Subject: Re: C++ FQA
Posted by mirek on Mon, 12 Nov 2007 21:43:31 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Mon, 12 November 2007 15:39Quote:
BTW, I did only that single benchmark D vs U++ - U++ was about 2x faster, but what was really shocking is that D consumed 5 times as much memory....

As said before, D has a known performance problem with built in hash maps. I guess the developers are more concerned with the shape of the language and haven't had time to optimize such details yet.

As for the memory consumption, I am not surprised at all. Mark and sweep abandons traditional memory paradigms as allocation when you need and freeing again when you are done with the object. As much as you can optimize these algorithms, allocation and memory freeing are very time consuming operations (relatively speaking of course). In mark and sweep, allocation consists of an if to see if there is enough space. If not, a huge block is allocated. If there is enough space, a simple pointer incrementation is performed. Very fast.

Sorry for being rude, but your understanding of actual GC (and especially conservative GC) is sort of lacking

Quote:
Another advantage is zero memory fragmentation is the mark-and-sweeper is also a moving-compactor.

Not in all cases and AFAIK, not in D. I might be wrong, but I am afraid that D, using conservative GC, is still prone to memory fragmentation.

Quote:
And if you make it generational too, you can really optimize it.

I am no expect in GC and perhaps I am wrong about this, but IMO generational GC and moving GC are mutually exclusive.

Quote:
So in theory, GC programs should run a lot faster. Memory is cheap, and if with 1-2 GB of memory extra you can gain sufficient extra speed, I think GC will continue to get more and more popular. But this is only theory, and that's why I'm interested in some scientifically sane benchmarks.

...or you can do twice as much work with existing hardware....

Anyway, I think the main argument w.r.t. GC is management of other resources than memory.

With GC, it is virtually impossible. And no, you cannot have destructors and GC working together.

I believe that with a couple of tricks, I am getting (with U++) more than I could get with GC - all resources are managed by program structure.

Mirek