Задание:

Представим на время, что вы black hat, и у вас в распоряжении оказались данные ~30 млн. заказов некоторого сервиса доставки еды за некоторый период 2021 и 2022 года.

Поверхностный анализ показал, что данные содержат 18 758 328 уникальных телефонов с полным именем клиента, а средняя длина полного имени - 20 символов (латинских или кириллических).

Допустим, вы хотите развернуть веб-сервис, который позволит по номеру телефона найти полное имя клиента. Но вы не хотите оставлять следы на диске или в базе данных - придется все держать в памяти. Но еще вы не хотите зря тратить крипту на слишком большой сервер.

Поэтому давайте оценим, сколько памяти займут эти данные:

1. если мы хотим реализовать поиск за постоянное время, т.е. O(1)?
2. если мы хотим занять как можно меньше памяти?
3. если мы хотим реализовать поиск за постоянное время, т.е. O(1)?

Для оценки необходимого объема памяти для хранения данных о 30 миллионах заказов, включая 18 758 328 уникальных телефонных номеров и связанных с ними полных имен в Java на 64-битной системе, необходимо учитывать несколько аспектов.

Каждый телефонный номер представляет собой уникальную строку из 10 цифр и хранится в Java как объект String. Этот объект включает заголовок объекта, ссылку на массив символов, поля для хэш-кода и длины, а также выравнивание. В результате, на каждый телефонный номер требуется 80 байт памяти.

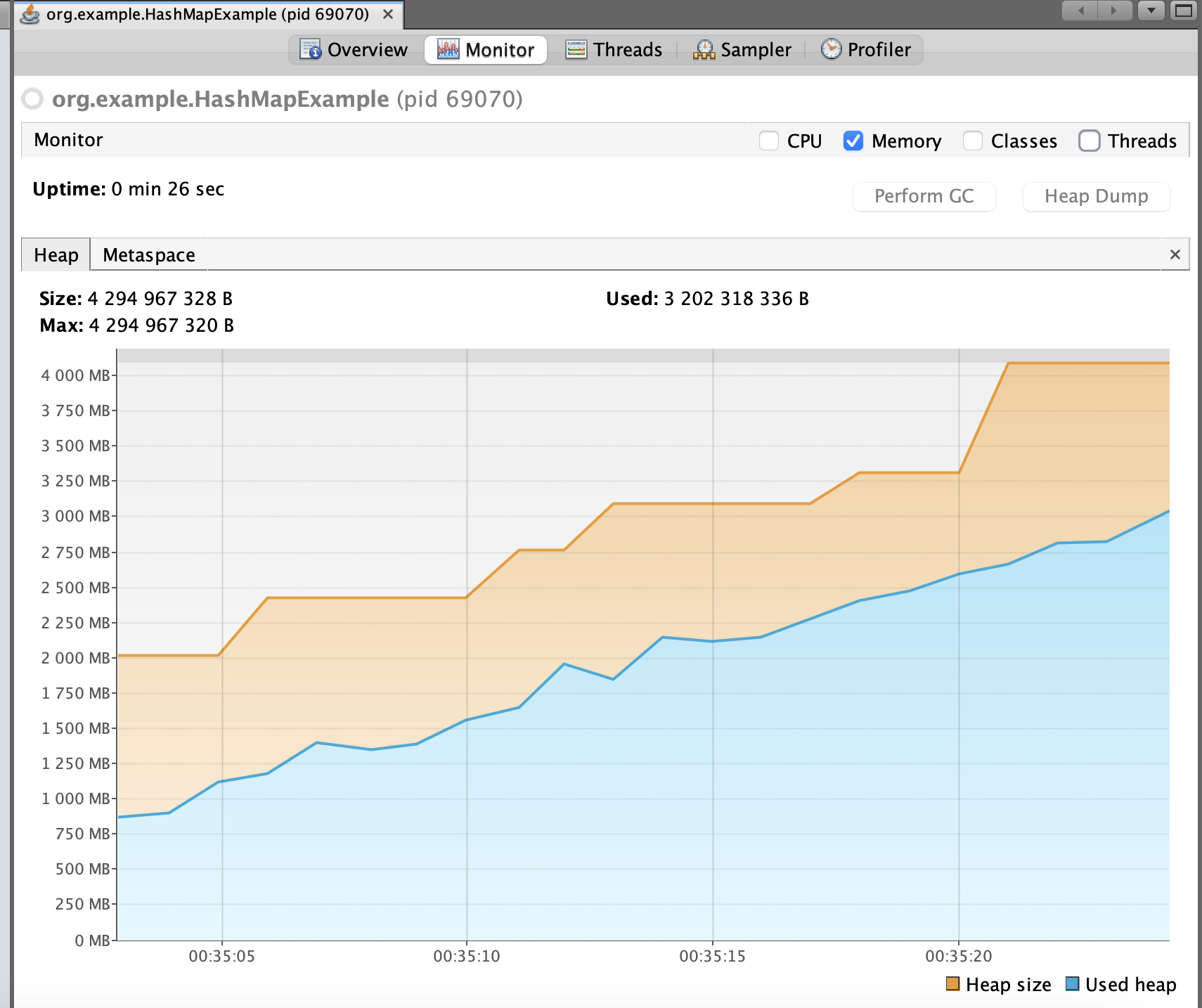
Полные имена, которые также хранятся как объекты String, состоят в среднем из 20 символов. Структура хранения аналогична телефонным номерам, и на каждое полное имя требуется 96 байт памяти.

Кроме того, каждая запись в HashMap, используемая для связывания телефонных номеров с полными именами, включает ссылки на ключ и значение, заголовок объекта Entry, поля для хэша и ссылки на следующий элемент, а также выравнивание. Это добавляет еще 40 байт на каждую запись.

Суммарно, для каждой уникальной пары телефонного номера и полного имени требуется 216 байт памяти. Учитывая, что таких пар 18 758 328, общий объем памяти составит около 4 051 798 848 байт, что эквивалентно примерно 3864 МБ.

Таким образом, для хранения данных о 18 758 328 уникальных телефонных номерах и связанных с ними полных имен в Java HashMap на 64-битной машине потребуется около 3864 МБ памяти, включая накладные расходы на объекты Java и данные.

Я написал программу, запустил программу. Затем я использовал VisualVM для мониторинга потребления памяти моей программой и провел измерения. Вот результаты моих замеров:



1. если мы хотим занять как можно меньше памяти?

Для эффективного хранения 18 758 328 записей, включая телефонные номера и полные имена, используем примитивные массивы. Телефонные номера хранятся в массиве long[], используя 8 байт на номер, что составляет ~143 МБ. Полные имена хранятся в массиве char[], с дополнительными массивами nameIndices и nameLengths для отслеживания начала и длины каждого имени. При средней длине имени 20 символов, общий объем для имен составит ~715 МБ. Массивы индексов и длин потребуют еще ~143 МБ. Общий расчетный объем памяти — около 1 ГБ. Этот подход минимизирует накладные расходы, обеспечивая более эффективное использование памяти.