

---

Subject: Re: Vector performance on a specific situation

Posted by [mirek](#) on Tue, 18 Jun 2013 08:52:34 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

crydev wrote on Tue, 18 June 2013 03:01Hello,

I have a question about the Vector's performance in a specific situation. I have a program that utilizes 8 threads on new systems, heavy utilization of paralellism. Say I have a Vector containing 300 items. I split the indexes of those items over 8 threads, meaning the Vector will be accessed from 8 threads simultaneously, but every thread accesses a different item. The same memory location is never modified.

I have read something about Vector cache lines. What is the performance of the U++ implementation of the Vector in this situation?

It all dependes on sizeof(T) etc... but if you are doing a lot of access to elements and distribute threads in nearby indicies, cacheline sharing between threads is indeed a big problem.

Note that trivial Vector->Array does not really help here, as individual elements will be likely allocated in the same cachelines cells.

So it all depends on what you are doing with elements. 300 cells does not sound like too many, indicating that per-cell computation is pretty heavy (if it there is any advantage to use multiple threads).

For more qualified reply I would need to know definition of T and some description about computation.

Mirek

---