
Subject: String.Compare optimised

Posted by [mirek](#) on Thu, 29 Oct 2020 11:59:46 GMT

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

Well, it looks like there was a slight regression in String.Compare performance in 2020.2 in comparison with 2020.1

It is now not only rectified, but the new code is even significantly faster.

Here is the final benchmark comparison between trunk and 2020.1 (and std::), sorting all words of Alice in wonderland:

* C:\upp\out\benchmarks\CLANGx64.Debug_Full.Release\Sort.exe 29.10.2020 12:52:10, user: cxi

AliceWords().GetCount() = 26467

TIMING Sort Vector<String>: 386.00 ms - 1.29 ms (386.00 ms / 300), min: 0.00 ns, max: 3.00 ms, nesting: 0 - 300

TIMING std::sort std::vector<std::string>: 887.00 ms - 2.96 ms (887.00 ms / 300), min: 2.00 ms, max: 6.00 ms, nesting: 0 - 300

* C:\upp\out\benchmarks_2020.1\CLANGx64\Sort.exe 29.10.2020 12:53:29, user: cxi

AliceWords().GetCount() = 26467

TIMING Sort Vector<String>: 499.00 ms - 1.66 ms (499.00 ms / 300), min: 1.00 ms, max: 3.00 ms, nesting: 0 - 300

TIMING std::sort std::vector<std::string>: 874.00 ms - 2.91 ms (874.00 ms / 300), min: 2.00 ms, max: 6.00 ms, nesting: 0 - 300

BTW, fast path of String comparison in inlined assembly now looks like this:

```
400047EB  mov rcx,[r12]
400047EF  bswap rcx
400047F2  mov rdx,[rdi]
400047F5  bswap rdx
400047F8  cmp rcx,rdx
400047FB  jnz 0x14000486d
400047FD  mov rcx,[r12+0x8]
40004802  bswap rcx
40004805  mov rdx,[rdi+0x8]
40004809  bswap rdx
4000480E  cmp rcx,rdx
```

