Project2 Report

Purpose of the experiment

Compare the efficiency of vector dot product operations in c++ and java and analyze the reasons.

Operating Environment

- OS:Mac OS
- Chip: Apple M1
- Total number of cores: 8 (4 performance and 4 power efficiency)
- Memory: 8 GB

Data Structure and Algorithms

In c++ and java code, the data types and algorithms used are kept consistent.

In both c++ and java, arrays are used to represent vectors. The arrays are filled with random number.

Program runtime includes only the process of multiplying vectors, not the process of generating arrays, reclaiming memory, and IO.

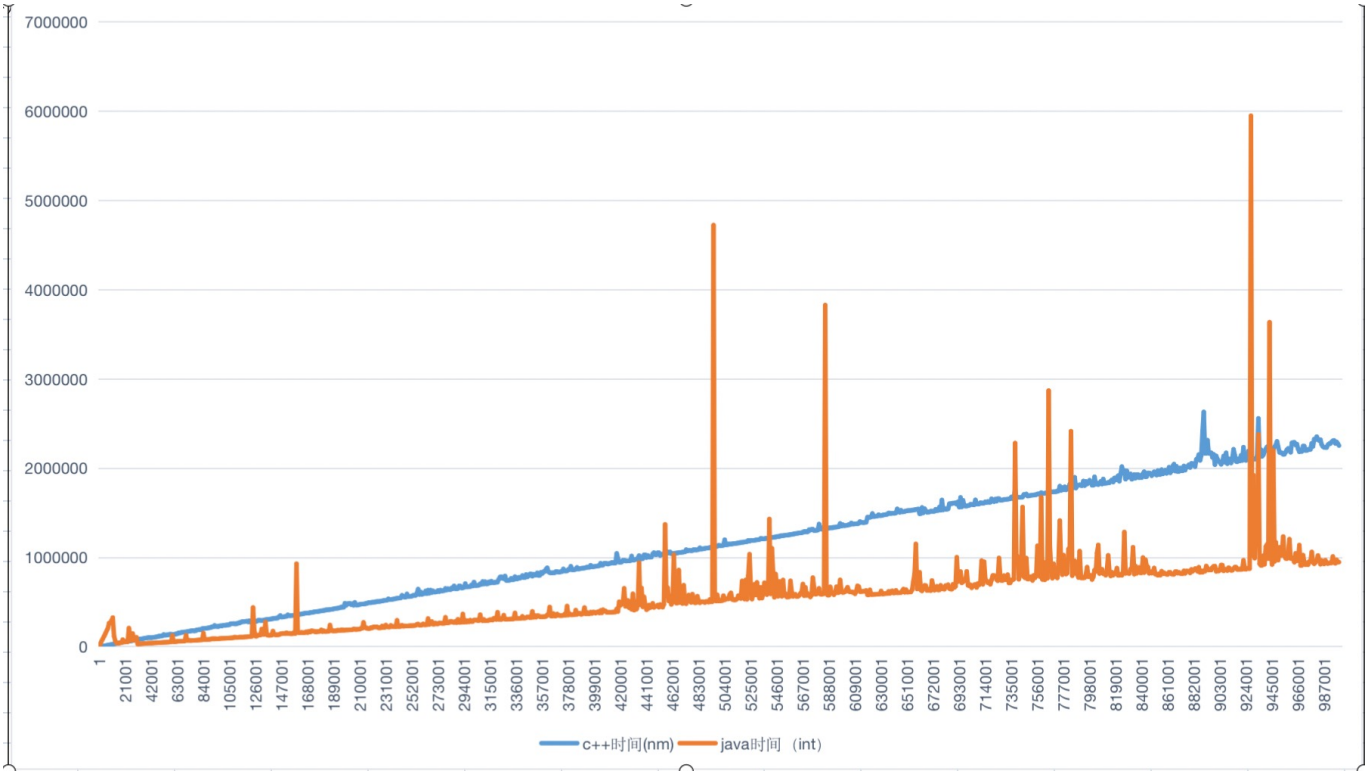
The program will record the running time for vectors of length 1 to 999001 in steps of 1000 and write it to a .csv file.

Finally, data processing software is used to generate line graphs to visually compare performance differences.

Methodology

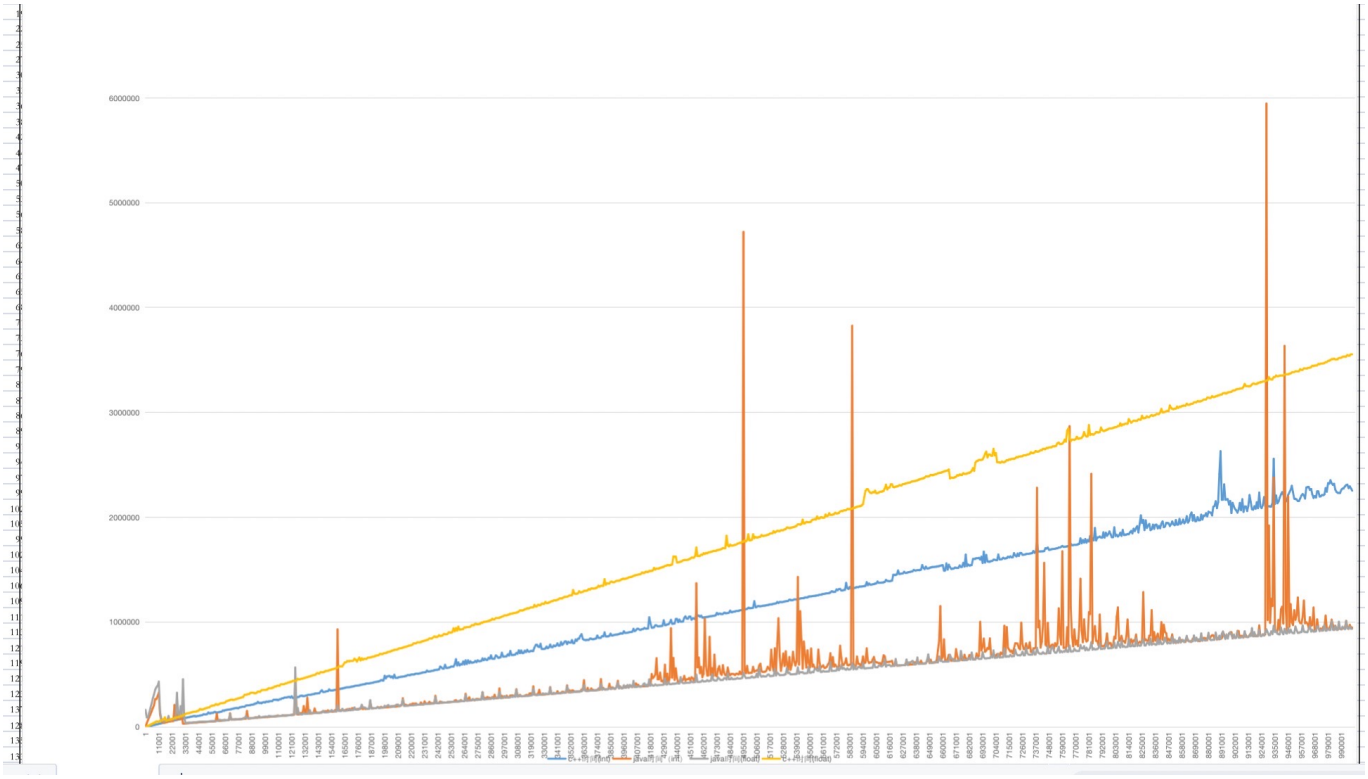
The control variable method was used to compare the efficiency of the two.

1.Run results for both languages,data type:int



From the experimental results, we can find that java runs slower when the amount of data is relatively small, but when the amount of data increases, java runs faster than C++, which is not in line with the original expectation.

2.The factor that changed in the first set of experiments was the data type of the array.Different data types may cause differences in runtime between the two languages, so change the data type to float.



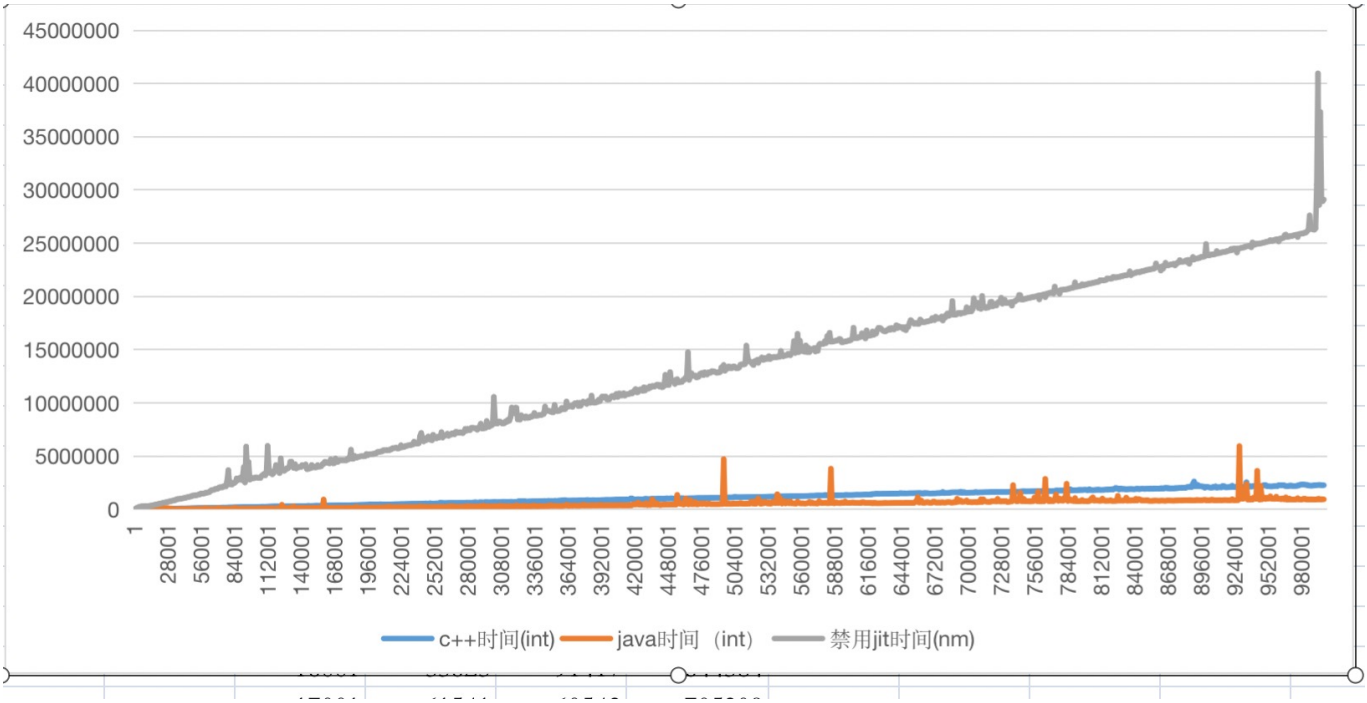
First of all, C++'s floating-point operations are significantly slower than integer operations, as expected.

Java's floating-point operations are not very significantly slower than plastic operations, which is relatively unexpectable.

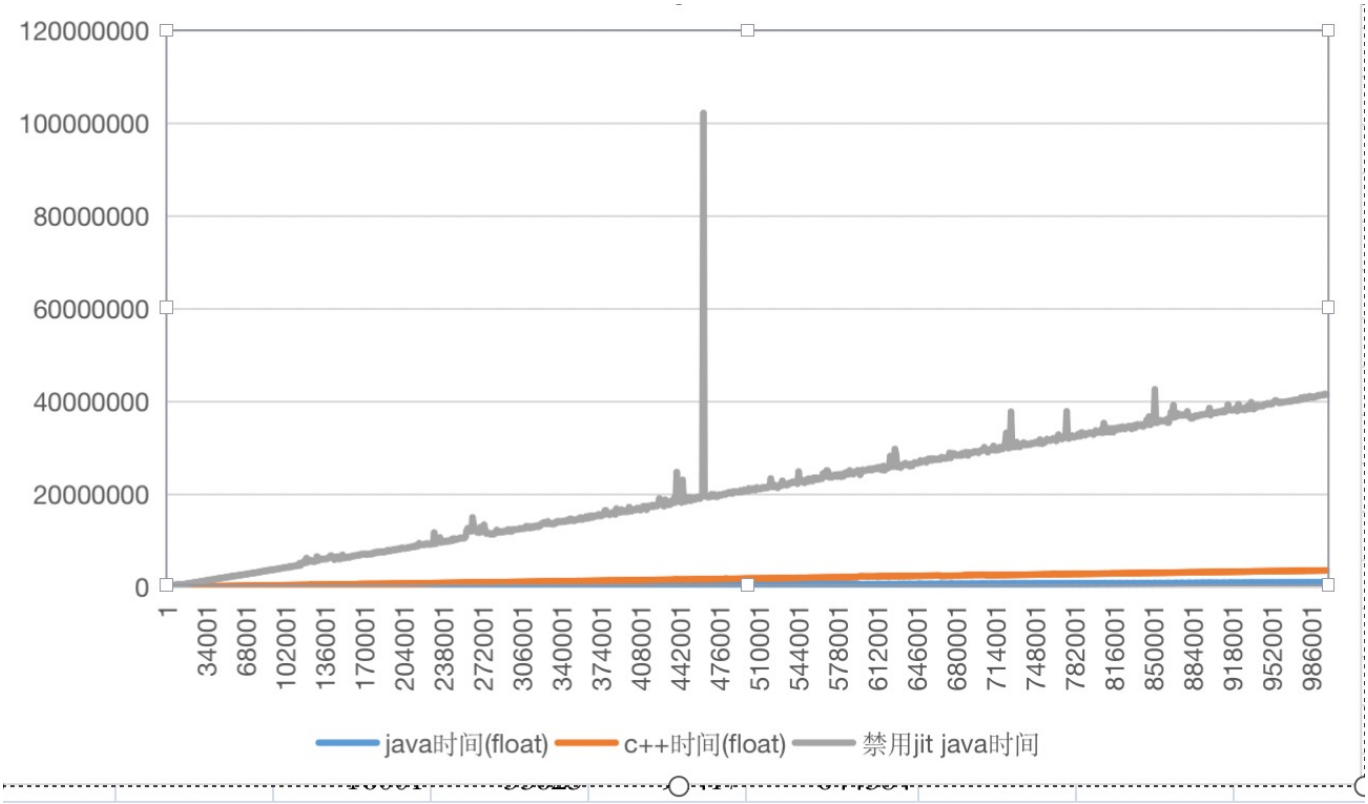
The floating point operations in C++ are also slower than Java and do not meet expectations.

3.JIT compilers usually monitor the code during program execution, compile frequently executed code fragments into native machine code, and cache these compilation results so that the compiled code can be called directly during the next execution, thus avoiding a repetitive interpretation and execution process. Considering that there is a lot of repetitive computation in the program, try disabling JIT and then see the change in java efficiency.

Integer calculation:



Float calculation:



After disabling JIT, Java's runtime increased substantially, far exceeding C++, as expected.

Conclusion

The Java compiler is very efficient.When java compiler optimization is disabled, C++ is more efficient than Java, but when compiler optimization is turned on, Java can even outperform C++.

More directions

- Java is sometimes more efficient than C++, probably because of the automatic garbage collection mechanism.
- May be related to thread process scheduling.