# Исследование оптимизации инвариантов цикла

## Тестовая инфраструктура

CPU: AMD Ryzen 7 2700 Eight-Core Processor 3.20 GHz

OS: Ubuntu 20.04.02

## Результаты тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| Файл | O0 | O1 | O1 (LICM отключён) |
| licm\_5 | 1,72 с | 0,99 с | 1,04 с |
| licm\_10 | 1,56 с | 0,69 с | 0,91 с |
| licm\_20 | 1,36 с | 0,73 с | 0,81 с |
| licm\_div\_5 | 1,73 с | 0,25 с | 1,73 с |
| licm\_div\_10 | 3,46 с | 0,34 с | 3,46 с |
| licm\_div\_20 | 6,93 с | 0,71 с | 6,89 с |

## Анализ результатов

### Различия между неоптимизированной O0 и оптимизированной O1 версиями

1. В неоптимизированной версии инвариантные, относительно итераций выражения вроде a1+a2 где a1 и a2 – неизменяемые входные аргументы, вычисляются на каждом шаге цикла. В оптимизированный версии эти значения вычисляются один раз за пределами цикла, а входе итераций подставляются готовые значения.
2. Код функции main содержит вычисление набора значений, в неоптимизированный версии, эти значения вычисляются при каждом запуске, а в оптимизированном – получены на этапе компиляции.
3. В оптимизированной версии вычисление инкремента и условия продолжения цикла встроено непосредственно в его тело, в неоптимизированной версии они вынесены в отдельные блоки.
4. С++ код функции func предполагает передачу параметров по значению, из-за этого в неоптимизированной версии происходит их копирование, но т.к. они являются неизменяемыми в ходе цикла, то в оптимизированной версии, по всей видимости, предоставляется некий "прямой" доступ к аргументам.

### Особенности версии с отключённым LICM

1. Отключение механизма LICM подразумевает собой отключение вынесения инвариантного кода за пределы цикла. В остальном, эта версия аналогична стандартной оптимизированной O1 версии.

### Общий вывод

Как показывают замеры, время выполнения программы при отключении LICM оптимизации, приближается ко времени исполнения программы, собираемой без оптимизаций вовсе. Это происходит потому, что в используемых примерах, доля инвариантного кода в циклах крайне высока, что делает влияние LICM оптимизации наиболее значимым (по сравнению с другими оптимизациями) на время выполнения программы.

## Приложение

### Приложение 1 полный вывод программы тестирования

Performance counter stats for 'build/licm\_5\_O0':

1 726,34 msec task-clock # 0,999 CPUs utilized

162 context-switches # 0,094 K/sec

0 cpu-migrations # 0,000 K/sec

895 page-faults # 0,518 K/sec

6 922 420 717 cycles # 4,010 GHz (83,32%)

959 588 stalled-cycles-frontend # 0,01% frontend cycles idle (83,32%)

5 705 496 389 stalled-cycles-backend # 82,42% backend cycles idle (83,32%)

10 807 521 473 instructions # 1,56 insn per cycle

# 0,53 stalled cycles per insn (83,33%)

801 509 450 branches # 464,284 M/sec (83,32%)

18 612 branch-misses # 0,00% of all branches (83,38%)

1,728830607 seconds time elapsed

1,722774000 seconds user

0,003997000 seconds sys

Performance counter stats for 'build/licm\_5\_O1':

996,73 msec task-clock # 0,999 CPUs utilized

96 context-switches # 0,096 K/sec

0 cpu-migrations # 0,000 K/sec

897 page-faults # 0,900 K/sec

4 040 298 008 cycles # 4,054 GHz (83,16%)

876 640 stalled-cycles-frontend # 0,02% frontend cycles idle (83,15%)

3 229 542 520 stalled-cycles-backend # 79,93% backend cycles idle (83,28%)

5 996 914 062 instructions # 1,48 insn per cycle

# 0,54 stalled cycles per insn (83,56%)

400 724 855 branches # 402,039 M/sec (83,55%)

13 948 branch-misses # 0,00% of all branches (83,30%)

0,997849069 seconds time elapsed

0,993385000 seconds user

0,004005000 seconds sys

Performance counter stats for 'build/licm\_5\_O1\_no\_licm':

1 041,66 msec task-clock # 0,998 CPUs utilized

97 context-switches # 0,093 K/sec

0 cpu-migrations # 0,000 K/sec

897 page-faults # 0,861 K/sec

4 174 406 971 cycles # 4,007 GHz (83,11%)

628 582 stalled-cycles-frontend # 0,02% frontend cycles idle (83,15%)

2 955 489 336 stalled-cycles-backend # 70,80% backend cycles idle (83,50%)

9 993 124 565 instructions # 2,39 insn per cycle

# 0,30 stalled cycles per insn (83,49%)

401 028 180 branches # 384,991 M/sec (83,49%)

13 872 branch-misses # 0,00% of all branches (83,25%)

1,043619413 seconds time elapsed

1,038080000 seconds user

0,004008000 seconds sys

Performance counter stats for 'build/licm\_10\_O0':

1 561,66 msec task-clock # 0,999 CPUs utilized

149 context-switches # 0,095 K/sec

0 cpu-migrations # 0,000 K/sec

1 188 page-faults # 0,761 K/sec

6 280 301 855 cycles # 4,022 GHz (83,36%)

1 618 062 stalled-cycles-frontend # 0,03% frontend cycles idle (83,36%)

5 421 496 423 stalled-cycles-backend # 86,33% backend cycles idle (83,36%)

9 406 385 137 instructions # 1,50 insn per cycle

# 0,58 stalled cycles per insn (83,36%)

401 486 290 branches # 257,089 M/sec (83,36%)

29 161 branch-misses # 0,01% of all branches (83,20%)

1,563440210 seconds time elapsed

1,558012000 seconds user

0,004005000 seconds sys

Performance counter stats for 'build/licm\_10\_O1':

689,51 msec task-clock # 0,998 CPUs utilized

65 context-switches # 0,094 K/sec

0 cpu-migrations # 0,000 K/sec

1 188 page-faults # 0,002 M/sec

2 755 106 616 cycles # 3,996 GHz (83,18%)

702 603 stalled-cycles-frontend # 0,03% frontend cycles idle (83,18%)

2 135 027 999 stalled-cycles-backend # 77,49% backend cycles idle (83,19%)

4 996 400 677 instructions # 1,81 insn per cycle

# 0,43 stalled cycles per insn (83,34%)

201 202 073 branches # 291,802 M/sec (83,76%)

10 770 branch-misses # 0,01% of all branches (83,34%)

0,690778371 seconds time elapsed

0,686095000 seconds user

0,004012000 seconds sys

Performance counter stats for 'build/licm\_10\_O1\_no\_licm':

913,88 msec task-clock # 0,999 CPUs utilized

86 context-switches # 0,094 K/sec

0 cpu-migrations # 0,000 K/sec

1 186 page-faults # 0,001 M/sec

3 659 988 440 cycles # 4,005 GHz (83,37%)

914 546 stalled-cycles-frontend # 0,02% frontend cycles idle (83,38%)

2 837 174 249 stalled-cycles-backend # 77,52% backend cycles idle (83,38%)

8 990 603 731 instructions # 2,46 insn per cycle

# 0,32 stalled cycles per insn (83,37%)

201 373 758 branches # 220,349 M/sec (83,38%)

12 261 branch-misses # 0,01% of all branches (83,12%)

0,915257514 seconds time elapsed

0,914414000 seconds user

0,000000000 seconds sys

Performance counter stats for 'build/licm\_20\_O0':

1 362,39 msec task-clock # 0,999 CPUs utilized

128 context-switches # 0,094 K/sec

0 cpu-migrations # 0,000 K/sec

2 063 page-faults # 0,002 M/sec

5 447 857 479 cycles # 3,999 GHz (83,28%)

2 837 788 stalled-cycles-frontend # 0,05% frontend cycles idle (83,28%)

4 684 402 111 stalled-cycles-backend # 85,99% backend cycles idle (83,27%)

8 701 145 995 instructions # 1,60 insn per cycle

# 0,54 stalled cycles per insn (83,28%)

201 970 296 branches # 148,247 M/sec (83,54%)

25 714 branch-misses # 0,01% of all branches (83,36%)

1,364293489 seconds time elapsed

1,350775000 seconds user

0,011989000 seconds sys

Performance counter stats for 'build/licm\_20\_O1':

730,38 msec task-clock # 0,998 CPUs utilized

69 context-switches # 0,094 K/sec

0 cpu-migrations # 0,000 K/sec

2 061 page-faults # 0,003 M/sec

2 918 600 837 cycles # 3,996 GHz (83,03%)

1 259 149 stalled-cycles-frontend # 0,04% frontend cycles idle (83,03%)

2 386 518 543 stalled-cycles-backend # 81,77% backend cycles idle (83,57%)

4 501 653 388 instructions # 1,54 insn per cycle

# 0,53 stalled cycles per insn (83,58%)

102 078 449 branches # 139,761 M/sec (83,58%)

17 656 branch-misses # 0,02% of all branches (83,20%)

0,731927130 seconds time elapsed

0,726749000 seconds user

0,003993000 seconds sys

Performance counter stats for 'build/licm\_20\_O1\_no\_licm':

814,08 msec task-clock # 0,998 CPUs utilized

78 context-switches # 0,096 K/sec

1 cpu-migrations # 0,001 K/sec

2 061 page-faults # 0,003 M/sec

3 203 342 249 cycles # 3,935 GHz (83,30%)

1 781 651 stalled-cycles-frontend # 0,06% frontend cycles idle (83,32%)

2 485 572 880 stalled-cycles-backend # 77,59% backend cycles idle (83,31%)

8 118 525 246 instructions # 2,53 insn per cycle

# 0,31 stalled cycles per insn (83,30%)

101 738 530 branches # 124,974 M/sec (83,35%)

18 312 branch-misses # 0,02% of all branches (83,43%)

0,815309864 seconds time elapsed

0,814369000 seconds user

0,000000000 seconds sys

Performance counter stats for 'build/licm\_div\_5\_O0':

1 732,63 msec task-clock # 0,999 CPUs utilized

160 context-switches # 0,092 K/sec

0 cpu-migrations # 0,000 K/sec

604 page-faults # 0,349 K/sec

7 022 950 415 cycles # 4,053 GHz (83,23%)

1 123 061 stalled-cycles-frontend # 0,02% frontend cycles idle (83,39%)

6 615 891 785 stalled-cycles-backend # 94,20% backend cycles idle (83,39%)

3 205 940 111 instructions # 0,46 insn per cycle

# 2,06 stalled cycles per insn (83,39%)

201 477 768 branches # 116,285 M/sec (83,38%)

34 415 branch-misses # 0,02% of all branches (83,23%)

1,734715800 seconds time elapsed

1,733130000 seconds user

0,000000000 seconds sys

Performance counter stats for 'build/licm\_div\_5\_O1':

253,39 msec task-clock # 0,996 CPUs utilized

24 context-switches # 0,095 K/sec

0 cpu-migrations # 0,000 K/sec

605 page-faults # 0,002 M/sec

1 010 439 462 cycles # 3,988 GHz (82,64%)

269 542 stalled-cycles-frontend # 0,03% frontend cycles idle (82,65%)

798 627 428 stalled-cycles-backend # 79,04% backend cycles idle (83,26%)

1 496 006 898 instructions # 1,48 insn per cycle

# 0,53 stalled cycles per insn (84,22%)

100 435 682 branches # 396,371 M/sec (84,22%)

5 705 branch-misses # 0,01% of all branches (83,01%)

0,254385395 seconds time elapsed

0,253817000 seconds user

0,000000000 seconds sys

Performance counter stats for 'build/licm\_div\_5\_O1\_no\_licm':

1 737,25 msec task-clock # 0,999 CPUs utilized

161 context-switches # 0,093 K/sec

0 cpu-migrations # 0,000 K/sec

607 page-faults # 0,349 K/sec

7 018 895 568 cycles # 4,040 GHz (83,20%)

761 728 stalled-cycles-frontend # 0,01% frontend cycles idle (83,25%)

6 701 394 060 stalled-cycles-backend # 95,48% backend cycles idle (83,43%)

3 305 727 897 instructions # 0,47 insn per cycle

# 2,03 stalled cycles per insn (83,43%)

101 598 222 branches # 58,482 M/sec (83,43%)

20 370 branch-misses # 0,02% of all branches (83,26%)

1,738914322 seconds time elapsed

1,737844000 seconds user

0,000000000 seconds sys

Performance counter stats for 'build/licm\_div\_10\_O0':

3 459,51 msec task-clock # 0,999 CPUs utilized

318 context-switches # 0,092 K/sec

0 cpu-migrations # 0,000 K/sec

1 089 page-faults # 0,315 K/sec

14 025 489 196 cycles # 4,054 GHz (83,30%)

1 472 040 stalled-cycles-frontend # 0,01% frontend cycles idle (83,36%)

13 607 567 332 stalled-cycles-backend # 97,02% backend cycles idle (83,36%)

5 712 736 776 instructions # 0,41 insn per cycle

# 2,38 stalled cycles per insn (83,36%)

202 604 945 branches # 58,565 M/sec (83,36%)

31 364 branch-misses # 0,02% of all branches (83,27%)

3,462131265 seconds time elapsed

3,459745000 seconds user

0,000000000 seconds sys

Performance counter stats for 'build/licm\_div\_10\_O1':

345,51 msec task-clock # 0,997 CPUs utilized

33 context-switches # 0,096 K/sec

0 cpu-migrations # 0,000 K/sec

1 088 page-faults # 0,003 M/sec

1 375 277 484 cycles # 3,980 GHz (82,65%)

377 338 stalled-cycles-frontend # 0,03% frontend cycles idle (83,03%)

1 067 652 746 stalled-cycles-backend # 77,63% backend cycles idle (83,80%)

2 497 187 431 instructions # 1,82 insn per cycle

# 0,43 stalled cycles per insn (83,80%)

100 816 361 branches # 291,790 M/sec (83,80%)

8 640 branch-misses # 0,01% of all branches (82,92%)

0,346651274 seconds time elapsed

Performance counter stats for 'build/licm\_div\_10\_O1\_no\_licm':

3 457,64 msec task-clock # 0,999 CPUs utilized

319 context-switches # 0,092 K/sec

0 cpu-migrations # 0,000 K/sec

1 088 page-faults # 0,315 K/sec

14 029 189 425 cycles # 4,057 GHz (83,35%)

1 354 842 stalled-cycles-frontend # 0,01% frontend cycles idle (83,35%)

13 505 806 770 stalled-cycles-backend # 96,27% backend cycles idle (83,35%)

5 811 732 041 instructions # 0,41 insn per cycle

# 2,32 stalled cycles per insn (83,35%)

102 643 770 branches # 29,686 M/sec (83,35%)

33 215 branch-misses # 0,03% of all branches (83,26%)

3,460297546 seconds time elapsed

3,457768000 seconds user

0,000000000 seconds sys

Performance counter stats for 'build/licm\_div\_20\_O0':

6 928,77 msec task-clock # 0,999 CPUs utilized

637 context-switches # 0,092 K/sec

0 cpu-migrations # 0,000 K/sec

2 061 page-faults # 0,297 K/sec

28 069 066 465 cycles # 4,051 GHz (83,33%)

8 195 776 stalled-cycles-frontend # 0,03% frontend cycles idle (83,33%)

27 317 301 428 stalled-cycles-backend # 97,32% backend cycles idle (83,32%)

10 723 654 228 instructions # 0,38 insn per cycle

# 2,55 stalled cycles per insn (83,33%)

204 334 365 branches # 29,491 M/sec (83,34%)

66 030 branch-misses # 0,03% of all branches (83,35%)

6,934686529 seconds time elapsed

6,923285000 seconds user

0,003997000 seconds sys

Performance counter stats for 'build/licm\_div\_20\_O1':

710,60 msec task-clock # 0,999 CPUs utilized

71 context-switches # 0,100 K/sec

0 cpu-migrations # 0,000 K/sec

2 062 page-faults # 0,003 M/sec

2 823 232 480 cycles # 3,973 GHz (83,13%)

1 634 127 stalled-cycles-frontend # 0,06% frontend cycles idle (83,12%)

2 304 871 209 stalled-cycles-backend # 81,64% backend cycles idle (83,13%)

4 504 363 849 instructions # 1,60 insn per cycle

# 0,51 stalled cycles per insn (83,65%)

101 961 636 branches # 143,486 M/sec (83,69%)

16 016 branch-misses # 0,02% of all branches (83,28%)

0,711592650 seconds time elapsed

0,702796000 seconds user

0,007986000 seconds sys

Performance counter stats for 'build/licm\_div\_20\_O1\_no\_licm':

6 887,10 msec task-clock # 0,999 CPUs utilized

660 context-switches # 0,096 K/sec

0 cpu-migrations # 0,000 K/sec

2 062 page-faults # 0,299 K/sec

28 073 058 606 cycles # 4,076 GHz (83,33%)

7 844 098 stalled-cycles-frontend # 0,03% frontend cycles idle (83,34%)

26 898 452 988 stalled-cycles-backend # 95,82% backend cycles idle (83,34%)

10 421 196 177 instructions # 0,37 insn per cycle

# 2,58 stalled cycles per insn (83,34%)

104 781 888 branches # 15,214 M/sec (83,34%)

121 364 branch-misses # 0,12% of all branches (83,30%)

6,892377577 seconds time elapsed

6,881982000 seconds user

0,003996000 seconds sys