Subject: Re: C++ FQA
Posted by mirek on Sun, 11 Nov 2007 08:54:11 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Sat, 10 November 2007 17:57luzr wrote on Sat, 10 November 2007 17:54cbpporter wrote on Sat, 10 November 2007 11:31
And I'm quite surprised to see people who don't like gc, but have nothing against reference counting, which is slower and almost impossible to use efficiently in a multi-threading application.

I mostly agree with this...

As with GC, refcount can be made thread safe, IMHO.

The things is that even thread unsafe reference counting is about as fast or slower than mark&sweep GC.

And, w.r.t. thread safety, the another trouble is that you cannot safely use atomic operations only when the object is really shared between threads (when it is needed).

Quote:
Even pick_ can be not thread safe, if it's bad coded, and must have some sort of synchronizing code to be thread safe.

Everything can be thread unsafe if you really try. Anyway, the simplest pick_ implementation is naturaly thread safe.

Quote:
I would not accept a language based mostly on GC, but I've got nothing against an optional gc among other management stuffs.

The problem is that this is not quite possible.

Quote:
BTW, I can't see how refcount can be slower than GC... maybe I'm wrong, but I'd like to have it explained !

OK, only think about the amount of operations that have to be performed in simple "return the value" scenario

RefCounted<Foo> Fn() {

```
   RefCounted<Foo> x = new Foo;
   .....
   return x;
}

void Fn2() {
   RefCounted<Foo> y = new Foo;
   ...
   y = Fn();
}
```

Mirek

---