

Abstract

Duplicated Strings: How are they Created and Cured?

MICHIHIRO HORIE^{1,a)} KAZUNORI OGATA^{1,b)} KIYOKUNI KAWACHIYA^{1,c)} TAMIYA ONODERA^{1,d)}

Presented: February 28, 2013

String data is often the most used data in recent Java applications, and it occupies the largest part of the Java heap. The practical problem is many wasteful duplicates in the values of the string objects. Such duplication consumes heap memory. To address this problem, we devised *Profile-Based String Deduplication* to minimize the duplication of string values. Our *StringProfiler* tool collects the calling contexts for the allocations of string objects during a trial program run. It determines which calling contexts created duplicate string values, and then it suggests which calling contexts can be optimized. Deduplication to reduce the duplications is done with a set of native methods that can return unified string objects without creating string objects. Finally, this presentation evaluates our approach by using actual Java applications. Our prototype implementation achieved a 9% reduction in the total size of the objects allocated in some standard Java benchmarks.

¹ IBM Research, Koto, Tokyo 135-8511, Japan

a) horie@jp.ibm.com

b) ogatak@jp.ibm.com

c) kawatiya@jp.ibm.com

d) tonodera@jp.ibm.com