

Projekt i EDAA35

Jämförelse av exekveringstider av C++ och Java

Samuel Rosenqvist, sa7638ro-s@student.lth.se, Rasmus Olsson, ra0011ol-s@student.lth.se,
Douglas Tönnesen, do3718to-s@student.lth.se

Sammanfattning—The abstract goes here.

I. INLEDNING

DENNA rapportens syfte är att undersöka och diskutera eventuella skillnader i exekveringstid mellan vanliga algoritmer implementerade i Java och C++. Jämförelsen kan underlätta valet av programspråk för en programmerare. Rapporten behandlar relativt grundläggande algoritmer och bör inte användas för att ta beslut angående valet till större program och projekt.

5 maj, 2017

II. BAKGRUND

Programspråk skiljer sig åt på flera sätt och det är därför nödvändigt att jämföra skillnaderna för att kunna fatta beslut angående vilket språk som är bäst lämpat för den givna situationen. Genom att jämföra exekveringstid kommer man ett steg närmare ett beslut. Exekveringstid är inte alltid högsta prioritet men när man väljer mellan språk som C++ och Java så spelar det ofta stor roll.

III. FRÅGESTÄLLNING

Med avseende endast på exekveringstid, vilket programspråk lämpar sig bäst för enkla algoritmer?

IV. METOD

Algoritmer som jämförts är C++ `std::sort` och Javas `Collections.sort`, samt implementering av merge sort och även beräkning av de första en miljon printalen. Exekveringstiden är uppmätt i programmen med Javas `System.nanoTime` och i C++ med `high_resolution_clock` från `chrono` biblioteket. Jämförelser och grafplottar har gjorts i R.

V. VALIDITETSDISKUSSION

Eftersom vår C++ kunskap ej är på samma nivå som vår Java är det möjligt att C++ inte utnyttjats till sin maximala potential. För att motverka detta har vi försökt använda implementeringar skrivna av vanare C++ användare.

Då vi ej har tillräcklig kunskap inom C++ kunde vi inte göra jämförelser på längre och mer komplicerade program, utan fick nöja oss med relativt enkla program vilket kan påverka resultaten.

VI. RESULTAT

VII. DISKUSSION

VIII. SLUTSATSER

BILAGA A

Appendix one text goes here.

BILAGA B

Appendix two text goes here.

REFERENSER

- [1] H. Kopka and P. W. Daly, *A Guide to L^AT_EX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.