

22.12.25

일주일 피드백

G1 GC

- 1) g1 gc가 GC를 발생시키는 REGION의 우선순위 기준 : region별로 garbage가 많은 순서대로
- 2) G1 GC도 기존과 같이 young/old로 구분되었으므로 객체가 할당될때 young부터 된다.
- 3) 각각의 reigon은 각각의 Thread를 가지고 있다.

plab(promotion local allocation buffers)

GC thread가 promotion시 thread마다 old generation의 일정 부분을 할당하고 다 사용하면 다시 buffer를 재할당한다.

- plab은 promotion 과정중 동기화 문제를 회피(old area)
- tlab은 young area의 fast allocation을 위한 것(eden area)

cms 순서

- 1) **initial mark** : GC Root로부터 바로 참조되거나, young 영역의 살아있는 객체로부터 바로 참조되는 old 영역의 객체를 mark (STW 발생)
- 2) **Concurrent mark** : initial mark단계에서 mark한 객체부터 시작해서 **old영역을 순회**하면서 살아있는 모든 객체들을 markt한다. (STW 발생하지 않고 애플리케이션 스레드를 멈추지 않고 동작)
- 3) **Concurrent reclean** :
Card라는 힙영역 안에 **dirty**로 표시한 객체로부터 참조되고 있는 객체를 mark (STW 발생하지 않고 애플리케이션 스레드를 멈추지 않고 동작)
- 4) **Concurrent aborable Preclean**

변화한 객체들을 계속 **스캔**

(STW 발생하지 않고 애플리케이션 스레드를 멈추지 않고 동작, Final Remark의 수행시간을 최대한 짧게 만든다)

5) Final Remark

STW를 통해 애플리케이션 스레드를 잠시 멈춤으로써 **객체들의 상태를 완전히 반영**한다. 이 단계를 young 영역이 거의 비워져있을 때 수행하게 하도록 하여 STW을 일으키는 동작이 연쇄적으로 발생하는 것을 방지

6) Concurrent sweep

애플리케이션 스레드와 병렬적으로 수행, 사용하지 않는 객체들을 정리하여 **빈공간을 확보**한다.

gc root set

gc root란? 가비지 컬렉션의 Root를 말한다. GC Root들은 힙 외부에서 접근할 수 있는 변수나 오브젝트를 뜻한다.

- JVM 메모리의 **Stack** 영역에 존재하는 참조 변수

Local variable Section, Operand Stack 에 Object 의 Reference 정보가 있다면 Reachable Object 다.

- Method Area의 **static** 데이터

Method Area 에 로딩된 클래스 중 constant pool 에 있는 Reference 정보를 토대로 Thread 에서 직접 참조하진 않지만 constant pool 을 통해 간접 link 를 하고 있는 Object 는 Reachable Object 이다.

- **JNI** 에 의해 생성된 객체들

1. 아직 Memory 에 남아있으며 Native Method Area 로 넘겨진 Object 의 Reference 가 JNI 형태로 참조관계가 있는 Object 는 Reachable Object 이다

<https://memostack.tistory.com/229>

메모리 단편화

메모리의 공간이 작은 조각으로 나뉘어져 **사용가능한 메모리가 충분히 존재하지만 할당이 불가능한 상태**

string - constant pool

Java에서의 String은 일반적인 Heap에 저장할 수 있고 또한 별도의 String constants Pool이라는 공간을 제공하고 있다. **String Pool을 이용하기 때문에 같은 값에 대해서는 새로운 메모리에 할당 없이 재사용 가능하다**는 장점을 가지고 있다.

syn

synchronize sequence number

서버가 공개키 전달하는 방식 이름

Server Key Exchange

4way hs 과정

1. 클라이언트가 서버에게 FIN 패킷을 보낸다.
2. 서버는 클라이언트에게 ACK를 보낸다.
3. 서버는 클라이언트에게 FIN 패킷을 보낸다.
4. 클라이언트는 서버에게 ACK를 보낸다.

RTO

정해진 시간(RTO, Retransmission Timeout)