


```

        score.end());

        std::cout << std::setw(GAP) << line
        << std::setw(GAP) << std::fixed <<
std::setprecision(1)
        << std::setw(GAP) << score[score.size() /
2].count()
        << std::endl;
    }
}

```

De size.cc:

```

#include <algorithm>    // nth_element
#include <array>        // array
#include <chrono>       // high_resolution_clock
#include <iomanip>       // setw
#include <iostream>     // cout
#include <vector>       // vector

using namespace std::chrono;

const unsigned MINSIZE = 1 << 10; // minimum line size to test:
1KB
const unsigned MAXSIZE = 1 << 26; // maximum line size to test:
32MB
const unsigned GAP = 12;          // gap for cout columns
const unsigned REP = 100;         // number of repetitions of
every test
const unsigned STEPS = 1e6;       // steps

int main()
{
    std::cout << "#"
        << std::setw(GAP - 1) << "line (B)"
        << std::setw(GAP) << "time (Âµs)"
        << std::endl;

    for (unsigned size = MINSIZE; size <= MAXSIZE; size *= 2)
    {
        std::vector<duration<double, std::micro>> score(REP);

        for (auto &s: score)
        {
            std::vector<char> bytes(size);

            auto start = high_resolution_clock::now();

            for (unsigned i = 0; i < STEPS; ++i)
                bytes[(i*64)&(size-1)]++;
        }
    }
}

```

```

        auto stop = high_resolution_clock::now();

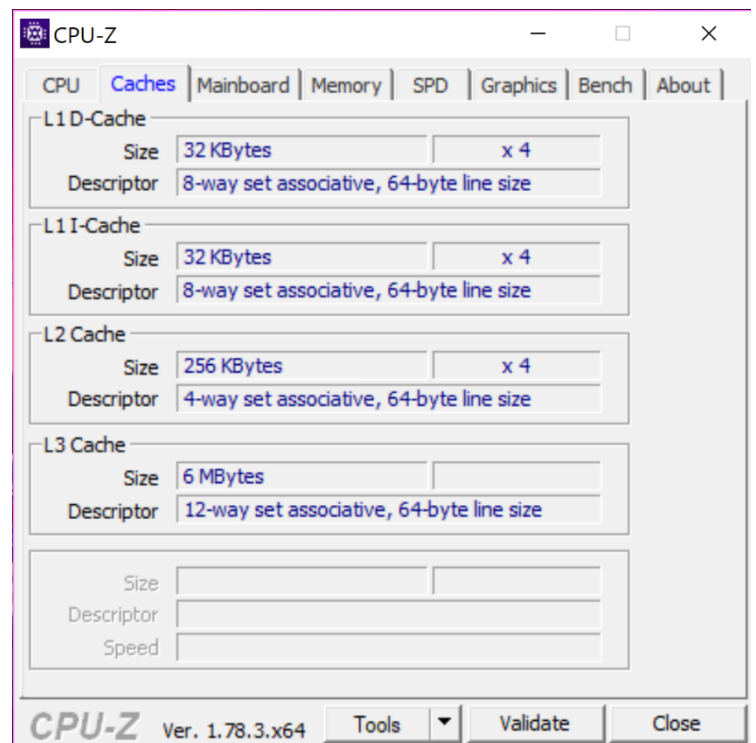
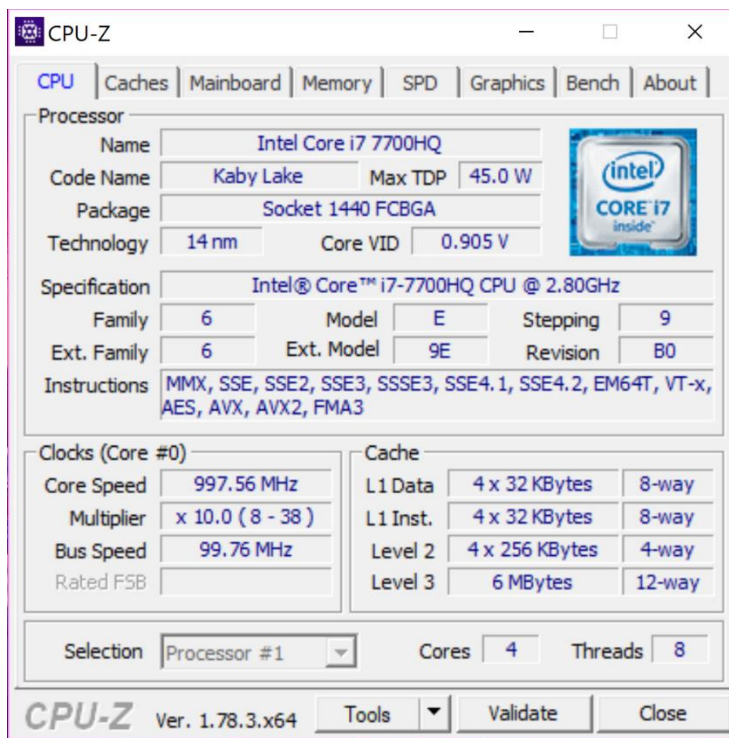
        s = stop - start;
    }

    std::nth_element(score.begin(),
                    score.begin() + score.size() / 2,
                    score.end());

    std::cout << std::setw(GAP) << size
               << std::setw(GAP) << std::fixed <<
std::setprecision(1)
               << std::setw(GAP) << score[score.size() /
2].count()
               << std::endl;
    }
}

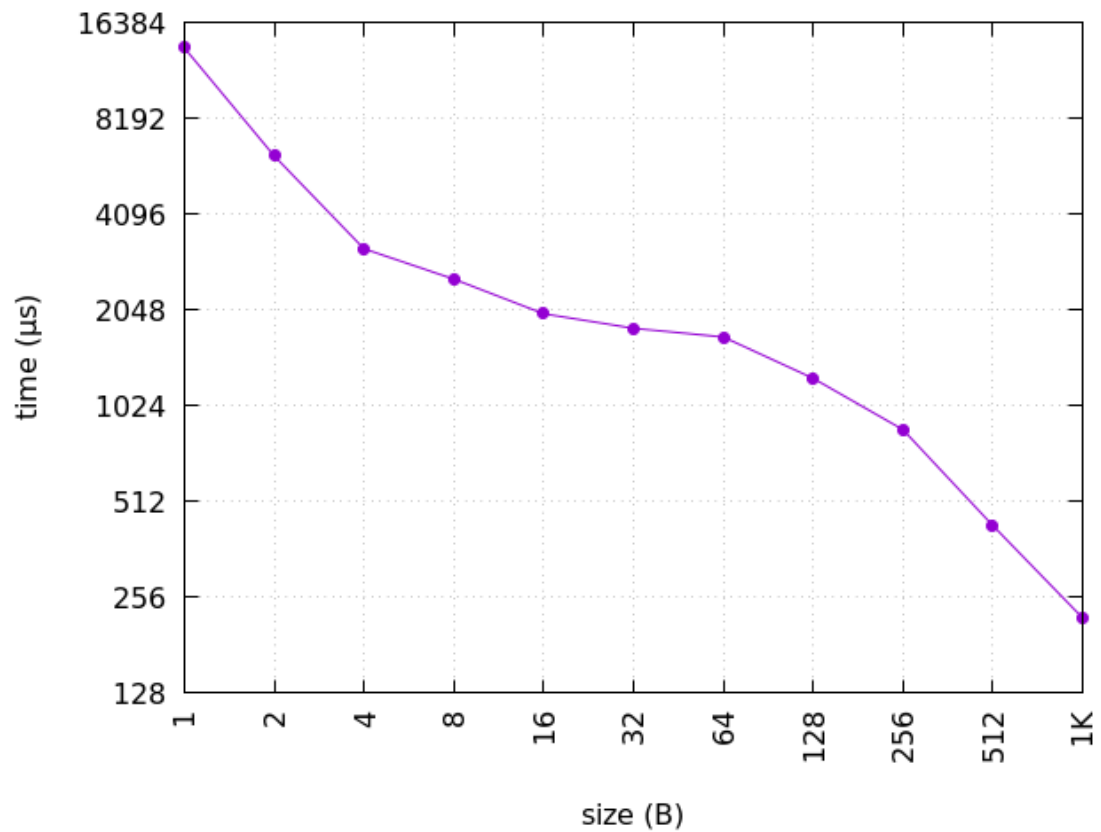
```

Mientras que las características de mi PC son algo complejas ya que estoy ejecutando Ubuntu en una maquina virtual de Vmware en la cual tengo asignados 4 núcleos lógicos (de los 8 disponibles) y 4gb de Ram DDR4 (de las 16gb disponibles), teniendo en cuenta eso adjunto la imagen de las características concretas del equipo realizadas con el programa CPU – Z:



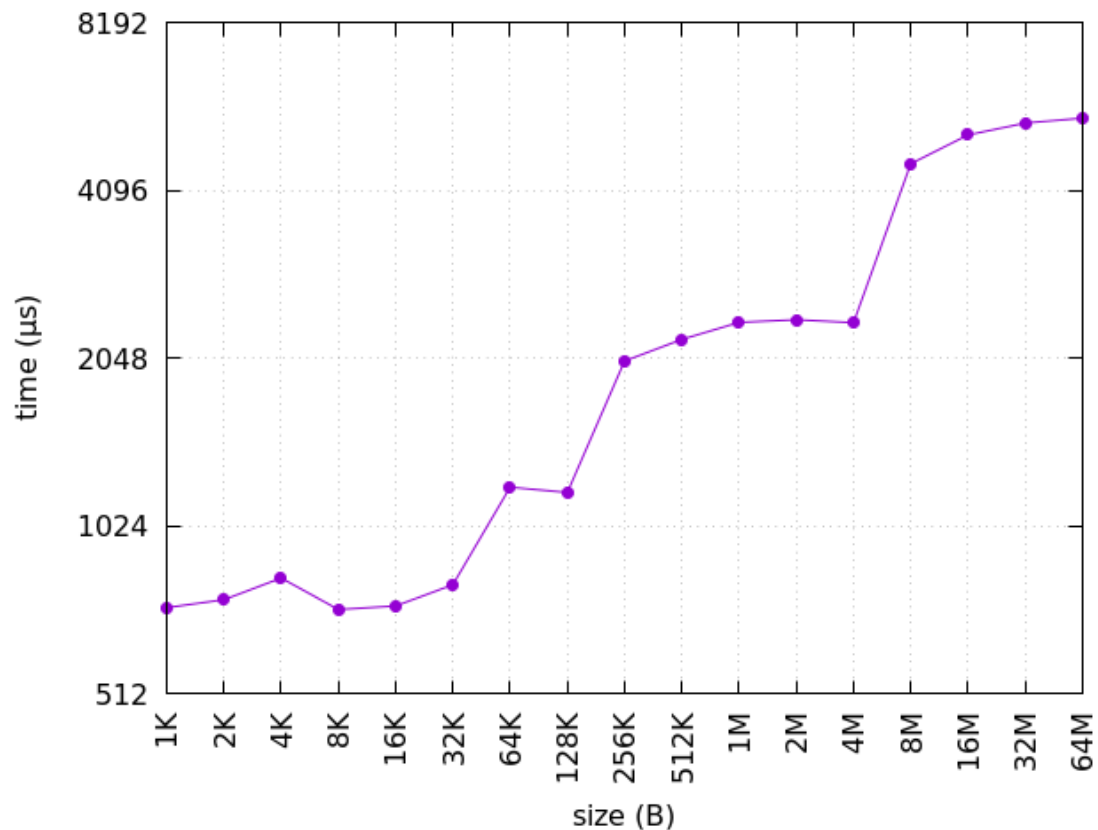
Y las gráficas y sus explicaciones:

La del line.cc:



En esta es fácilmente observable que tiene una tendencia descendente, y eso se puede explicar teniendo en cuenta que cuanto mayor es el tamaño de línea, más cantidad de información puede pasar al mismo tiempo, por lo que traspaso de información tarda menos tiempo.

En cuanto a la del size.cc:



Esta es algo más compleja ya que al tratarse de una máquina virtual que usa una memoria virtual dada, el traslado de información no es completamente normal, aunque si es observable más claramente en algunos puntos, sobre todo en el primero cuando pasamos de memoria L1 a L2 al llegar a los 32KB vemos claramente un salto que indica un uso de la memoria mas lento, aunque hay algunos picos extraños donde mejora el rendimiento normalmente tiende a reducirse, vemos otros dos saltos, uno en los 128KB cuando se llena la L2 y otro en torno a los 4MB ya que al llegar a los 6MB se llena la L3.