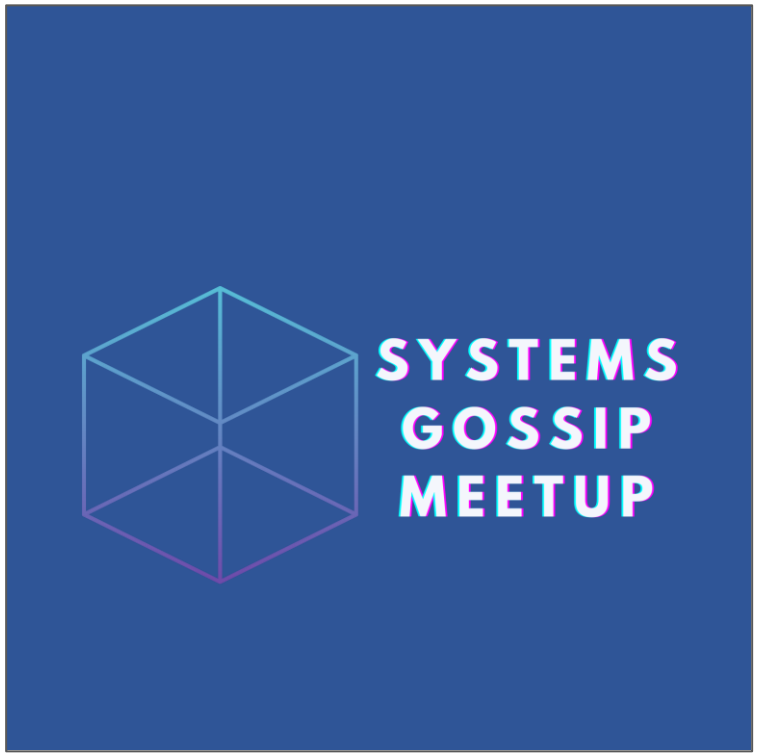


Learning large systems using peer-to-peer gossip

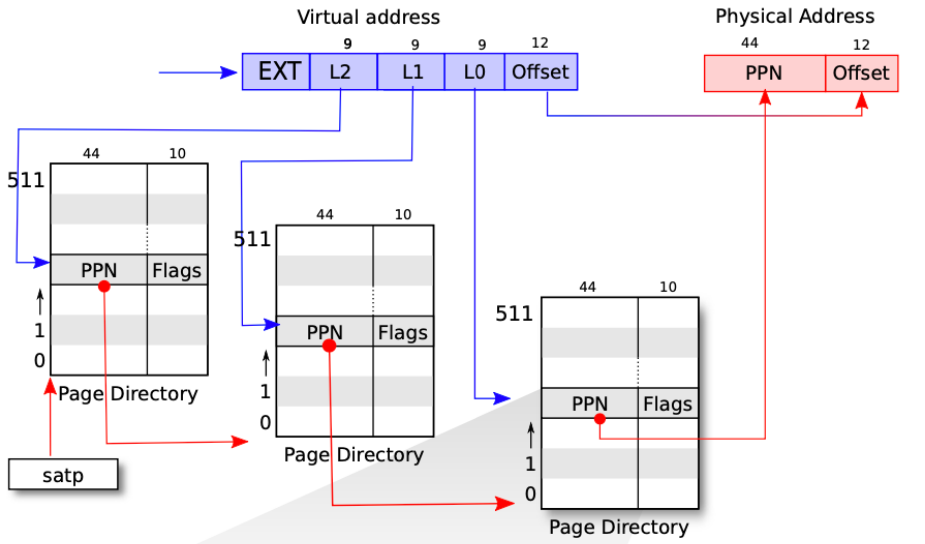
Policy Against Harassment at ACM Activities

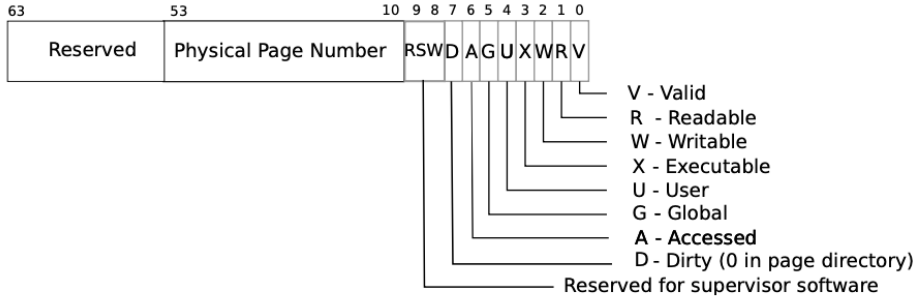
OS Meetup wants to encourage and preserve this open exchange of ideas, which requires an environment that enables all to participate without fear of personal harassment. We define harassment to include specific unacceptable factors and behaviors listed in the ACM's policy against harassment. Unacceptable behavior will not be tolerated.
https://www.acm.org/about-acm/policy-against-harassment

Virtual Memory Primitives for User Programs

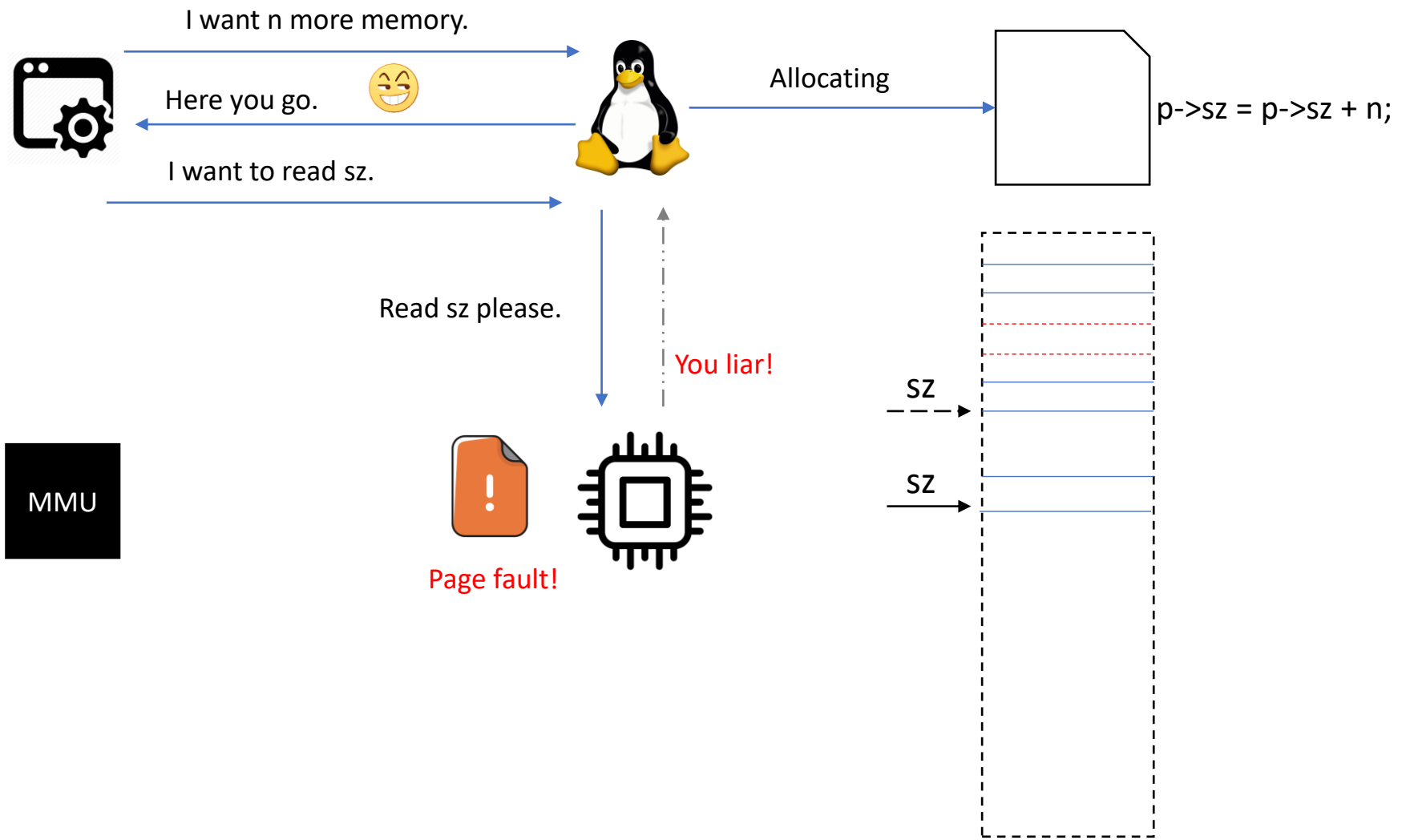
- Recall:
  - Page table
  - Trap
- Virtual Memory Primitives
  - mmap, mprotect, munmap
- Garbage Collection
  - Introduction
  - Using virtual memory primitives
- JVM Garbage Collection

Recall: Page Table





Recall: Page Fault



Virtual Memory Primitives: mmap

```
void *mmap(void *addr, size_t length, int prot,
int flags, int fd, off_t offset);
```

addr: if addr is null, kernel will pick and address for you

length: length of the mapping, must great than 0

prot: PROT_NONE, PROT_READ, PROT_WRITE, PROT_EXEC

flags : MAP_SHARED, MAP_PRIVATE

fd : file descriptor

offset : Map starting position of the target

Virtual Memory Primitives: munmap

```
int munmap(void *addr, size_t length);
```

addr: start position

length: size to unmap

Credit to: [How to use mmap function in C language? (linuxhint.com)]

Virtual Memory Primitives: mprotect

```
int mprotect(void *addr, size_t len, int prot);
```

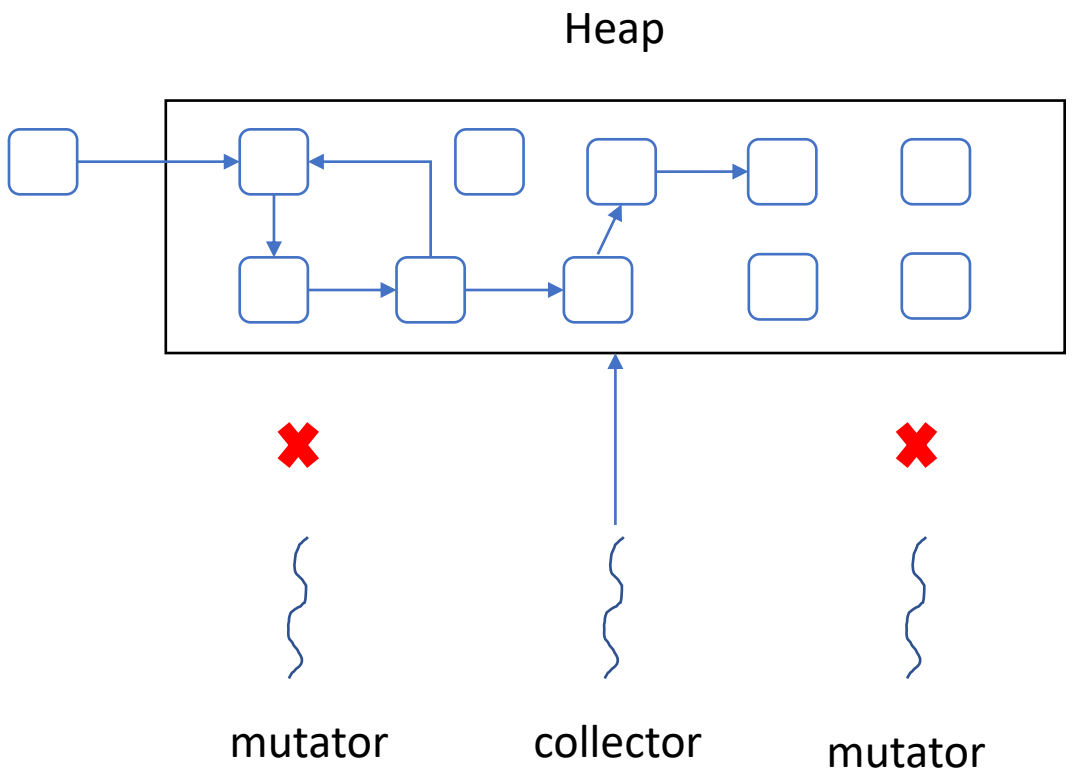
addr: start position

len: [addr, addr + len – 1]

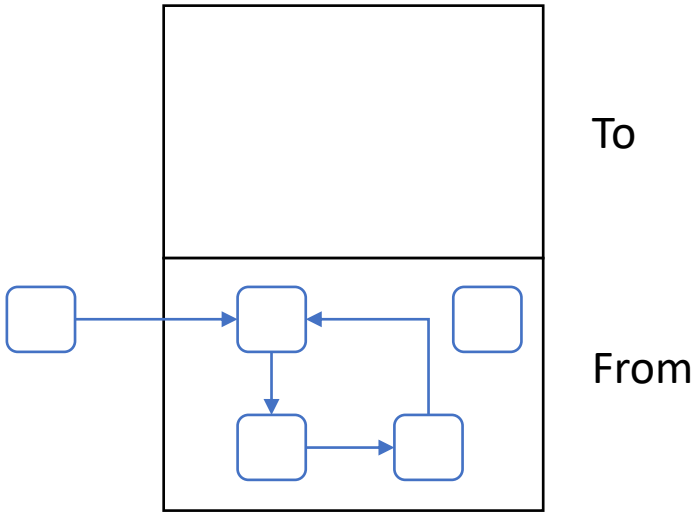
prot: PROT_NONE, PROT_READ, PROT_WRITE, PROT_EXEC

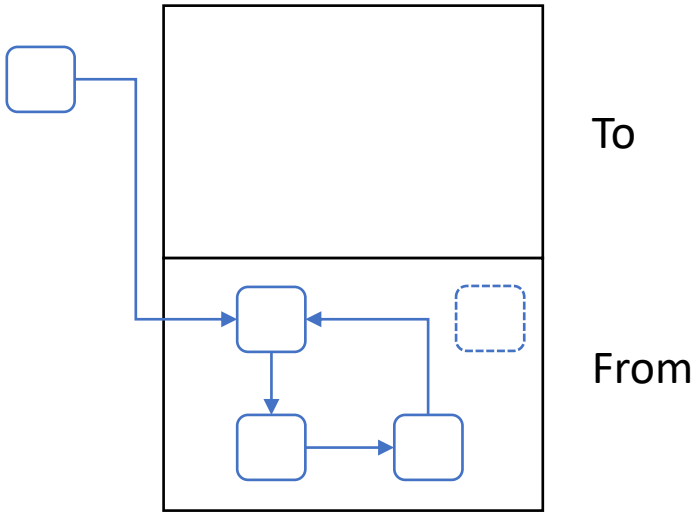
Credit to: mprotect() - Unix, Linux System Call (tutorialspoint.com)

Garbage Collection: Basic

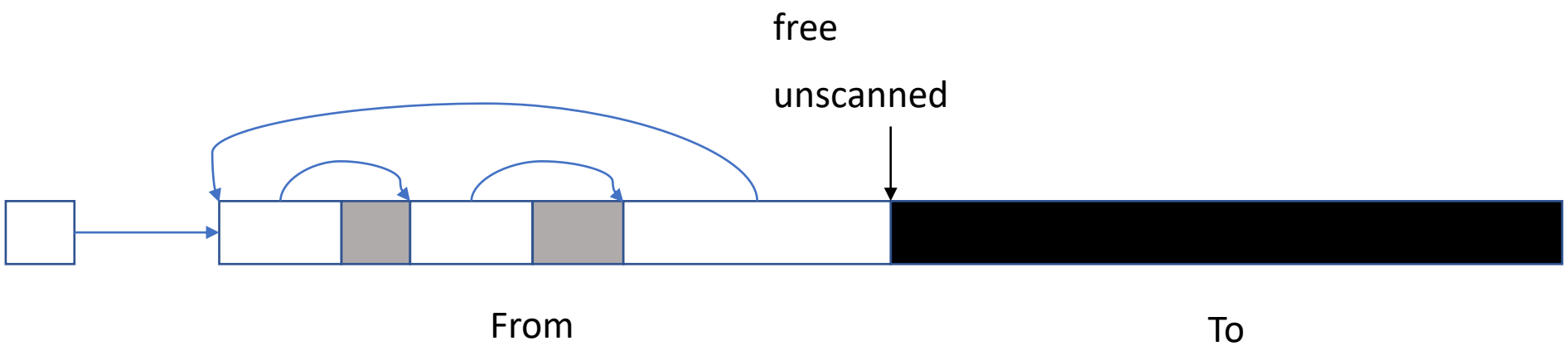


Garbage Collection: Baker's Algorithm



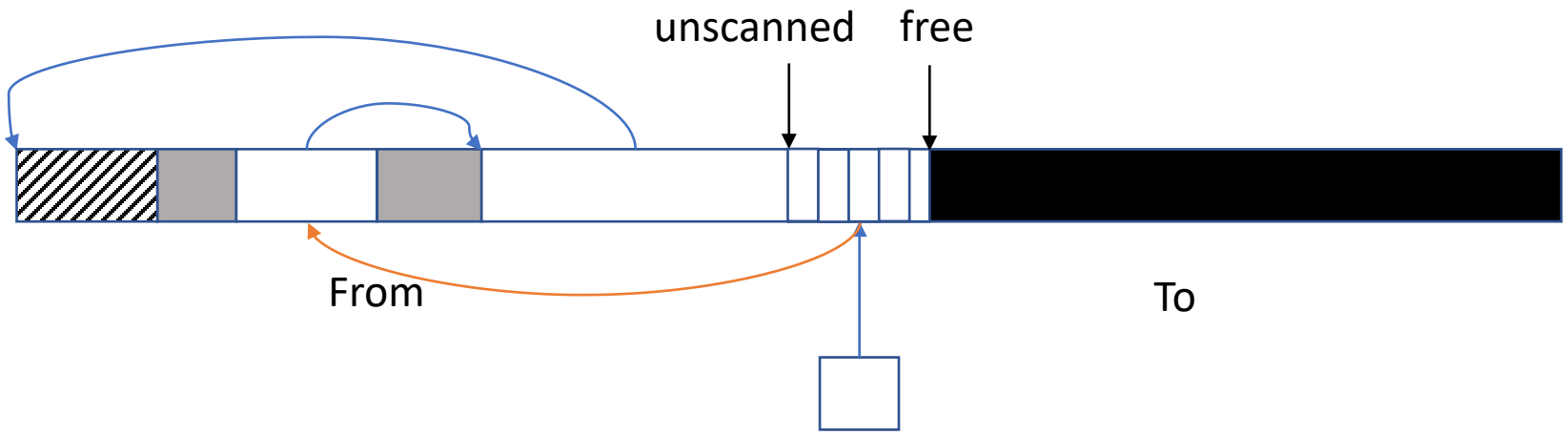


Garbage Collection: Baker's Algorithm



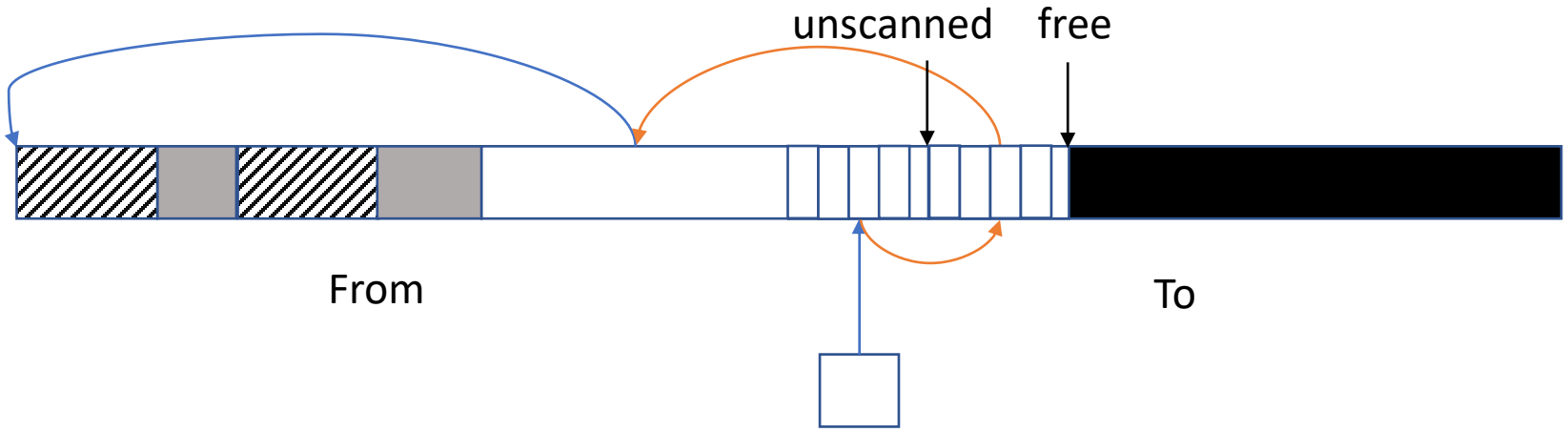
Credit to: cs426-15.pdf (uiuc.edu)

Garbage Collection: Baker's Algorithm



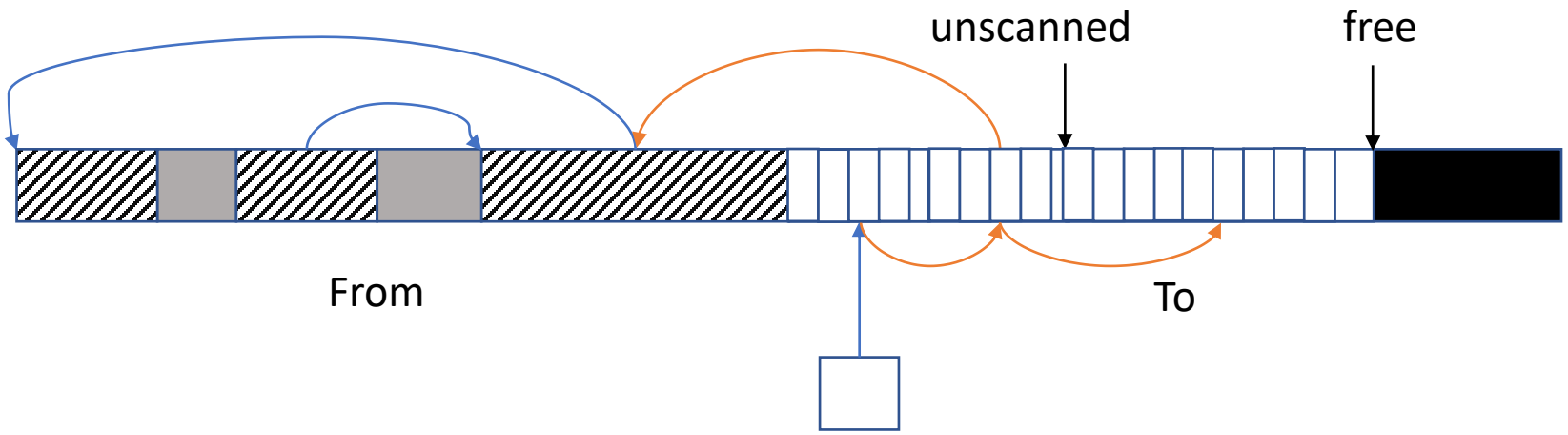
Credit to: cs426-15.pdf (uiuc.edu)

Garbage Collection: Baker's Algorithm



Credit to: cs426-15.pdf (uiuc.edu)

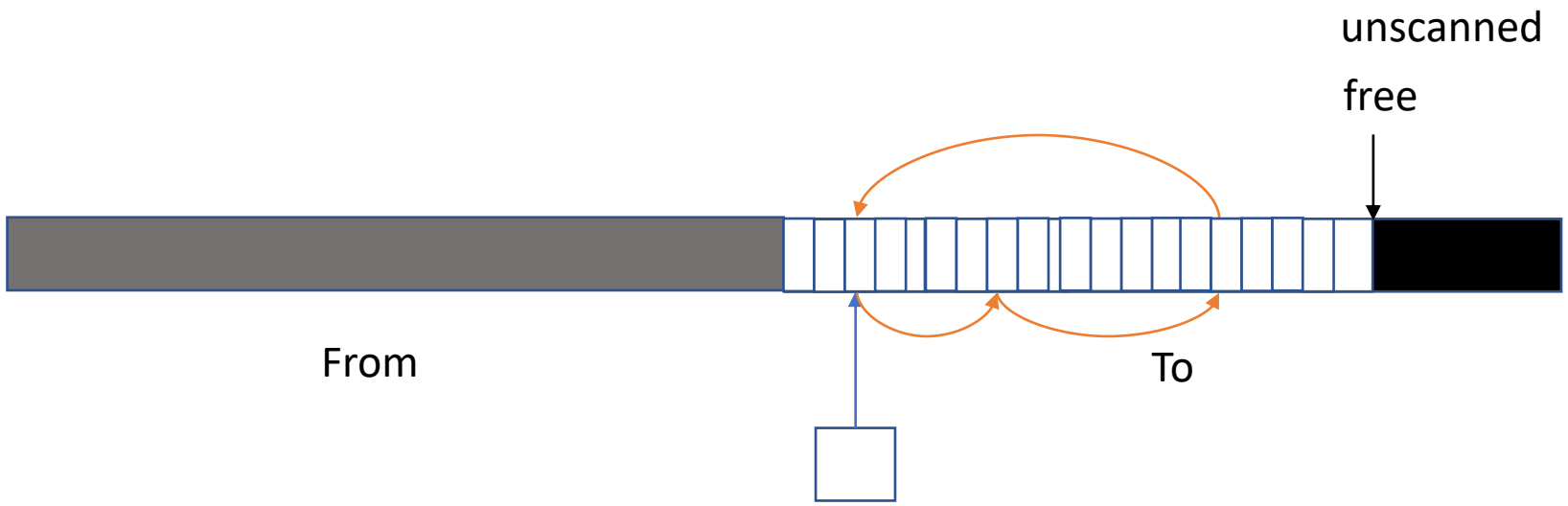
Garbage Collection: Baker's Algorithm



Credit to: cs426-15.pdf (uiuc.edu)

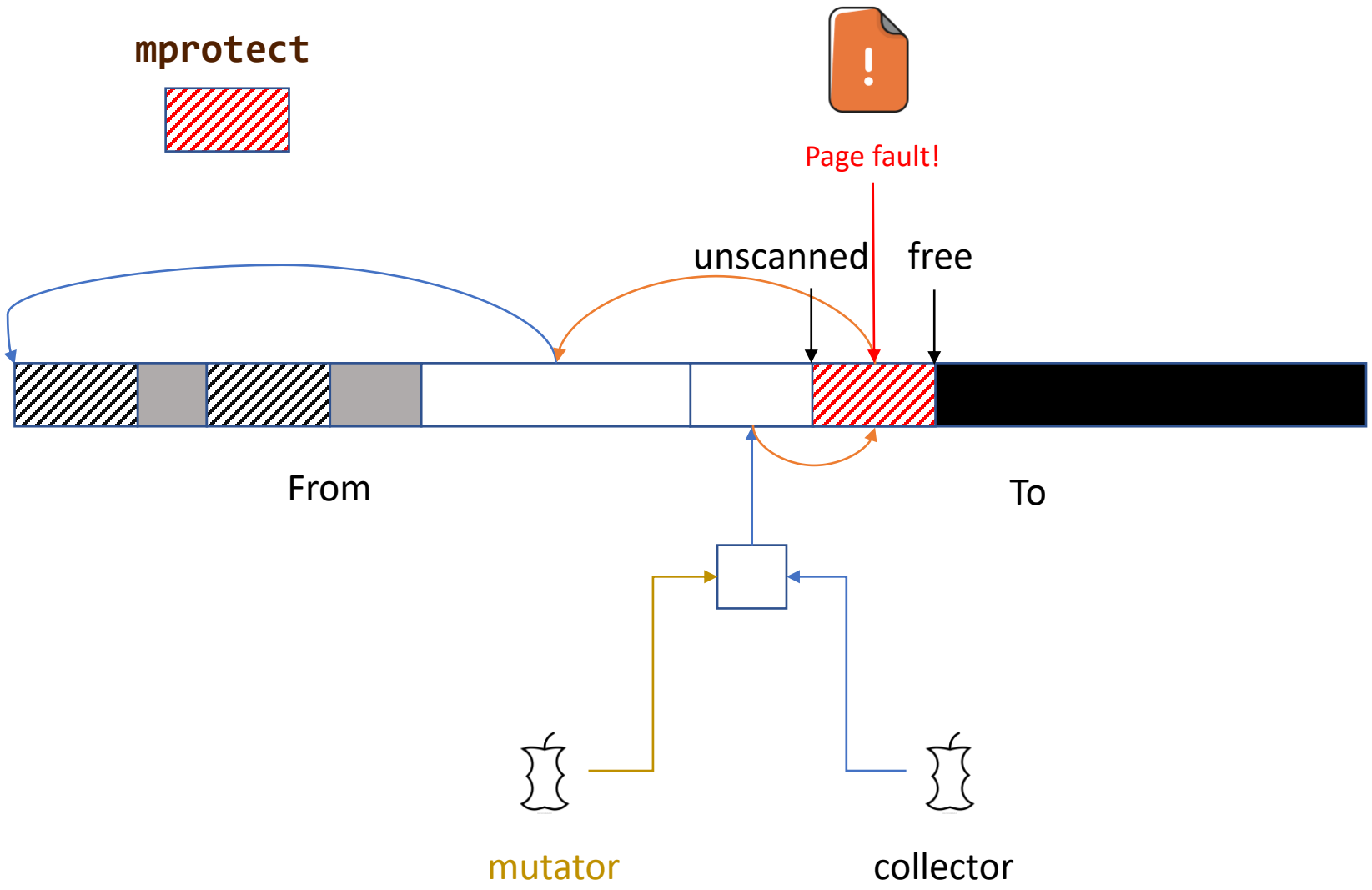
Garbage Collection: Baker's Algorithm

Will we have fraction holes in the to section after cleaning?

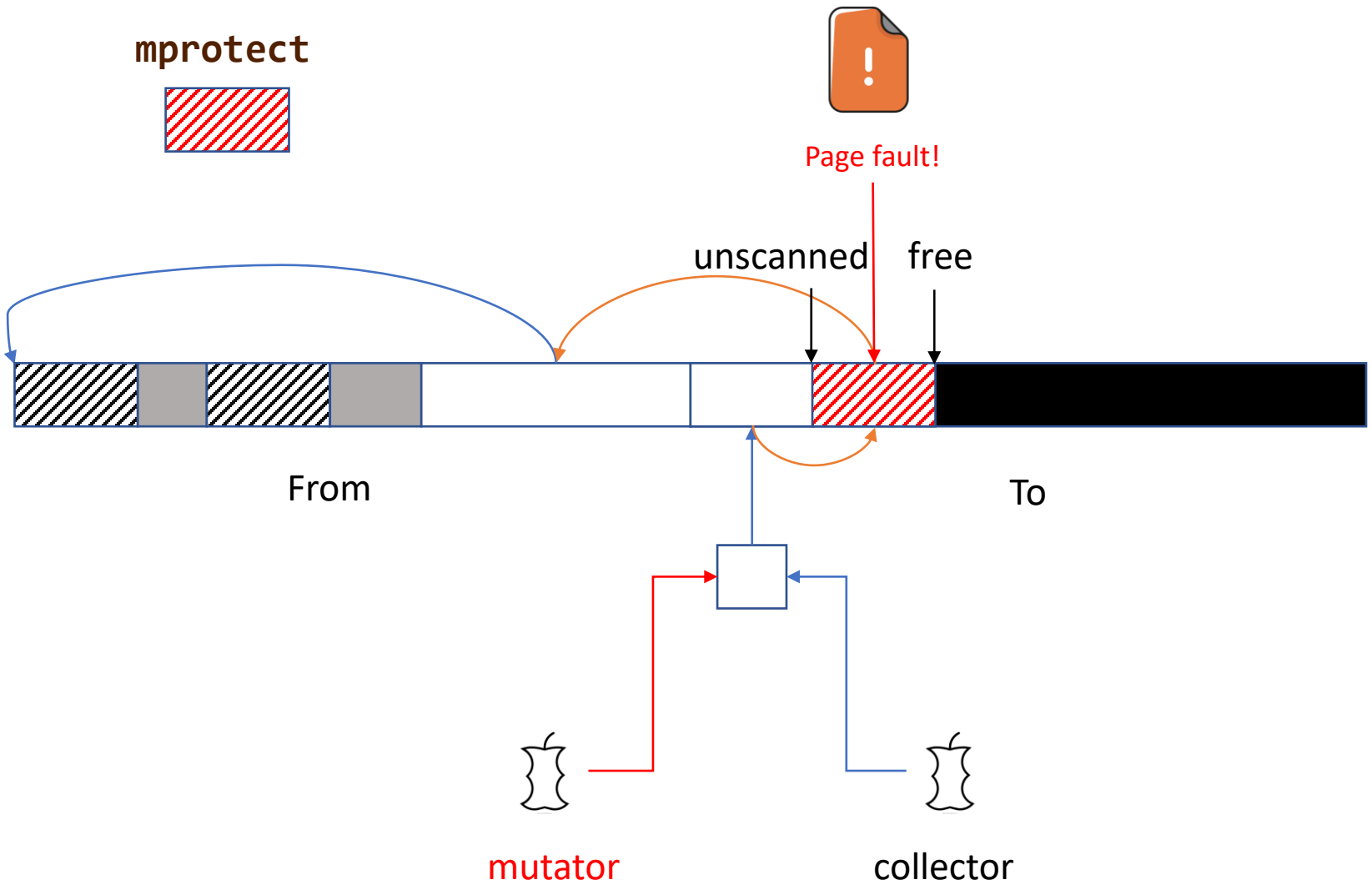


Credit to: cs426-15.pdf (uiuc.edu)

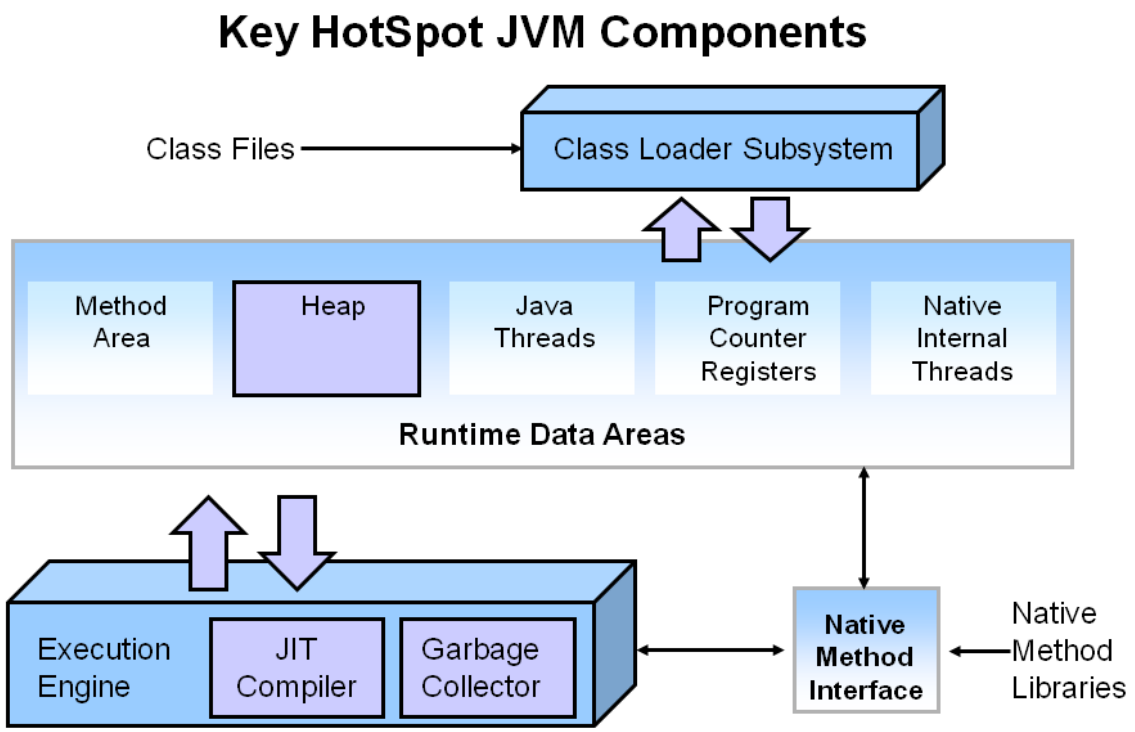
Garbage Collection: Baker's Algorithm



Garbage Collection: Baker's Algorithm

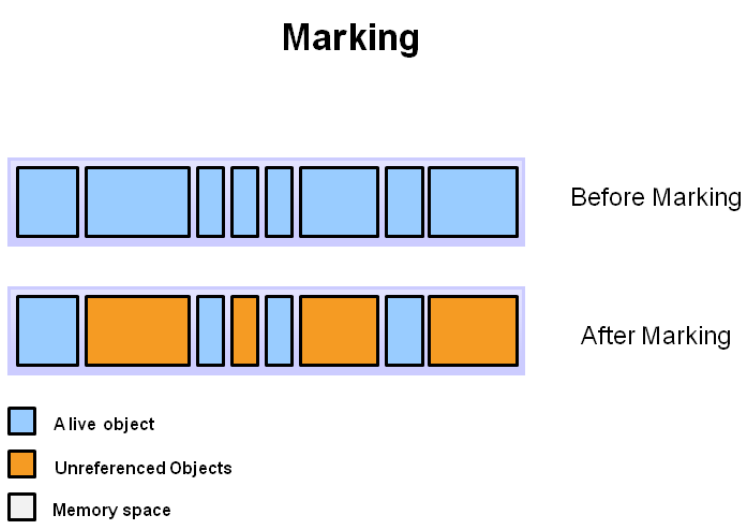


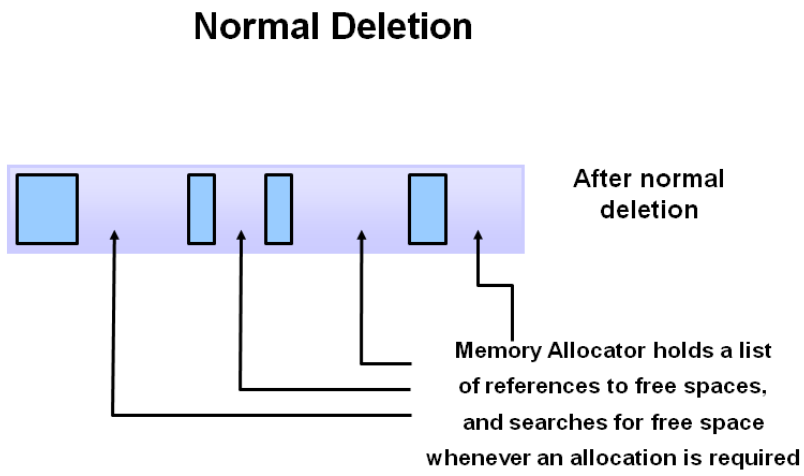
Garbage Collection: JVM



Credit to: [Java Garbage Collection Basics (oracle.com)]

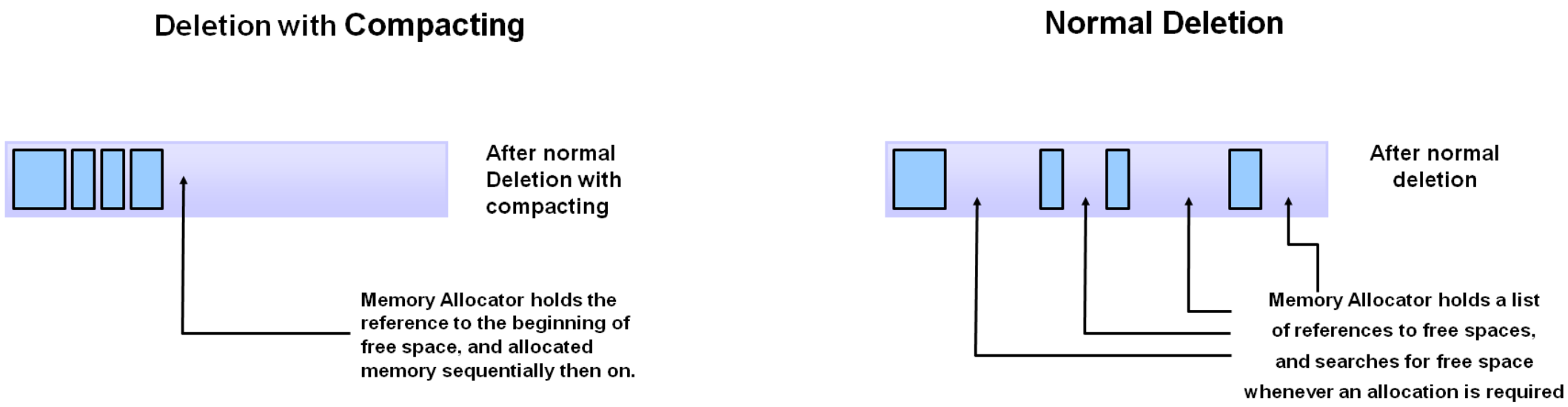
Garbage Collection: JVM





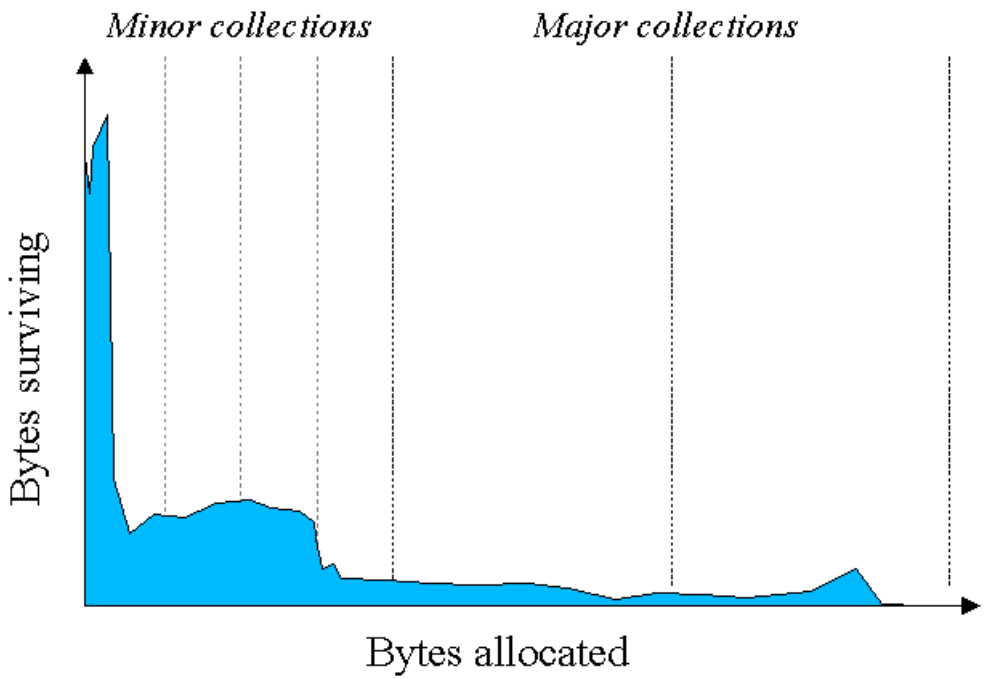
Credit to: [Java Garbage Collection Basics (oracle.com)]

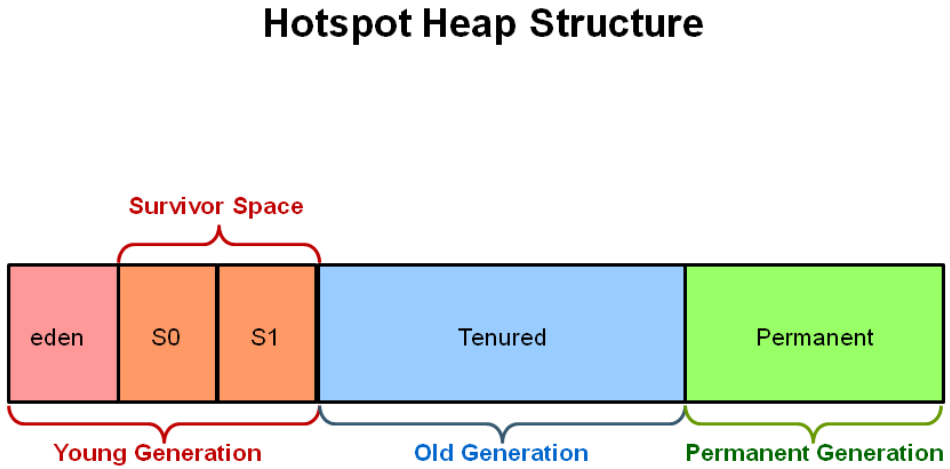
Garbage Collection: JVM



Credit to: [Java Garbage Collection Basics (oracle.com)]

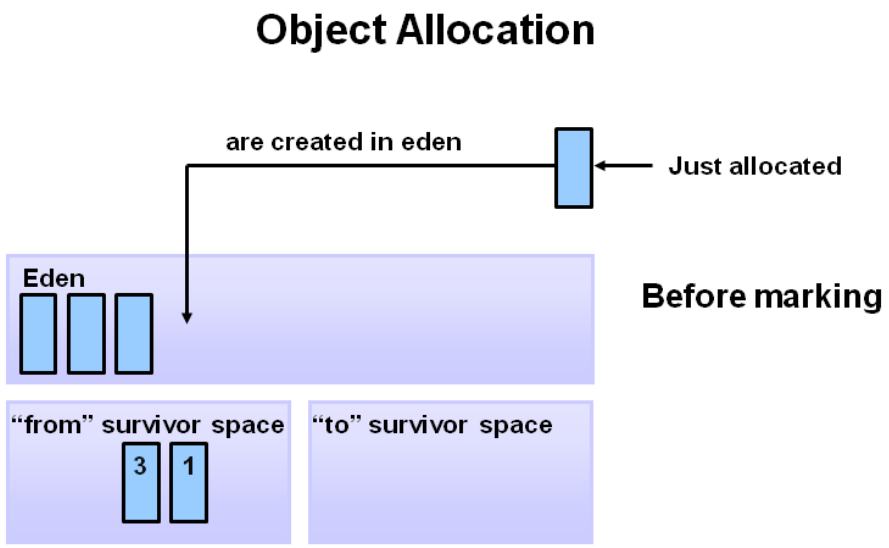
Garbage Collection: JVM

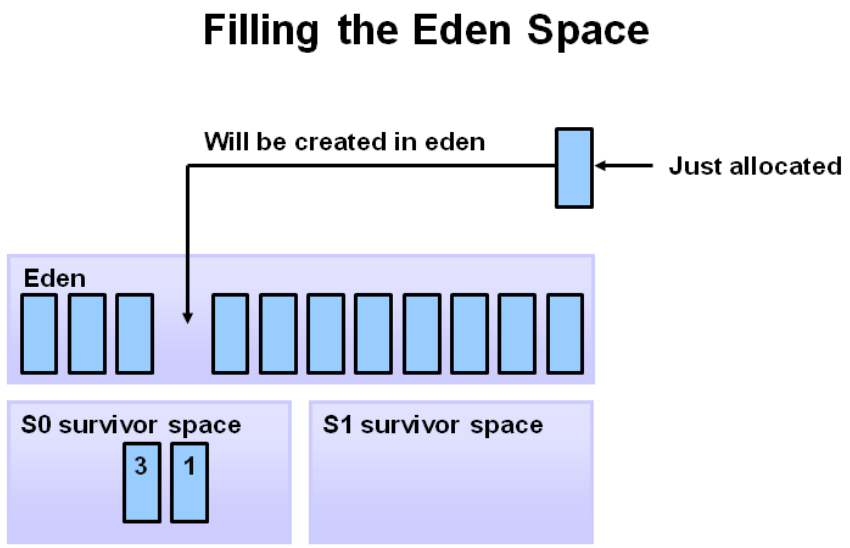




Credit to: [Java Garbage Collection Basics (oracle.com)]

Garbage Collection: JVM

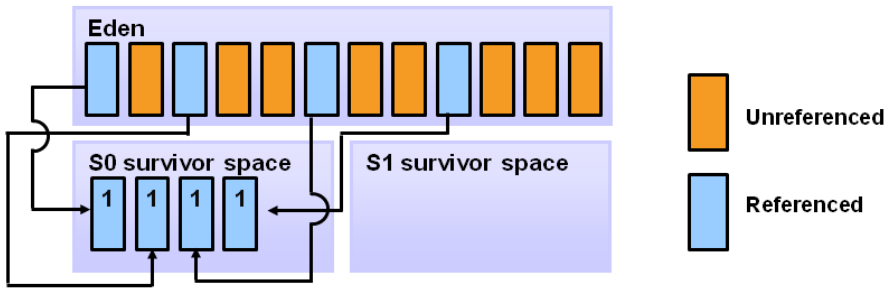




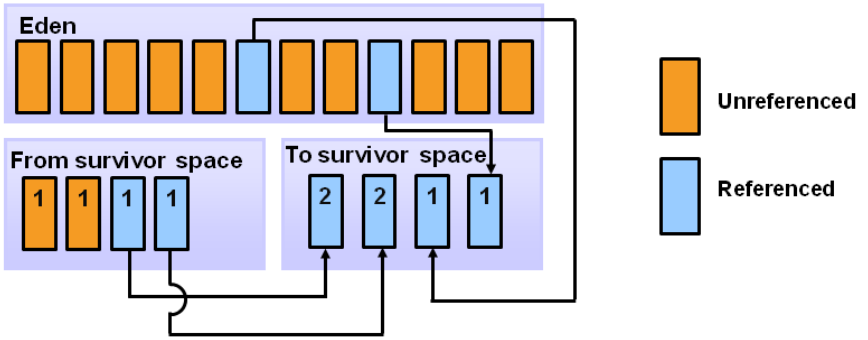
Credit to: [Java Garbage Collection Basics (oracle.com)]

Garbage Collection: JVM

Copying Referenced Objects

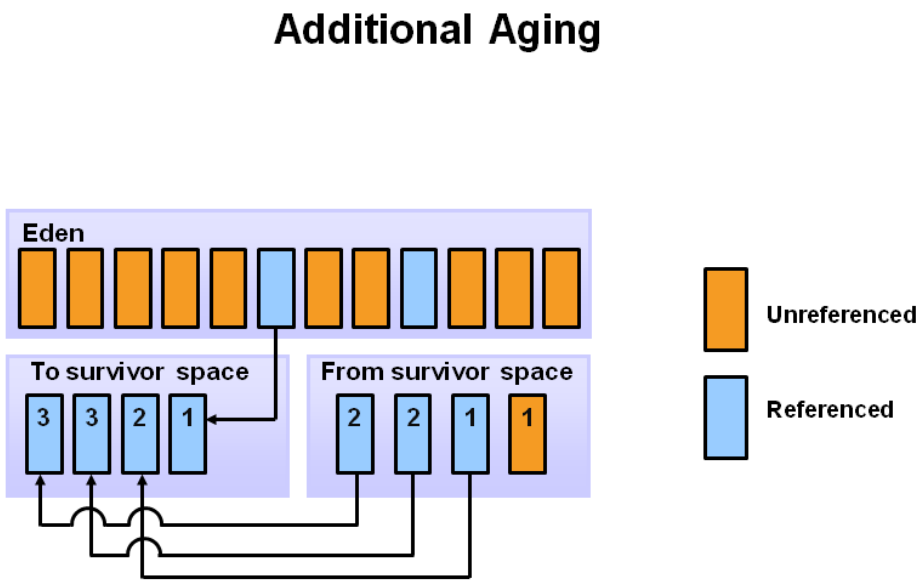


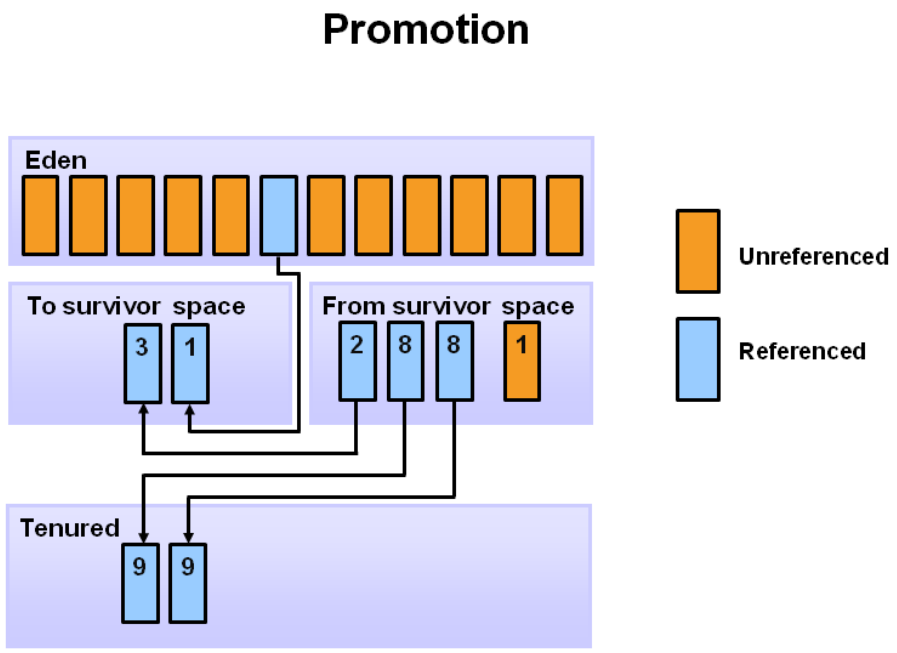
Object Aging



Credit to: [Java Garbage Collection Basics (oracle.com)]

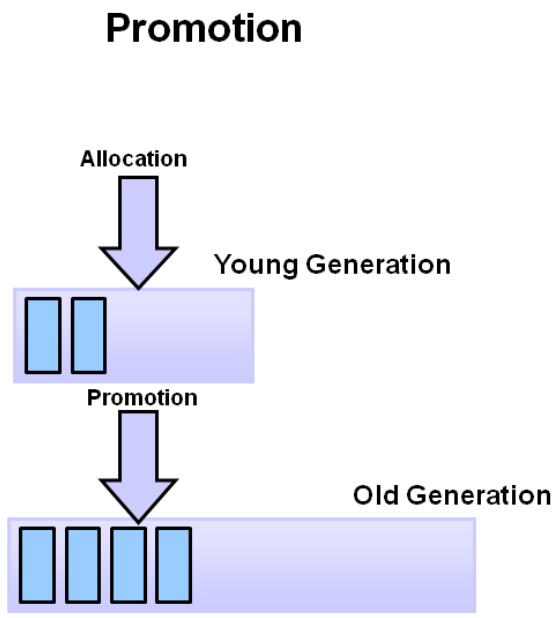
Garbage Collection: JVM

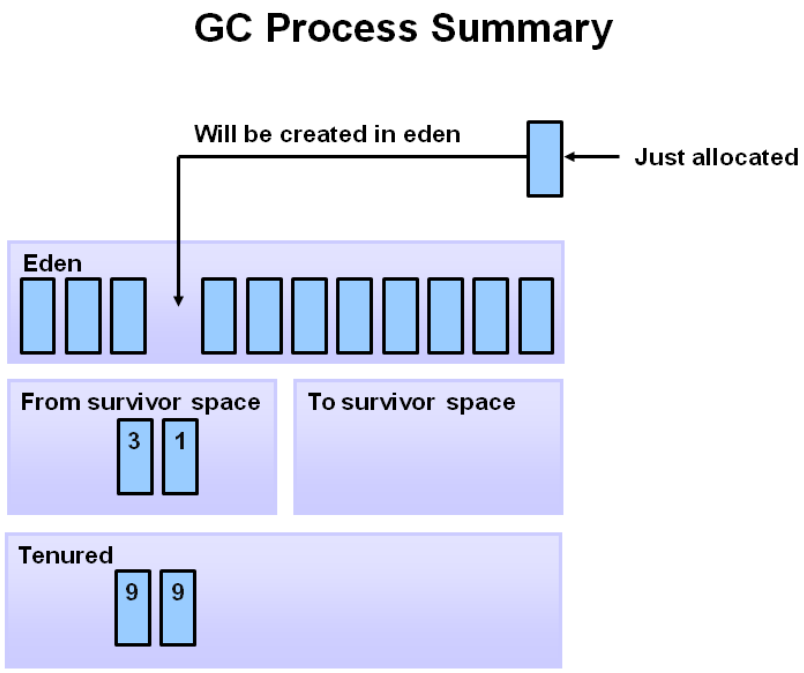




Credit to: [Java Garbage Collection Basics (oracle.com)]

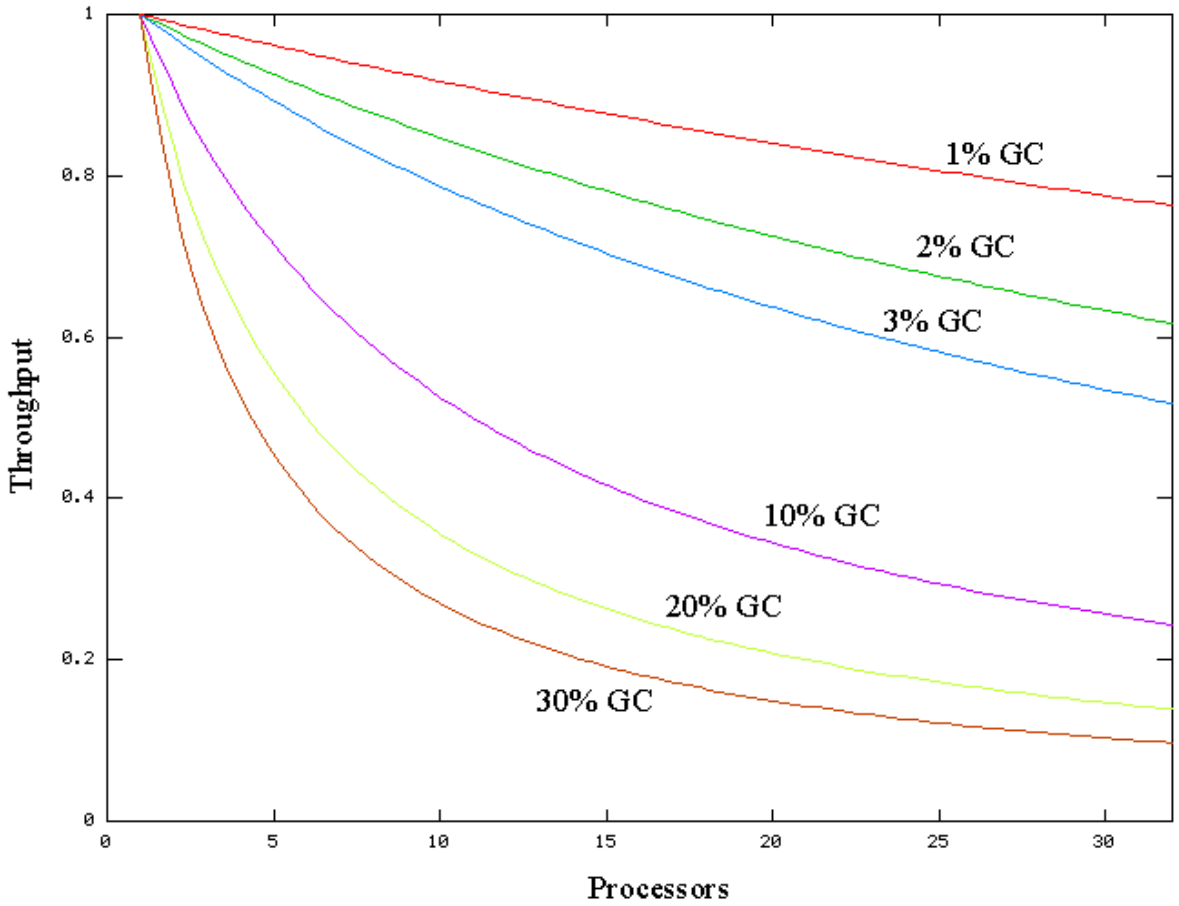
Garbage Collection: JVM





Credit to: [Java Garbage Collection Basics (oracle.com)]

Garbage Collection: JVM



Credit to: [Java SE 6 HotSpot[tm] Virtual Machine Garbage Collection Tuning (oracle.com)]

Garbage Collection: GC Tuning

Responsiveness: *Large pause times are not acceptable*

- How quickly a desktop UI responds to an event
- How fast a website returns a page
- How fast a database query is returned

Throughput: *High pause time are acceptable*

- The number of transactions completed in a given time.
- The number of jobs that a batch program can complete in an hour.
- The number of database queries that can be completed in an hour.

Garbage Collection: Faster

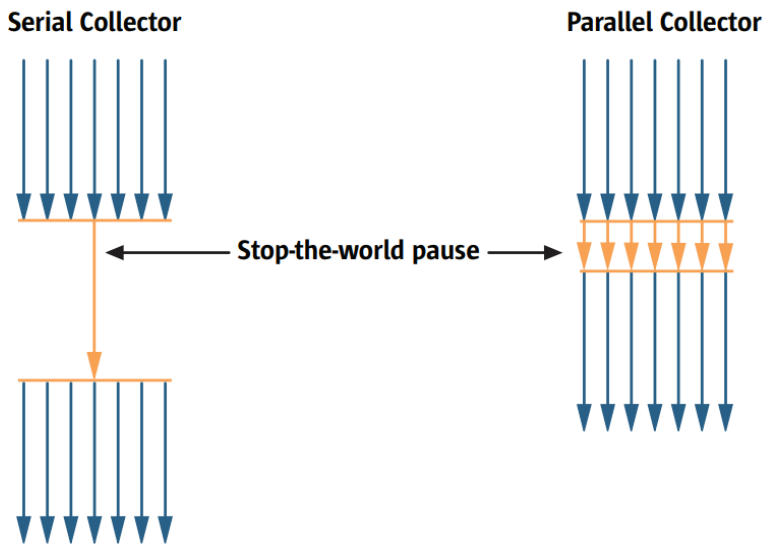


Figure 6. Comparison between serial and parallel young generation collection

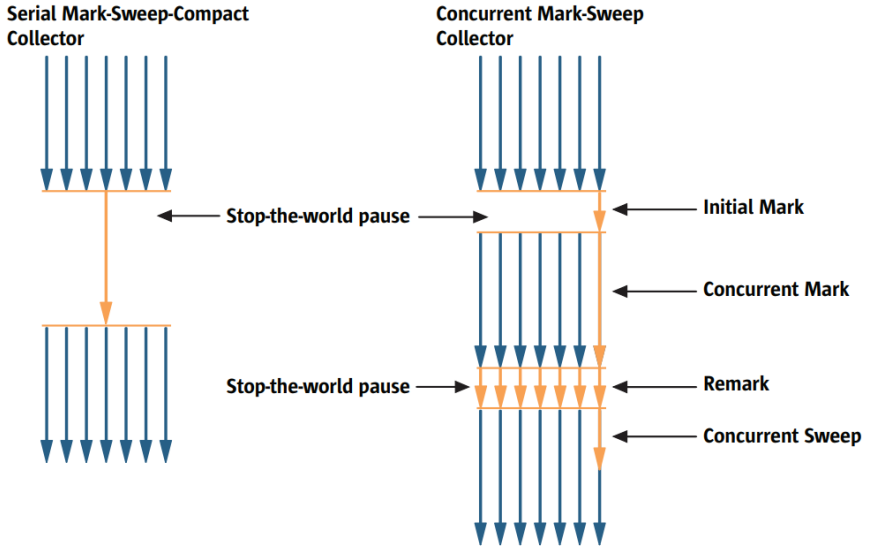


Figure 7. Comparison between serial and CMS old generation collection

Garbage Collection: Faster

Ensure that as few objects as possible are reachable during the garbage collection. The fewer objects that are alive, the less there is to be marked.

If the young generation is too small, objects are tenured prematurely to the old generation.

If the young generation is too large, too many objects are alive (undead) and the GC cycle will take too long.

Summary

- Page table
- Page allocation: fork
- Memory sharing between parent and child process
- Copy on write: fork
- Page fault: lazy allocation
- Next
  - Hongwang Li on Intrrupts