Arquitectura de Computadores

GRADO EN INGENIERÍA INFORMÁTICA

Laboratorio 2: valgrind

Curso 2020/2021

Jorge Rodríguez Fraile, 100405951, Grupo 81, 100405951@alumnos.uc3m.es

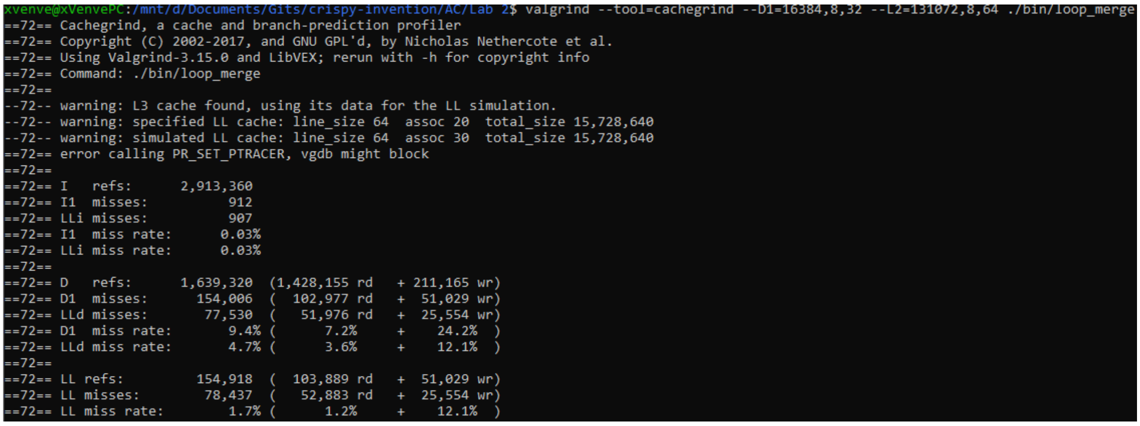
Índice

Tarea 1: Fusión de bucles

loop_merge.cpp y loop_merge-opt.cpp

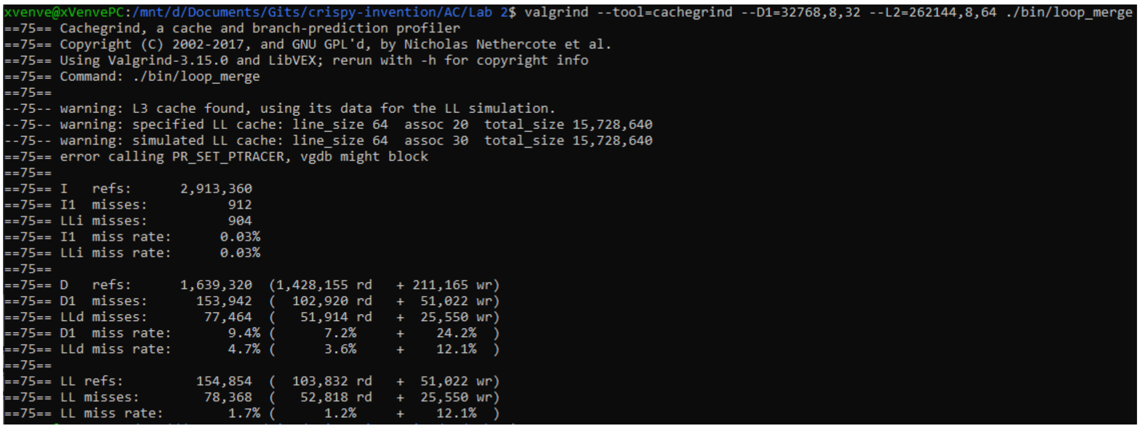
1. loop_merge

L1 de 16 KiB con tamaño de línea de 32 B, L2 de 128 KiB con tamaño de línea de 64B



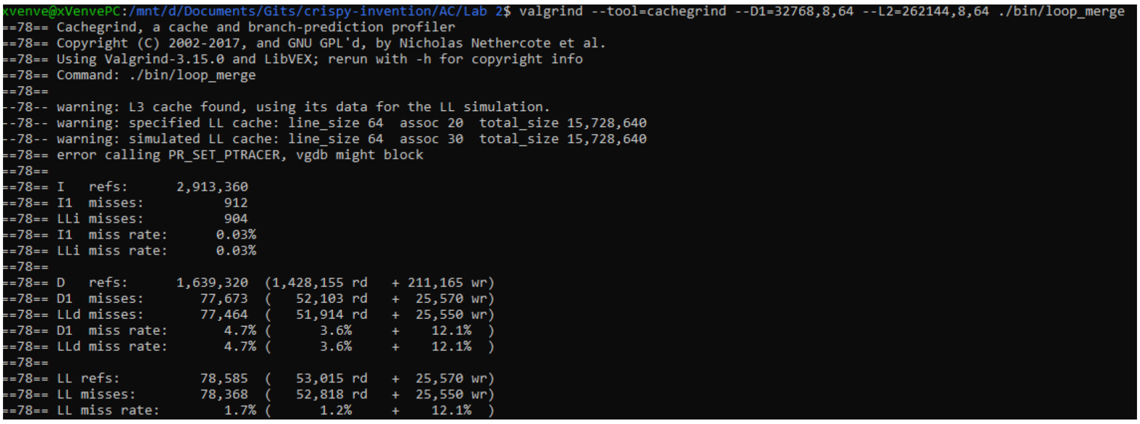
```
xvenve@xVenvePC:/mnt/d/Documents/Gits/crispy-invention/AC/Lab 2$ valgrind --tool=cachegrind --D1=16384,8,32 --L2=131072,8,64 ./bin/loop_merge
==72== Cachegrind, a cache and branch-prediction profiler
==72== Copyright (C) 2002-2017, and GNU GPL'd, by Nicholas Nethercote et al.
==72== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==72== Command: ./bin/loop_merge
==72==
--72-- warning: L3 cache found, using its data for the LL simulation.
--72-- warning: specified LL cache: line_size 64  assoc 20  total_size 15,728,640
--72-- warning: simulated LL cache: line_size 64  assoc 30  total_size 15,728,640
==72== error calling PR_SET_PTRACER, vgdb might block
==72==
==72== I   refs:      2,913,360
==72== I1  misses:          912
==72== LLi misses:          907
==72== I1  miss rate:      0.03%
==72== LLi miss rate:      0.03%
==72==
==72== D   refs:      1,639,320  (1,428,155 rd   + 211,165 wr)
==72== D1  misses:      154,006  (  102,977 rd   +  51,029 wr)
==72== LLd misses:       77,530  (   51,976 rd   +  25,554 wr)
==72== D1  miss rate:       9.4% (      7.2%     +    24.2%  )
==72== LLd miss rate:       4.7% (      3.6%     +    12.1%  )
==72==
==72== LL refs:         154,918  (  103,889 rd   +  51,029 wr)
==72== LL misses:        78,437  (   52,883 rd   +  25,554 wr)
==72== LL miss rate:        1.7% (      1.2%     +    12.1%  )
```

L1 de 32 KiB con tamaño de línea de 32 B, L2 de 256 KiB con tamaño de línea de 64B



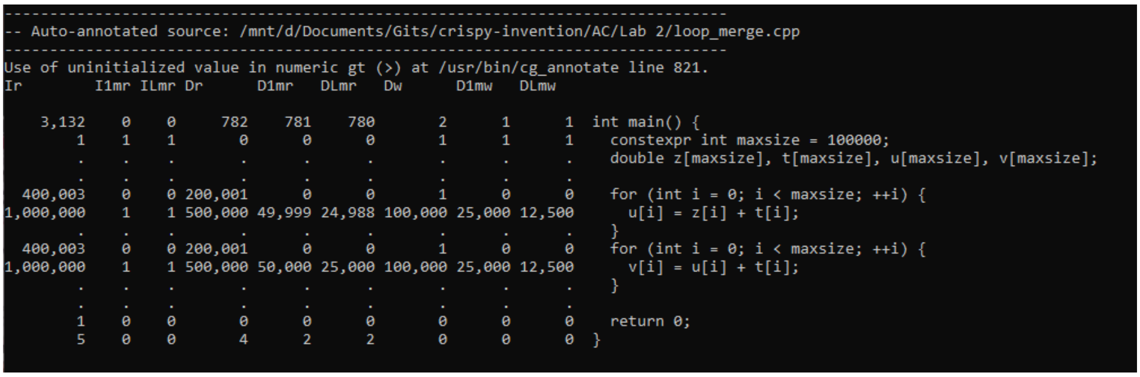
```
xvenve@xVenvePC:/mnt/d/Documents/Gits/crispy-invention/AC/Lab 2$ valgrind --tool=cachegrind --D1=32768,8,32 --L2=262144,8,64 ./bin/loop_merge
==75== Cachegrind, a cache and branch-prediction profiler
==75== Copyright (C) 2002-2017, and GNU GPL'd, by Nicholas Nethercote et al.
==75== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==75== Command: ./bin/loop_merge
==75==
--75-- warning: L3 cache found, using its data for the LL simulation.
--75-- warning: specified LL cache: line_size 64  assoc 20  total_size 15,728,640
--75-- warning: simulated LL cache: line_size 64  assoc 30  total_size 15,728,640
==75== error calling PR_SET_PTRACER, vgdb might block
==75==
==75== I   refs:      2,913,360
==75== I1  misses:          912
==75== LLi misses:          904
==75== I1  miss rate:      0.03%
==75== LLi miss rate:      0.03%
==75==
==75== D   refs:      1,639,320  (1,428,155 rd   + 211,165 wr)
==75== D1  misses:      153,942  (  102,920 rd   +  51,022 wr)
==75== LLd misses:       77,464  (   51,914 rd   +  25,550 wr)
==75== D1  miss rate:       9.4% (      7.2%     +    24.2%  )
==75== LLd miss rate:       4.7% (      3.6%     +    12.1%  )
==75==
==75== LL refs:         154,854  (  103,832 rd   +  51,022 wr)
==75== LL misses:        78,368  (   52,818 rd   +  25,550 wr)
==75== LL miss rate:        1.7% (      1.2%     +    12.1%  )
```

L1 de 32 KiB con tamaño de línea de 64 B, L2 de 256 KiB con tamaño de línea de 64B

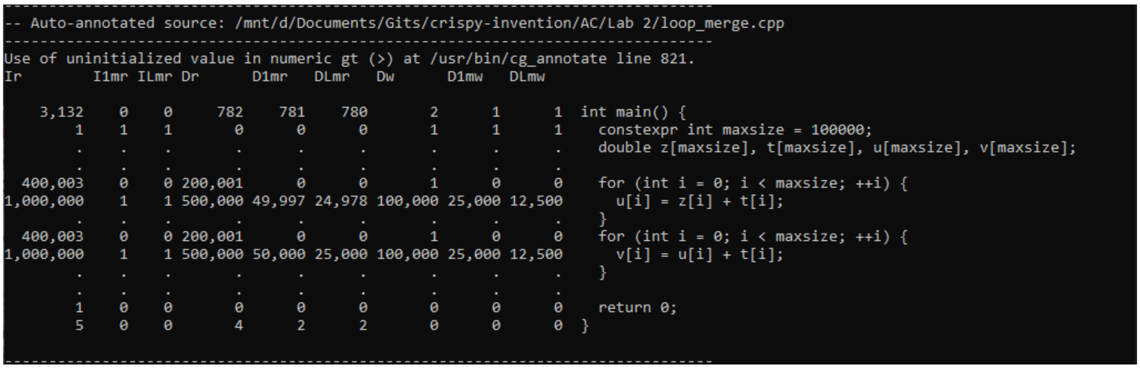


```
xvenve@xVenvePC:/mnt/d/Documents/Gits/crispy-invention/AC/Lab 2$ valgrind --tool=cachegrind --D1=32768,8,64 --L2=262144,8,64 ./bin/loop_merge
==78== Cachegrind, a cache and branch-prediction profiler
==78== Copyright (C) 2002-2017, and GNU GPL'd, by Nicholas Nethercote et al.
==78== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==78== Command: ./bin/loop_merge
==78==
--78-- warning: L3 cache found, using its data for the LL simulation.
--78-- warning: specified LL cache: line_size 64  assoc 20  total_size 15,728,640
--78-- warning: simulated LL cache: line_size 64  assoc 30  total_size 15,728,640
==78== error calling PR_SET_PTRACER, vgdb might block
==78==
==78== I   refs:      2,913,360
==78== I1  misses:          912
==78== LLi misses:          904
==78== I1  miss rate:      0.03%
==78== LLi miss rate:      0.03%
==78==
==78== D   refs:      1,639,320  (1,428,155 rd   + 211,165 wr)
==78== D1  misses:       77,673  (   52,103 rd   +  25,570 wr)
==78== LLd misses:       77,464  (   51,914 rd   +  25,550 wr)
==78== D1  miss rate:       4.7% (      3.6%     +    12.1%  )
==78== LLd miss rate:       4.7% (      3.6%     +    12.1%  )
==78==
==78== LL refs:          78,585  (   53,015 rd   +  25,570 wr)
==78== LL misses:        78,368  (   52,818 rd   +  25,550 wr)
==78== LL miss rate:        1.7% (      1.2%     +    12.1%  )
```

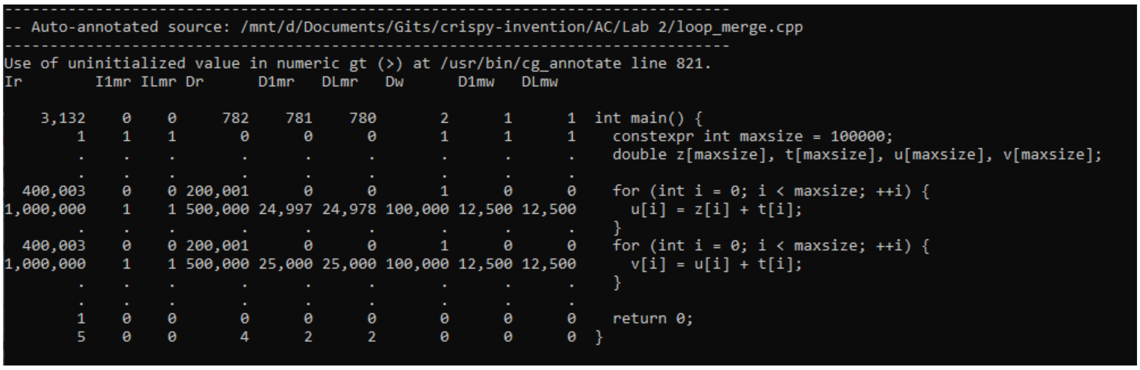
2. cg_annotate



```
--------------------------------------------------------------------------------
-- Auto-annotated source: /mnt/d/Documents/Gits/crispy-invention/AC/Lab 2/loop_merge.cpp
--------------------------------------------------------------------------------
Use of uninitialized value in numeric gt (>) at /usr/bin/cg_annotate line 821.
Ir        I1mr ILmr Dr      D1mr   DLmr   Dw      D1mw   DLmw

    3,132    0    0     782    781    780       2      1      1  int main() {
        1    1    1       0      0      0       1      1      1    constexpr int maxsize = 100000;
        .    .    .       .      .      .       .      .      .    double z[maxsize], t[maxsize], u[maxsize], v[maxsize];
        .    .    .       .      .      .       .      .      .
  400,003    0    0 200,001      0      0       1      0      0    for (int i = 0; i < maxsize; ++i) {
1,000,000    1    1 500,000 49,999 24,988 100,000 25,000 12,500      u[i] = z[i] + t[i];
        .    .    .       .      .      .       .      .      .    }
  400,003    0    0 200,001      0      0       1      0      0    for (int i = 0; i < maxsize; ++i) {
1,000,000    1    1 500,000 50,000 25,000 100,000 25,000 12,500      v[i] = u[i] + t[i];
        .    .    .       .      .      .       .      .      .    }
        .    .    .       .      .      .       .      .      .
        1    0    0       0      0      0       0      0      0    return 0;
        5    0    0       4      2      2       0      0      0  }
```



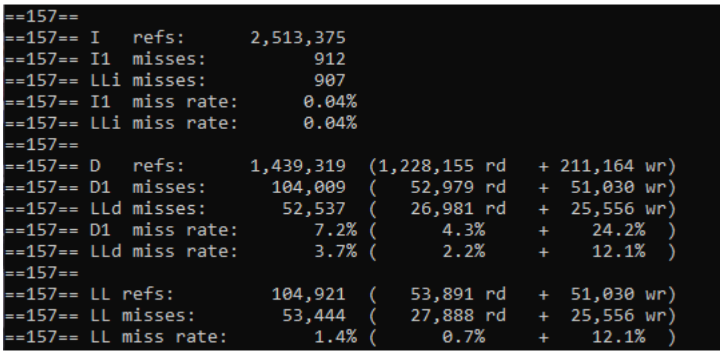
```
--------------------------------------------------------------------------------
-- Auto-annotated source: /mnt/d/Documents/Gits/crispy-invention/AC/Lab 2/loop_merge.cpp
--------------------------------------------------------------------------------
Use of uninitialized value in numeric gt (>) at /usr/bin/cg_annotate line 821.
Ir        I1mr ILmr Dr      D1mr   DLmr   Dw      D1mw   DLmw

    3,132    0    0     782    781    780       2      1      1  int main() {
        1    1    1       0      0      0       1      1      1    constexpr int maxsize = 100000;
        .    .    .       .      .      .       .      .      .    double z[maxsize], t[maxsize], u[maxsize], v[maxsize];
        .    .    .       .      .      .       .      .      .
  400,003    0    0 200,001      0      0       1      0      0    for (int i = 0; i < maxsize; ++i) {
1,000,000    1    1 500,000 49,997 24,978 100,000 25,000 12,500      u[i] = z[i] + t[i];
        .    .    .       .      .      .       .      .      .    }
  400,003    0    0 200,001      0      0       1      0      0    for (int i = 0; i < maxsize; ++i) {
1,000,000    1    1 500,000 50,000 25,000 100,000 25,000 12,500      v[i] = u[i] + t[i];
        .    .    .       .      .      .       .      .      .    }
        .    .    .       .      .      .       .      .      .
        1    0    0       0      0      0       0      0      0    return 0;
        5    0    0       4      2      2       0      0      0  }

--------------------------------------------------------------------------------
```



```
--------------------------------------------------------------------------------
-- Auto-annotated source: /mnt/d/Documents/Gits/crispy-invention/AC/Lab 2/loop_merge.cpp
--------------------------------------------------------------------------------
Use of uninitialized value in numeric gt (>) at /usr/bin/cg_annotate line 821.
Ir        I1mr ILmr Dr      D1mr   DLmr   Dw      D1mw   DLmw

    3,132    0    0     782    781    780       2      1      1  int main() {
        1    1    1       0      0      0       1      1      1    constexpr int maxsize = 100000;
        .    .    .       .      .      .       .      .      .    double z[maxsize], t[maxsize], u[maxsize], v[maxsize];
        .    .    .       .      .      .       .      .      .
  400,003    0    0 200,001      0      0       1      0      0    for (int i = 0; i < maxsize; ++i) {
1,000,000    1    1 500,000 24,997 24,978 100,000 12,500 12,500      u[i] = z[i] + t[i];
        .    .    .       .      .      .       .      .      .    }
  400,003    0    0 200,001      0      0       1      0      0    for (int i = 0; i < maxsize; ++i) {
1,000,000    1    1 500,000 25,000 25,000 100,000 12,500 12,500      v[i] = u[i] + t[i];
        .    .    .       .      .      .       .      .      .    }
        .    .    .       .      .      .       .      .      .
        1    0    0       0      0      0       0      0      0    return 0;
        5    0    0       4      2      2       0      0      0  }
```

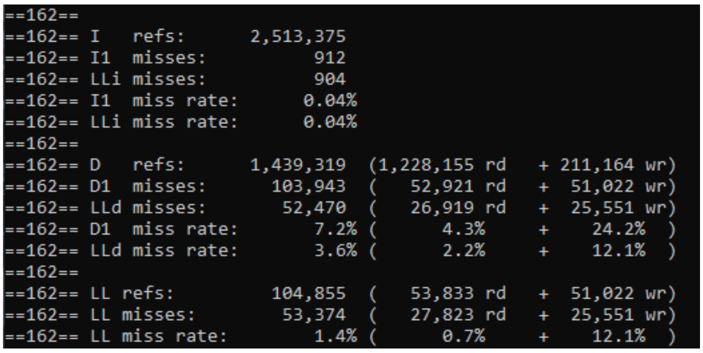
3. loop_merge-opt

L1 de 16 KiB con tamaño de línea de 32 B, L2 de 128 KiB con tamaño de línea de 64B



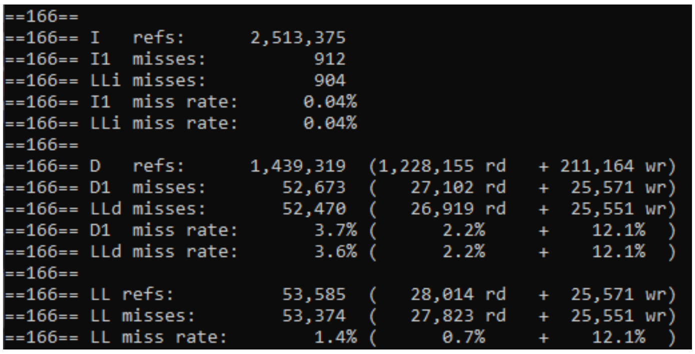
```
==157==
==157== I   refs:      2,513,375
==157== I1  misses:          912
==157== LLi misses:          907
==157== I1  miss rate:      0.04%
==157== LLi miss rate:      0.04%
==157==
==157== D   refs:      1,439,319  (1,228,155 rd   + 211,164 wr)
==157== D1  misses:      104,009  (   52,979 rd   +  51,030 wr)
==157== LLd misses:       52,537  (   26,981 rd   +  25,556 wr)
==157== D1  miss rate:       7.2% (      4.3%     +    24.2%  )
==157== LLd miss rate:       3.7% (      2.2%     +    12.1%  )
==157==
==157== LL refs:         104,921  (   53,891 rd   +  51,030 wr)
==157== LL misses:        53,444  (   27,888 rd   +  25,556 wr)
==157== LL miss rate:        1.4% (      0.7%     +    12.1%  )
```

L1 de 32 KiB con tamaño de línea de 32 B, L2 de 256 KiB con tamaño de línea de 64B



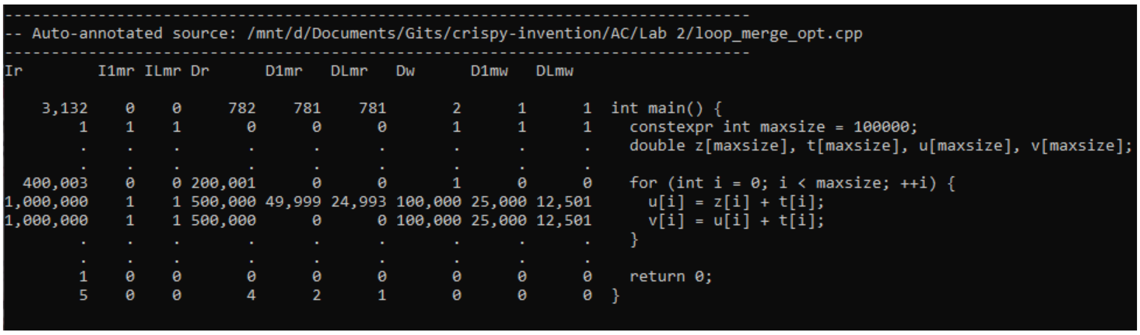
```
==162==
==162== I   refs:      2,513,375
==162== I1  misses:          912
==162== LLi misses:          904
==162== I1  miss rate:      0.04%
==162== LLi miss rate:      0.04%
==162==
==162== D   refs:      1,439,319  (1,228,155 rd   + 211,164 wr)
==162== D1  misses:      103,943  (   52,921 rd   +  51,022 wr)
==162== LLd misses:       52,470  (   26,919 rd   +  25,551 wr)
==162== D1  miss rate:       7.2% (      4.3%     +    24.2%  )
==162== LLd miss rate:       3.6% (      2.2%     +    12.1%  )
==162==
==162== LL refs:         104,855  (   53,833 rd   +  51,022 wr)
==162== LL misses:        53,374  (   27,823 rd   +  25,551 wr)
==162== LL miss rate:        1.4% (      0.7%     +    12.1%  )
```

L1 de 32 KiB con tamaño de línea de 64 B, L2 de 256 KiB con tamaño de línea de 64B

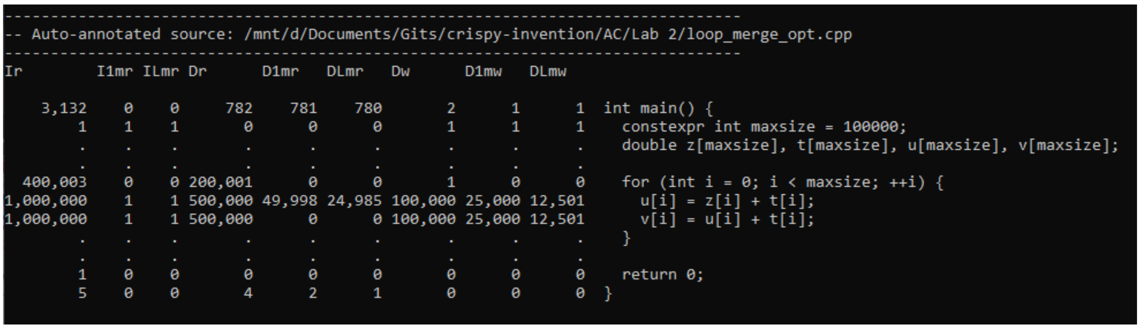


```
==166==
==166== I   refs:      2,513,375
==166== I1  misses:          912
==166== LLi misses:          904
==166== I1  miss rate:      0.04%
==166== LLi miss rate:      0.04%
==166==
==166== D   refs:      1,439,319  (1,228,155 rd   + 211,164 wr)
==166== D1  misses:       52,673  (   27,102 rd   +  25,571 wr)
==166== LLd misses:       52,470  (   26,919 rd   +  25,551 wr)
==166== D1  miss rate:       3.7% (      2.2%     +    12.1%  )
==166== LLd miss rate:       3.6% (      2.2%     +    12.1%  )
==166==
==166== LL refs:          53,585  (   28,014 rd   +  25,571 wr)
==166== LL misses:        53,374  (   27,823 rd   +  25,551 wr)
==166== LL miss rate:        1.4% (      0.7%     +    12.1%  )
```

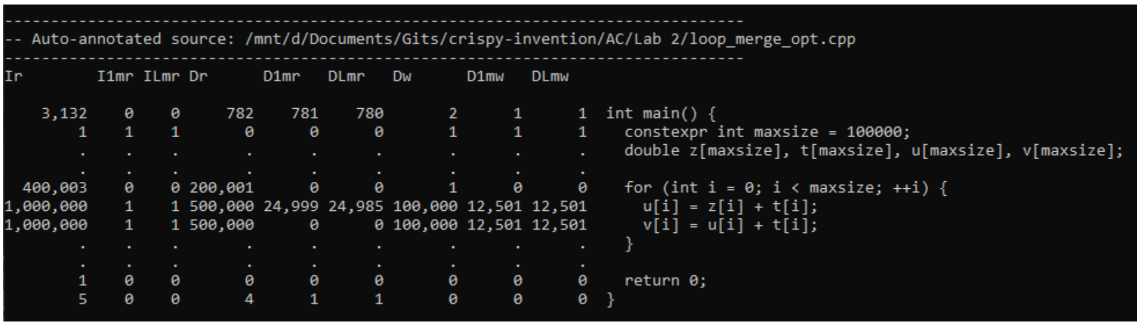
4. cg_annotate



```
--------------------------------------------------------------------------------
-- Auto-annotated source: /mnt/d/Documents/Gits/crispy-invention/AC/Lab 2/loop_merge_opt.cpp
--------------------------------------------------------------------------------
Ir        I1mr ILmr Dr      D1mr   DLmr   Dw      D1mw   DLmw

    3,132    0    0     782    781    781       2      1      1  int main() {
        1    1    1       0      0      0       1      1      1    constexpr int maxsize = 100000;
        .    .    .       .      .      .       .      .      .    double z[maxsize], t[maxsize], u[maxsize], v[maxsize];
        .    .    .       .      .      .       .      .      .
  400,003    0    0 200,001      0      0       1      0      0    for (int i = 0; i < maxsize; ++i) {
1,000,000    1    1 500,000 49,999 24,993 100,000 25,000 12,501      u[i] = z[i] + t[i];
1,000,000    1    1 500,000      0      0 100,000 25,000 12,501      v[i] = u[i] + t[i];
        .    .    .       .      .      .       .      .      .    }
        .    .    .       .      .      .       .      .      .
        1    0    0       0      0      0       0      0      0    return 0;
        5    0    0       4      2      1       0      0      0  }
```



```
--------------------------------------------------------------------------------
-- Auto-annotated source: /mnt/d/Documents/Gits/crispy-invention/AC/Lab 2/loop_merge_opt.cpp
--------------------------------------------------------------------------------
Ir        I1mr ILmr Dr      D1mr   DLmr   Dw      D1mw   DLmw

    3,132    0    0     782    781    780       2      1      1  int main() {
        1    1    1       0      0      0       1      1      1    constexpr int maxsize = 100000;
        .    .    .       .      .      .       .      .      .    double z[maxsize], t[maxsize], u[maxsize], v[maxsize];
        .    .    .       .      .      .       .      .      .
  400,003    0    0 200,001      0      0       1      0      0    for (int i = 0; i < maxsize; ++i) {
1,000,000    1    1 500,000 49,998 24,985 100,000 25,000 12,501      u[i] = z[i] + t[i];
1,000,000    1    1 500,000      0      0 100,000 25,000 12,501      v[i] = u[i] + t[i];
        .    .    .       .      .      .       .      .      .    }
        .    .    .       .      .      .       .      .      .
        1    0    0       0      0      0       0      0      0    return 0;
        5    0    0       4      2      1       0      0      0  }
```



```
--------------------------------------------------------------------------------
-- Auto-annotated source: /mnt/d/Documents/Gits/crispy-invention/AC/Lab 2/loop_merge_opt.cpp
--------------------------------------------------------------------------------
Ir        I1mr ILmr Dr      D1mr   DLmr   Dw      D1mw   DLmw

    3,132    0    0     782    781    780       2      1      1  int main() {
        1    1    1       0      0      0       1      1      1    constexpr int maxsize = 100000;
        .    .    .       .      .      .       .      .      .    double z[maxsize], t[maxsize], u[maxsize], v[maxsize];
        .    .    .       .      .      .       .      .      .
  400,003    0    0 200,001      0      0       1      0      0    for (int i = 0; i < maxsize; ++i) {
1,000,000    1    1 500,000 24,999 24,985 100,000 12,501 12,501      u[i] = z[i] + t[i];
1,000,000    1    1 500,000      0      0 100,000 12,501 12,501      v[i] = u[i] + t[i];
        .    .    .       .      .      .       .      .      .    }
        .    .    .       .      .      .       .      .      .
        1    0    0       0      0      0       0      0      0    return 0;
        5    0    0       4      1      1       0      0      0  }
```

5. Compare ambos resultados.

Diferencia en el ratio de fallos en L1: La principal diferencia se aprecia cuando doblamos el tamaño de línea, pero no afecta el aumento del tamaño de cache ya que no hay remplazado por falta de espacio.

9.4	7.2	2.2
9.4	7.2	2.2
4.7	3.7	1

Diferencia en el ratio de fallos en L2:

4.7	3.7	1
4.7	3.6	1
4.7	3.6	1.1

Diferencia en el ratio de fallos local en L2:

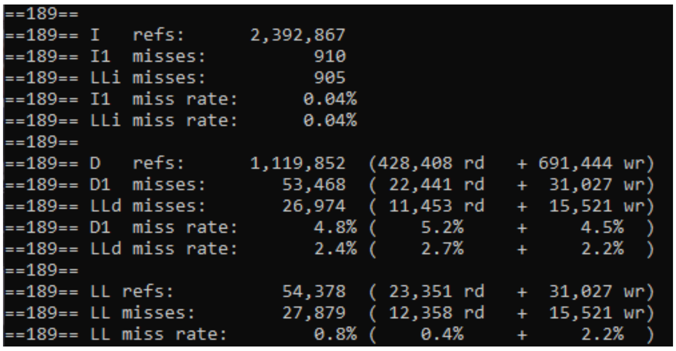
1.7	1.4	0.3
1.7	1.4	0.3
1.7	1.4	0.3

Tarea 2: Acceso secuencial

access_seq.cpp y access_strided.cpp

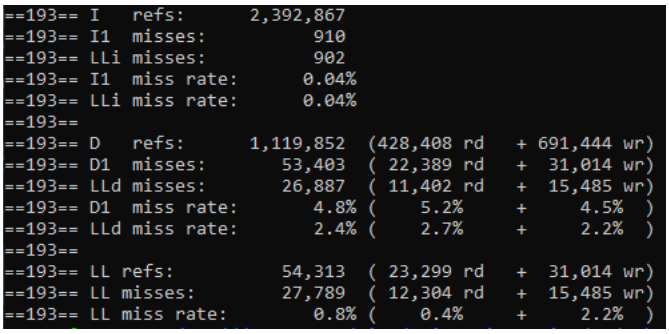
1. access_seq

L1 de 16 KiB con tamaño de línea de 32 B, L2 de 128 KiB con tamaño de línea de 64B



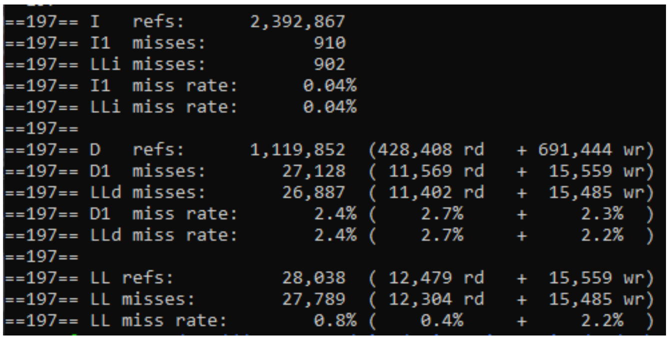
```
==189==
==189== I   refs:      2,392,867
==189== I1  misses:          910
==189== LLi misses:          905
==189== I1  miss rate:      0.04%
==189== LLi miss rate:      0.04%
==189==
==189== D   refs:      1,119,852  (428,408 rd   + 691,444 wr)
==189== D1  misses:       53,468  ( 22,441 rd   +  31,027 wr)
==189== LLd misses:       26,974  ( 11,453 rd   +  15,521 wr)
==189== D1  miss rate:       4.8% (    5.2%     +     4.5%  )
==189== LLd miss rate:       2.4% (    2.7%     +     2.2%  )
==189==
==189== LL refs:          54,378  ( 23,351 rd   +  31,027 wr)
==189== LL misses:        27,879  ( 12,358 rd   +  15,521 wr)
==189== LL miss rate:        0.8% (    0.4%     +     2.2%  )
```

L1 de 32 KiB con tamaño de línea de 32 B, L2 de 256 KiB con tamaño de línea de 64B



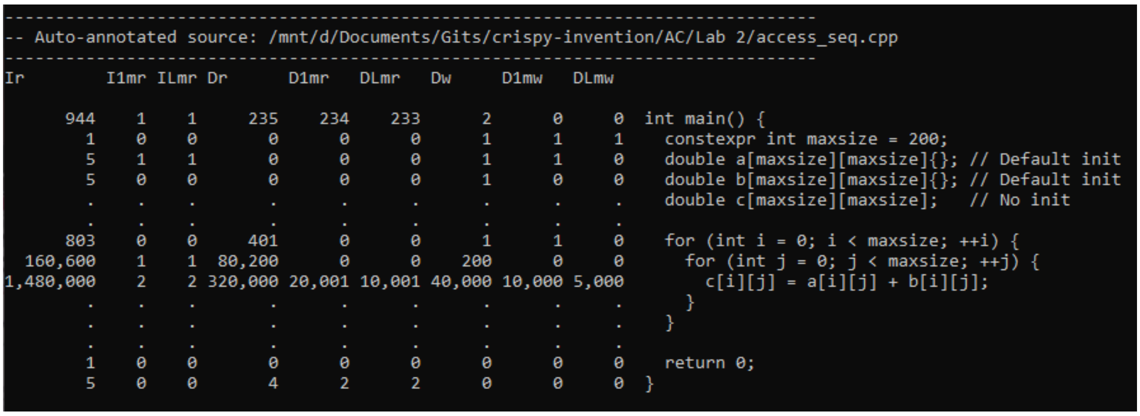
```
==193== I   refs:      2,392,867
==193== I1  misses:          910
==193== LLi misses:          902
==193== I1  miss rate:      0.04%
==193== LLi miss rate:      0.04%
==193==
==193== D   refs:      1,119,852  (428,408 rd   + 691,444 wr)
==193== D1  misses:       53,403  ( 22,389 rd   +  31,014 wr)
==193== LLd misses:       26,887  ( 11,402 rd   +  15,485 wr)
==193== D1  miss rate:       4.8% (    5.2%     +     4.5%  )
==193== LLd miss rate:       2.4% (    2.7%     +     2.2%  )
==193==
==193== LL refs:          54,313  ( 23,299 rd   +  31,014 wr)
==193== LL misses:        27,789  ( 12,304 rd   +  15,485 wr)
==193== LL miss rate:        0.8% (    0.4%     +     2.2%  )
```

L1 de 32 KiB con tamaño de línea de 64 B, L2 de 256 KiB con tamaño de línea de 64B

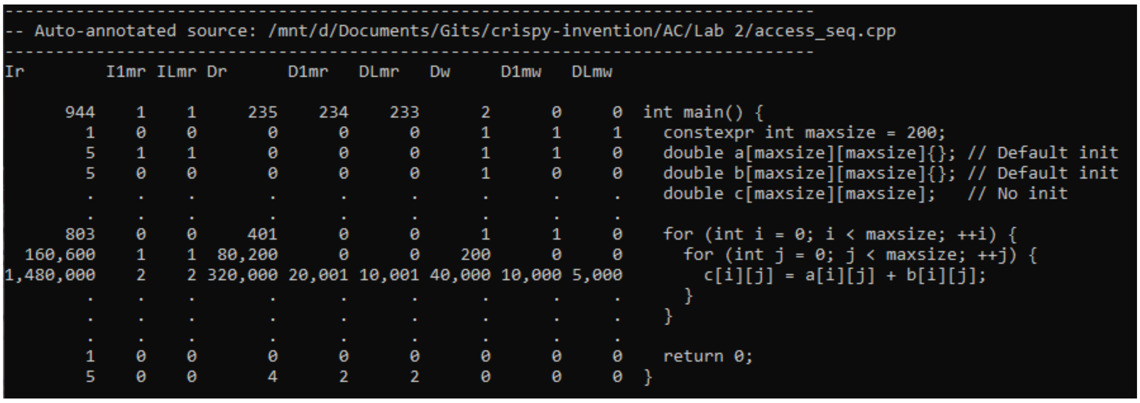


```
==197== I   refs:      2,392,867
==197== I1  misses:          910
==197== LLi misses:          902
==197== I1  miss rate:      0.04%
==197== LLi miss rate:      0.04%
==197==
==197== D   refs:      1,119,852  (428,408 rd   + 691,444 wr)
==197== D1  misses:       27,128  ( 11,569 rd   +  15,559 wr)
==197== LLd misses:       26,887  ( 11,402 rd   +  15,485 wr)
==197== D1  miss rate:       2.4% (    2.7%     +     2.3%  )
==197== LLd miss rate:       2.4% (    2.7%     +     2.2%  )
==197==
==197== LL refs:          28,038  ( 12,479 rd   +  15,559 wr)
==197== LL misses:        27,789  ( 12,304 rd   +  15,485 wr)
==197== LL miss rate:        0.8% (    0.4%     +     2.2%  )
```

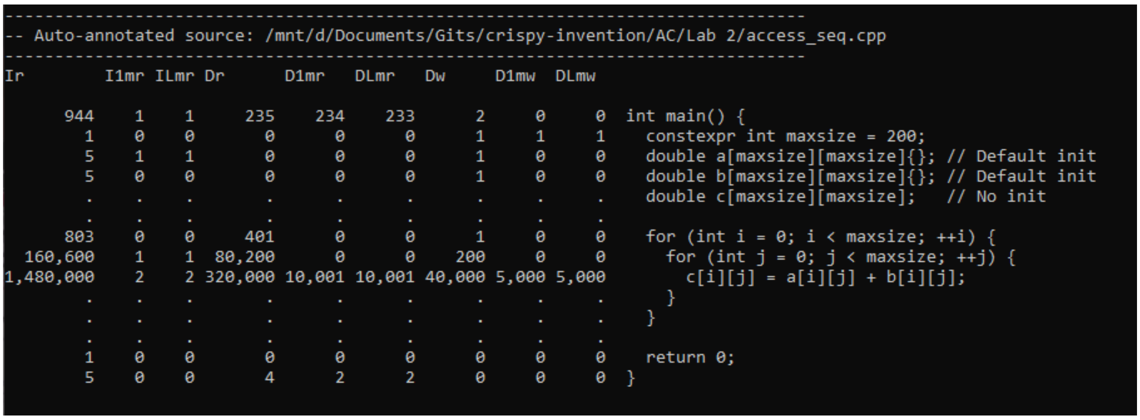
2. cg_annotate



```
--------------------------------------------------------------------------------
-- Auto-annotated source: /mnt/d/Documents/Gits/crispy-invention/AC/Lab 2/access_seq.cpp
--------------------------------------------------------------------------------
Ir        I1mr ILmr Dr      D1mr   DLmr   Dw     D1mw   DLmw

      944    1    1     235    234    233      2      0     0 int main() {
        1    0    0       0      0      0      1      1     1   constexpr int maxsize = 200;
        5    1    1       0      0      0      1      1     0   double a[maxsize][maxsize]{}; // Default init
        5    0    0       0      0      0      1      0     0   double b[maxsize][maxsize]{}; // Default init
        .    .    .       .      .      .      .      .     .   double c[maxsize][maxsize];   // No init
        .    .    .       .      .      .      .      .     .
      803    0    0     401      0      0      1      1     0   for (int i = 0; i < maxsize; ++i) {
  160,600    1    1  80,200      0      0    200      0     0     for (int j = 0; j < maxsize; ++j) {
1,480,000    2    2 320,000 20,001 10,001 40,000 10,000 5,000       c[i][j] = a[i][j] + b[i][j];
        .    .    .       .      .      .      .      .     .     }
        .    .    .       .      .      .      .      .     .   }
        .    .    .       .      .      .      .      .     .
        1    0    0       0      0      0      0      0     0   return 0;
        5    0    0       4      2      2      0      0     0 }
```



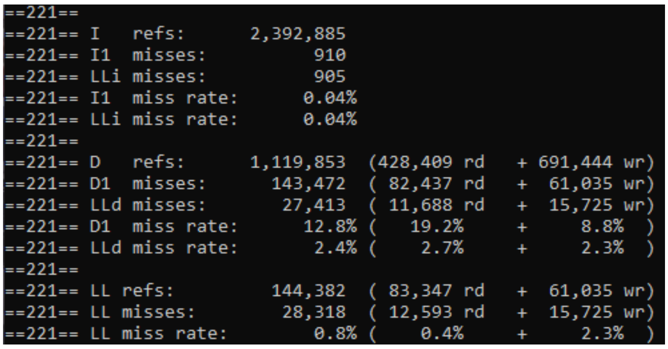
```
--------------------------------------------------------------------------------
-- Auto-annotated source: /mnt/d/Documents/Gits/crispy-invention/AC/Lab 2/access_seq.cpp
--------------------------------------------------------------------------------
Ir        I1mr ILmr Dr      D1mr   DLmr   Dw     D1mw   DLmw

      944    1    1     235    234    233      2      0     0  int main() {
        1    0    0       0      0      0      1      1     1    constexpr int maxsize = 200;
        5    1    1       0      0      0      1      1     0    double a[maxsize][maxsize]{}; // Default init
        5    0    0       0      0      0      1      0     0    double b[maxsize][maxsize]{}; // Default init
        .    .    .       .      .      .      .      .     .    double c[maxsize][maxsize];   // No init
        .    .    .       .      .      .      .      .     .
      803    0    0     401      0      0      1      1     0    for (int i = 0; i < maxsize; ++i) {
  160,600    1    1  80,200      0      0    200      0     0      for (int j = 0; j < maxsize; ++j) {
1,480,000    2    2 320,000 20,001 10,001 40,000 10,000 5,000        c[i][j] = a[i][j] + b[i][j];
        .    .    .       .      .      .      .      .     .      }
        .    .    .       .      .      .      .      .     .    }
        .    .    .       .      .      .      .      .     .
        1    0    0       0      0      0      0      0     0    return 0;
        5    0    0       4      2      2      0      0     0  }
```



```
--------------------------------------------------------------------------------
-- Auto-annotated source: /mnt/d/Documents/Gits/crispy-invention/AC/Lab 2/access_seq.cpp
--------------------------------------------------------------------------------
Ir        I1mr ILmr Dr      D1mr   DLmr   Dw     D1mw  DLmw

      944    1    1     235    234    233      2     0     0  int main() {
        1    0    0       0      0      0      1     1     1    constexpr int maxsize = 200;
        5    1    1       0      0      0      1     0     0    double a[maxsize][maxsize]{}; // Default init
        5    0    0       0      0      0      1     0     0    double b[maxsize][maxsize]{}; // Default init
        .    .    .       .      .      .      .     .     .    double c[maxsize][maxsize];   // No init
        .    .    .       .      .      .      .     .     .
      803    0    0     401      0      0      1     0     0    for (int i = 0; i < maxsize; ++i) {
  160,600    1    1  80,200      0      0    200     0     0      for (int j = 0; j < maxsize; ++j) {
1,480,000    2    2 320,000 10,001 10,001 40,000 5,000 5,000        c[i][j] = a[i][j] + b[i][j];
        .    .    .       .      .      .      .     .     .      }
        .    .    .       .      .      .      .     .     .    }
        .    .    .       .      .      .      .     .     .
        1    0    0       0      0      0      0     0     0    return 0;
        5    0    0       4      2      2      0     0     0  }
```

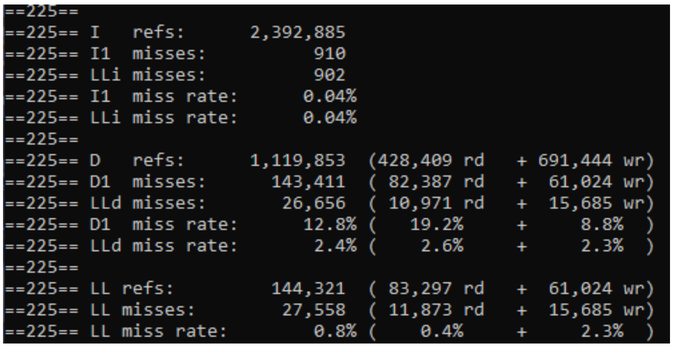
3. access_strided

L1 de 16 KiB con tamaño de línea de 32 B, L2 de 128 KiB con tamaño de línea de 64B



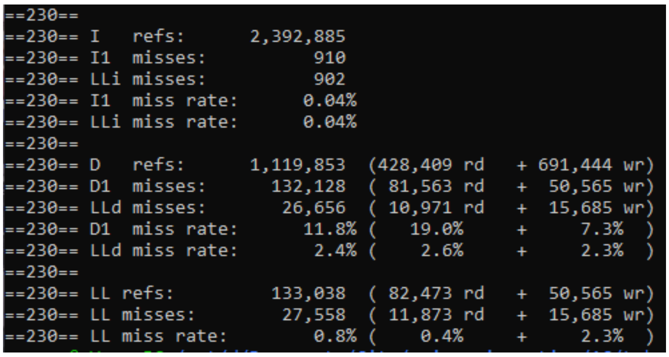
```
==221==
==221== I   refs:      2,392,885
==221== I1  misses:          910
==221== LLi misses:          905
==221== I1  miss rate:      0.04%
==221== LLi miss rate:      0.04%
==221==
==221== D   refs:      1,119,853  (428,409 rd   + 691,444 wr)
==221== D1  misses:      143,472  ( 82,437 rd   +  61,035 wr)
==221== LLd misses:       27,413  ( 11,688 rd   +  15,725 wr)
==221== D1  miss rate:      12.8% (   19.2%     +     8.8%  )
==221== LLd miss rate:       2.4% (    2.7%     +     2.3%  )
==221==
==221== LL refs:         144,382  ( 83,347 rd   +  61,035 wr)
==221== LL misses:        28,318  ( 12,593 rd   +  15,725 wr)
==221== LL miss rate:        0.8% (    0.4%     +     2.3%  )
```

L1 de 32 KiB con tamaño de línea de 32 B, L2 de 256 KiB con tamaño de línea de 64B



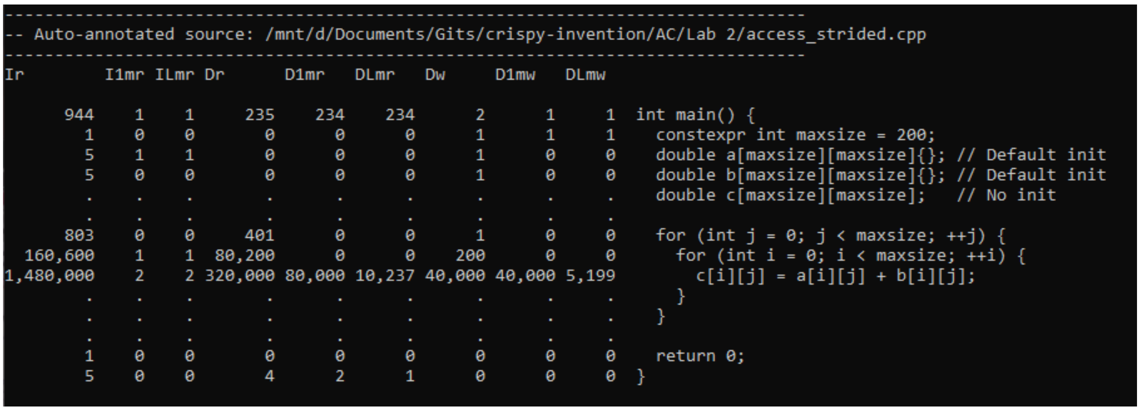
```
==225==
==225== I   refs:      2,392,885
==225== I1  misses:          910
==225== LLi misses:          902
==225== I1  miss rate:      0.04%
==225== LLi miss rate:      0.04%
==225==
==225== D   refs:      1,119,853  (428,409 rd   + 691,444 wr)
==225== D1  misses:      143,411  ( 82,387 rd   +  61,024 wr)
==225== LLd misses:       26,656  ( 10,971 rd   +  15,685 wr)
==225== D1  miss rate:      12.8% (   19.2%     +     8.8%  )
==225== LLd miss rate:       2.4% (    2.6%     +     2.3%  )
==225==
==225== LL refs:         144,321  ( 83,297 rd   +  61,024 wr)
==225== LL misses:        27,558  ( 11,873 rd   +  15,685 wr)
==225== LL miss rate:        0.8% (    0.4%     +     2.3%  )
```

L1 de 32 KiB con tamaño de línea de 64 B, L2 de 256 KiB con tamaño de línea de 64B

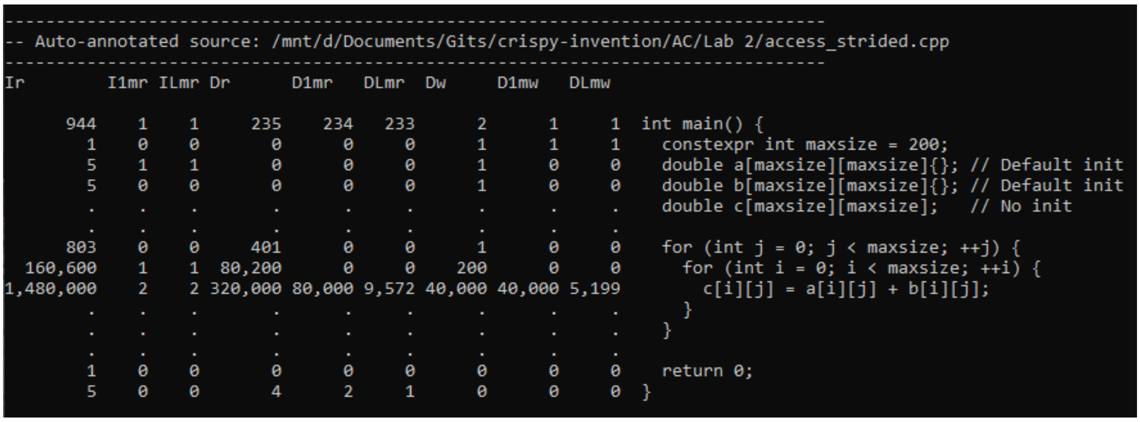


```
==230==
==230== I   refs:      2,392,885
==230== I1  misses:          910
==230== LLi misses:          902
==230== I1  miss rate:      0.04%
==230== LLi miss rate:      0.04%
==230==
==230== D   refs:      1,119,853  (428,409 rd   + 691,444 wr)
==230== D1  misses:      132,128  ( 81,563 rd   +  50,565 wr)
==230== LLd misses:       26,656  ( 10,971 rd   +  15,685 wr)
==230== D1  miss rate:      11.8% (   19.0%     +     7.3%  )
==230== LLd miss rate:       2.4% (    2.6%     +     2.3%  )
==230==
==230== LL refs:         133,038  ( 82,473 rd   +  50,565 wr)
==230== LL misses:        27,558  ( 11,873 rd   +  15,685 wr)
==230== LL miss rate:        0.8% (    0.4%     +     2.3%  )
```

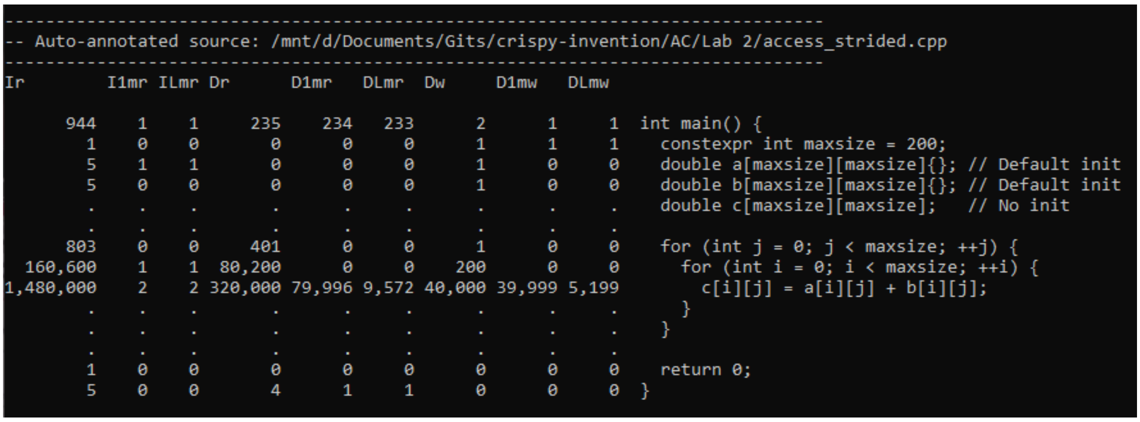
4. cg_annotate



```
--------------------------------------------------------------------------------
-- Auto-annotated source: /mnt/d/Documents/Gits/crispy-invention/AC/Lab 2/access_strided.cpp
--------------------------------------------------------------------------------
Ir        I1mr ILmr Dr      D1mr   DLmr   Dw     D1mw   DLmw

      944    1    1     235    234    234      2      1    1  int main() {
        1    0    0       0      0      0      1      1    1    constexpr int maxsize = 200;
        5    1    1       0      0      0      1      0    0    double a[maxsize][maxsize]{}; // Default init
        5    0    0       0      0      0      1      0    0    double b[maxsize][maxsize]{}; // Default init
        .    .    .       .      .      .      .      .    .    double c[maxsize][maxsize];   // No init
        .    .    .       .      .      .      .      .    .
      803    0    0     401      0      0      1      0    0    for (int j = 0; j < maxsize; ++j) {
  160,600    1    1  80,200      0      0    200      0    0      for (int i = 0; i < maxsize; ++i) {
1,480,000    2    2 320,000 80,000 10,237 40,000 40,000 5,199        c[i][j] = a[i][j] + b[i][j];
        .    .    .       .      .      .      .      .    .      }
        .    .    .       .      .      .      .      .    .    }
        .    .    .       .      .      .      .      .    .
        1    0    0       0      0      0      0      0    0    return 0;
        5    0    0       4      2      1      0      0    0  }
```



```
--------------------------------------------------------------------------------
-- Auto-annotated source: /mnt/d/Documents/Gits/crispy-invention/AC/Lab 2/access_strided.cpp
--------------------------------------------------------------------------------
Ir        I1mr ILmr Dr      D1mr   DLmr  Dw     D1mw   DLmw

      944    1    1     235    234   233      2      1    1  int main() {
        1    0    0       0      0     0      1      1    1    constexpr int maxsize = 200;
        5    1    1       0      0     0      1      0    0    double a[maxsize][maxsize]{}; // Default init
        5    0    0       0      0     0      1      0    0    double b[maxsize][maxsize]{}; // Default init
        .    .    .       .      .     .      .      .    .    double c[maxsize][maxsize];   // No init
        .    .    .       .      .     .      .      .    .
      803    0    0     401      0     0      1      0    0    for (int j = 0; j < maxsize; ++j) {
  160,600    1    1  80,200      0     0    200      0    0      for (int i = 0; i < maxsize; ++i) {
1,480,000    2    2 320,000 80,000 9,572 40,000 40,000 5,199        c[i][j] = a[i][j] + b[i][j];
        .    .    .       .      .     .      .      .    .      }
        .    .    .       .      .     .      .      .    .    }
        .    .    .       .      .     .      .      .    .
        1    0    0       0      0     0      0      0    0    return 0;
        5    0    0       4      2     1      0      0    0  }
```



```
--------------------------------------------------------------------------------
-- Auto-annotated source: /mnt/d/Documents/Gits/crispy-invention/AC/Lab 2/access_strided.cpp
--------------------------------------------------------------------------------
Ir        I1mr ILmr Dr      D1mr   DLmr  Dw     D1mw   DLmw

      944    1    1     235    234   233      2      1    1  int main() {
        1    0    0       0      0     0      1      1    1    constexpr int maxsize = 200;
        5    1    1       0      0     0      1      0    0    double a[maxsize][maxsize]{}; // Default init
        5    0    0       0      0     0      1      0    0    double b[maxsize][maxsize]{}; // Default init
        .    .    .       .      .     .      .      .    .    double c[maxsize][maxsize];   // No init
        .    .    .       .      .     .      .      .    .
      803    0    0     401      0     0      1      0    0    for (int j = 0; j < maxsize; ++j) {
  160,600    1    1  80,200      0     0    200      0    0      for (int i = 0; i < maxsize; ++i) {
1,480,000    2    2 320,000 79,996 9,572 40,000 39,999 5,199        c[i][j] = a[i][j] + b[i][j];
        .    .    .       .      .     .      .      .    .      }
        .    .    .       .      .     .      .      .    .    }
        .    .    .       .      .     .      .      .    .
        1    0    0       0      0     0      0      0    0    return 0;
        5    0    0       4      1     1      0      0    0  }
```

5. Compare ambos resultados.

Diferencia en el ratio de fallos en L1: Acceder por fila en vez de por columna no aprovecha la localidad espacial.

El inicial aprovecha el tamaño de línea aumentado, pero el optimizado no aprovecha ese tamaño de línea.

4.8	12.8	-8
4.8	12.8	-8
2.4	11.8	-9.4

Diferencia en el ratio de fallos en L2:

2.4	2.4	0
2.4	2.4	0
2.4	2.4	0

Diferencia en el ratio de fallos local en L2:

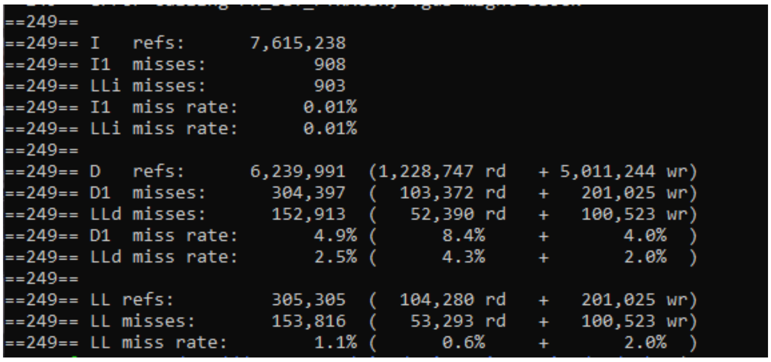
0.8	0.8	0
0.8	0.8	0
0.8	0.8	0

Tarea 3: Estructuras y arrays

soa.cpp y aos.cpp

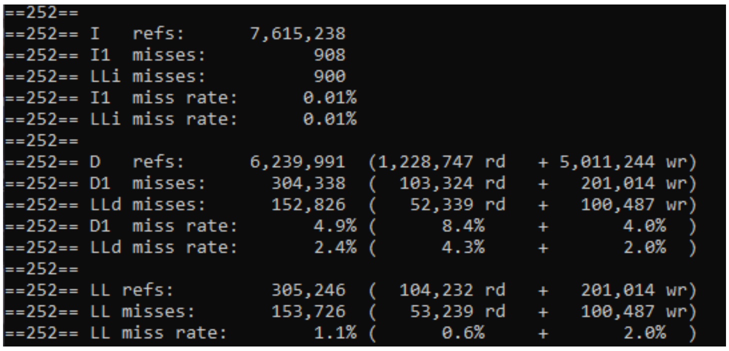
1. soa

L1 de 16 KiB con tamaño de línea de 32 B, L2 de 128 KiB con tamaño de línea de 64B



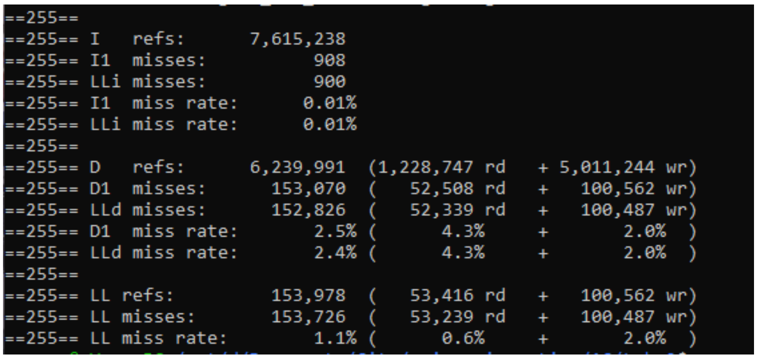
```
==249==
==249== I   refs:      7,615,238
==249== I1  misses:          908
==249== LLi misses:          903
==249== I1  miss rate:      0.01%
==249== LLi miss rate:      0.01%
==249==
==249== D   refs:      6,239,991  (1,228,747 rd   + 5,011,244 wr)
==249== D1  misses:      304,397  (  103,372 rd   +   201,025 wr)
==249== LLd misses:      152,913  (   52,390 rd   +   100,523 wr)
==249== D1  miss rate:       4.9% (      8.4%     +       4.0%  )
==249== LLd miss rate:       2.5% (      4.3%     +       2.0%  )
==249==
==249== LL refs:         305,305  (  104,280 rd   +   201,025 wr)
==249== LL misses:       153,816  (   53,293 rd   +   100,523 wr)
==249== LL miss rate:        1.1% (      0.6%     +       2.0%  )
```

L1 de 32 KiB con tamaño de línea de 32 B, L2 de 256 KiB con tamaño de línea de 64B



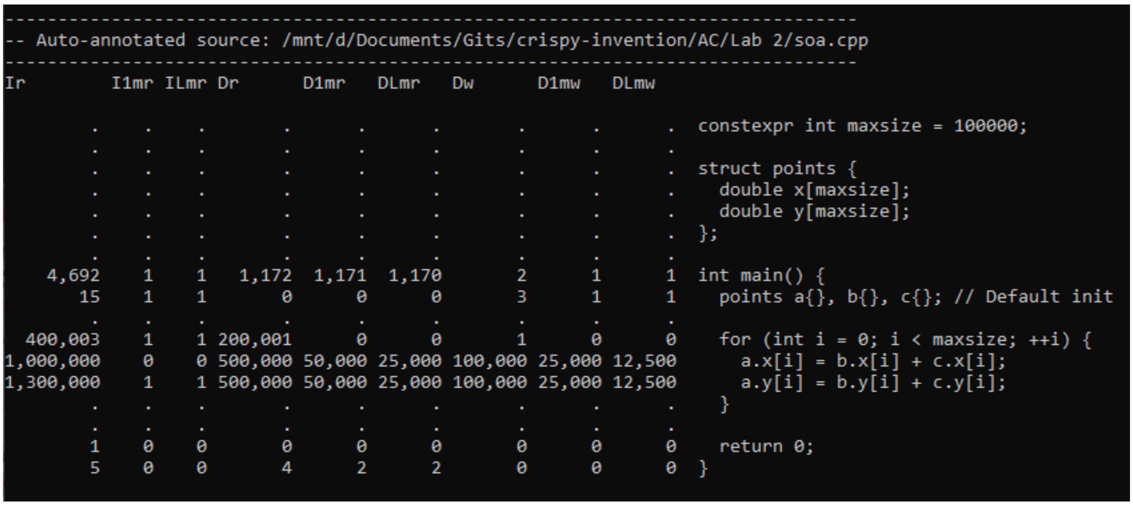
```
==252==
==252== I   refs:      7,615,238
==252== I1  misses:          908
==252== LLi misses:          900
==252== I1  miss rate:      0.01%
==252== LLi miss rate:      0.01%
==252==
==252== D   refs:      6,239,991  (1,228,747 rd   + 5,011,244 wr)
==252== D1  misses:      304,338  (  103,324 rd   +   201,014 wr)
==252== LLd misses:      152,826  (   52,339 rd   +   100,487 wr)
==252== D1  miss rate:       4.9% (      8.4%     +       4.0%  )
==252== LLd miss rate:       2.4% (      4.3%     +       2.0%  )
==252==
==252== LL refs:         305,246  (  104,232 rd   +   201,014 wr)
==252== LL misses:       153,726  (   53,239 rd   +   100,487 wr)
==252== LL miss rate:        1.1% (      0.6%     +       2.0%  )
```

L1 de 32 KiB con tamaño de línea de 64 B, L2 de 256 KiB con tamaño de línea de 64B

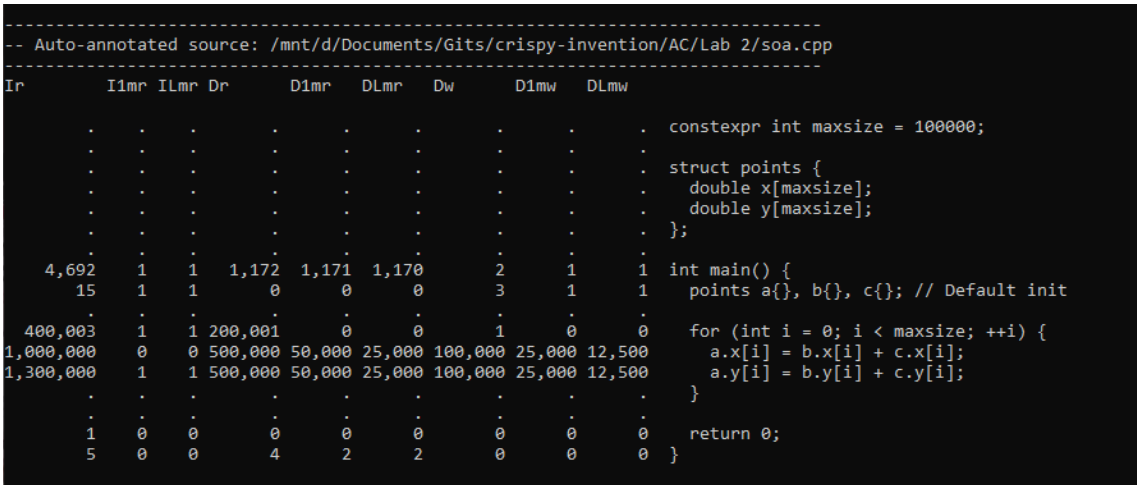


```
==255==
==255== I   refs:      7,615,238
==255== I1  misses:          908
==255== LLi misses:          900
==255== I1  miss rate:      0.01%
==255== LLi miss rate:      0.01%
==255==
==255== D   refs:      6,239,991  (1,228,747 rd   + 5,011,244 wr)
==255== D1  misses:      153,070  (   52,508 rd   +   100,562 wr)
==255== LLd misses:      152,826  (   52,339 rd   +   100,487 wr)
==255== D1  miss rate:       2.5% (      4.3%     +       2.0%  )
==255== LLd miss rate:       2.4% (      4.3%     +       2.0%  )
==255==
==255== LL refs:         153,978  (   53,416 rd   +   100,562 wr)
==255== LL misses:       153,726  (   53,239 rd   +   100,487 wr)
==255== LL miss rate:        1.1% (      0.6%     +       2.0%  )
```

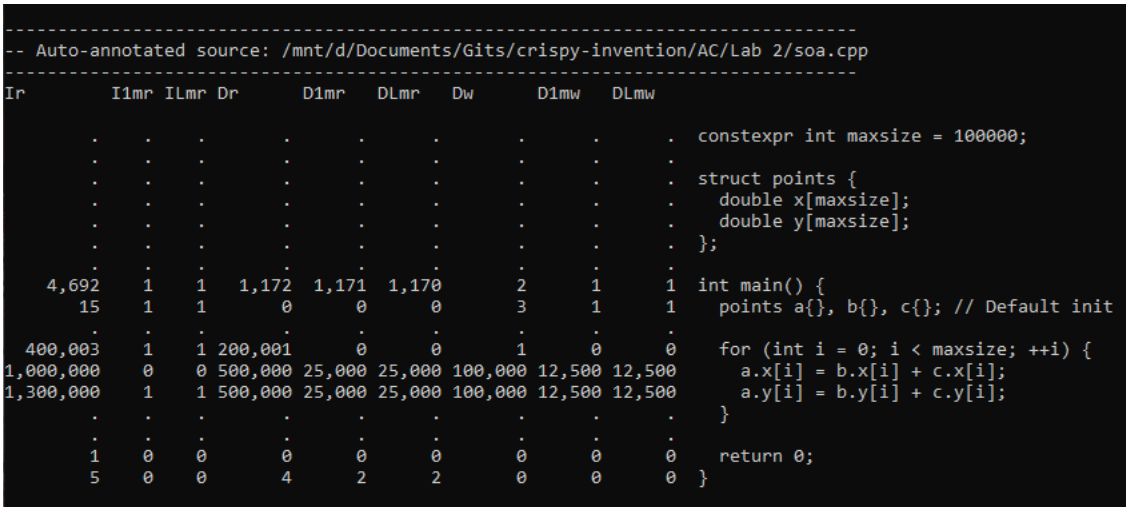
2. cg_annotate



```
--------------------------------------------------------------------------------
-- Auto-annotated source: /mnt/d/Documents/Gits/crispy-invention/AC/Lab 2/soa.cpp
--------------------------------------------------------------------------------
Ir        I1mr ILmr Dr      D1mr   DLmr   Dw      D1mw   DLmw

        .    .    .       .      .      .       .      .      .  constexpr int maxsize = 100000;
        .    .    .       .      .      .       .      .      .
        .    .    .       .      .      .       .      .      .  struct points {
        .    .    .       .      .      .       .      .      .    double x[maxsize];
        .    .    .       .      .      .       .      .      .    double y[maxsize];
        .    .    .       .      .      .       .      .      .  };
        .    .    .       .      .      .       .      .      .
    4,692    1    1   1,172  1,171  1,170       2      1      1  int main() {
       15    1    1       0      0      0       3      1      1    points a{}, b{}, c{}; // Default init
        .    .    .       .      .      .       .      .      .
  400,003    1    1 200,001      0      0       1      0      0    for (int i = 0; i < maxsize; ++i) {
1,000,000    0    0 500,000 50,000 25,000 100,000 25,000 12,500      a.x[i] = b.x[i] + c.x[i];
1,300,000    1    1 500,000 50,000 25,000 100,000 25,000 12,500      a.y[i] = b.y[i] + c.y[i];
        .    .    .       .      .      .       .      .      .    }
        .    .    .       .      .      .       .      .      .
        1    0    0       0      0      0       0      0      0    return 0;
        5    0    0       4      2      2       0      0      0  }
```



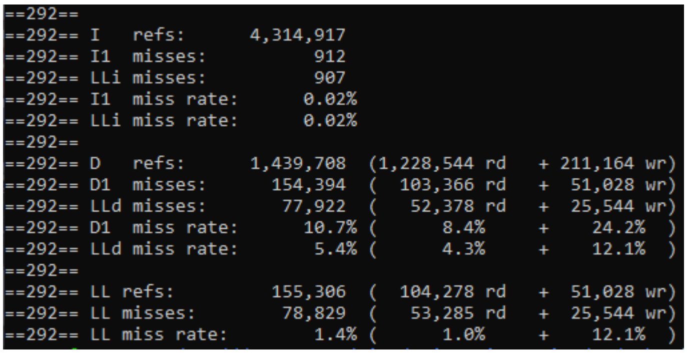
```
--------------------------------------------------------------------------------
-- Auto-annotated source: /mnt/d/Documents/Gits/crispy-invention/AC/Lab 2/soa.cpp
--------------------------------------------------------------------------------
Ir        I1mr ILmr Dr      D1mr   DLmr   Dw      D1mw   DLmw

        .    .    .       .      .      .       .      .      .  constexpr int maxsize = 100000;
        .    .    .       .      .      .       .      .      .
        .    .    .       .      .      .       .      .      .  struct points {
        .    .    .       .      .      .       .      .      .    double x[maxsize];
        .    .    .       .      .      .       .      .      .    double y[maxsize];
        .    .    .       .      .      .       .      .      .  };
        .    .    .       .      .      .       .      .      .
    4,692    1    1   1,172  1,171  1,170       2      1      1  int main() {
       15    1    1       0      0      0       3      1      1    points a{}, b{}, c{}; // Default init
        .    .    .       .      .      .       .      .      .
  400,003    1    1 200,001      0      0       1      0      0    for (int i = 0; i < maxsize; ++i) {
1,000,000    0    0 500,000 50,000 25,000 100,000 25,000 12,500      a.x[i] = b.x[i] + c.x[i];
1,300,000    1    1 500,000 50,000 25,000 100,000 25,000 12,500      a.y[i] = b.y[i] + c.y[i];
        .    .    .       .      .      .       .      .      .    }
        .    .    .       .      .      .       .      .      .
        1    0    0       0      0      0       0      0      0    return 0;
        5    0    0       4      2      2       0      0      0  }
```



```
--------------------------------------------------------------------------------
-- Auto-annotated source: /mnt/d/Documents/Gits/crispy-invention/AC/Lab 2/soa.cpp
--------------------------------------------------------------------------------
Ir        I1mr ILmr Dr      D1mr   DLmr   Dw      D1mw   DLmw

        .    .    .       .      .      .       .      .      .  constexpr int maxsize = 100000;
        .    .    .       .      .      .       .      .      .
        .    .    .       .      .      .       .      .      .  struct points {
        .    .    .       .      .      .       .      .      .    double x[maxsize];
        .    .    .       .      .      .       .      .      .    double y[maxsize];
        .    .    .       .      .      .       .      .      .  };
        .    .    .       .      .      .       .      .      .
    4,692    1    1   1,172  1,171  1,170       2      1      1  int main() {
       15    1    1       0      0      0       3      1      1    points a{}, b{}, c{}; // Default init
        .    .    .       .      .      .       .      .      .
  400,003    1    1 200,001      0      0       1      0      0    for (int i = 0; i < maxsize; ++i) {
1,000,000    0    0 500,000 25,000 25,000 100,000 12,500 12,500      a.x[i] = b.x[i] + c.x[i];
1,300,000    1    1 500,000 25,000 25,000 100,000 12,500 12,500      a.y[i] = b.y[i] + c.y[i];
        .    .    .       .      .      .       .      .      .    }
        .    .    .       .      .      .       .      .      .
        1    0    0       0      0      0       0      0      0    return 0;
        5    0    0       4      2      2       0      0      0  }
```

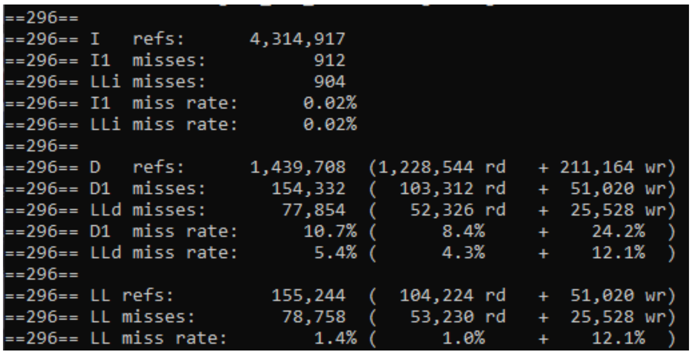
3. aos

L1 de 16 KiB con tamaño de línea de 32 B, L2 de 128 KiB con tamaño de línea de 64B



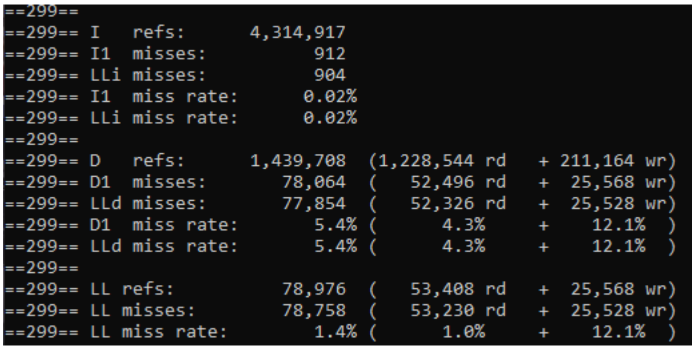
```
==292==
==292== I   refs:      4,314,917
==292== I1  misses:          912
==292== LLi misses:          907
==292== I1  miss rate:      0.02%
==292== LLi miss rate:      0.02%
==292==
==292== D   refs:      1,439,708  (1,228,544 rd   + 211,164 wr)
==292== D1  misses:      154,394  (  103,366 rd   +  51,028 wr)
==292== LLd misses:       77,922  (   52,378 rd   +  25,544 wr)
==292== D1  miss rate:      10.7% (      8.4%     +    24.2%  )
==292== LLd miss rate:       5.4% (      4.3%     +    12.1%  )
==292==
==292== LL refs:         155,306  (  104,278 rd   +  51,028 wr)
==292== LL misses:        78,829  (   53,285 rd   +  25,544 wr)
==292== LL miss rate:        1.4% (      1.0%     +    12.1%  )
```

L1 de 32 KiB con tamaño de línea de 32 B, L2 de 256 KiB con tamaño de línea de 64B



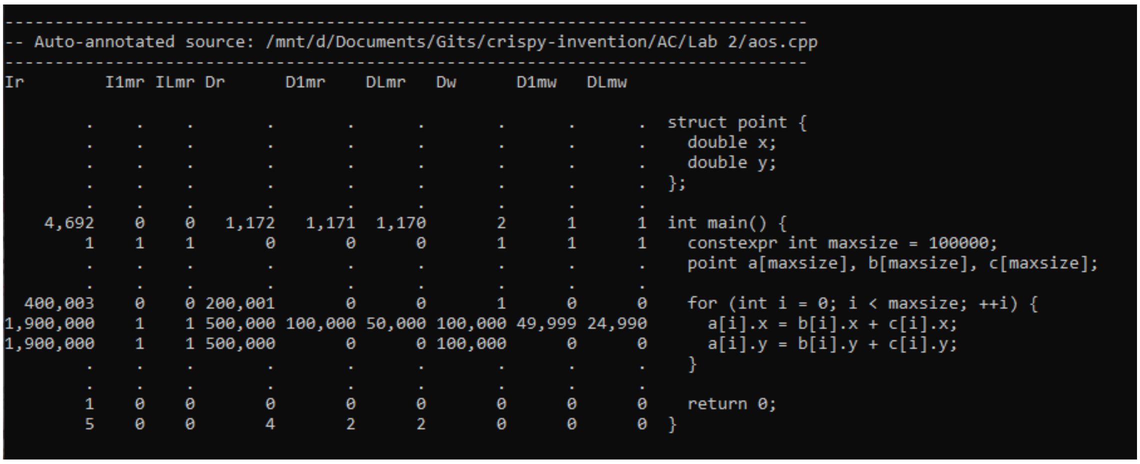
```
==296==
==296== I   refs:      4,314,917
==296== I1  misses:          912
==296== LLi misses:          904
==296== I1  miss rate:      0.02%
==296== LLi miss rate:      0.02%
==296==
==296== D   refs:      1,439,708  (1,228,544 rd   + 211,164 wr)
==296== D1  misses:      154,332  (  103,312 rd   +  51,020 wr)
==296== LLd misses:       77,854  (   52,326 rd   +  25,528 wr)
==296== D1  miss rate:      10.7% (      8.4%     +    24.2%  )
==296== LLd miss rate:       5.4% (      4.3%     +    12.1%  )
==296==
==296== LL refs:         155,244  (  104,224 rd   +  51,020 wr)
==296== LL misses:        78,758  (   53,230 rd   +  25,528 wr)
==296== LL miss rate:        1.4% (      1.0%     +    12.1%  )
```

L1 de 32 KiB con tamaño de línea de 64 B, L2 de 256 KiB con tamaño de línea de 64B

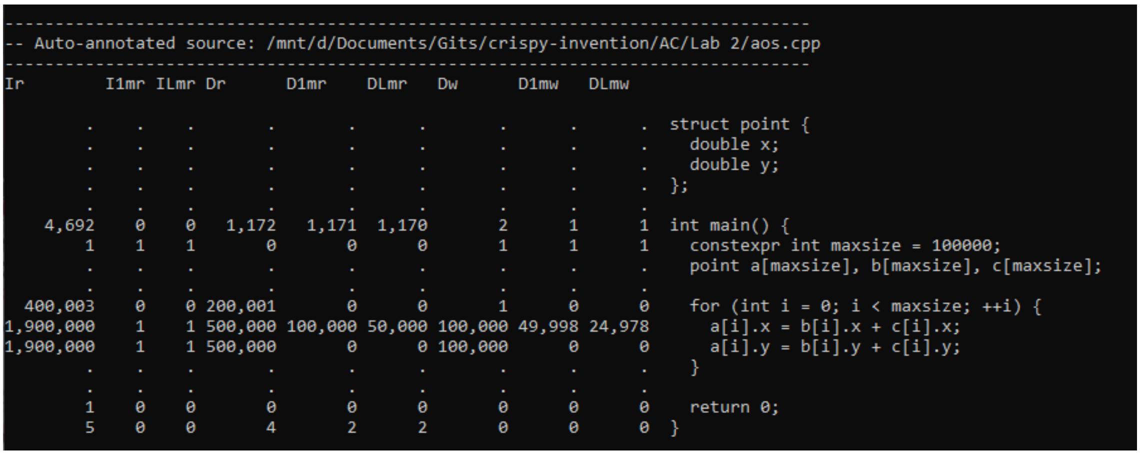


```
==299==
==299== I   refs:      4,314,917
==299== I1  misses:          912
==299== LLi misses:          904
==299== I1  miss rate:      0.02%
==299== LLi miss rate:      0.02%
==299==
==299== D   refs:      1,439,708  (1,228,544 rd   + 211,164 wr)
==299== D1  misses:       78,064  (   52,496 rd   +  25,568 wr)
==299== LLd misses:       77,854  (   52,326 rd   +  25,528 wr)
==299== D1  miss rate:       5.4% (      4.3%     +    12.1%  )
==299== LLd miss rate:       5.4% (      4.3%     +    12.1%  )
==299==
==299== LL refs:          78,976  (   53,408 rd   +  25,568 wr)
==299== LL misses:        78,758  (   53,230 rd   +  25,528 wr)
==299== LL miss rate:        1.4% (      1.0%     +    12.1%  )
```

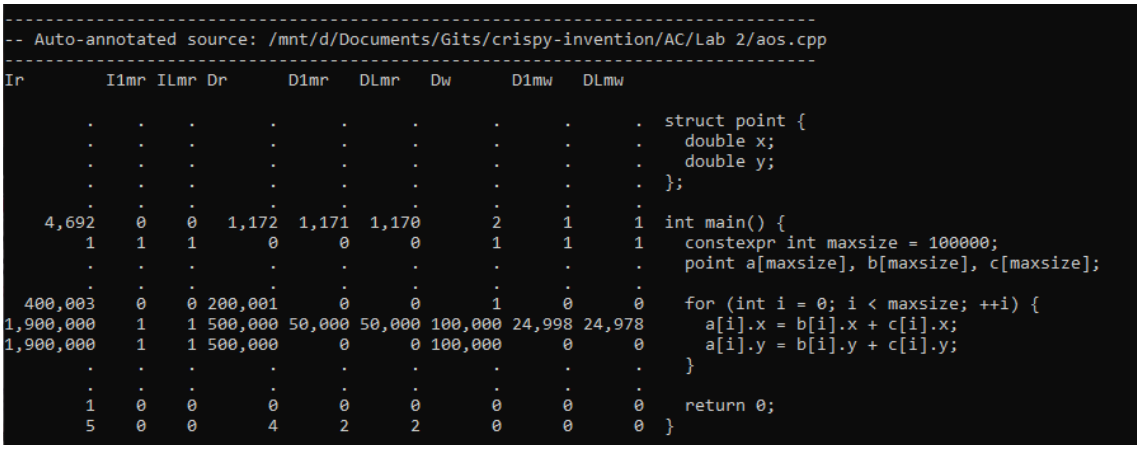
4. cg_annotate



```
--------------------------------------------------------------------------------
-- Auto-annotated source: /mnt/d/Documents/Gits/crispy-invention/AC/Lab 2/aos.cpp
--------------------------------------------------------------------------------
Ir        I1mr ILmr Dr      D1mr    DLmr   Dw      D1mw   DLmw

        .    .    .       .       .      .       .      .      .  struct point {
        .    .    .       .       .      .       .      .      .    double x;
        .    .    .       .       .      .       .      .      .    double y;
        .    .    .       .       .      .       .      .      .  };
        .    .    .       .       .      .       .      .      .
    4,692    0    0   1,172   1,171  1,170       2      1      1  int main() {
        1    1    1       0       0      0       1      1      1    constexpr int maxsize = 100000;
        .    .    .       .       .      .       .      .      .    point a[maxsize], b[maxsize], c[maxsize];
        .    .    .       .       .      .       .      .      .
  400,003    0    0 200,001       0      0       1      0      0    for (int i = 0; i < maxsize; ++i) {
1,900,000    1    1 500,000 100,000 50,000 100,000 49,999 24,990      a[i].x = b[i].x + c[i].x;
1,900,000    1    1 500,000       0      0 100,000      0      0      a[i].y = b[i].y + c[i].y;
        .    .    .       .       .      .       .      .      .    }
        .    .    .       .       .      .       .      .      .
        1    0    0       0       0      0       0      0      0    return 0;
        5    0    0       4       2      2       0      0      0  }
```



```
--------------------------------------------------------------------------------
-- Auto-annotated source: /mnt/d/Documents/Gits/crispy-invention/AC/Lab 2/aos.cpp
--------------------------------------------------------------------------------
Ir        I1mr ILmr Dr      D1mr    DLmr   Dw      D1mw   DLmw

        .    .    .       .       .      .       .      .      .  struct point {
        .    .    .       .       .      .       .      .      .    double x;
        .    .    .       .       .      .       .      .      .    double y;
        .    .    .       .       .      .       .      .      .  };
        .    .    .       .       .      .       .      .      .
    4,692    0    0   1,172   1,171  1,170       2      1      1  int main() {
        1    1    1       0       0      0       1      1      1    constexpr int maxsize = 100000;
        .    .    .       .       .      .       .      .      .    point a[maxsize], b[maxsize], c[maxsize];
        .    .    .       .       .      .       .      .      .
  400,003    0    0 200,001       0      0       1      0      0    for (int i = 0; i < maxsize; ++i) {
1,900,000    1    1 500,000 100,000 50,000 100,000 49,998 24,978      a[i].x = b[i].x + c[i].x;
1,900,000    1    1 500,000       0      0 100,000      0      0      a[i].y = b[i].y + c[i].y;
        .    .    .       .       .      .       .      .      .    }
        .    .    .       .       .      .       .      .      .
        1    0    0       0       0      0       0      0      0    return 0;
        5    0    0       4       2      2       0      0      0  }
```



```
--------------------------------------------------------------------------------
-- Auto-annotated source: /mnt/d/Documents/Gits/crispy-invention/AC/Lab 2/aos.cpp
--------------------------------------------------------------------------------
Ir        I1mr ILmr Dr      D1mr   DLmr   Dw      D1mw   DLmw

        .    .    .       .      .      .       .      .      .  struct point {
        .    .    .       .      .      .       .      .      .    double x;
        .    .    .       .      .      .       .      .      .    double y;
        .    .    .       .      .      .       .      .      .  };
        .    .    .       .      .      .       .      .      .
    4,692    0    0   1,172  1,171  1,170       2      1      1  int main() {
        1    1    1       0      0      0       1      1      1    constexpr int maxsize = 100000;
        .    .    .       .      .      .       .      .      .    point a[maxsize], b[maxsize], c[maxsize];
        .    .    .       .      .      .       .      .      .
  400,003    0    0 200,001      0      0       1      0      0    for (int i = 0; i < maxsize; ++i) {
1,900,000    1    1 500,000 50,000 50,000 100,000 24,998 24,978      a[i].x = b[i].x + c[i].x;
1,900,000    1    1 500,000      0      0 100,000      0      0      a[i].y = b[i].y + c[i].y;
        .    .    .       .      .      .       .      .      .    }
        .    .    .       .      .      .       .      .      .
        1    0    0       0      0      0       0      0      0    return 0;
        5    0    0       4      2      2       0      0      0  }
```

5. Compare ambos resultados.

Diferencia en el ratio de fallos en L1:

4.9	10.7	-5.8
4.9	10.7	-5.8
2.5	5.4	-2.9

Diferencia en el ratio de fallos en L2:

2.5	5.4	-2.9
2.4	5.4	-3
2.4	5.4	-3

Diferencia en el ratio de fallos local en L2:

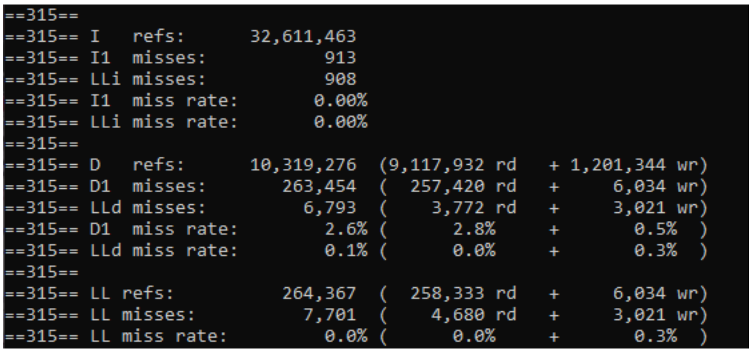
1.1	1.4	-0.3
1.1	1.4	-0.3
1.1	1.4	-0.3

Tarea 4: Producto de matrices

product.cpp y producto_block.cpp

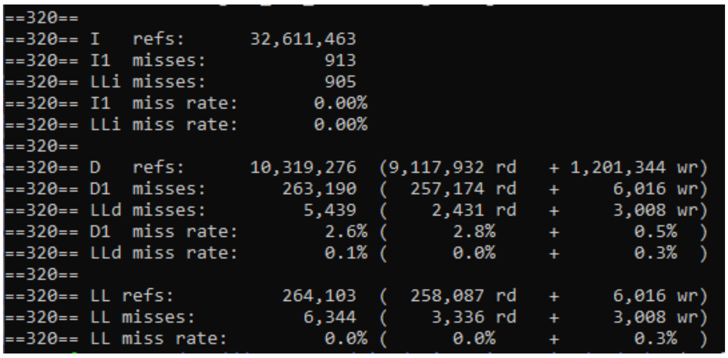
1. product.cpp

L1 de 16 KiB con tamaño de línea de 32 B, L2 de 128 KiB con tamaño de línea de 64B



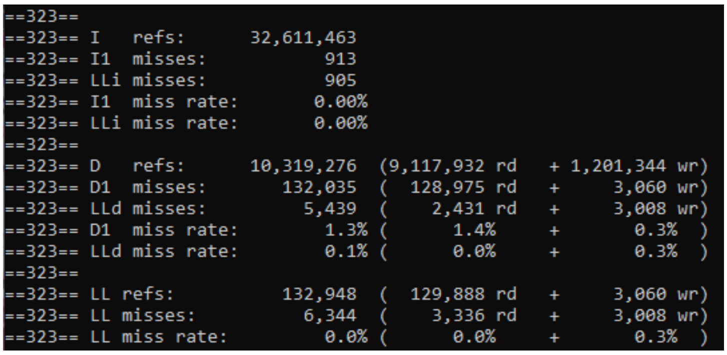
```
==315==
==315== I   refs:      32,611,463
==315== I1  misses:           913
==315== LLi misses:           908
==315== I1  miss rate:       0.00%
==315== LLi miss rate:       0.00%
==315==
==315== D   refs:      10,319,276  (9,117,932 rd   + 1,201,344 wr)
==315== D1  misses:       263,454  (  257,420 rd   +     6,034 wr)
==315== LLd misses:         6,793  (    3,772 rd   +     3,021 wr)
==315== D1  miss rate:        2.6% (      2.8%     +       0.5%  )
==315== LLd miss rate:        0.1% (      0.0%     +       0.3%  )
==315==
==315== LL refs:          264,367  (  258,333 rd   +     6,034 wr)
==315== LL misses:          7,701  (    4,680 rd   +     3,021 wr)
==315== LL miss rate:         0.0% (      0.0%     +       0.3%  )
```

L1 de 32 KiB con tamaño de línea de 32 B, L2 de 256 KiB con tamaño de línea de 64B



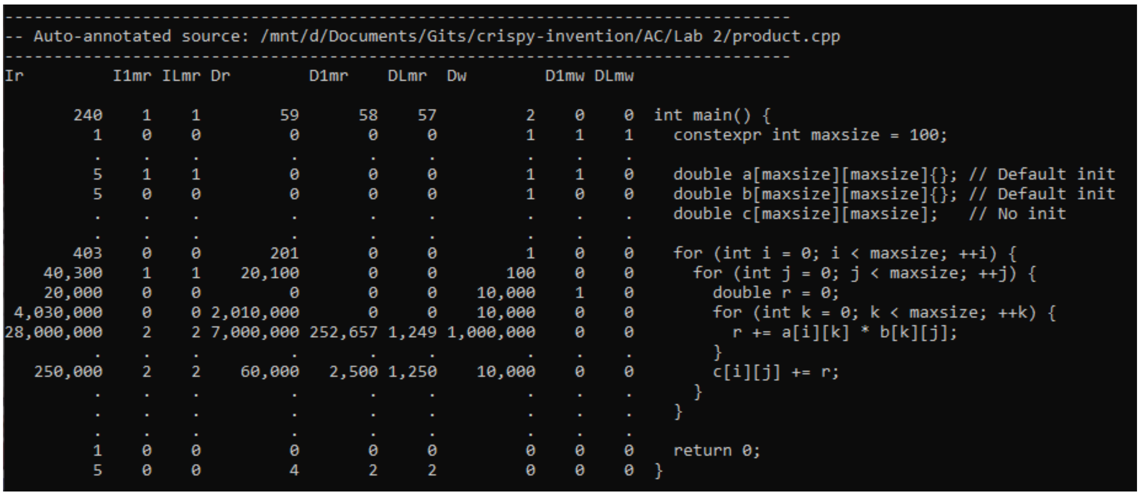
```
==320==
==320== I   refs:      32,611,463
==320== I1  misses:           913
==320== LLi misses:           905
==320== I1  miss rate:       0.00%
==320== LLi miss rate:       0.00%
==320==
==320== D   refs:      10,319,276  (9,117,932 rd   + 1,201,344 wr)
==320== D1  misses:       263,190  (  257,174 rd   +     6,016 wr)
==320== LLd misses:         5,439  (    2,431 rd   +     3,008 wr)
==320== D1  miss rate:        2.6% (      2.8%     +       0.5%  )
==320== LLd miss rate:        0.1% (      0.0%     +       0.3%  )
==320==
==320== LL refs:          264,103  (  258,087 rd   +     6,016 wr)
==320== LL misses:          6,344  (    3,336 rd   +     3,008 wr)
==320== LL miss rate:         0.0% (      0.0%     +       0.3%  )
```

L1 de 32 KiB con tamaño de línea de 64 B, L2 de 256 KiB con tamaño de línea de 64B

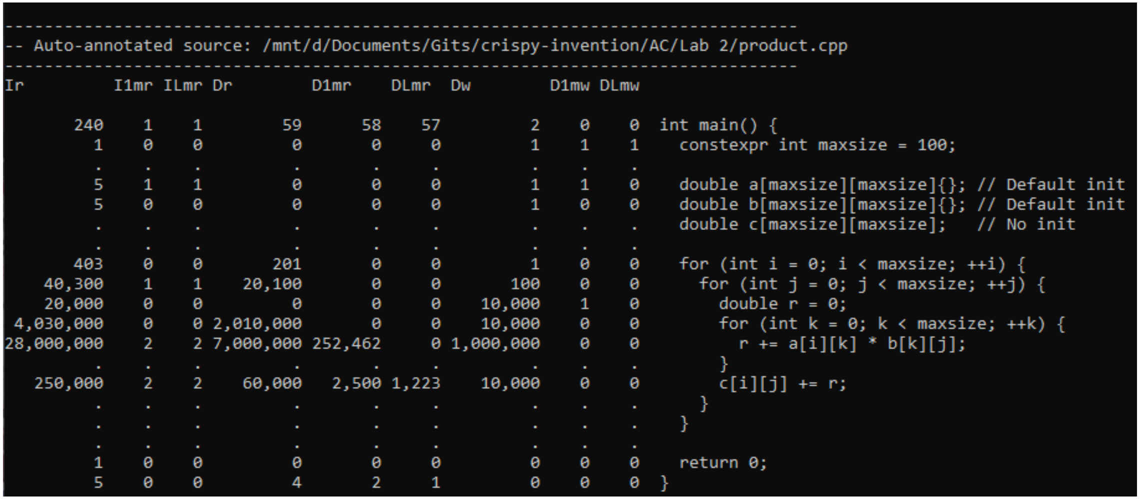


```
==323==
==323== I   refs:      32,611,463
==323== I1  misses:           913
==323== LLi misses:           905
==323== I1  miss rate:       0.00%
==323== LLi miss rate:       0.00%
==323==
==323== D   refs:      10,319,276  (9,117,932 rd   + 1,201,344 wr)
==323== D1  misses:       132,035  (  128,975 rd   +     3,060 wr)
==323== LLd misses:         5,439  (    2,431 rd   +     3,008 wr)
==323== D1  miss rate:        1.3% (      1.4%     +       0.3%  )
==323== LLd miss rate:        0.1% (      0.0%     +       0.3%  )
==323==
==323== LL refs:          132,948  (  129,888 rd   +     3,060 wr)
==323== LL misses:          6,344  (    3,336 rd   +     3,008 wr)
==323== LL miss rate:         0.0% (      0.0%     +       0.3%  )
```

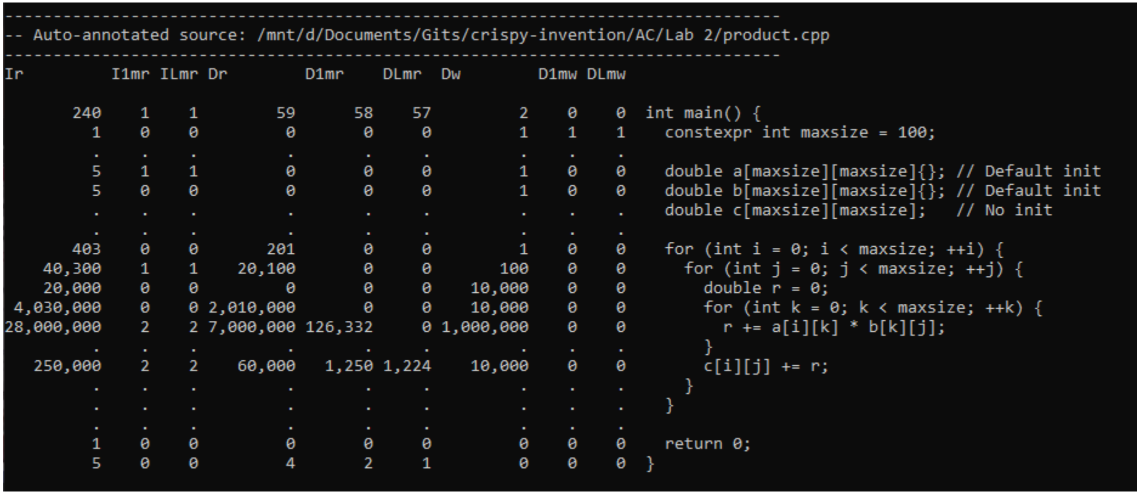
2. cg_annotate



```
--------------------------------------------------------------------------------
-- Auto-annotated source: /mnt/d/Documents/Gits/crispy-invention/AC/Lab 2/product.cpp
--------------------------------------------------------------------------------
Ir         I1mr ILmr Dr        D1mr    DLmr  Dw        D1mw DLmw

       240    1    1        59      58    57         2    0    0  int main() {
         1    0    0         0       0     0         1    1    1    constexpr int maxsize = 100;
         .    .    .         .       .     .         .    .    .
         5    1    1         0       0     0         1    1    0    double a[maxsize][maxsize]{}; // Default init
         5    0    0         0       0     0         1    0    0    double b[maxsize][maxsize]{}; // Default init
         .    .    .         .       .     .         .    .    .    double c[maxsize][maxsize];   // No init
         .    .    .         .       .     .         .    .    .
       403    0    0       201       0     0         1    0    0    for (int i = 0; i < maxsize; ++i) {
    40,300    1    1    20,100       0     0       100    0    0      for (int j = 0; j < maxsize; ++j) {
    20,000    0    0         0       0     0    10,000    1    0        double r = 0;
 4,030,000    0    0 2,010,000       0     0    10,000    0    0        for (int k = 0; k < maxsize; ++k) {
28,000,000    2    2 7,000,000 252,657 1,249 1,000,000    0    0          r += a[i][k] * b[k][j];
         .    .    .         .       .     .         .    .    .        }
   250,000    2    2    60,000   2,500 1,250    10,000    0    0        c[i][j] += r;
         .    .    .         .       .     .         .    .    .      }
         .    .    .         .       .     .         .    .    .    }
         .    .    .         .       .     .         .    .    .
         1    0    0         0       0     0         0    0    0    return 0;
         5    0    0         4       2     2         0    0    0  }
```



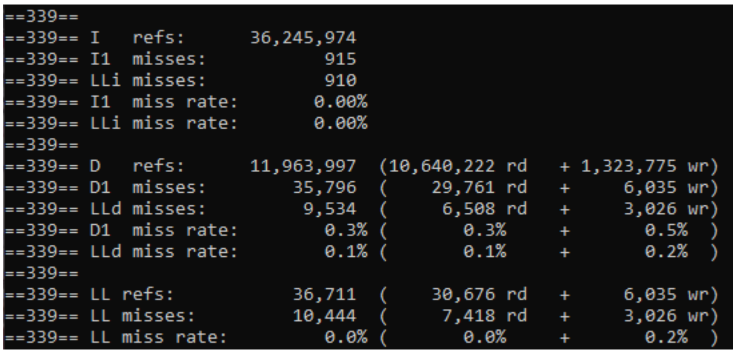
```
--------------------------------------------------------------------------------
-- Auto-annotated source: /mnt/d/Documents/Gits/crispy-invention/AC/Lab 2/product.cpp
--------------------------------------------------------------------------------
Ir         I1mr ILmr Dr        D1mr    DLmr  Dw        D1mw DLmw

       240    1    1        59      58    57         2    0    0  int main() {
         1    0    0         0       0     0         1    1    1    constexpr int maxsize = 100;
         .    .    .         .       .     .         .    .    .
         5    1    1         0       0     0         1    1    0    double a[maxsize][maxsize]{}; // Default init
         5    0    0         0       0     0         1    0    0    double b[maxsize][maxsize]{}; // Default init
         .    .    .         .       .     .         .    .    .    double c[maxsize][maxsize];   // No init
         .    .    .         .       .     .         .    .    .
       403    0    0       201       0     0         1    0    0    for (int i = 0; i < maxsize; ++i) {
    40,300    1    1    20,100       0     0       100    0    0      for (int j = 0; j < maxsize; ++j) {
    20,000    0    0         0       0     0    10,000    1    0        double r = 0;
 4,030,000    0    0 2,010,000       0     0    10,000    0    0        for (int k = 0; k < maxsize; ++k) {
28,000,000    2    2 7,000,000 252,462     0 1,000,000    0    0          r += a[i][k] * b[k][j];
         .    .    .         .       .     .         .    .    .        }
   250,000    2    2    60,000   2,500 1,223    10,000    0    0        c[i][j] += r;
         .    .    .         .       .     .         .    .    .      }
         .    .    .         .       .     .         .    .    .    }
         .    .    .         .       .     .         .    .    .
         1    0    0         0       0     0         0    0    0    return 0;
         5    0    0         4       2     1         0    0    0  }
```



```
--------------------------------------------------------------------------------
-- Auto-annotated source: /mnt/d/Documents/Gits/crispy-invention/AC/Lab 2/product.cpp
--------------------------------------------------------------------------------
Ir         I1mr ILmr Dr        D1mr    DLmr  Dw        D1mw DLmw

       240    1    1        59      58    57         2    0    0  int main() {
         1    0    0         0       0     0         1    1    1    constexpr int maxsize = 100;
         .    .    .         .       .     .         .    .    .
         5    1    1         0       0     0         1    0    0    double a[maxsize][maxsize]{}; // Default init
         5    0    0         0       0     0         1    0    0    double b[maxsize][maxsize]{}; // Default init
         .    .    .         .       .     .         .    .    .    double c[maxsize][maxsize];   // No init
         .    .    .         .       .     .         .    .    .
       403    0    0       201       0     0         1    0    0    for (int i = 0; i < maxsize; ++i) {
    40,300    1    1    20,100       0     0       100    0    0      for (int j = 0; j < maxsize; ++j) {
    20,000    0    0         0       0     0    10,000    0    0        double r = 0;
 4,030,000    0    0 2,010,000       0     0    10,000    0    0        for (int k = 0; k < maxsize; ++k) {
28,000,000    2    2 7,000,000 126,332     0 1,000,000    0    0          r += a[i][k] * b[k][j];
         .    .    .         .       .     .         .    .    .        }
   250,000    2    2    60,000   1,250 1,224    10,000    0    0        c[i][j] += r;
         .    .    .         .       .     .         .    .    .      }
         .    .    .         .       .     .         .    .    .    }
         .    .    .         .       .     .         .    .    .
         1    0    0         0       0     0         0    0    0    return 0;
         5    0    0         4       2     1         0    0    0  }
```

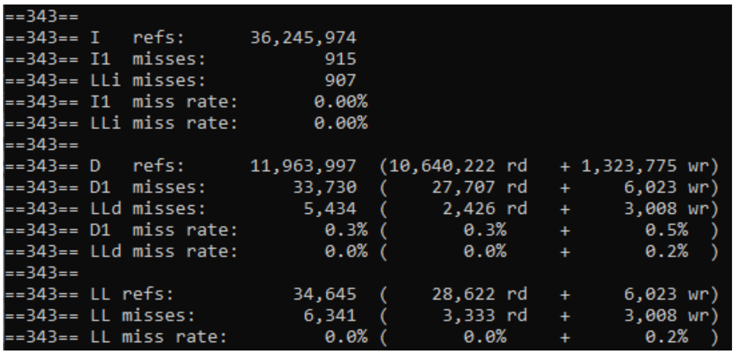
3. producto_block.cpp

L1 de 16 KiB con tamaño de línea de 32 B, L2 de 128 KiB con tamaño de línea de 64B



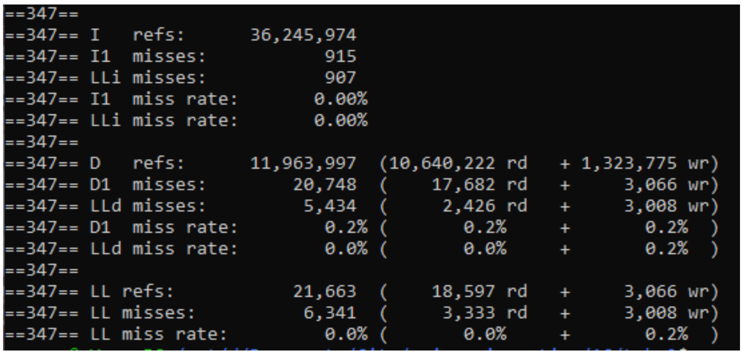
```
==339==
==339== I   refs:      36,245,974
==339== I1  misses:           915
==339== LLi misses:           910
==339== I1  miss rate:       0.00%
==339== LLi miss rate:       0.00%
==339==
==339== D   refs:      11,963,997  (10,640,222 rd   + 1,323,775 wr)
==339== D1  misses:        35,796  (    29,761 rd   +     6,035 wr)
==339== LLd misses:         9,534  (     6,508 rd   +     3,026 wr)
==339== D1  miss rate:        0.3% (       0.3%     +       0.5%  )
==339== LLd miss rate:        0.1% (       0.1%     +       0.2%  )
==339==
==339== LL refs:           36,711  (    30,676 rd   +     6,035 wr)
==339== LL misses:         10,444  (     7,418 rd   +     3,026 wr)
==339== LL miss rate:         0.0% (       0.0%     +       0.2%  )
```

L1 de 32 KiB con tamaño de línea de 32 B, L2 de 256 KiB con tamaño de línea de 64B



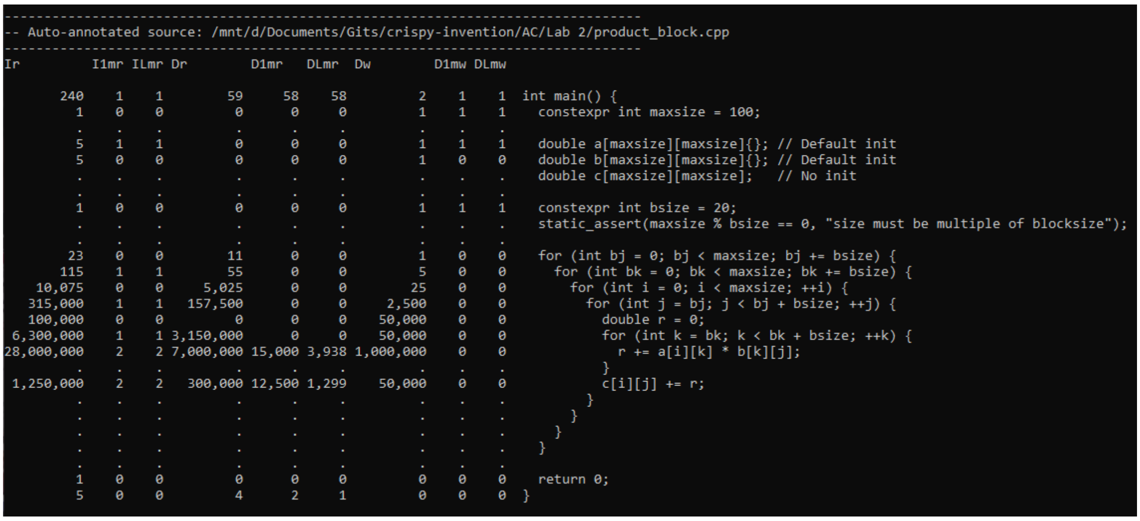
```
==343==
==343== I   refs:      36,245,974
==343== I1  misses:           915
==343== LLi misses:           907
==343== I1  miss rate:       0.00%
==343== LLi miss rate:       0.00%
==343==
==343== D   refs:      11,963,997  (10,640,222 rd   + 1,323,775 wr)
==343== D1  misses:        33,730  (    27,707 rd   +     6,023 wr)
==343== LLd misses:         5,434  (     2,426 rd   +     3,008 wr)
==343== D1  miss rate:        0.3% (       0.3%     +       0.5%  )
==343== LLd miss rate:        0.0% (       0.0%     +       0.2%  )
==343==
==343== LL refs:           34,645  (    28,622 rd   +     6,023 wr)
==343== LL misses:          6,341  (     3,333 rd   +     3,008 wr)
==343== LL miss rate:         0.0% (       0.0%     +       0.2%  )
```

L1 de 32 KiB con tamaño de línea de 64 B, L2 de 256 KiB con tamaño de línea de 64B

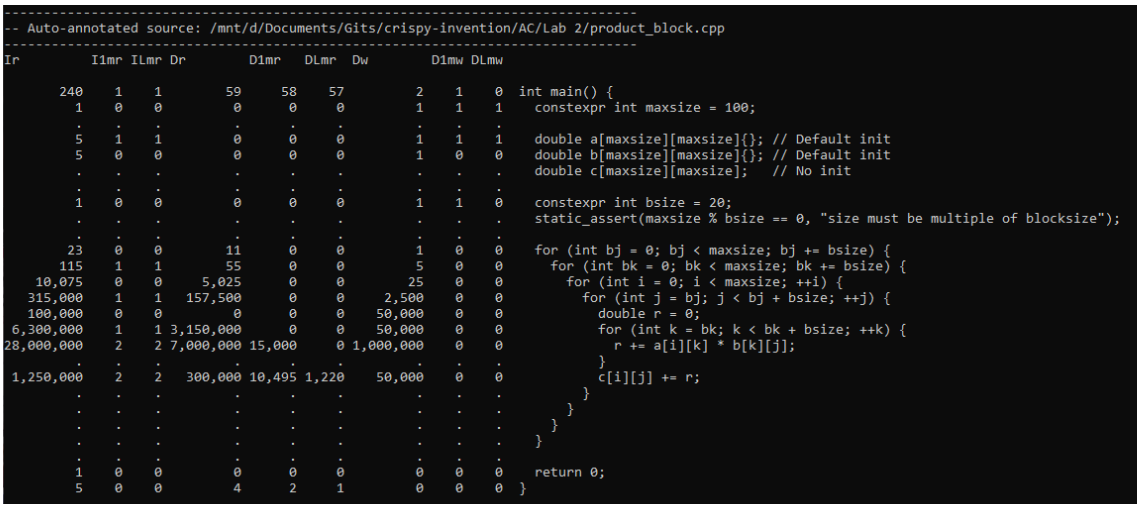


```
==347==
==347== I   refs:      36,245,974
==347== I1  misses:           915
==347== LLi misses:           907
==347== I1  miss rate:       0.00%
==347== LLi miss rate:       0.00%
==347==
==347== D   refs:      11,963,997  (10,640,222 rd   + 1,323,775 wr)
==347== D1  misses:        20,748  (    17,682 rd   +     3,066 wr)
==347== LLd misses:         5,434  (     2,426 rd   +     3,008 wr)
==347== D1  miss rate:        0.2% (       0.2%     +       0.2%  )
==347== LLd miss rate:        0.0% (       0.0%     +       0.2%  )
==347==
==347== LL refs:           21,663  (    18,597 rd   +     3,066 wr)
==347== LL misses:          6,341  (     3,333 rd   +     3,008 wr)
==347== LL miss rate:         0.0% (       0.0%     +       0.2%  )
```

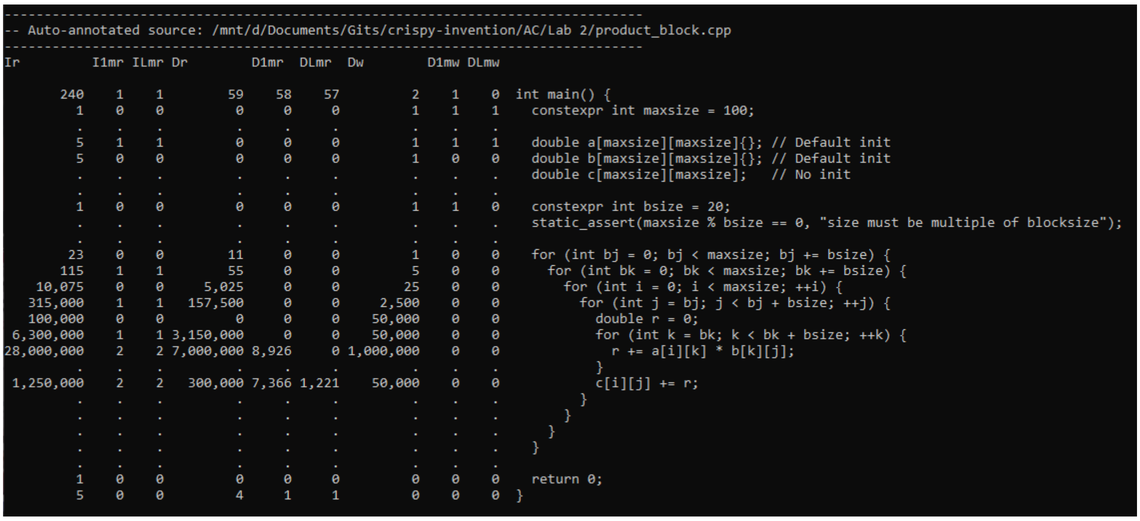
4. cg_annotate



```
--------------------------------------------------------------------------------
-- Auto-annotated source: /mnt/d/Documents/Gits/crispy-invention/AC/Lab 2/product_block.cpp
--------------------------------------------------------------------------------
Ir         I1mr ILmr Dr         D1mr   DLmr  Dw         D1mw DLmw

       240    1    1        59     58    58         2    1    1 int main() {
         1    0    0         0      0     0         1    1    1   constexpr int maxsize = 100;
         .    .    .         .      .     .         .    .    .
         5    1    1         0      0     0         1    1    1   double a[maxsize][maxsize]{}; // Default init
         5    0    0         0      0     0         1    0    0   double b[maxsize][maxsize]{}; // Default init
         .    .    .         .      .     .         .    .    .   double c[maxsize][maxsize];   // No init
         .    .    .         .      .     .         .    .    .
         1    0    0         0      0     0         1    1    1   constexpr int bsize = 20;
         .    .    .         .      .     .         .    .    .   static_assert(maxsize % bsize == 0, "size must be multiple of blocksize");
         .    .    .         .      .     .         .    .    .
        23    0    0        11      0     0         1    0    0   for (int bj = 0; bj < maxsize; bj += bsize) {
       115    1    1        55      0     0         5    0    0     for (int bk = 0; bk < maxsize; bk += bsize) {
    10,075    0    0     5,025      0     0        25    0    0       for (int i = 0; i < maxsize; ++i) {
   315,000    1    1   157,500      0     0     2,500    0    0         for (int j = bj; j < bj + bsize; ++j) {
   100,000    0    0         0      0     0    50,000    0    0           double r = 0;
 6,300,000    1    1 3,150,000      0     0    50,000    0    0           for (int k = bk; k < bk + bsize; ++k) {
28,000,000    2    2 7,000,000 15,000 3,938 1,000,000    0    0             r += a[i][k] * b[k][j];
         .    .    .         .      .     .         .    .    .           }
 1,250,000    2    2   300,000 12,500 1,299    50,000    0    0           c[i][j] += r;
         .    .    .         .      .     .         .    .    .         }
         .    .    .         .      .     .         .    .    .       }
         .    .    .         .      .     .         .    .    .     }
         .    .    .         .      .     .         .    .    .   }
         .    .    .         .      .     .         .    .    .
         1    0    0         0      0     0         0    0    0   return 0;
         5    0    0         4      2     1         0    0    0 }
```



```
--------------------------------------------------------------------------------
-- Auto-annotated source: /mnt/d/Documents/Gits/crispy-invention/AC/Lab 2/product_block.cpp
--------------------------------------------------------------------------------
Ir         I1mr ILmr Dr         D1mr   DLmr  Dw         D1mw DLmw

       240    1    1        59     58    57         2    1    0 int main() {
         1    0    0         0      0     0         1    1    1   constexpr int maxsize = 100;
         .    .    .         .      .     .         .    .    .
         5    1    1         0      0     0         1    1    1   double a[maxsize][maxsize]{}; // Default init
         5    0    0         0      0     0         1    0    0   double b[maxsize][maxsize]{}; // Default init
         .    .    .         .      .     .         .    .    .   double c[maxsize][maxsize];   // No init
         .    .    .         .      .     .         .    .    .
         1    0    0         0      0     0         1    1    0   constexpr int bsize = 20;
         .    .    .         .      .     .         .    .    .   static_assert(maxsize % bsize == 0, "size must be multiple of blocksize");
         .    .    .         .      .     .         .    .    .
        23    0    0        11      0     0         1    0    0   for (int bj = 0; bj < maxsize; bj += bsize) {
       115    1    1        55      0     0         5    0    0     for (int bk = 0; bk < maxsize; bk += bsize) {
    10,075    0    0     5,025      0     0        25    0    0       for (int i = 0; i < maxsize; ++i) {
   315,000    1    1   157,500      0     0     2,500    0    0         for (int j = bj; j < bj + bsize; ++j) {
   100,000    0    0         0      0     0    50,000    0    0           double r = 0;
 6,300,000    1    1 3,150,000      0     0    50,000    0    0           for (int k = bk; k < bk + bsize; ++k) {
28,000,000    2    2 7,000,000 15,000     0 1,000,000    0    0             r += a[i][k] * b[k][j];
         .    .    .         .      .     .         .    .    .           }
 1,250,000    2    2   300,000 10,495 1,220    50,000    0    0           c[i][j] += r;
         .    .    .         .      .     .         .    .    .         }
         .    .    .         .      .     .         .    .    .       }
         .    .    .         .      .     .         .    .    .     }
         .    .    .         .      .     .         .    .    .   }
         .    .    .         .      .     .         .    .    .
         1    0    0         0      0     0         0    0    0   return 0;
         5    0    0         4      2     1         0    0    0 }
```



```
--------------------------------------------------------------------------------
-- Auto-annotated source: /mnt/d/Documents/Gits/crispy-invention/AC/Lab 2/product_block.cpp
--------------------------------------------------------------------------------
Ir         I1mr ILmr Dr         D1mr  DLmr  Dw         D1mw DLmw

       240    1    1        59     58    57         2    1    0 int main() {
         1    0    0         0      0     0         1    1    1   constexpr int maxsize = 100;
         .    .    .         .      .     .         .    .    .
         5    1    1         0      0     0         1    1    1   double a[maxsize][maxsize]{}; // Default init
         5    0    0         0      0     0         1    0    0   double b[maxsize][maxsize]{}; // Default init
         .    .    .         .      .     .         .    .    .   double c[maxsize][maxsize];   // No init
         .    .    .         .      .     .         .    .    .
         1    0    0         0      0     0         1    1    0   constexpr int bsize = 20;
         .    .    .         .      .     .         .    .    .   static_assert(maxsize % bsize == 0, "size must be multiple of blocksize");
         .    .    .         .      .     .         .    .    .
        23    0    0        11      0     0         1    0    0   for (int bj = 0; bj < maxsize; bj += bsize) {
       115    1    1        55      0     0         5    0    0     for (int bk = 0; bk < maxsize; bk += bsize) {
    10,075    0    0     5,025      0     0        25    0    0       for (int i = 0; i < maxsize; ++i) {
   315,000    1    1   157,500      0     0     2,500    0    0         for (int j = bj; j < bj + bsize; ++j) {
   100,000    0    0         0      0     0    50,000    0    0           double r = 0;
 6,300,000    1    1 3,150,000      0     0    50,000    0    0           for (int k = bk; k < bk + bsize; ++k) {
28,000,000    2    2 7,000,000  8,926     0 1,000,000    0    0             r += a[i][k] * b[k][j];
         .    .    .         .      .     .         .    .    .           }
 1,250,000    2    2   300,000  7,366 1,221    50,000    0    0           c[i][j] += r;
         .    .    .         .      .     .         .    .    .         }
         .    .    .         .      .     .         .    .    .       }
         .    .    .         .      .     .         .    .    .     }
         .    .    .         .      .     .         .    .    .   }
         .    .    .         .      .     .         .    .    .
         1    0    0         0      0     0         0    0    0   return 0;
         5    0    0         4      1     1         0    0    0 }
```

5. Compare ambos resultados.

Diferencia en el ratio de fallos en L1: Mejora la localidad espacial en el caso optimizado.

2.6	0.3	2.3
2.6	0.3	2.3
1.3	0.2	1.1

Diferencia en el ratio de fallos en L2:

0.1	0.1	0.1
0.1	0.0	0.1
0.1	0.0	0.1

Diferencia en el ratio de fallos local en L2:

0.0	0.0	0.0
0.0	0.0	0.0
0.0	0.0	0.0