



Санкт-Петербургский государственный университет
Кафедра системного программирования

Методы работы с утечками памяти в JVM

Слугин Александр Николаевич, группа 21.Б10-мм

Научный руководитель: С.Ю. Сартасов, старший преподаватель кафедры системного программирования

Санкт-Петербург
2024

OutOfMemoryError — одна из самых распространённых ошибок во время выполнения программ на JVM.

Причины возникновения:

- Недостаток памяти для первоначальной загрузки классов среды выполнения Java
- Недостаточное пространство в куче для размещения нового объекта

Инструменты для борьбы с утечками памяти не внедрены в промышленные версии JVM.

Постановка задачи

Целью работы является исследование целесообразности применения способов обнаружения и предотвращения утечек памяти в программах, исполняющихся при помощи виртуальной машины Java.

Задачи:

- Провести обзор методов, которые позволяют обнаруживать и предотвращать утечку памяти
- Определить методологию и сравнить рассмотренные методы
- Составить требования к методу для предотвращения утечек памяти

В следующем семестре:

- Реализовать метод и проанализировать влияние на работу программы

Существующие подходы

- Анализ состояния программы
- Визуализация состояния кучи
- Статический анализ исходного кода
- Обнаружение неиспользуемых объектов в куче
- Анализ роста используемой памяти

Многие инструменты реализованы в Jikes RVM — исследовательской версии виртуальной машины Java.

Bell — Bit Encoding Leak Location

Вероятностный подход для определения места выделения объекта.

- Кодирует информацию в один бит и хранит его в заголовке объекта
- Был реализован в Jikes RVM
- Используется в инструментах для обнаружения утечек памяти

Инструмент для обнаружения утечек памяти.

- Определяет неиспользуемые объекты
- Использует Bell для получения места выделения и места последнего использования объекта
- Хранит информацию в четырёх битах в заголовке объекта
- В реализации Jikes RVM не требует дополнительной памяти, в других JVM использует 6,25% памяти вне кучи
- Увеличивает время выполнения программы на 29%, но при использовании статического профилировщика — на 11%

Инструмент для предотвращения утечек памяти во время выполнения программы для языков со сборщиком мусора.

- Перемещает на диск неиспользуемые объекты
- Использует Sleigh для обнаружения неиспользуемых объектов
- Имеет механизм для возвращения объектов в кучу
- Реализован в Jikes RVM
- При наличии утечек памяти увеличивает производительность приложений на 27-46%, требует 8,5-9 Мб виртуальной памяти
- В приложениях без утечек памяти увеличивает время выполнения на 23,7% и не использует память

Инструмент для борьбы с утечками памяти во время выполнения программы.

- Перемещает на диск неиспользуемые объекты
 - ▶ Помечает неиспользуемые объекты и ссылки с помощью одного бита
 - ▶ При обращении к объекту изменяет значение бита
- Неактивен до 50% использования памяти, после 80% начинает перемещать объекты на диск
- Реализован в Jikes RVM
- Для пяти из десяти приложений удалось справиться с утечками памяти
- Увеличивает время выполнения программы в среднем на 6%

Фреймворк для разработки программного обеспечения с возможностью самовосстановления.

- Предотвращение утечек памяти — одна из возможностей
- ObjectDumpHealer — компонент, который отслеживает использование памяти и может перемещать объекты из виртуальной памяти на диск
- Классы объектов необходимо пометить аннотацией @Dumpable
- Увеличивает время выполнения программы на 2,5%
- Недостаток: обрабатывает только объекты, помеченные аннотацией

Инструмент для выявления утечек памяти во время выполнения программы для языков со сборщиком мусора.

- Определяет классы и места выделения объектов, которые приводят к увеличению используемой памяти
 - ▶ Строит взвешенный граф, узлы отвечают за один класс объектов, рёбра — за ссылки между объектами разных классов
 - ▶ После каждой сборки мусора сравнивает граф с предыдущим и находит классы объектов, размер которых увеличился
- Реализован в Jikes RVM
- Для всех приложений точно определил наличие утечек памяти
- Увеличивает время выполнения на 2,3%, дополнительно использует менее 0,5% виртуальной памяти

Сравнение инструментов

Метод	Определяет утёкшие объекты	Предотвращает сбой программы	Требует модификации сборщика мусора	Требует модификации компилятора	Требует модификации исходного кода	Издержки по времени	Издержки по памяти
Sleigh	Да	Нет	Да	Нет	Нет	29% (м.б. уменьшено до 11%)	0
LeakSurvivor	Да	Да	Да	Нет	Нет	23,7% (2,5% добавляет к Sleigh)	0-9 Мб
Melt	Да	Да	Да	Да	Нет	6%	-
Panacea	Нет	Да	Нет	Нет	Да	2,5%	-
Cork	Да (Только класс объекта)	Нет	Да	Нет	Нет	2,3%	1%

Таблица 1: Сравнение инструментов для борьбы с утечками памяти

План реализации

В дальнейшем планируется реализовать метод для предотвращения утечек памяти в OpenJDK¹.

Метод будет выполнять следующие задачи:

- Обнаружение объектов, которые являются утечкой памяти
- Информирование о возможной утечке
- Перемещение обнаруженных объектов на диск с возможностью их восстановления
- Удаление объектов с диска при переполнении диска

¹OpenJDK — версия среды разработки Java (Java Development Kit) с открытым исходным кодом.

- Проведен обзор инструментов для обнаружения и предотвращения утечек памяти
- Выполнено сравнение рассмотренных инструментов
- Составлены требования к методу для предотвращения утечек памяти

В дальнейшем планируется:

- Реализовать метод для предотвращения утечек памяти в OpenJDK
- Проанализировать влияние реализованного метода на выполнение программы