Don’t save anything you can recalculate

devblogs.microsoft.com/oldnewthing/20041220-00

December 20, 2004

Raymond Chen

Nowadays, a major barrier to performance for many classes of programs is paging. We saw earlier this year that paging can kill a server. Today, another example of how performance became tied to paging.

The principle is “Don’t save anything you can recalculate.” This of course, seems counterintuitive: Shouldn’t you save the answer so you don’t have to recalculate it?

The answer is, “It depends.”

If recalculating the answer isn’t very expensive and has good data locality, then you may be better off recalculating it than saving it, especially if saving it **reduces** locality. For example, if the result is stored in a separate object, you now have to touch a second object—risking a page fault—to get the saved answer.

Last time, we saw how Windows 95 applied this principle so that rebasing a DLL didn’t thrash your machine. I’m told that the Access team used this principle to reap significant performance gains. Instead of caching results, they just threw them away and recalculated them the next time they were needed.

Whether this technique works for you is hard to predict. If your program is processor-bound, then caching computations is probably a good idea. But if your program is memory-bound, then you may be better off getting rid of the cache, since the cache is just creating more memory pressure.

Raymond Chen

Follow