

Abstract

Spin-Less Work-Stealing for Copying GC in OpenJDK

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

Presented: November 1, 2017

Spin loop is a well-known technique to keep a thread active without a context switch. In OpenJDK, spin loop is used on top of work-stealing in parallel GC. When a GC thread fails to steal a task multiple times continuously, it starts to spin in a loop for a short period, and then retries to steal a task. Since spin loop consumes the CPU resource, when the number of CPU cores is not enough for running the parallel GC threads, GC performance can be degraded seriously. Even if there is no performance impact with enough CPU cores, it is a waste of resource. In this presentation, we propose spin-less work-stealing, which has mechanisms to reduce the cost of spin loop with keeping the GC performance. We evaluated our approach by using SPECjvm2008.

¹ IBM Research – Tokyo, IBM Japan Ltd., Chuo, Tokyo 103–8510, Japan

a) horie@jp.ibm.com

b) horii@jp.ibm.com

c) ogatak@jp.ibm.com

d) tonodera@jp.ibm.com