

УДК 004.4

В.А. Кравец

СРАВНЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ ВИРТУАЛЬНЫХ МАШИН JVM ДЛЯ ЯЗЫКА ПРОГРАММИРОВАНИЯ JAVA И CLR ДЛЯ ЯЗЫКА ПРОГРАММИРОВАНИЯ C#

Современный инструментарий разработчика программного обеспечения состоит из достаточно разнообразных по своему назначению модулей. Одним из основных компонентов среды разработки является виртуальная машина языка, являющаяся средой исполнения исходного программного кода. Рассматриваются две среды исполнения с сопутствующими им языками, являющиеся на сегодняшний день лидирующими в сфере разработки программного обеспечения, – Java Virtual Machine (JVM) с языком Java и Common Language Runtime (CLR) с языком C#.

Ключевые слова: JVM, CLR, Java, C#, виртуальная машина, сортировка данных.

Если сравнивать современную технологию разработки и технологию разработки 90-х, 70-х или 50-х гг. прошлого века, то можно заметить качественный рост в развитии технологии программного обеспечения. С 50-х годов идет планомерный рост в обеспечении сферы разработки и разработчиков новыми типами инструментов разработки. Одним из достижений сферы разработки программного обеспечения стало появление в начале 90-х годов XX века независимых от программной и аппаратной платформ сред выполнения исходного программного кода – виртуальных машин.

Виртуальная машина в контексте языков программирования – это особая программная среда, эмулирующая идеальную среду выполнения промежуточного программного кода, не зависящую от аппаратной платформы или операционной системы.

Первой языковой виртуальной машиной, применяемой в промышленной разработке, стала JVM – Java Virtual Machine («Виртуальная машина языка Java») от компании Sun Microsystems. Данная виртуальная машина была разработана в 1995 г. как основной компонент в программной платформе Java. Основное назначение JVM – интерпретация промежуточного байт-кода, скомпилированного из исходного кода, написанного на языке программирования Java, с помощью компилятора Java. [1]

Виртуальные машины Java доступны для всех существующих ныне семейств операционных систем, – Windows, BSD, Linux, Solaris, Unix, и др., а также для всех аппаратных платформ, на которых могут функционировать эти операционные системы. Для примера высокой степени распространенности виртуальной машины JVM и языка Java возьмем статистику распространения языков программирования по индексу TIOBE. Так, на март 2013 года доля языка Java соответственно равнялась 18.156%, или первое место в индексе. [2]

Поэтому можно утверждать, что и Java, и JVM имеют достаточно высокую степень распространенности в сфере разработки программного обеспечения.

Основным конкурентом программной платформы Java является среда исполнения CLR вместе с языком программирования C# от корпорации Microsoft. Разработка данной платформы связана с «патентными войнами» между компанией Sun и корпорацией Microsoft, итогом которых стал запрет для корпорации Microsoft разрабатывать свою версию JVM. В результате была разработана компонентная среда .NET Framework, а также виртуальная машина CLR (Common Language Runtime – «Общезыковая исполняющая среда») с языком C#. [3]

Так же как и JVM, в CLR происходит интерпретация байт-кода, скомпилированного из исходного кода на языке .NET Framework (например, C#). В то же время компиляция исходного кода происходит в промежуточный код на языке CIL, который и является фактически байт-кодом.

В отличие от JVM, CLR и .NET Framework реализованы только для платформы Windows, в то время как на других платформах имеются лишь проекты разработки по переносу CLR. [4]

Если руководствоваться индексом ТЮВЕ, можно сделать вывод об относительно высоком положении языка C# среди языков программирования по применимости, – пятое место или 6.597%. [2]

Кроме непосредственно статистических данных по виртуальным машинам JVM и CLR, для оценки их вычислительной мощности были проведены эксперименты по измерению скорости работы приложений, разработанных соответственно на языках Java и C#. Наиболее оптимальными для решения данной задачи были признаны алгоритмы сортировки данных вследствие их универсальности для большинства языков программирования, а также высокой степени применимости в промышленной разработке. Объектом сортировки данных были выбраны массивы, наличествующие как тип данных в обоих языках программирования.

Для разработки тестовых приложений были приняты следующие алгоритмы сортировки данных:

- 1) пузырьковый метод;
- 2) сортировка выбором;
- 3) быстрая сортировка, или сортировка Хоара. [5]

Базовой системой для запуска приложений выбрана операционная система Windows 7, как удовлетворяющая обеим средам исполнения. В качестве сред разработки были использованы Visual Studio 2008 для языка C#, Geany 1.23 и NetBeans 7.3 для языка Java. Аппаратные характеристики машины, на которой запускались тестовые приложения: мобильный процессор Intel Core I7-3520M, 2.9 GHz, 8 GiB ОЗУ.

Сортировка методом пузырька является одним из базовых методов сортировки массивов, однако не рекомендуется к применению в сортировке больших объемов данных из-за невысокой скорости. Сортировка методом пузырька заключается в обмене соседних элементов, удовлетворяющих условию сортировки. В приведенной ниже таблице показаны результаты сортировки массивов размерностью от 1000 до 100000 элементов.

Таблица 1

Время выполнения сортировки методом пузырька
для массивов различных размерностей

Количество элементов	Время работы C# + CLR + .NET Framework 3.5 (Release) (сек.)	Время работы Java 1.7 + JVM (сек.)
1000	0,002±0,001	0,015±0,006
2000	0,009±0,006	0,018±0,006
5000	0,027±0,003	0,050±0,005
10000	0,125±0,011	0,188±0,007
20000	0,558±0,010	0,710±0,010
50000	3,645±0,018	4,424±0,021
100000	14,85±0,02	17,66±0,03

Так же, как и алгоритм сортировки пузырьком, алгоритм сортировки выбором является неоптимальным в плане использования в промышленном программировании, но подходит для тестирования виртуальных машин (табл. 2).

Таблица 2

Время выполнения сортировки выбором
для массивов различных размерностей

Количество элементов	Время работы C# + CLR + .NET Framework 3.5 (Release) (сек.)	Время работы Java 1.7 + JVM (сек.)
1000	0,002±0,001	0,016±0,004
2000	0,009±0,006	0,018±0,005
5000	0,017±0,008	0,023±0,013
10000	0,058±0,007	0,064±0,007
20000	0,182±0,012	0,248±0,010
50000	1,071±0,012	1,453±0,021
100000	4,274±0,011	5,94±0,21
200000	17,15±0,04	21,97±1,10

Основные шаги алгоритма таковы:

- 1) найти номер минимального элемента в текущем списке данных;
- 2) поменять местами минимальный элемент с первым неотсортированным элементом;
- 3) продолжить сортировать хвост списка без отсортированного элемента.

Быстрая сортировка, или сортировка Хоара – один из самых быстрых алгоритмов сортировки данных. Данный алгоритм является базовым в промышленном программировании.

Основные шаги алгоритма таковы:

- 1) выбрать элемент, называемый опорным;
- 2) сравнить все остальные элементы с опорным, на основании сравнения разбить множество на три – «меньшие опорного», «равные» и «большие», расположить их в порядке меньшие – равные – большие;
- 3) повторить рекурсивно для «меньших» и «больших».

Обычно в качестве опорного элемента выбирается срединный элемент массива.

В ниже приведенной таблице идет сравнение скорости реализаций алгоритма. В отличие от первых алгоритмов, верхний порог количества элементов может быть увеличен на два порядка (табл. 3).

Таблица 3

Время выполнения быстрой сортировки Хоара
для массивов различных размерностей

Количество элементов	Время работы C# + CLR + .NET Framework 3.5 (Release) (сек.)	Время работы Java 1.7 + JVM (сек.)
50000	0,009±0,007	0,024±0,007
100000	0,012±0,004	0,024±0,007
500000	0,039±0,008	0,055±0,008
1000000	0,086±0,008	0,101±0,008
5000000	0,430±0,008	0,524±0,008
10000000	0,898±0,008	1,094±0,016

Анализируя результаты тестирования, можно утверждать, что скорость работы виртуальной машины CLR и языка C# выше, чем виртуальной машины JVM и языка Java. Следует учитывать, что в тестировании использовалась старшая версия виртуальной машины CLR - версия 3.5 2008 года выпуска, и младшая версия виртуальной машины JVM – версия 1.7 2012 года выпуска.

Поэтому можно утверждать, что, несмотря на больший технологический возраст, виртуальная машина CLR и язык программирования C# являются более производительными в области сортировки данных, нежели конкурирующая платформа Java.

Библиографический список

1. The Java® Virtual Machine Specification [Электронный ресурс]. Электрон. дан. Режим доступа: <http://docs.oracle.com/javase/specs/jvms/se7/html>. Загл. с экрана.
2. TIOBE Software [Электронный ресурс]. Tiobe index. Электрон. дан. Режим доступа: <http://docs.oracle.com/javase/specs/jvms/se7/html/jvms-1.html#jvms-1.1>. Загл. с экрана.
3. Среда CLR [Электронный ресурс]. Электрон. дан. Режим доступа: <http://msdn.microsoft.com/ru-ru/library/8bs2ecf4.aspx>. Загл. с экрана.
4. Mono [Электронный ресурс]. Электрон. дан. Режим доступа: http://www.mono-project.com/Main_Page - Загл. с экрана.
5. Алгоритмы сортировки [Электронный ресурс]. Викиучебник. Электрон. дан. Режим доступа: http://ru.wikibooks.org/wiki/Алгоритмы_сортировки. Загл. с экрана.

КРАВЕЦ Василий Александрович – магистрант факультета информационных технологий Костанайского государственного университета им. А. Байтурсынова.