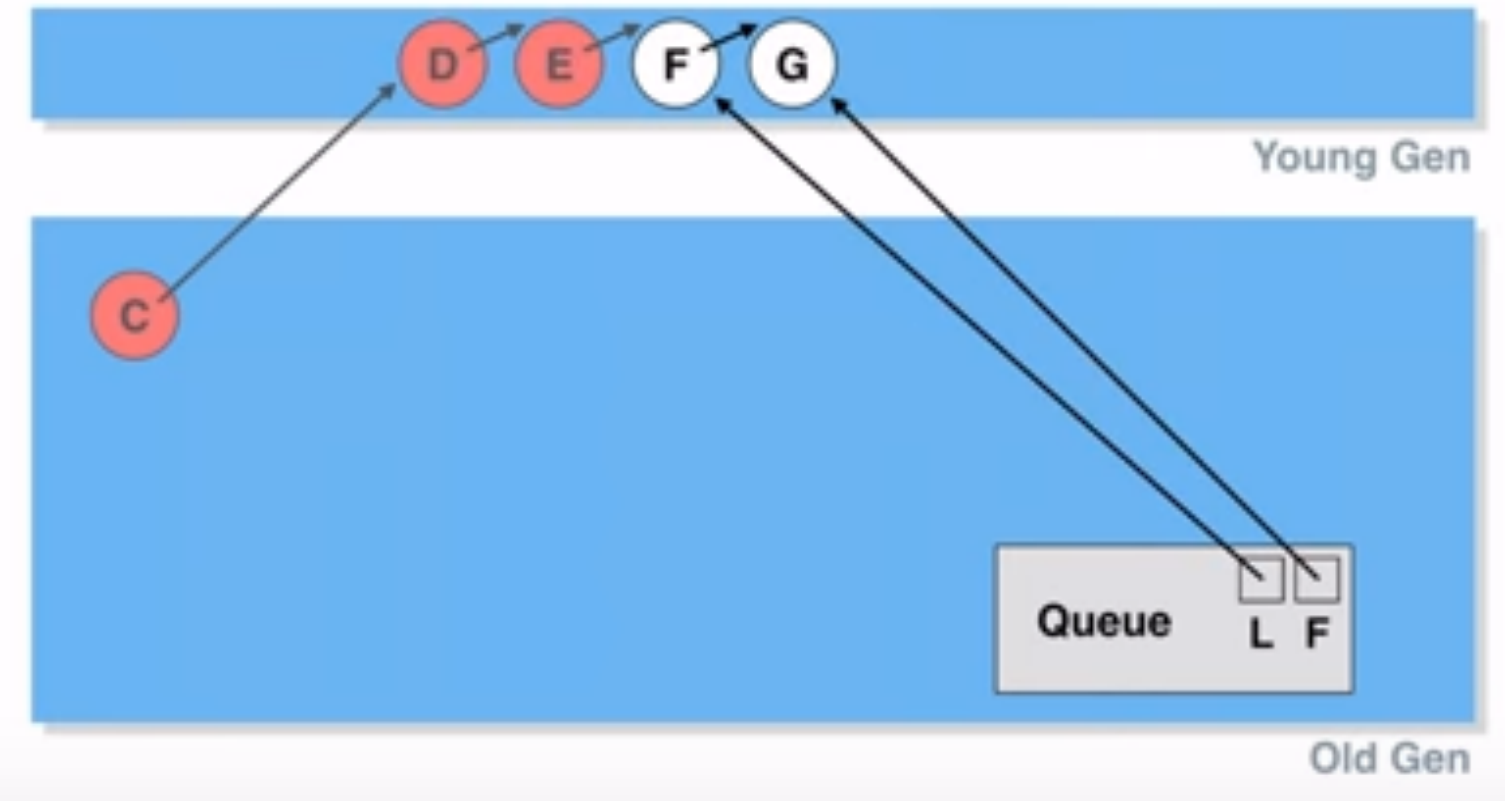
Nepotism



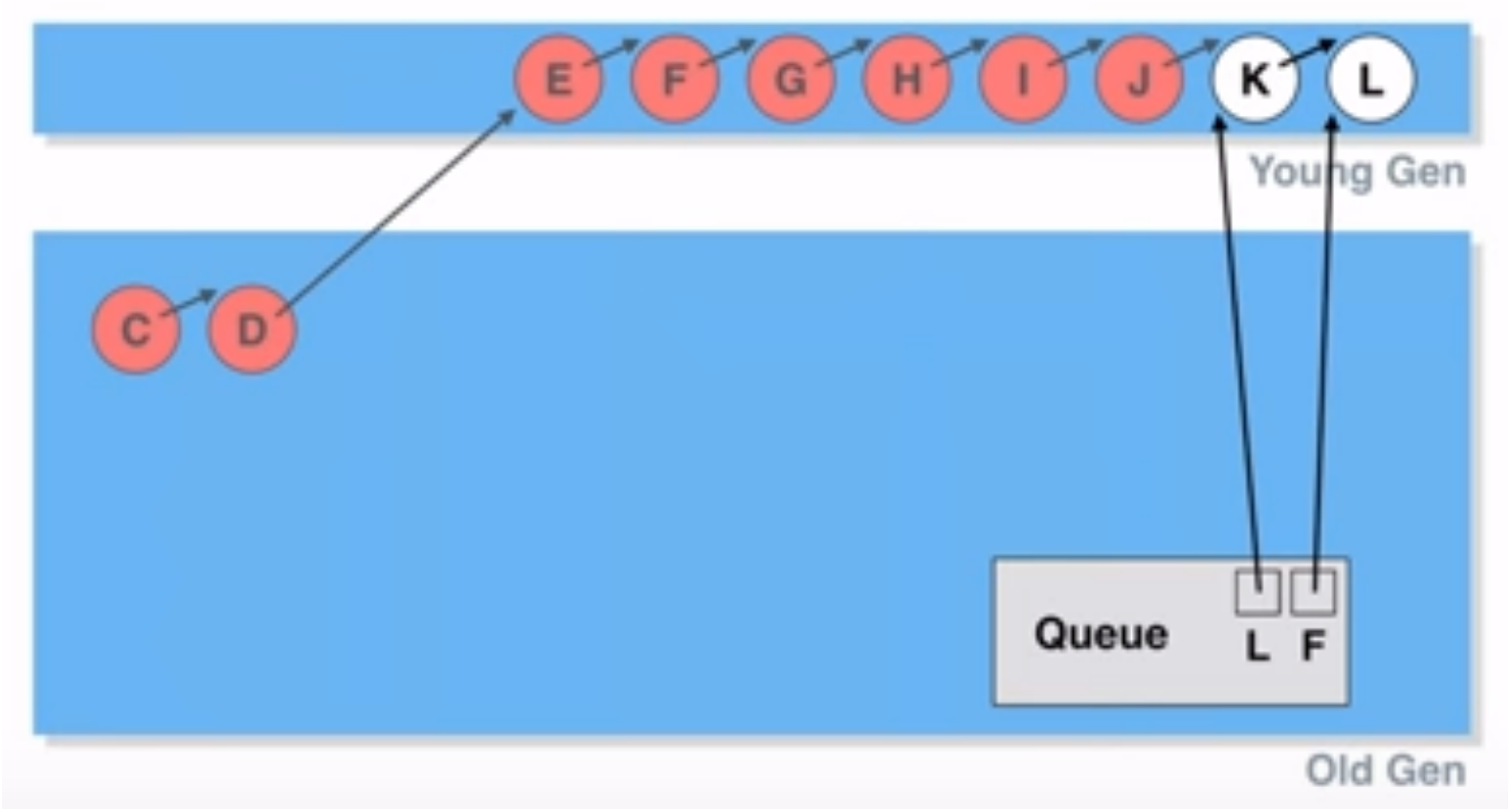
Nepotism



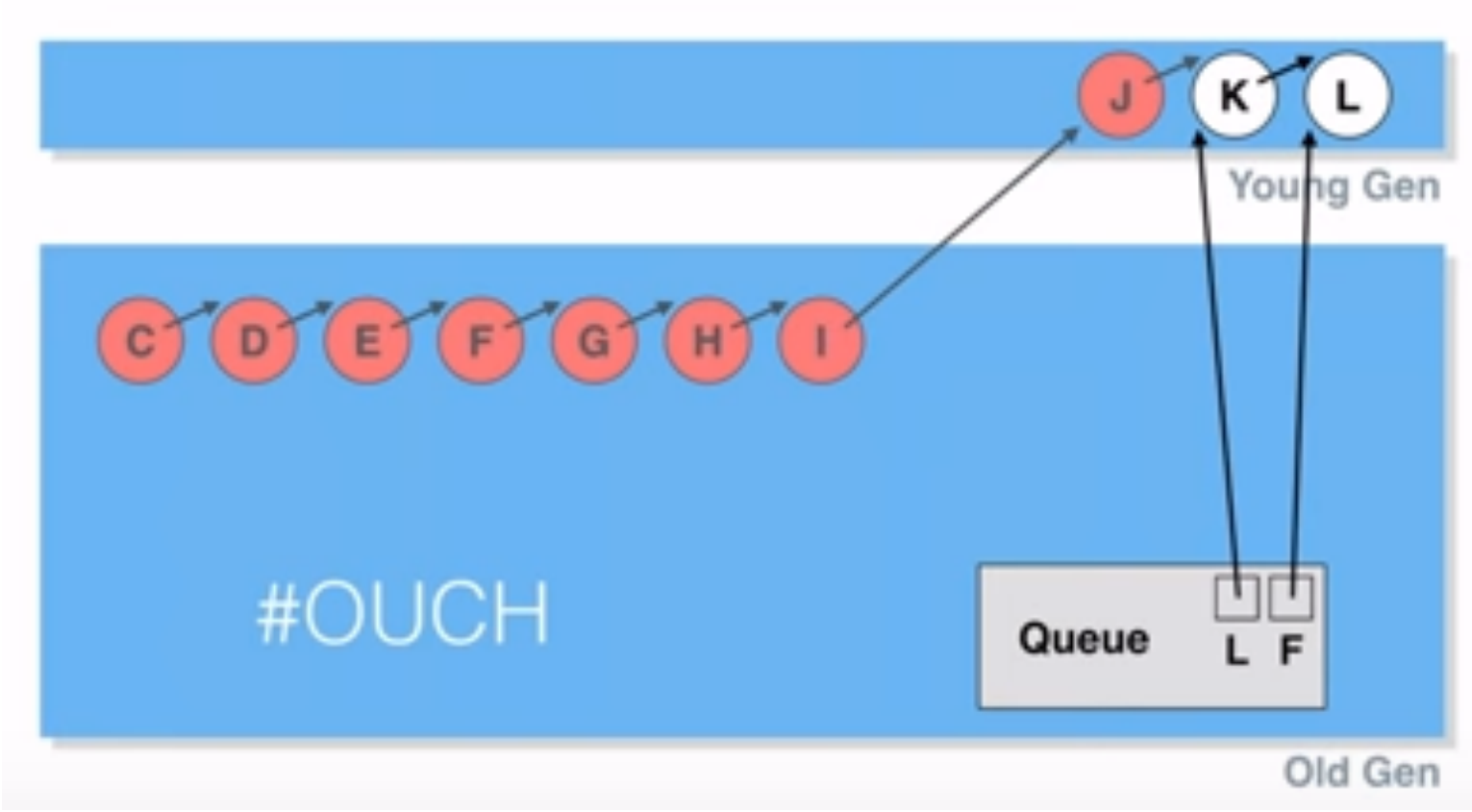
Nepotism



Nepotism



Nepotism



Escape Analysis

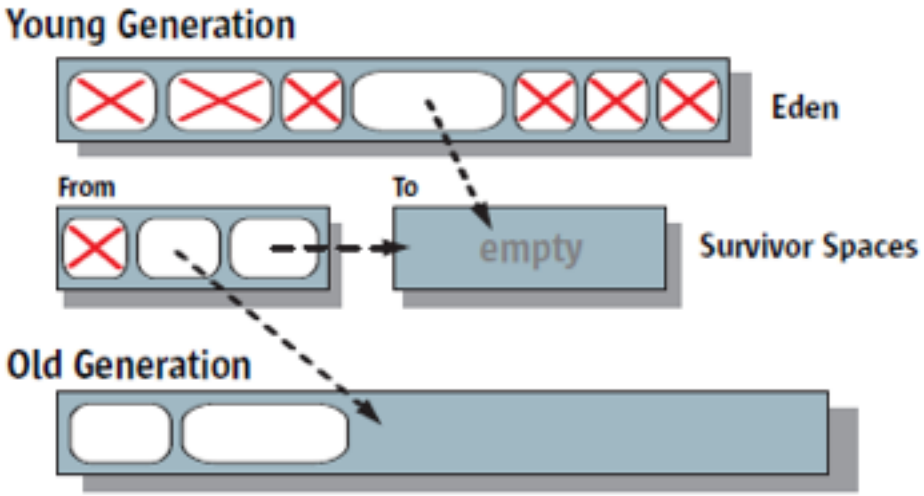
- Scalar Replacement
- Stack Allocation
- Lock Elision

Two regions collectors

- Serial collector

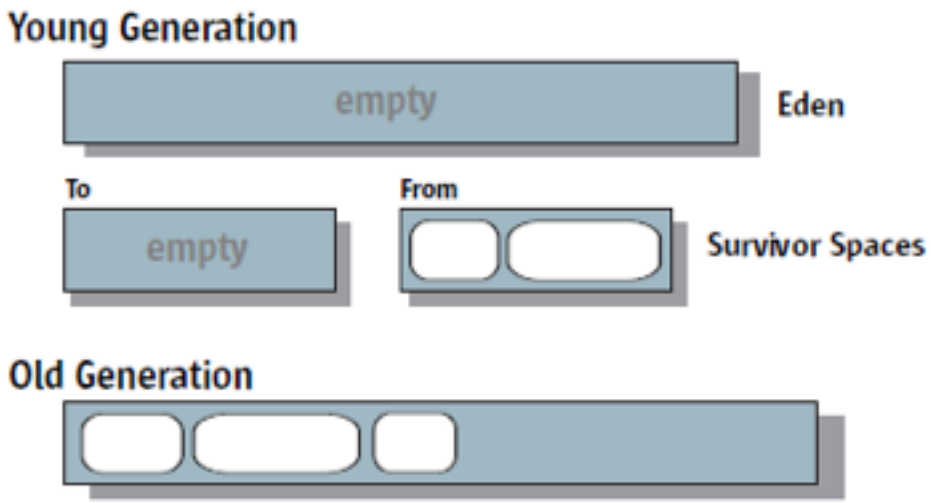
Serial collector

Young generation collection via copy



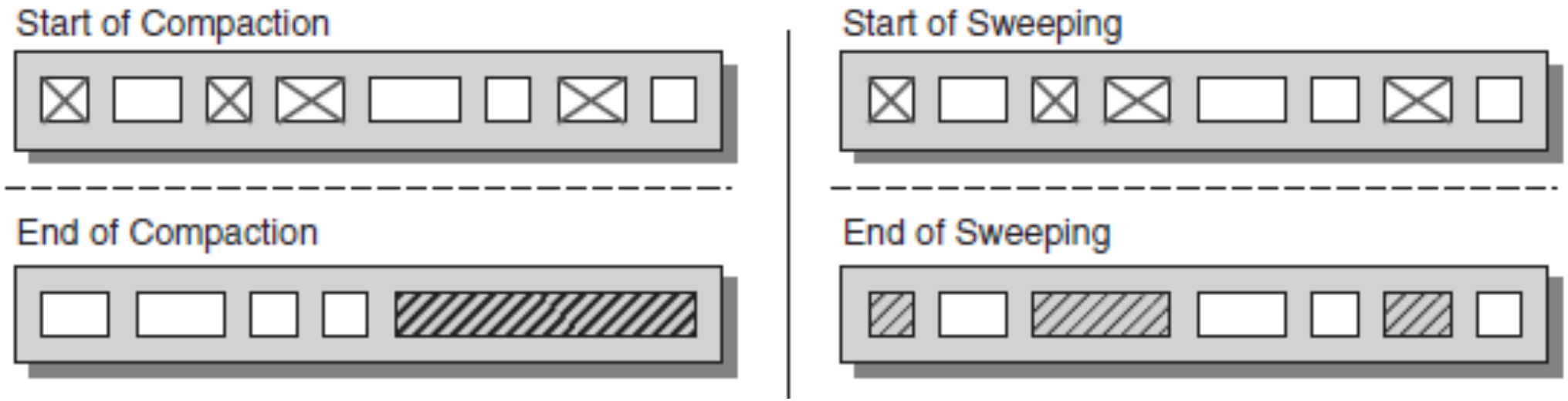
Serial collector

Young generation collection via copy



Serial collector

Old generation collection via mark/sweep/compact

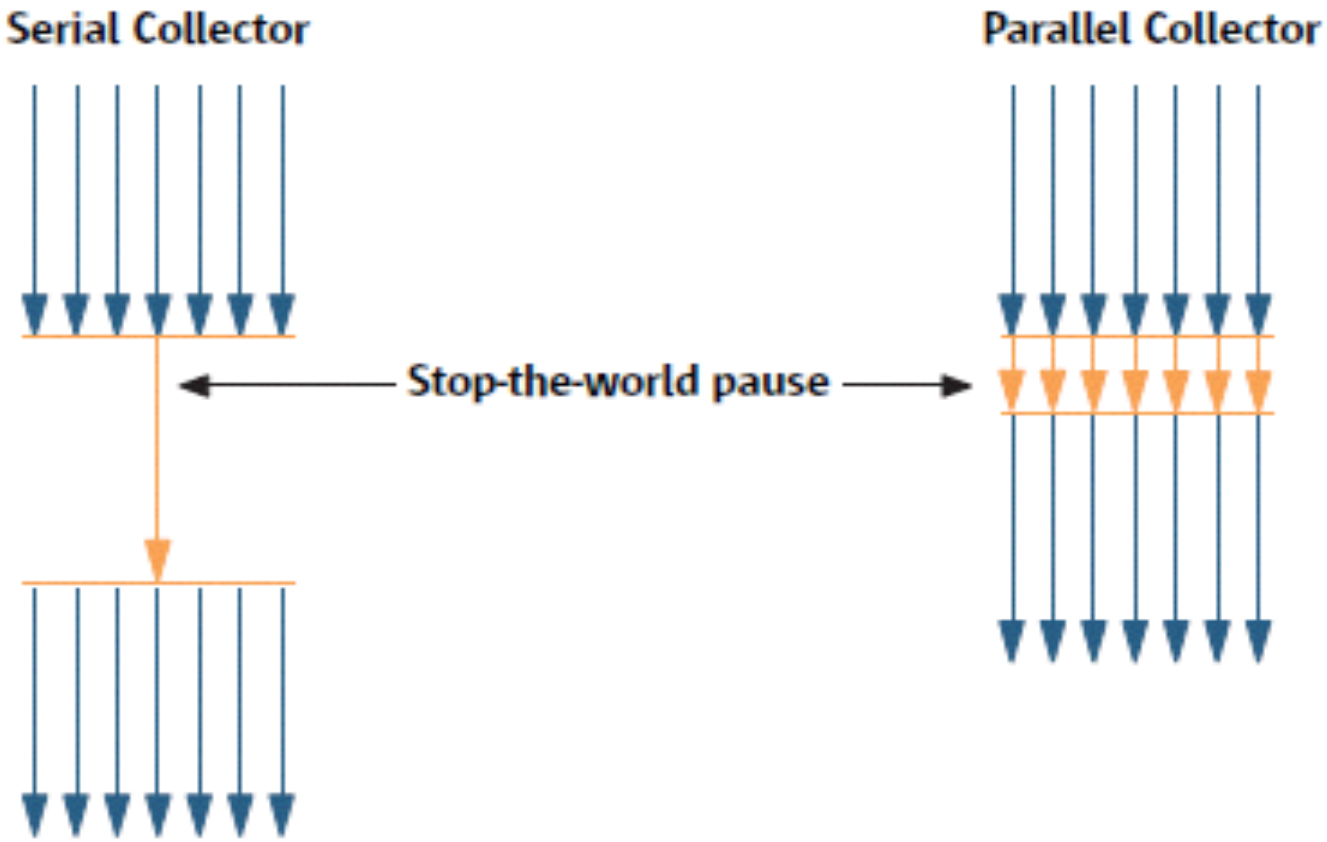


Two regions collectors

- Parallel collector

Parallel collector

Young generation collection



Parallel collector

Old generation collection is the same mark/sweep/compact as the serial collection

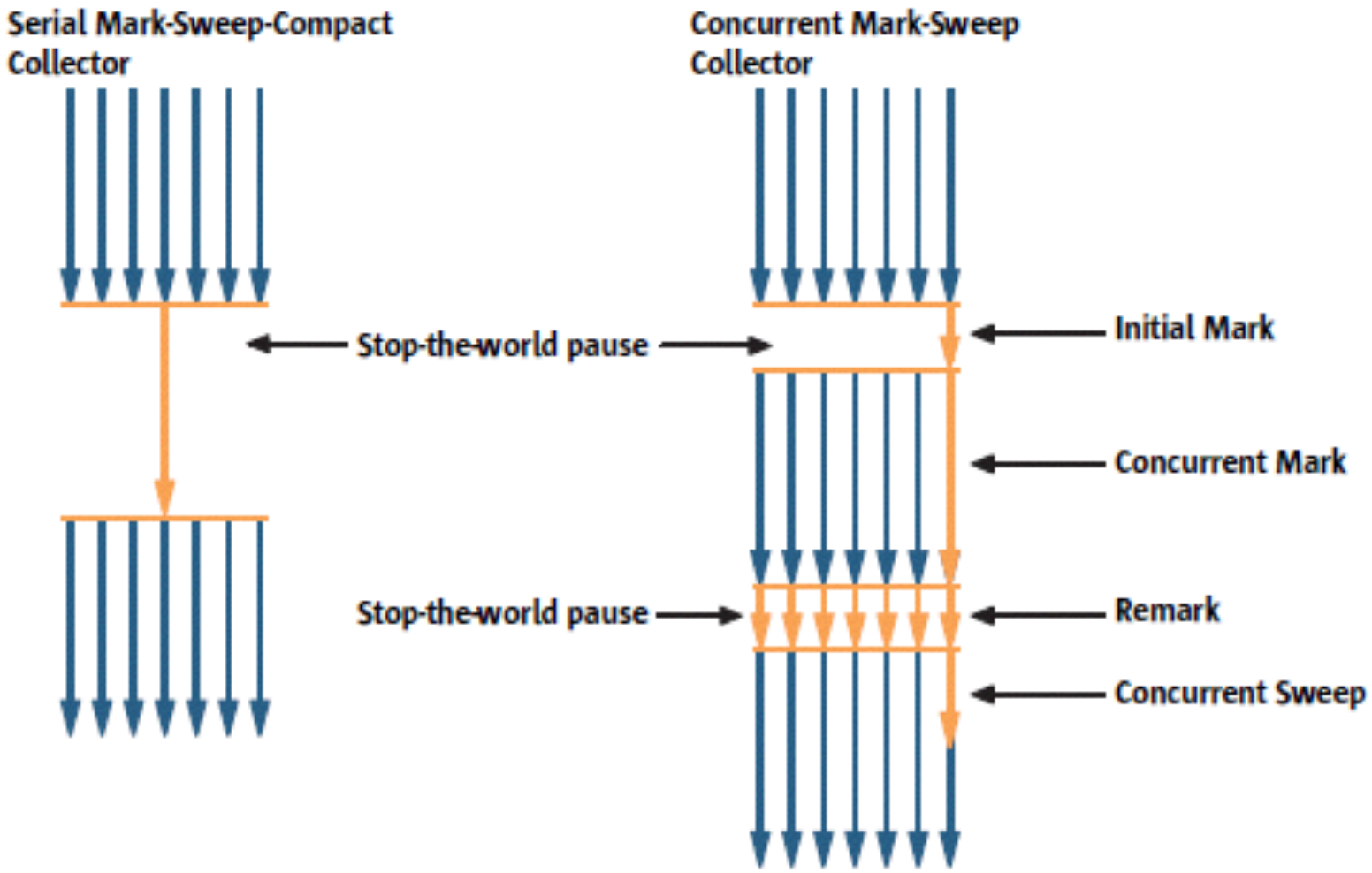
Two regions collectors

- Concurrent Mark-Sweep collector

Concurrent Mark-Sweep collector

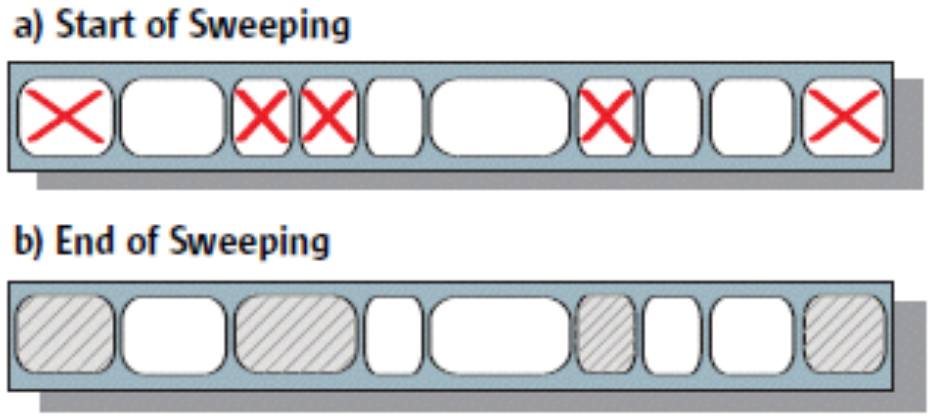
Young generation collection is the same as in parallel collector

Old generation collection



Concurrent Mark-Sweep collector

Old generation collection: sweeping no compacting



G1

Multi-region collector

G1

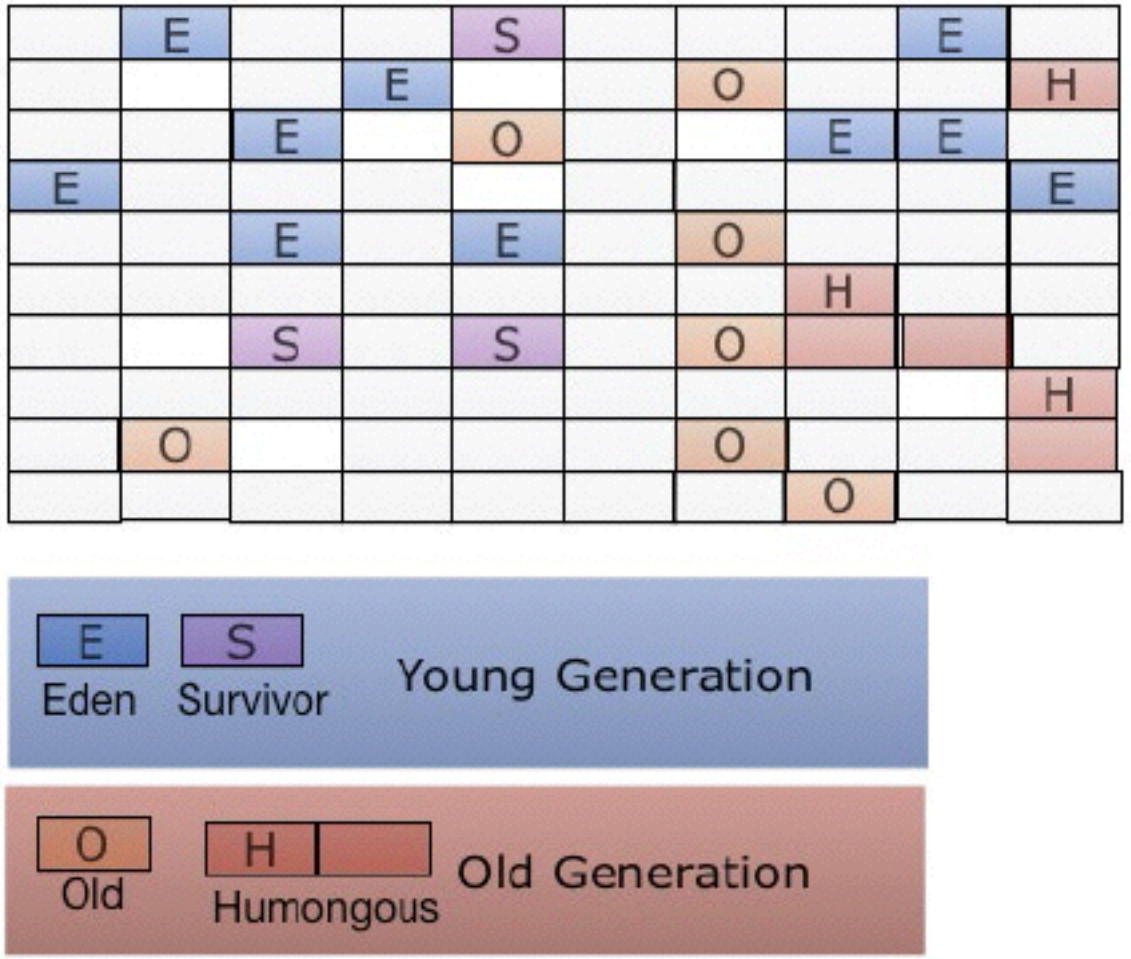
Multi-region collector

- Xmx32G
- -XX:MaxGCPauseMillis=100

G1

- G1ConcMarkStepDurationMillis
- G1ConcRSHotCardLimit
- G1ConcRSLogCacheSize
- G1ConcRefinementGreenZone
- G1ConcRefinementRedZone
- G1ConcRefinementServiceIntervalMillis
- G1ConcRefinementThreads
- G1ConcRefinementThresholdStep
- G1ConcRefinementYellowZone
- G1ConfidencePercent
- G1HeapRegionSize
- G1HeapWastePercent
- G1MixedGCCountTarget
- G1RSetRegionEntries
- G1RSetScanBlockSize
- G1RSetSparseRegionEntries
- G1RSetUpdatingPauseTimePercent
- G1RefProcDrainInterval
- G1ReservePercent
- G1SATBBufferEnqueueingThresholdPercent
- G1SATBBufferSize
- G1UpdateBufferSize
- G1UseAdaptiveConcRefinement
- UseG1GC

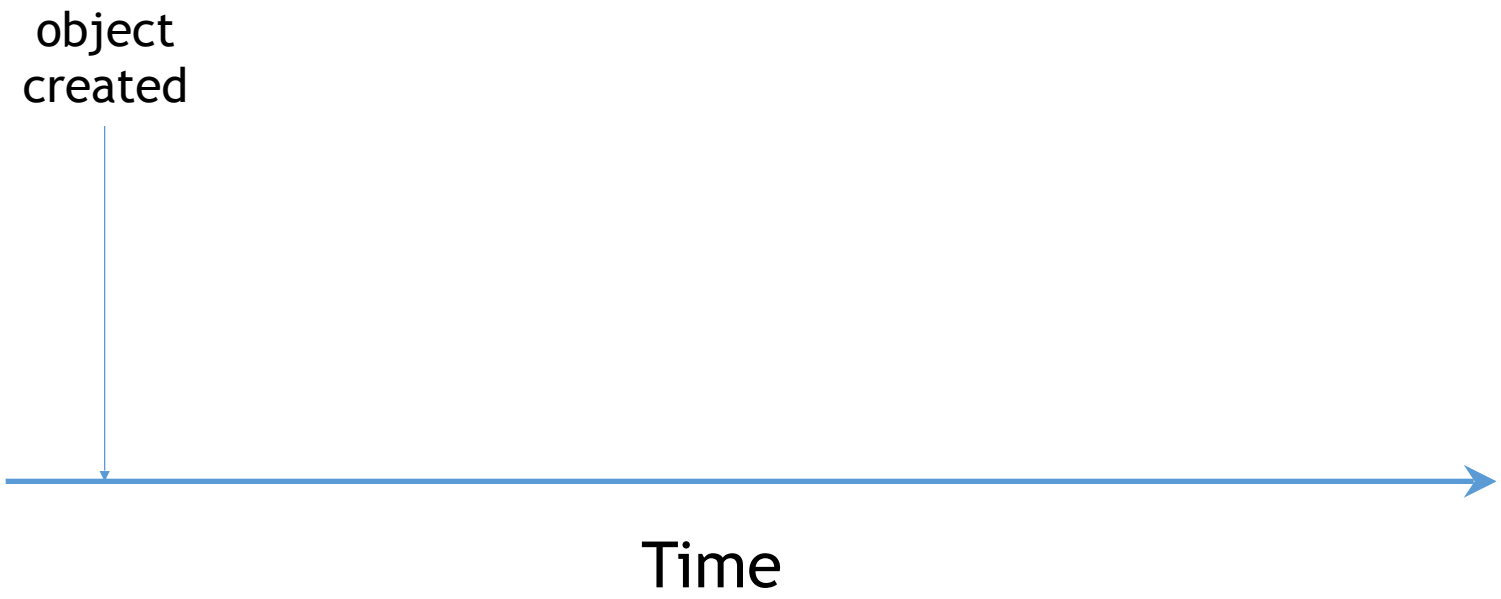
G1


E
S
E
O
H
E
O
E
E
E
E
E
O
E
S
S
O
H
O
O
O
E
S
Eden
Survivor
Young Generation
O
H
Old
Humongous
Old Generation

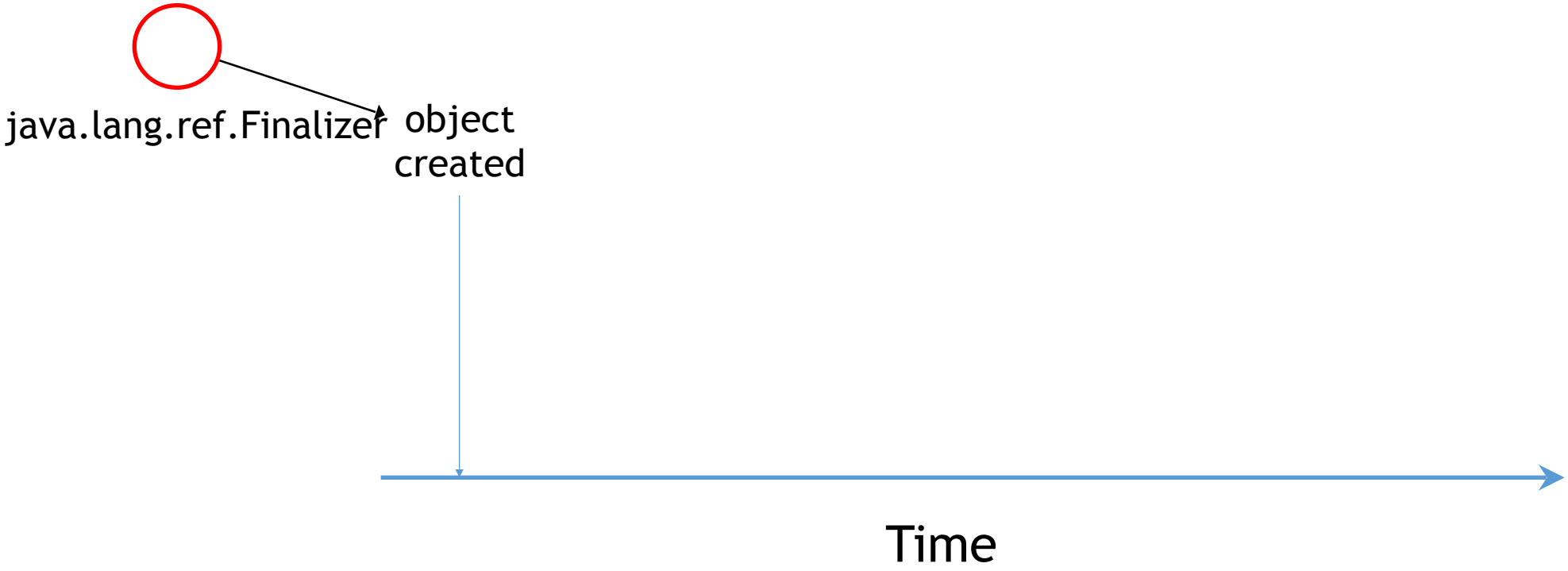
finally

finalize()
- Don’t do it
- It’s very slow

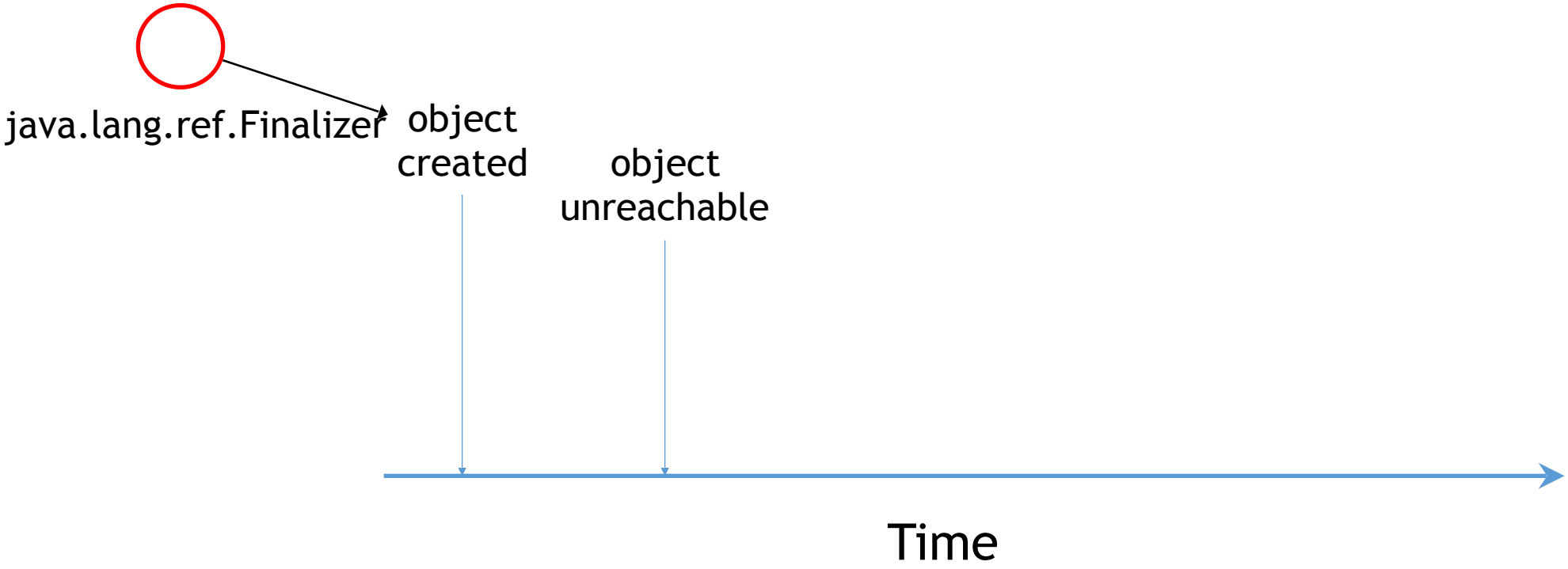
finally



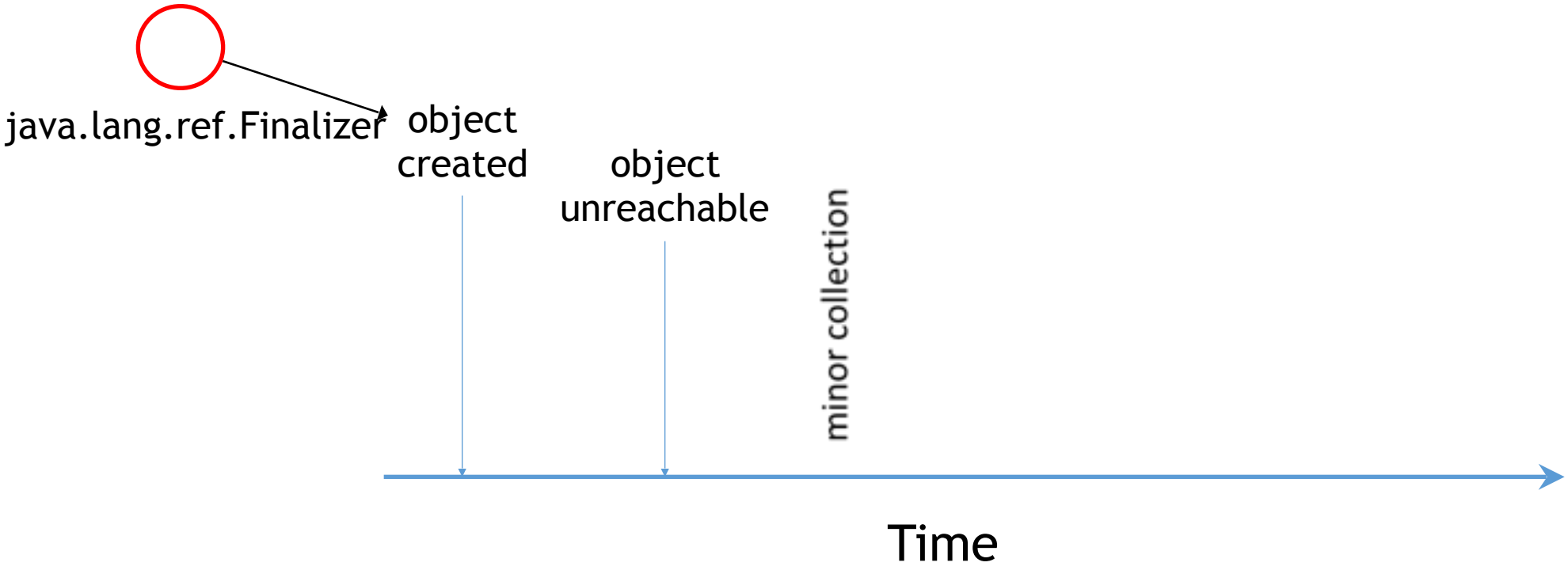
finally



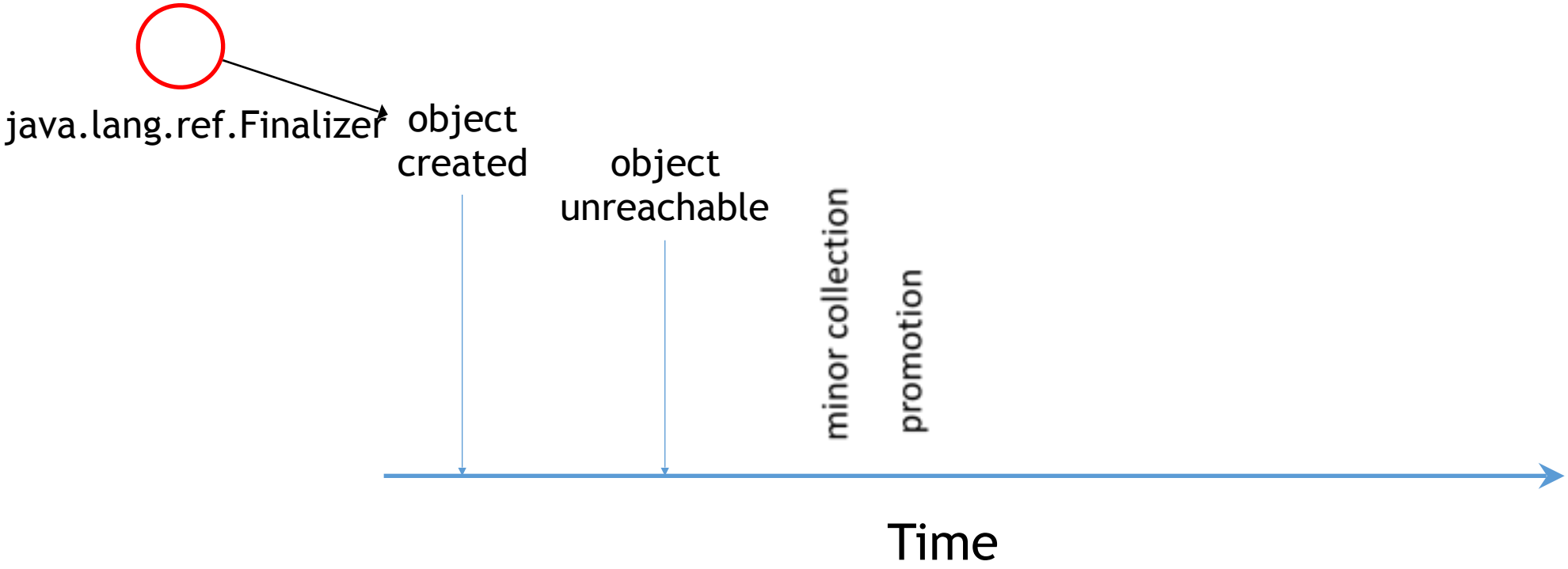
finally


java.lang.ref.Finalizer
object created
object unreachable
Time

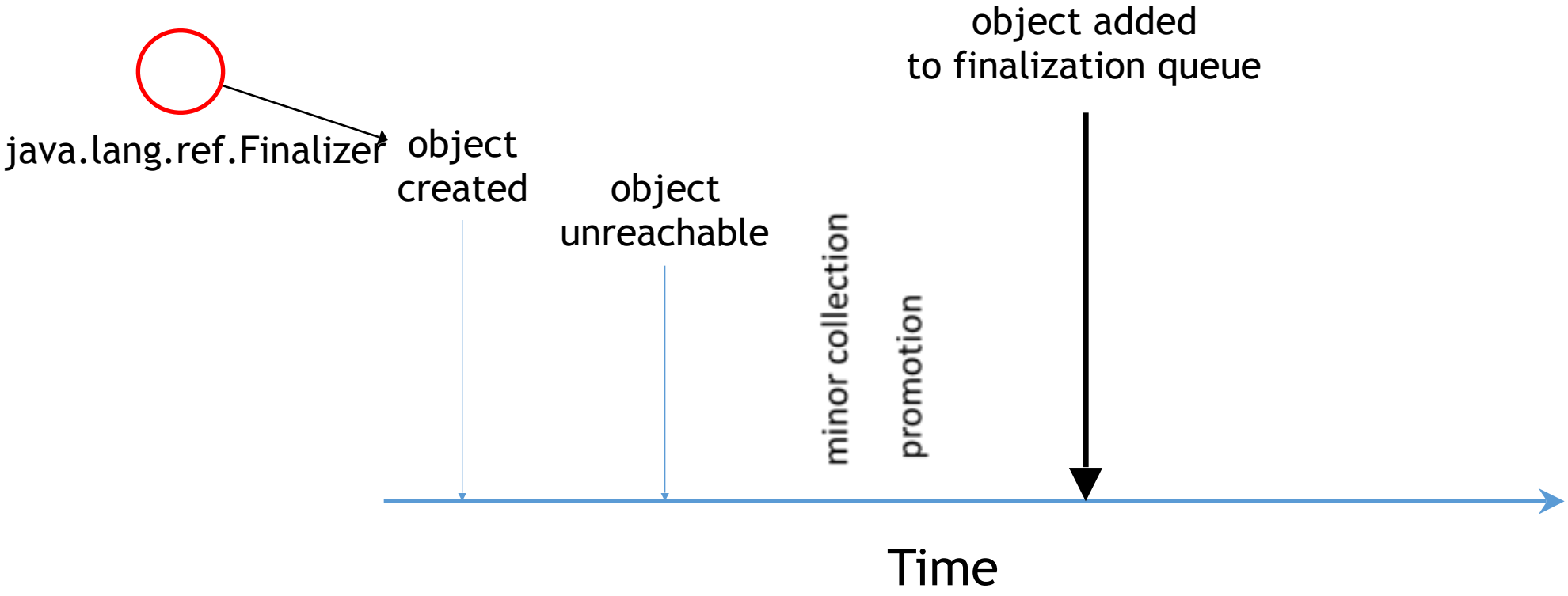
finally


java.lang.ref.Finalizer
object created
object unreachable
minor collection
Time

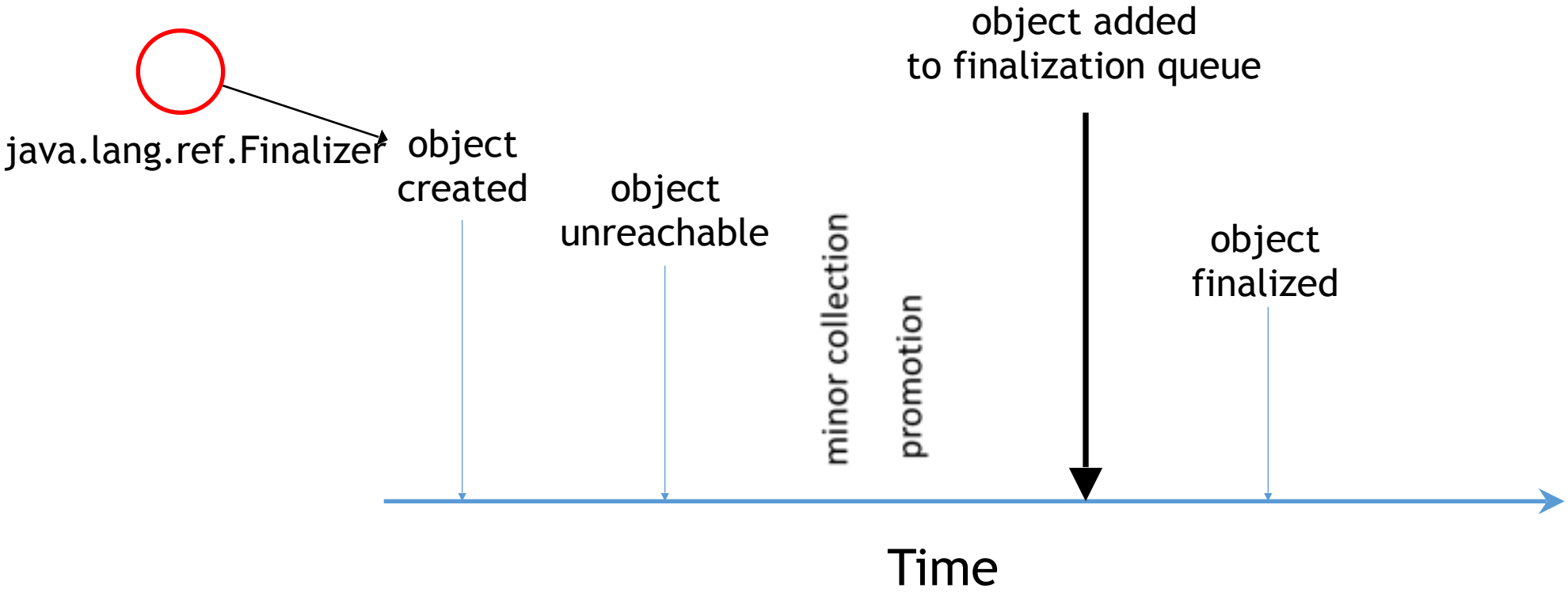
finally


java.lang.ref.Finalizer
object created
object unreachable
minor collection
promotion
Time

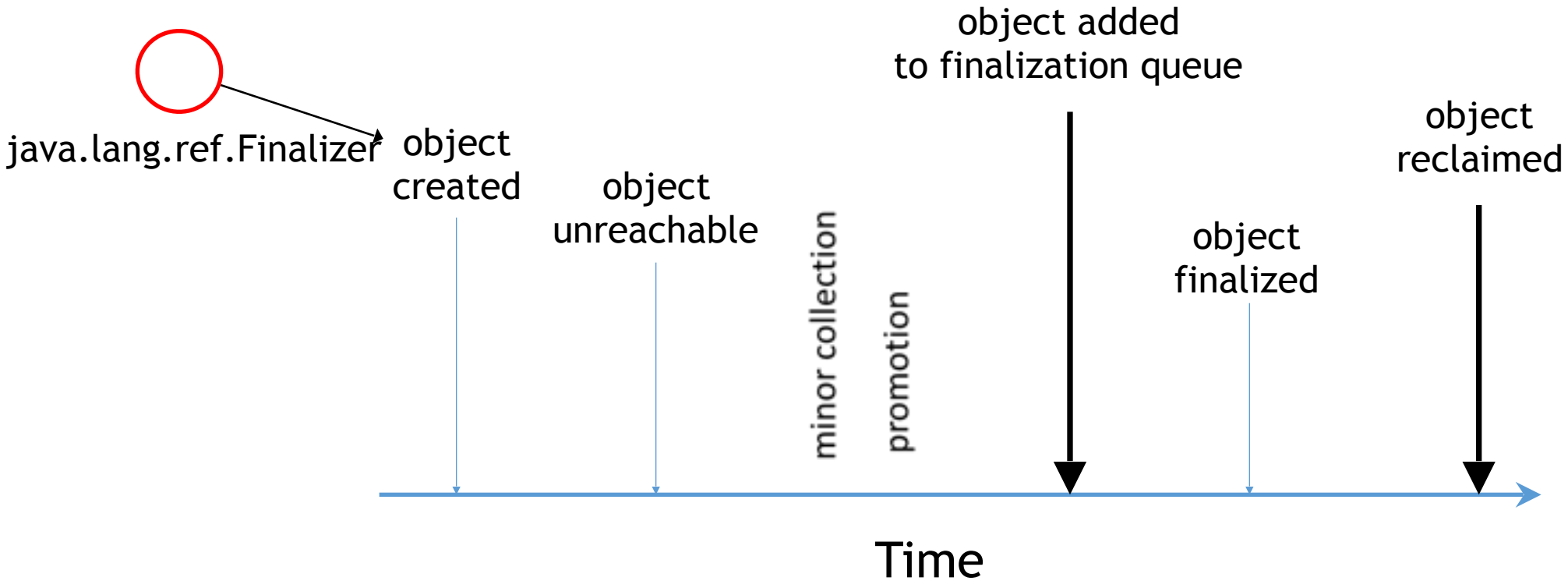
finally


java.lang.ref.Finalizer
object
created
object
unreachable
minor collection
promotion
object added
to finalization queue
Time

finally


java.lang.ref.Finalizer
object
created
object
unreachable
minor collection
promotion
object added
to finalization queue
object
finalized
Time

finally


java.lang.ref.Finalizer
object created
object unreachable
minor collection
promotion
object added to finalization queue
object finalized
object reclaimed
Time