

G1 garbage collector

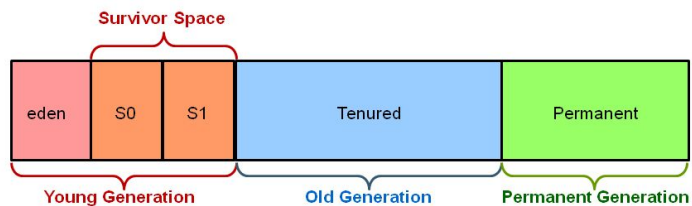
G1

- фоновый и параллельный
- высокая предсказуемость работы
- копирование и уплотнение
- сборка мусора с поколениями

Поколения

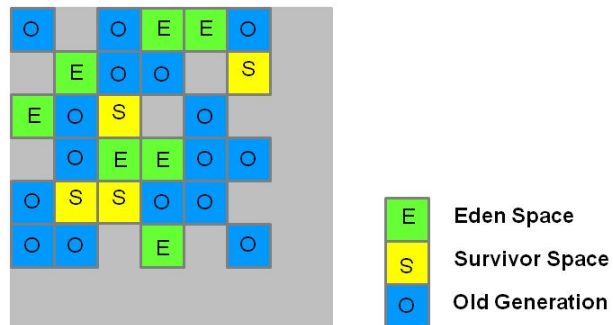
Прошлые GC

Hotspot Heap Structure



G1 GC

G1 Heap Allocation



Особенности

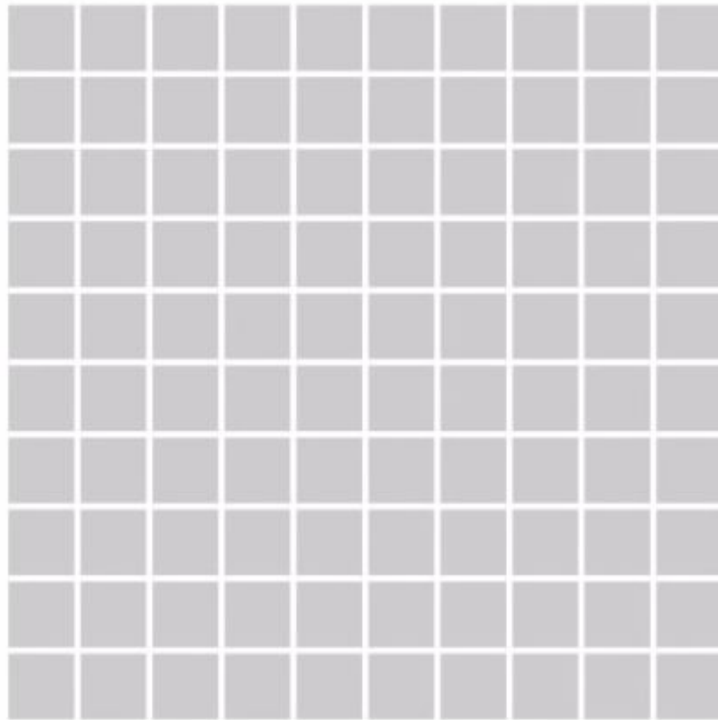
- куча состоит из регионов
- нет физического разделения между young и old поколениями
- принадлежность регионов определяется динамически
- для каждого региона известно, где находятся объекты, которые на него ссылаются:
 - remembered sets
 - независимая сборка регионов, высокая степень параллелизма
 - уменьшение времени пауз

Особенности

- garbage-first - сперва самые мусорные регионы, а лучше вообще без живых объектов
- маркировка, эвакуация и уплотнение живых объектов
 - collection sets
 - меньшая фрагментация
- мягкое требование к real-time

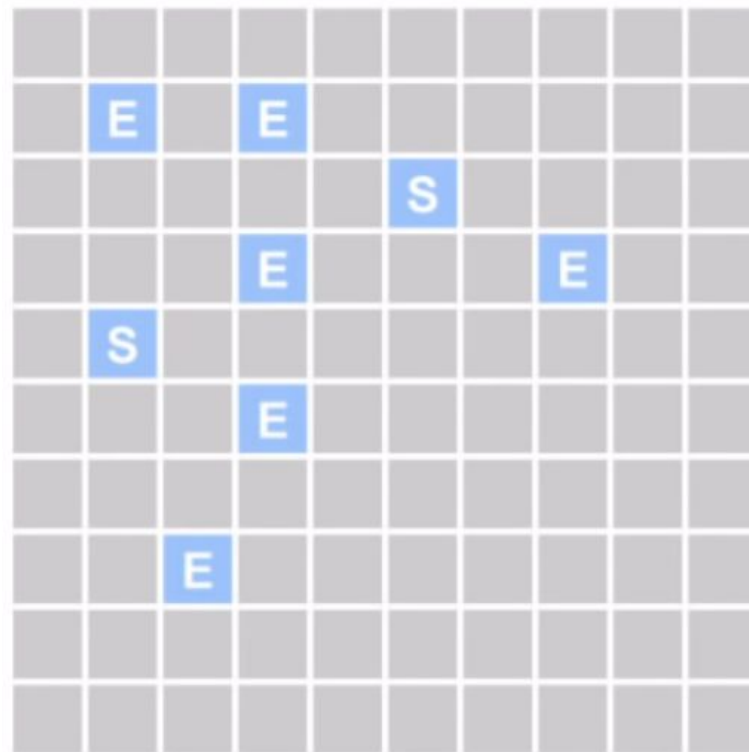
Куча

- регионы фиксированного размера
 - 1 - 32 МБ
 - около 2000 по умолчанию в JVM



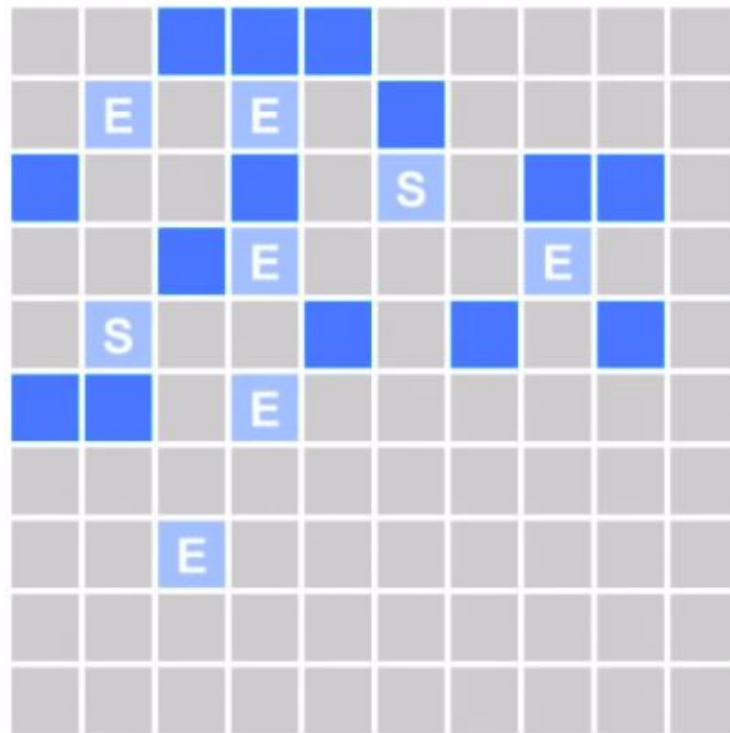
Молодое поколение

- набор регионов
 - eden, survivor
- выбирается динамически



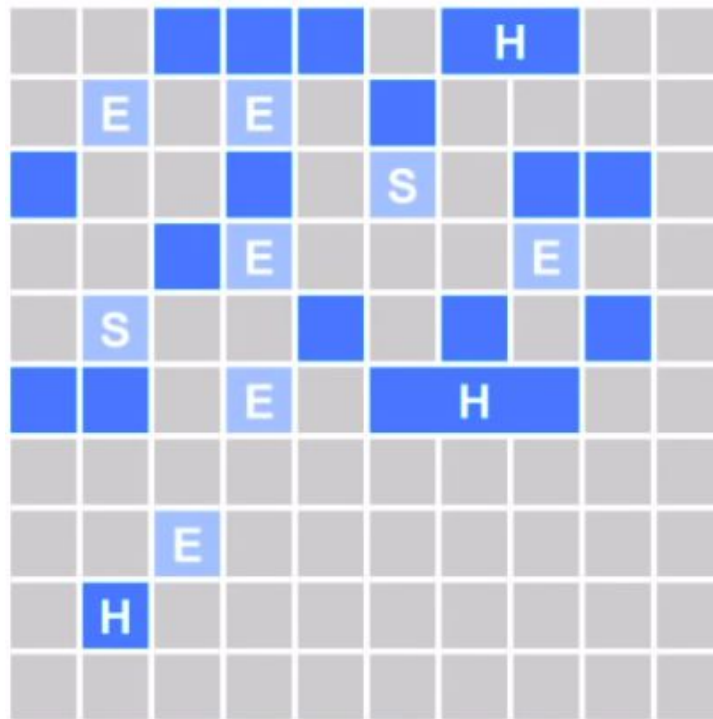
Старое поколение

- набор регионов
 - old
- выбирается динамически



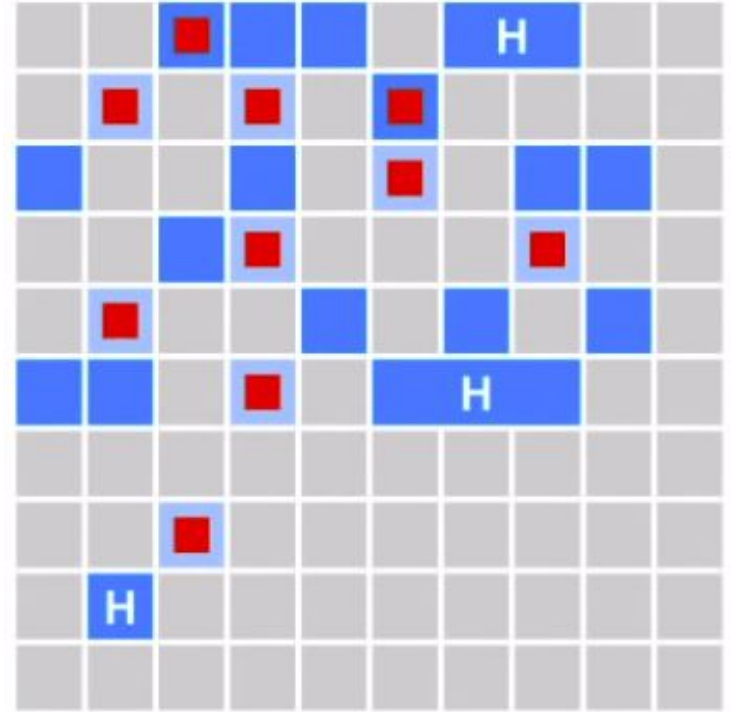
Большие объекты

- не помещаются в один регион
 - humongous
- набор смежных регионов
- замедляют работу, лучше избегать



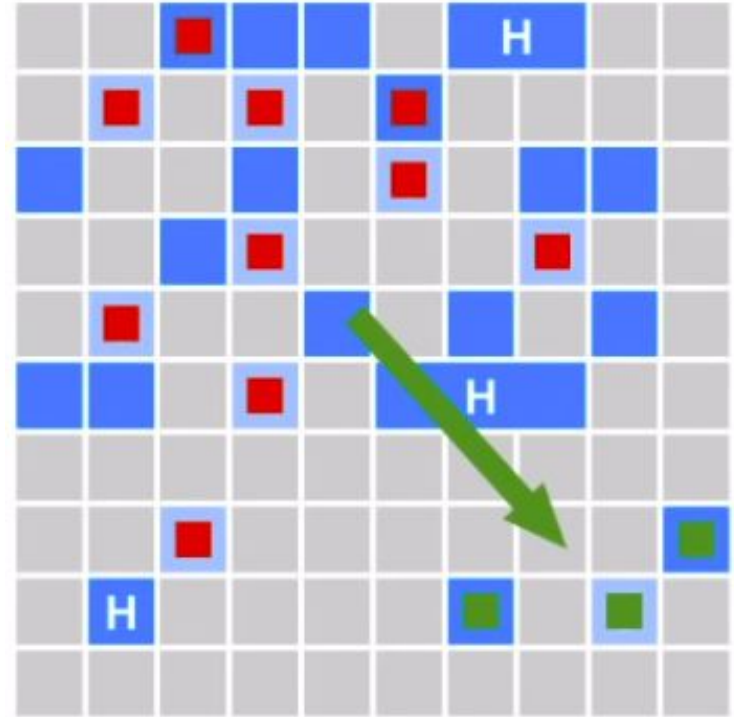
Collection sets

- регионы, в которых будет происходить сборка мусора
- молодое поколение собирается всегда
- иногда собирается часть регионов старого поколения
 - с помощью конкурентной маркировки выбираются наиболее подходящие



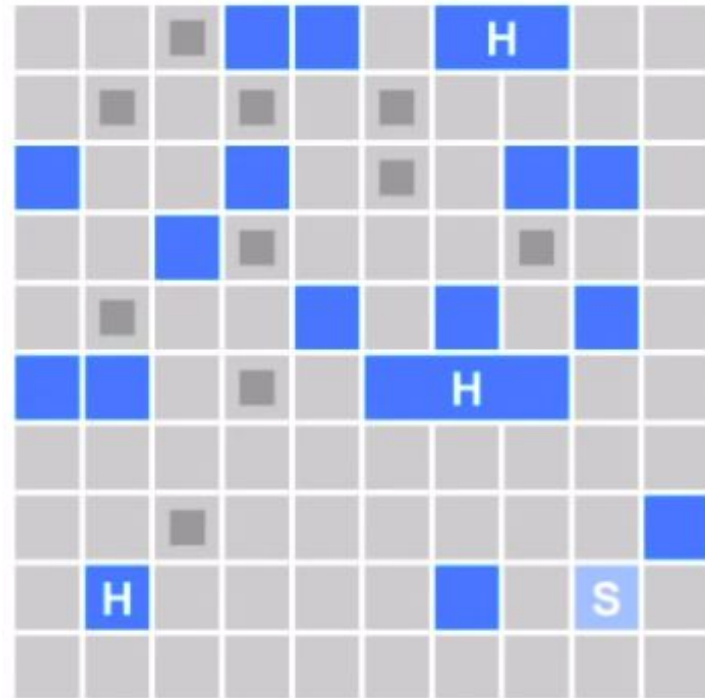
Сборка мусора

- молодое и старшее поколение
- копирование живых объектов из collection set (FROM-пространство) в регионы, составляющие часть ТО-пространства (пустые)
 - так появляются новые survivor и old регионы



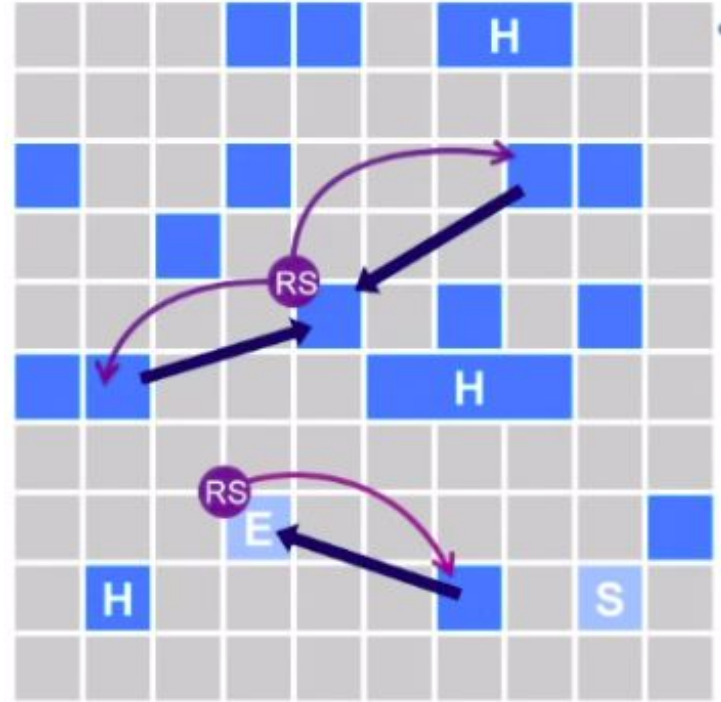
Освобождение памяти

- очищаются регионы из collection set
- FROM-пространство > TO-пространства для эффективной работы
- уплотнение за счет копирования



Remembered sets

- хеш-таблицы с информацией о регионах, которые ссылаются на объекты из данного региона
- позволяет собирать регионы независимо и параллельно
- RSets поддерживаются:
 - между старым и молодым поколением
 - между регионами старого поколения
- облегчают проверку достижимости



Барьер на запись

- дополнительный код, исполняемый при изменении значения поля объекта с указателем
 - специфичен для каждого GC
 - прозрачен для приложения
 - исполняется интерпретатором
- широко используется G1, в частности, для поддержания RSets
 - где теперь выжившие объекты находятся после сборки мусора?
- пример из реализации RSets:

```
a.f = x;
```

```
a.f = x;  
card_table[index_for(&a.f)] = DIRTY_VALUE;
```

Типы барьеров в G1

- pre-barrier
 - сохраняет предыдущее значение поля
 - поддерживает корректность фоновой маркировки
 - применяется в технологии Snapshot-At-The-Beginning (SATB)
- post-barrier
 - поддержка актуальности RSet

Фоновая маркировка

- периодически все живые объекты в куче помечаются
 - обновляется информация о достижимости в регионах
 - освобождаются регионы без живых объектов
 - разбиваются циклические зависимости среди недостижимых объектов
- конкурентная и параллельная одновременно
- использует pre-barrier
- стартует по достижении порогового значения занятости кучи (Initiating Heap Occupancy)

Весь алгоритм

