Memory Manager Performance Tests

Xiaoyuan Lu
Yongfeng Zhang

February 19, 2014

Abstract

This report first compares the performance of different free list implementation algorithms by calculating the hole number and hole sizes left by each of them after going through some workload. Results show that during a relatively heavy workload (we tried to use up the memory), allocation failure (not enough memory) didn't occur for first fit, best fit and worst fit, while next fit has two failurs and random fit has an average of 2.2 failures. In general, worst fit and first fit have the best performance among other algorithms. We then compared the performance of buddy allocator and free list allocators. Under different kinds of workloads, they show very different performances.

1 Test for Free List Allocators

1.1 Workload Design

The memory pool set up is of size 1KB. We ran the same workload once for first fit, next fit, best fit and worst fit. For random fit, we ran it ten times. The workload is designed to have both large memory requests (up to 256 bytes) and small memory requests (down to 8 bytes). The workload is a mix of memory allocations and memory frees, ended with several small memory allocations to use up the memory. We tried to allocate as much memory as possible, until memalloc failure first appears for some allocator. The purpose is to test their performances at their limitations.

1.2 Performance Analysis

We concluded the test results in the chart below. You can get the same results running test1.c.

Allocator	Number of holes	Min hole size	Max hole size	Average hole size	SD of hole size	Number of failures
First Fit	3	2	10	5.3333	3.3993	0
Next Fit	2	22	72	47.0000	25.0000	2
Best Fit	1	56	56	56.0000	0.0000	0
Worst Fit	3	2	18	10.0000	6.5320	0

The ten test results for the random fit are as follows:

Tests	Number of holes	Min hole size	Max hole size	Average hole size	SD of hole size	Number of failures
One	1	96	96	96.0000	0.0000	2
Two	2	2	42	22.0000	20.0000	0
Three	2	8	112	60.0000	52.0000	2
Four	3	4	140	54.6667	60.6923	4
Five	3	2	44	16.6667	19.3448	1
Six	2	4	126	65.0000	61.0000	4
Seven	3	4	132	74.6667	53.0995	5
Eight	2	4	92	48.0000	44.0000	2
Nine	2	10	18	14.0000	4.0000	1
Ten	1	46	46	46.0000	0.0000	1
Ave.	2	18	84.8	49.7000	31.4137	2.2

As we can see in the charts, as far as hole size is concerned, first fit has the best overall performance, followed by worst fit. Their difference is actually quite small, so I believe as the number of tests increases, these two fits will have very similar performances. Both next fit and best fit have large hole sizes compared to the former two. Besides, next fit failed two of the memory allocations. Surprizingly enough, random fit almost has the worst performance. It has the largest maximum hole size, standard deviation and number of failures. Therefore, in general, fist fit and worst fit have the best performance, while choosing the nodes randomly is the worst idea.

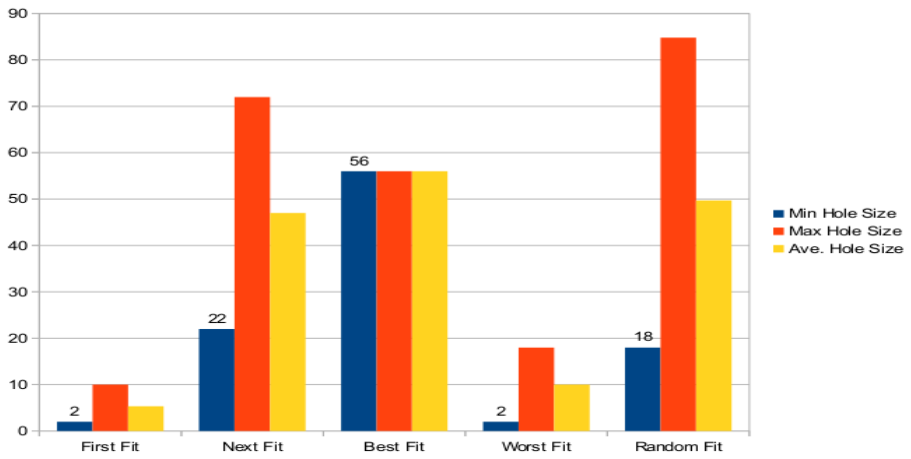

Figure 1: Hole Size for Free List Allocators