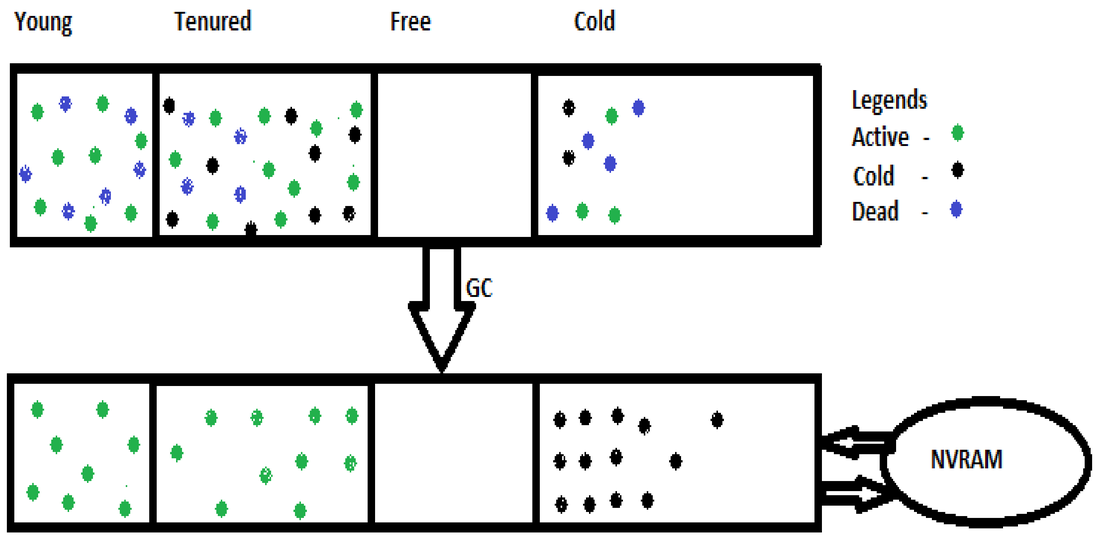
Garbage Collection of Cold Regions

Abhijit Taware, Kenneth B. Kent & Gerhard W. Dueck
University of New Brunswick, Faculty of Computer Science
Charlie Gracie
IBM Canada
{ataware, gdueck, ken}@unb.ca charlie_gracie@ca.ibm.com

Introduction

- Balanced GC divides the heap into multiple regions.
- Surviving objects age after each garbage cycle.
- Young objects reside in Eden space and older in the tenured space.
- Cold objects are alive, but very rarely accessed.
- Many different types of objects can become cold, such as partial error strings, logging objects or properties objects.
- Cold objects are overhead for GC. Such objects are moved to a designated cold region. GC can speed up by excluding cold regions during cleanup



Problem statement

- The cold region may get swapped out to disk due to lower access frequency.
- Some long running applications may end up having dead objects in cold region.
- Analyze benchmarks to see under which conditions cold regions should undergo garbage collection.
- Prototype implementation in open J9.

Estimated cold regions

- Cold regions need not be part of the partial GC.
- The table above gives estimated cold regions for various specifications.
- Specjbb2005 shows cold regions contribute to 13.96%, which can help reduce GC pause time.

Benchmark Name	Total regions	Estimated cold regions	Ratio
SPECjbb2005	1,024	143	13.96%
SPECjvm2008 compiler.compiler	1,024	14	1.37%
SPECjvm2008 compiler.sunflow	1,024	13	1.29%
SPECjvm2008 derby	1,076	107	9.94%
SPECjvm2008 sunflow	1,024	37	3.61%
SPECjvm2008 xml.transform	1,024	42	4.10%
SPECjvm2008 xml.validation	1,024	8	0.78%

Barry Zhou

Cold region classification

The cold regions could be in RAM, persistent memory or swapped out. The decision to GC cold regions can be taken based on following cost analysis –

- Regions in RAM: Should be GC'ed if high number of dead objects are suspected.
- Regions in Persistent Memory: The trade-off between how much space can be reclaimed in relation to the delay in performing the GC along with the wear on the persistent device.
- Paged out regions: Though cost of GC is high, it might help in some cases to free up the heap.

Cost considerations

- Number of cold regions are specified when JVM starts, so no runtime overhead is involved.
- Cold regions increase the number of copies during partial GC.
- Global GC does not have extra overhead due to cold regions.