

Benchmarks Java vs MS C# vs MS C++

Test env: Windows 7 x64, 4 GB RAM, Intel Core2 Quad Q9550 (2,83 GHz)

Java: jdk-1_6_0_16

MS C#: MS VS 2008SP1 Express, .NET 3.5SP1 (console app)

MS C++ 2008 (console app)

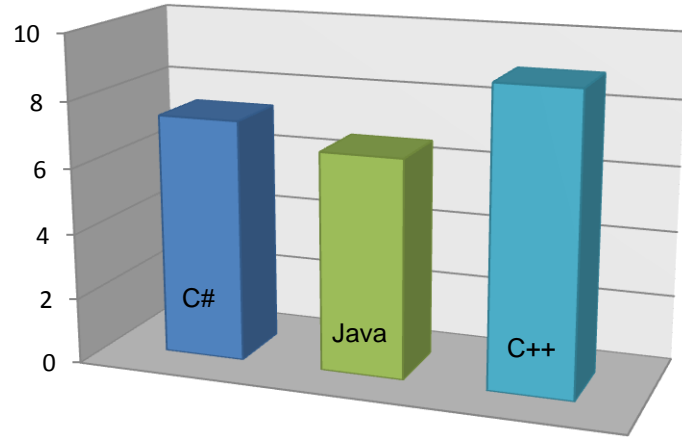
Note that all tests have been implemented without any language specific optimizations (i.e. using chars in C++ rather than STL strings).

Feel free to send any comments to: mtrzaska@pjwstk.edu.pl

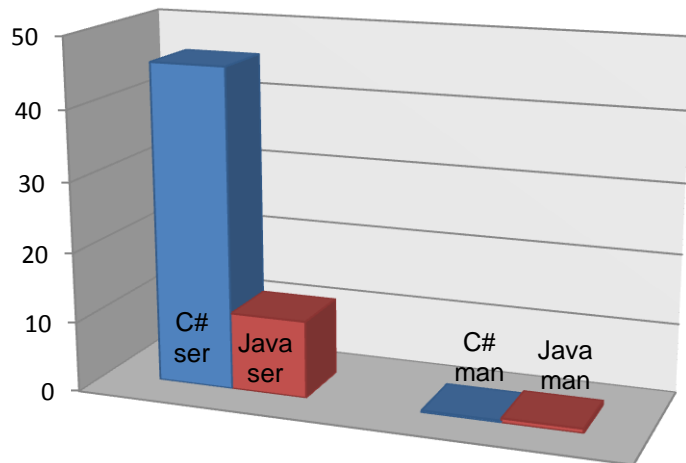
| Benchmark | Description (all tests have been compiled in release mode and started without debugging) | MS C++ 2008 | | | |
|----------------------------|--|-----------------------------------|-----------------------------------|------------------------------------|-----------------------------------|
| | | MS C# 3.0 (.NET 3.5SP1) | Java* | std::map | stdext::hash_map |
| [seconds] (less is better) | | | | | |
| #1 | Tests speed of working mapping collections (Hashtable for C#, TreeMap for Java). First 5 000 000 items is added to a collection. Then 500 000 items is retrieved based on a key and added to the second collection. The result is a total time of operation. Note that in case of Java we use Integer (object) rather than int (value). | 10,42 | 6,58 | NA | NA |
| #2 | Same as benchmark #1 except that generic collections have been used: Dictionary for C#, generic HashMap for Java, STL map for C++ (two types). The time for C++ does not include time for releasing memory (same in C# and Java). | 7,37 | 6,65 | 9,00 | 9,00 |
| #3w | The test starts with operation encapsulated in benchmark #2. After populating a second collection it is serialized (binary mode) or manually written to a disk file. | Serialization | Maunal | Serialization | Maunal |
| | | 11,86 | 7,68 | 20,52 | 7,01 |
| #3w | | File size: 28 390 349 bytes | File size: 14 388 893 bytes | File size: (21 389 128 bytes | File size: 14 961 597 bytes |
| #3r | Reading a collection from a disk file generated in benchmark 3w. | 45,68 | 0,29 | 11,03 | 0,63 |

* Java VM required increasing heap momory (VM parameter: -Xms512M -Xmx1024M)

Benchmark #2 (generic (template) collections)



Benchmark #3r - reading (serialization and manual)



Benchmark #3w - writing (serialization and manual)

