# Introduction

Different from C/C++, java do not need program to release memory explicitly because there are garbage collectors inside java virtual machine. The garbage collection mechanism improves productivity and reduces the possibility of memory leak significantly. However, this mechanism may cause performance issue sometimes. After J2SE 1.2 is released, JVM garbage collection performance has been improved significantly. However, since there is no setting that fit all applications, developers need to change the JVM setting according to their scenarios or application features. The garbage collection principle and the corresponding options for garbage collection performance tuning will be introduced as below.

# Performance Measurement

Before we think about performance tuning, we should think about how performance of garbage collection is measured. Usually, there are 2 primary measures: throughput and pauses.

Throughput is the percentage of total time not spent in garbage collection. Lower through put means less system resources are available for program execution. The symptoms are: longer average response time, support less concurrent users. Pauses are the times when an application is frozen during garbage collection. Usually, pauses do not impact the average performance of an application but it is very important to the applications that are very sensitive to response time. For example, if the system requirement of an application is that the max response time is 3 seconds and the max pause time of garbage collection is also 3 seconds, this system requirement cannot be met.

In general, for most of the applications, developer need to trade-off between throughput and pauses. For example, a very large heap may maximize throughput but cause long pause time. A small heap can minimize pause time but cause lower throughput and garbage is collected more frequently.

# Generations

Since most of the java objects live in a short period, JVM separate the heap into 3 generation: young, tenured and permanent.

There are one eden space and 2 survivor spaces in the young generation. The size of those 2 survivor spaces are the same and one of them is empty at any time.

Most of the objects are initially allocated in eden space.

Empty survivor space

Eden

Survivor

Survivor

Objects are allocated here

Objects that are copy to this survivor space in the last minor GC

When the young generation is going to be filled up, a minor garbage collection will be caused. The collector will find all live objects in young generation and copy them to the empty survivor space.

It is going to be filled up. Minor GC is triggered

Eden

Survivor

Survivor

Copy live objects

The objects that are old enough will be promoted to tenured generation. Then, all objects in eden space will be cleared. The original non-empty survivor space will become an empty survivor that will be the live objects destination for next minor garbage collection.

Live objects are copied to here

Empty survivor space

It is cleaned and ready for allocation

Survivor

Survivor

Eden

Tenured