Subject: Re: Core 2019
Posted by mirek on Sun, 09 Jun 2019 21:11:41 GMT
View Forum Message <> Reply to Message

Novo wrote on Sun, 09 June 2019 21:39mirek wrote on Sun, 09 June 2019 12:54Looking at peak profile, it looks like there are very little "small" blocks and most of memory is in those 80 "huge" (that means >64KB) blocks.

Can that be correct?

Mirek
It is hard to tell. I'm not controlling that.
Another problem is that all allocations/deallocations happen in CoWork's threads. I cannot call RDUMP(*PeakMemoryProfile()) inside of CoWork because it will be called at least 181363 times ...

Why would you want to? Peak is really peak, it is profile at the moment when there is maximum memory use.

One caveat about profile is that it is only profile of current thread for small and large blocks. But our problem is with huge blocks anyway.

Quote:
The app is parsing Wikipedia XML dump. It is decompressing a bz2 archive and parsing chunks of XML. After that my own parser is parsing Mediawiki text.
As a first pass my parser is building a list of tokens organized as a Vector<> (I'm not inserting in the middle :))
My parser is avoiding memory allocation at all possible costs. I'm calling Vector::SetCountR and reusing these vectors. When I need to deal with String I'm using my own not owning data string class.

Well, maybe there can also be an interference with MemoryTryRealloc (as those Vectors grow). Perhaps you can test what happens if

```
bool  MemoryTryRealloc(void *ptr, size_t& newsize) {
 return false; // (((dword)(uintptr_t)ptr) & 16) && MemoryTryRealloc__(ptr, newsize);
}
```

Quote:
Unfortunately, I cannot control memory allocation with XmlParser. I have to relay on the default allocator.

There are not many... BTW, are you parsing memory - XmlParser(const char *), or streams -

XmlParser(Stream& in) ?

Mirek