String Deduplication in Virtual Machines

Konstantin Nasartschuk, Kenneth B. Kent, Dane Henshall
University of New Brunswick, IBM Canada
Faculty of Computer Science
{kons.na, ken}@unb.ca, dane_henshall@ca.ibm.com

Background

Research in automatic memory management aims to optimize the heap structure in order to perform less collection, shorter collection phases and to result in less fragmentation. However the information needed in order to accomplish this increases the heap size and is therefore an addition to the problem. An attempt to make up for the additional space needed is to provide a better execution time.

The Java heap structure shows that a large percentage of objects stored on the heap are strings. The number averages at 20% in most applications, but can get as high as 40% in special cases such as the J2EE application server. This project aims to use the heap object structure in general and the string object structure specifically in order to improve the memory use of the Java VM.

Related Work

String objects in Java are immutable. Immutable objects have the specific property that they can never change their content. If a change of the object is requested by the application, a new object with that content is created and a reference to the new object is written into the application reference. This is used to gain an advantage by multiple mechanisms in the Java VM such as the string table or the `.substring()` command.

Research on strings and string deduplication took several different approaches. As strings build the majority of the heap, researchers suggested for example to collect strings first in order to check if this would free enough space to continue execution. By doing this, the authors were able to reduce the garbage collection time as the not all objects had to be processed. Other research suggest a deduplication based on offline profiling. It demonstrates a considerable speedup once the training stage is completed and the JVM is trained on an application.

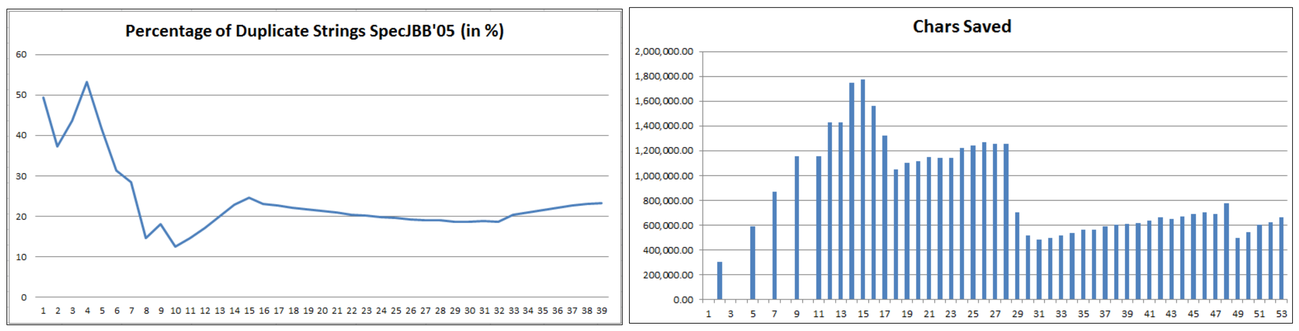

Figure 1: String Duplicates in specJBB 2005

Motivation

The string table approach currently used in virtual machines creates a large benefit for string objects during allocation time and decreases the memory usage of the application. However, it does not cover all of the strings created. Strings created using the `new` key word as well as constructed strings remain unaffected.

An initial analysis of the objects on the heap while running SPECjbb2005 and SPECjbb2013 revealed that some applications have a large amount of duplicate strings even after the deduplication performed by the string table and `.substring()`. Figure 1 and show the result of a complete analysis of all alive strings after each garbage collection. The graphs show the percentage of duplicate strings encountered as well as the possible heap space reduction for SPECjbb2005.

Approach

Our approach proposes an additional step during the traversal of the objects during the garbage collection phase. The idea is to store and compare all string objects encountered. In cases where duplicate string are found, they can be deduplicated. The flow of garbage collection will change to be as shown in Figure 2. The deduplication is performed by reusing the character arrays used for a string object while keeping all other header field parameters as they were. This method mantains the behavior described in the language specification, but potentially reduces the memory used by string objects.

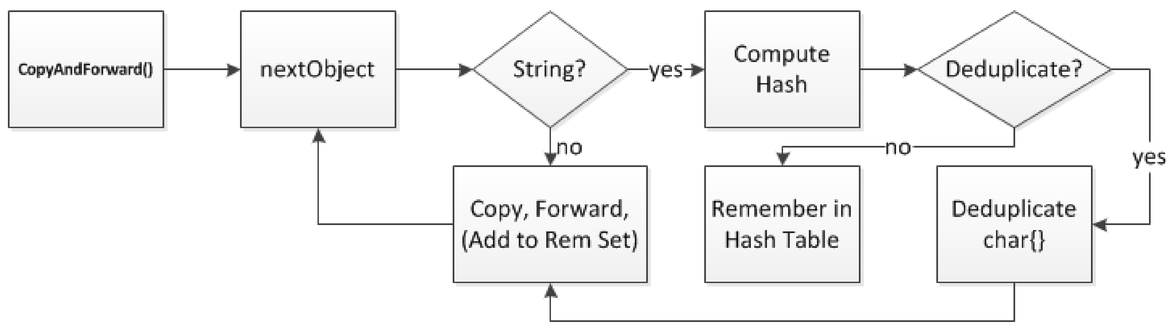

Figure 2: String Deduplication Flow

Future Work

The current state of the project is debugging and the evaluation process can begin in near future. The most interesting metrics are the memory impact and the computation time cost of the additional program flow.