ACCU 2022

MEASURING THE OVERHEAD OF IOSTREAMS

WHAT IS THE DIFFERENCE?

AHTO TRUU

UNIVERSITY OF TARTU
1632

Measuring the Overhead of C++ iostreams

Ahto Truu

MEMBER OF JURY, ESTONIAN OLYMPIAD IN INFORMATICS
SOFTWARE ARCHITECT, GUARDTIME

ACCU Conference, 07-APR-2021


accu
2021
VIRTUAL EVENT
Bloomberg
Engineering
undo
mosaic
CONSULTANTS TO FINANCIAL SERVICES
Programming as a Sport:
What Do You Mean?
Ahto Truu

Programming Competitions

Designing and implementing
correct and efficient algorithms
to solve given data processing problems.

How Does the Ranking Work?

Correctness

- “black box” testing with a set of test cases

Efficiency

- execution time and memory limits per test case
- increasingly large and/or difficult test cases

Algorithmic Complexity

- Suppose a problem that transforms a sequence of N integers
- Want to distinguish between O(N) and $O(N^2)$ algorithms
- But the real running time is
  - $A+B \cdot N+C \cdot N$ for the O(N) algorithm
  - $A+B \cdot N+C \cdot N^2$ for the $O(N^2)$ algorithm
  - … with typically B >> C
- So, how large an N do we need?

Concrete Complexity

- Even worse, the real real running time is
  - $A+B_1 \cdot N+C_1 \cdot N$ for the O(N) algorithm
  - $A+B_2 \cdot N+C_2 \cdot N^2$ for the $O(N^2)$ algorithm
- Where B_1 and B_2 depend on non-essential technical details

Test Setup

- Generate N·M integers
- Measure the time to copy them from stdin to stdout
  - … using stdio from C
  - … using cstdio from C++
  - … using C++ iostreams
    - … with or without `ios::sync_with_stdio(false)`
    - … with or without `cin.tie(nullptr)`
    - … with `'\n'` or `endl` as line terminator
  - `./source $N $M >input.txt && time -p ./pipe <input.txt >output.txt`
  - `./source $N $M | (time -p ./pipe) | ./sink >/dev/null`
- GCC 9.2.0 on Ubuntu 20.04, all compiled with -O2

Results

- Tests with 1M/2M lines, 10/20 numbers per line
- No difference between using cstdio from C and C++
- C++ iostreams with `'\n'`
  - … about 11× slower on files and 8× slower on pipes by default
  - … about 9× slower on files and 7× slower on pipes with unsync
  - … about 1.5-2× slower with untie
  - … about 10-20% faster with untie+unsync
- No difference between `'\n'` and endl
  - … except endl about 2× slower with untie+unsync
- No difference between 1M·20 and 2M·10 numbers
  - … except 2M·10 about 20% slower with untie+endl

UNIVERSITY OF TARTU
1632

Thank You!

Questions, comments?

ahto.truu@ut.ee
@ahtotruu