line.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 MAXLINE = 1024; // maximun line size to test
const unsigned GAP = 12;       // gap for cout columns
const unsigned REP = 100;      // number of repetitions of every test

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

        for (unsigned line = 1; line <= MAXLINE; line <<= 1) // line in bytes
        {
                std::vector<duration<double, std::micro>> score(REP);

                for (auto &s: score)
                {
                        std::vector<char> bytes(1 << 24); // 16MB
                        auto start = high_resolution_clock::now();

                        for (unsigned i = 0; i < bytes.size(); i += line)
                                bytes[i] ^= line;

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

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; // minimun line size to test:  1KB
const unsigned MAXSIZE = 1 << 26; // maximun 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()
{
        unsigned int num_iter_vector;
        std::cout << "#"
                << std::setw(GAP - 1) << "line (B)"
                << std::setw(GAP    ) << "time (µs)"
                << 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();
                if (STEPS*64/size > 1)
                        num_iter_vector = STEPS*64/size;
                else
                        num_iter_vector = 1;
                for (unsigned int j=0; j < num_iter_vector; j++)
                        for (unsigned int k = 0; k < size; k += 64)
                                bytes[k]++;

                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;
        }
}
```

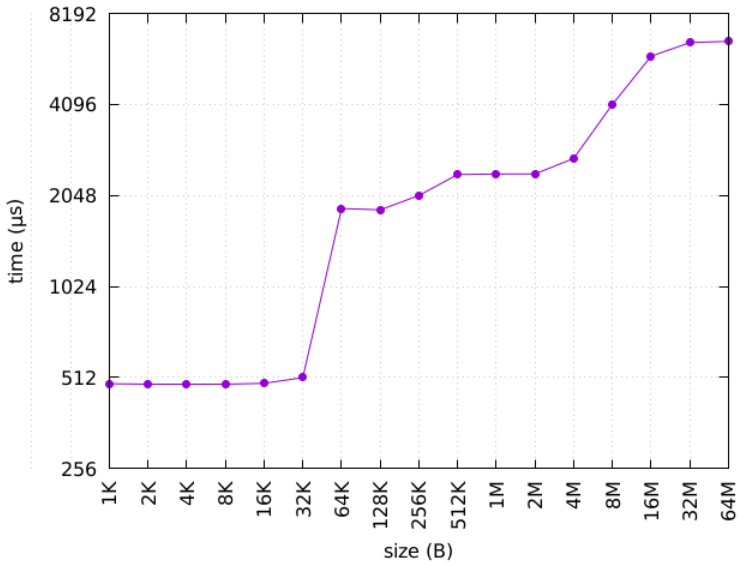
Salida de lspcu

```
Arquitectura:                        x86_64
modo(s) de operación de las CPUs:    32-bit, 64-bit
Orden de los bytes:                  Little Endian
Tamaños de las direcciones:          39 bits physical, 48 bits virtual
CPU(s):                              8
Lista de la(s) CPU(s) en línea:      0-7
Hilo(s) de procesamiento por núcleo: 2
Núcleo(s) por «socket»:              4
«Socket(s)»                          1
Modo(s) NUMA:                        1
ID de fabricante:                    GenuineIntel
Familia de CPU:                      6
Modelo:                              60
Nombre del modelo:                   Intel(R) Core(TM) i7-4720HQ CPU @ 2.60GHz
Revisión:                            3
CPU MHz:                             900.174
CPU MHz máx.:                        3600,0000
CPU MHz mín.:                        800,0000
BogoMIPS:                            5189.81
Virtualización:                      VT-x
Caché L1d:                           32K
Caché L1i:                           32K
Caché L2:                            256K
Caché L3:                            6144K
CPU(s) del nodo NUMA 0:              0-7
Indicadores:                         fpu vme de pse tsc msr pae mce cx8 apic sep mtrr
pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe
1gb rdtscp lm constant_tsc arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc c
puid aperfmperf pni pclmulqdq dtes64 monitor ds_cpl vmx est tm2 ssse3 sdbg fma cx16 xt
pr pdcm pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c r
drand lahf_lm abm cpuid_fault epb invpcid_single pti ssbd ibrs ibpb stibp tpr_shadow v
nmi flexpriority ept vpid ept_ad fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid
xsaveopt dtherm ida arat pln pts flush_l1d
```

Processor number i7-4720HQ

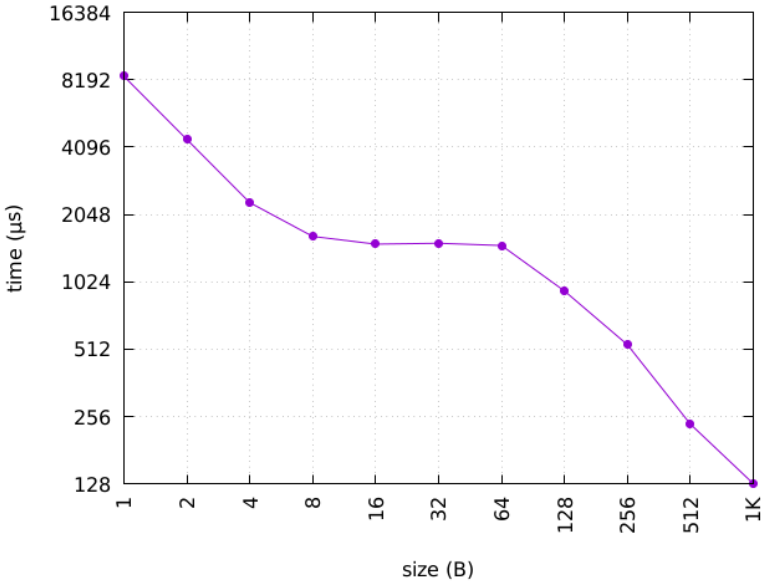
Processor number	i7-4720HQ
Family	Core i7 Mobile
Technology (micron)	0.022
Processor speed (GHz)	2.6
L2 cache size (KB)	1024
L3 cache size (MB)	6
The number of cores	4
EM64T	Supported
HyperThreading technology	Supported
Virtualization technology	Supported
Enhanced SpeedStep technology	Supported
Execute-Disable bit feature	Supported

[Processor numbers of mobile Intel CPUs]



En la gráfica correspondiente al programa Size.cc podemos observar los saltos en el tiempo cada vez que cambiamos a usar otra caché.

Los saltos se producen a los 32K, a los 256K y a los 6MB.



En la gŕafica correspondiente al programa Line.cc podemos observar la reducción en el tiempo de acceso a medida que aumenta el tamaño de bloque, al ser el porcentaje de cache miss menor.