

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА(ЛЕНИНА)**  
**Кафедра Вычислительной техники**

**ОТЧЕТ**  
**По практической работе**  
**По дисциплине «Параллельные алгоритмы и системы»**

Выполнил

Студент гр.9891

Преподаватель

Палунин А.И.

Пазников А. А.

Санкт-Петербург

2022

### **Цель работы:**

1. Собрать тесты из пакета SPEC CPU 2017
2. Запустить тесты
3. Провести анализ эффективности с помощью утилиты perf (cache-misses);
4. Найти места в коде, где больше всего промахов.

### **Содержание:**

• Тест nab_r	3
• Тест deepsjeng	4
• Тест exchange2	7
• Тест cactuBSSN	9

Из пакета были скомпилированы следующие тесты:

- **nab\_r** (файл, запускающий тест находится в директории /src/nab\_s). Сборка была осуществлена с помощью скрипта находящегося в этой же папке: simple-build-nab-s-644.sh. Для запуска теста в аргументах командной строки указываем наименование тестовых файлов (они расположены в каталоге src/hkrdenq). Результат работы теста представлен на рисунке 1.

```
ivi@Mitsumi-Two:~/cpu2017/benchspec/CPU/544.nab_r/src$ ./nab_s hkrdenq hkrdenq
nabmd hkrdenq 0

Reading .pdb file (hkrdenq/hkrdenq.pdb)
Reading .prm file (hkrdenq/hkrdenq.prm)
title:

      iter      Total
ff:      0      -680.86
Initial energy is -680.8606420
Starting molecular dynamics with Born solvation energy...

      iter      Total
ff:      0      -680.86
ff: 1000      -647.08

...Done, md returns 0
      iter      Total
ff:      0      -200.56
Initial energy is -200.5567510
Starting molecular dynamics with in vacuo non-bonded energy...

      iter      Total
ff:      0      -200.56
ff: 1000      -477.69

...Done, md returns 0
ivi@Mitsumi-Two:~/cpu2017/benchspec/CPU/544.nab_r/src$
```

Рисунок 1. Запуск теста nab\_r

Результат работы утилиты perf представлен на рисунке 2. Наибольшее число промахов в данном тесте присутствует на команде `mov %eax, -0x21c(%rbp)`. Найдем в исходных файлах теста код, соответствующий этой команде – рисунок 3.

```
Samples: 126K of event 'cache-misses', 4000 Hz, Event count (approx.): 190883315
nabmd_omp_fn.0 /home/ivi/cpu2017/benchspec/CPU/544.nab_r/src/nab_s [Percent: local period]
Percent
      mov     -0x140(%rbp),%rax
      add     %rdx,%rax
      sub     -0x190(%rbp),%xmm0
      mov     %xmm0,(%rax)
      jmp     17fc
      if (pearlist[j] == NULL) {
0,27 1a74: mov     -0x204(%rbp),%eax
0,03      cltq
      lea     0x0(,%rax,8),%rdx
0,10      mov     -0x130(%rbp),%rax
0,13      add     %rdx,%rax
0,06      mov     (%rax),%rax
0,62      test    %rax,%rax
      j     28a5
      i = pearlist[j][k];
0,10 1a9a: mov     -0x204(%rbp),%eax
0,28      cltq
      lea     0x0(,%rax,8),%rdx
      mov     -0x130(%rbp),%rax
0,06      add     %rdx,%rax
0,20      mov     (%rax),%rax
0,09      mov     -0x210(%rbp),%edx
      movslq  %edx,%rdx
0,03      shl     $0x2,%rdx
0,14      add     %rdx,%rax
0,05      mov     (%rax),%eax
3,24      mov     %eax,-0x21c(%rbp)
      i34 = dim * i;
0,20      mov     dim,%eax
0,18      imul    -0x21c(%rbp),%eax
0,35      mov     %eax,-0x1e8(%rbp)
      iaci = prm->Ntypes * (prm->Iac[i] - 1);
```

Рисунок 2.

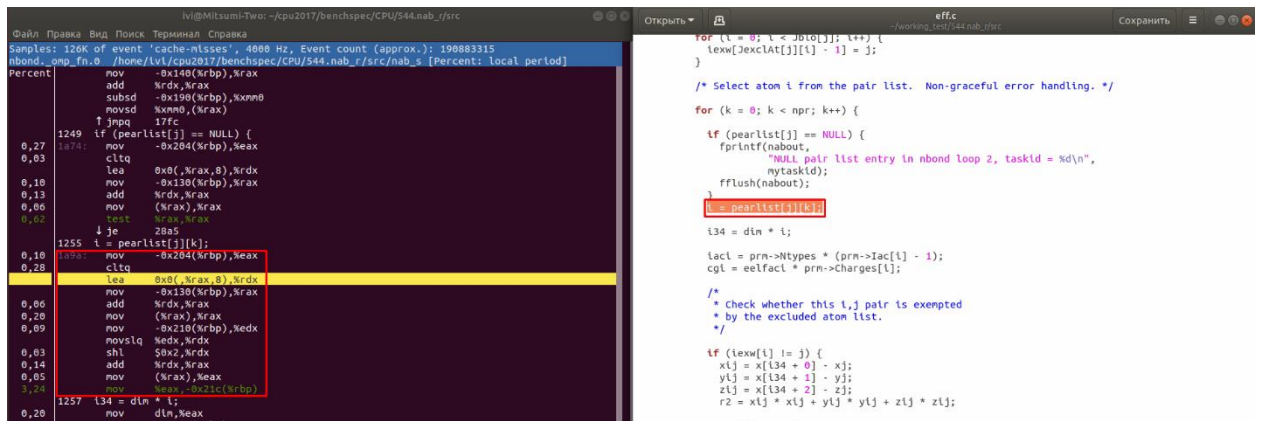


Рисунок 3.

Как видно из кода, промахи присутствуют в момент присвоения переменной  $i$  значения из массива `pearlist`.

Ещё промахи присутствуют при выполнении копирования одного участка памяти в виде двойных слов (DWORD) в другой, об этом нам говорит ассемблерная команда `movsd`. Она участвует в вычислении переменной  $r2inv = 1.0/r2$ . Рисунок 4.



Рисунок 4.

Больше в данном тесте не было найдено большого количества промахов.

Следующим был скомпилирован тест:

- **deepsjeng**. Сборка теста была осуществлена с использованием команды `gcc -g *.c -o test`. Запускающим файлом является файл `test`. В качестве аргумента при запуске тест принимает файл `test.txt` в котором содержится шахматная позиция и глубина анализа позиции. Результат работы теста представлен на рисунке 5.

Результат работы утилиты `perf` представлен на рисунке 6. Как видно из рисунка `cache-misses` чаще всего встречается в `search` и `qsearch`. Посмотрим на их ассемблерный код и найдем соответствующие участки кода в исходных файлах.

Сначала проверим `search_root` (рисунок 7). Как видим из ассемблерного кода проблемы возникают при изменении знака числа. Этому соответствует в исходном файле `search.cpp` функция `rootmovesearch()`.

```
Файл Правка Вид Поиск Терминал Справка
ivi@mitsumi-two:~/working_test/deepsjeng_r/src$ sudo perf record -g -e cache-misses ./test test.txt
[sudo] пароль для ivi:
Deep Sjeng version 3.2 SPEC, Copyright (C) 2000-2009 Gian-Carlo Pascutto
Allocated 15000000 hash entries, totalling 720000000 bytes.
Analyzing 15 plies...

+-----+
| *R | | | *Q | *K | | | *R |
+-----+
| *P | *P | *P | | *B | | *P | *P |
+-----+
| | | | *N | | *P | | | |
+-----+
| | | | *P | P | | *N | | |
+-----+
| | | | P | | | *B | | |
+-----+
| | | | P | B | | N | N | |
+-----+
| P | P | | | | | P | P |
+-----+
| R | | | B | Q | K | | | R |
+-----+
a b c d e f g h

Opening phase.
Time for move : 99999999
3 -43 0 750 0-0 ??
3 -18 0 1153 0-0 Nx3+ Rx3 Bx3 Qx3
3 -10 0 1733 Bx5 Bx5 h3 Bf4
3 -10 0 2353 Bx5 Bx5 h3 Bf4
4 -11 0 3161 Bx5 Bx5 h3 Bf4 hxg4 Bxg3+ Kd2
4 -11 0 5561 Bx5 Bx5 h3 Bf4 hxg4 Bxg3+ Kd2
5 1 0 8798 Bx5 Bx5 Qb3 Bx3 gx3 Qc8 Qa3 Be3
5 1 0 9543 Bx5 Bx5 Qb3 Bx3 gx3 Qc8 Qa3 Be3
6 31 0 11662 Bx5 !!
6 35 0 13203 Bx5 Bx5 h3 Bf4 hxg4 Bxg3+ Kd2 h6 Kc2 0-0
6 35 0 14196 Bx5 Bx5 h3 Bf4 hxg4 Bxg3+ Kd2 h6 Kc2 0-0
7 35 0 16333 Bx5 Bx5 h3 Bf4 hxg4 Bxg3+ Kd2 h6 Kc2 0-0
7 35 0 21839 Bx5 Bx5 h3 Bf4 hxg4 Bxg3+ Kd2 h6 Kc2 0-0
8 58 0 31092 Bx5 Bx5 h3 Bf4 hxg4 Bxg3+ Kd2 h6 Bg6+ Kd7 Qb3 Kc8 Kc2
8 58 0 33574 Bx5 Bx5 h3 Bf4 hxg4 Bxg3+ Kd2 h6 Bg6+ Kd7 Qb3 Kc8 Kc2
9 56 0 85090 Bx5 Bx5 h3 Bf4 hxg4 Bxg3+ Kd2 Qe7 Bg6+ Kd7 Bxh7 Bf4+ Kc2 Raf8 Qe2 Kc8 Qb5 Be3
9 56 0 92018 Bx5 Bx5 h3 Bf4 hxg4 Bxg3+ Kd2 Qe7 Bg6+ Kd7 Bxh7 Bf4+ Kc2 Raf8 Qe2 Kc8 Qb5 Be3
10 57 0 201398 Bx5 Bx5 h3 Bf4 hxg4 Bxg3+ Kd2 Qe7 Bg6+ Kd7 Bxh7 Raf8 Qb3 Kc8 Kc2 Rf4 Bb5 Bxh1 Bxh1 Bxg4
10 57 0 254780 Bx5 Bx5 h3 Bf4 hxg4 Bxg3+ Kd2 Qe7 Bg6+ Kd7 Bxh7 Raf8 Qb3 Kc8 Kc2 Rf4 Bb5 Bxh1 Bxh1 Bxg4
11 49 0 459068 Bx5 Bx5 h3 Bf4 hxg4 Bxg3+ Kd2 Qe7 Bg6+ Kd7 Bxh7 Raf8 Qb3 Kc8 Kc2 Qf7 Raf1 Qf4 Kb1 Qxg4
11 49 0 495980 Bx5 Bx5 h3 Bf4 hxg4 Bxg3+ Kd2 Qe7 Bg6+ Kd7 Bxh7 Raf8 Qb3 Kc8 Kc2 Qf7 Raf1 Qf4 Kb1 Qxg4
12 49 0 1247935 Bx5 Bx5 h3 Bx3 Qx3 Qe7 Qh5+ Kd7 h4 Bf4 0-0 g5 Ne2 Be3+ Kh1 Kc8 Rf7
12 49 0 1321354 Bx5 Bx5 h3 Bx3 Qx3 Qe7 Qh5+ Kd7 h4 Bf4 0-0 g5 Ne2 Be3+ Kh1 Kc8 Rf7
13 30 0 2441024 Bx5 Bx5 h3 Bx3 Qx3 Qe7 Qh5+ Kd7 h4 Bf4 Qq4 Bxg3+ Qxg3 Raf8 0-0-0 Kc8 Rdf1 g6 Bb5 Rx1+ Rx1 Rf8 Rxf
13 Qx8 Qb5 Qf2 Bxc6 bxc6
13 41 0 4073389 h3 Bx3 gx3 Nf7 f4 g6 Rg1 Qd7 Bd2 0-0-0 Qe2 Rde8 0-0-0 Rhf8 Kb1 a6
13 41 0 4074107 h3 Bx3 gx3 Nf7 f4 g6 Rg1 Qd7 Bd2 0-0-0 Qe2 Rde8 0-0-0 Rhf8 Kb1 a6
14 40 0 5455288 h3 Bx3 gx3 Nf7 f4 g6 Rg1 Qd7 Be3 0-0-0 Qe2 Rhf8 0-0-0 Rde8 Kb1 Kb8 Qf3
14 40 0 5681349 h3 Bx3 gx3 Nf7 f4 g6 Rg1 Qd7 Be3 0-0-0 Qe2 Rhf8 0-0-0 Rde8 Kb1 Kb8 Qf3
15 47 0 8915006 h3 Bx3 gx3 Nf7 f4 Bh4 Rg1 0-0 Be3 Nh6 Qf3 Nf5 Bx5 Rx5 0-0-0 Bxg3 Qxg3 g6 Kb1 Qe7 Rg2 Raf8
15 47 0 9568427 h3 Bx3 gx3 Nf7 f4 Bh4 Rg1 0-0 Be3 Nh6 Qf3 Nf5 Bx5 Rx5 0-0-0 Bxg3 Qxg3 g6 Kb1 Qe7 Rg2 Raf8

Nodes: 9568427 (65.27% qnodes)
[ perf record: Woken up 73 times to write data ]
[ perf record: Captured and wrote 18,489 MB perf.data (85982 samples) ]
```

Рисунок 5. Результат работы теста deepsjeng

```
ivi@mitsumi-two: ~/working_test/deepsjeng_r/src
Samples: 85K of event 'cache-misses', Event count (approx.): 198594439
Children Self Command Shared Object Symbol
+ 98,47% 0,00% test [unknown] [k] 0x4096258d4c544155
+ 98,47% 0,00% test libc-2.27.so [.] __libc_start_main
+ 98,47% 0,00% test test [.] main
+ 91,34% 0,00% test test [.] run_epd_testsuite
+ 85,47% 0,00% test test [.] think
+ 85,46% 0,01% test test [.] search_root
- 85,43% 5,37% test test [.] search
+ 80,06% search
- 5,37% 0x4096258d4c544155
__libc_start_main
main
run_epd_testsuite
think
search_root
rootmovesearch
+ search
+ 85,42% 0,00% test test [.] rootmovesearch
- 47,58% 5,38% test test [.] qsearch
+ 42,20% qsearch
- 5,38% 0x4096258d4c544155
__libc_start_main
main
run_epd_testsuite
think
search_root
rootmovesearch
search
+ search
- 26,55% 26,25% test test [.] ProbeTT
26,25% 0x4096258d4c544155
__libc_start_main
main
run_epd_testsuite
think
- search_root
- 26,25% rootmovesearch
- search
- 26,23% search
- 26,22% search
- 26,10% search
- 25,79% search
- 25,03% search
+ 24,14% search
0,53% qsearch
```

Рисунок 6. Результат работы утилиты perf report.

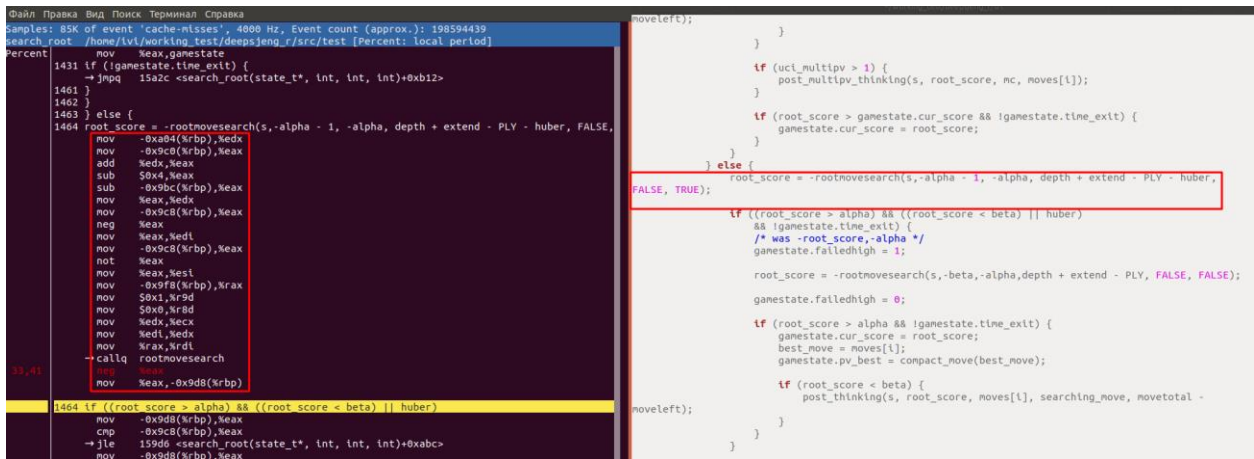


Рисунок 7.

На рисунке 8 видим проблему в том же файле `search.cpp` при работе цикла `for(i=0;i<MOVE_BUF;i++)`.

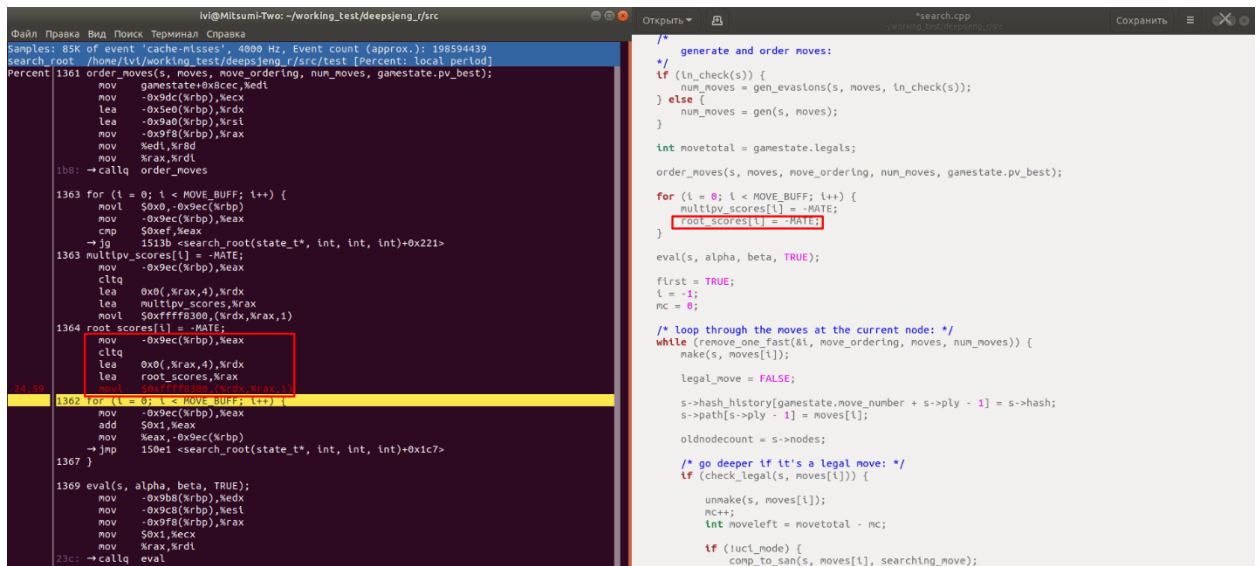


Рисунок 8.

Также в файле `search.cpp` присутствует проблема в функции `rootmovesearch()` в моменте присвоения переменной `res` значения из функции `search()`. Рисунок 9.

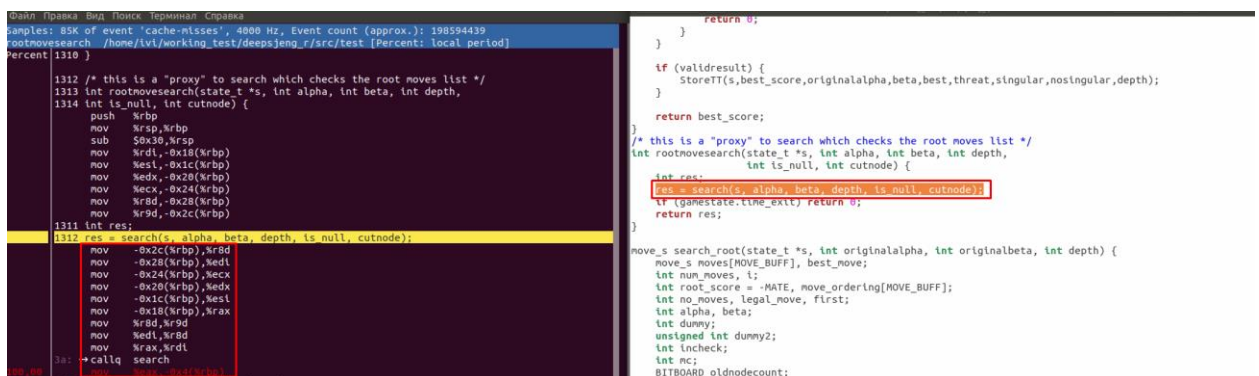


Рисунок 9.

Остальные cache-misses, которые видно на рисунке 6 находятся в файле search.cpp в функциях rootmovesearch() и search().

Следующим был скомпилирован тест:

- **exchange2\_r**. Тест был скомпилирован командой `gfortran -g *.F90 -o test`. Запускающим файлом является `test`. При запуске теста в качестве аргумента принимается файл (`puzzles.txt`), содержащий набор sudoku. Результат работы теста представлен на рисунке 10.

```
lv@mitsumi-two:~/working_test/exchange2_r/src$ sudo perf record -g -e cache-misses ./test puzzles.txt
Number of puzzles (0 - 54)? 1
Puzzle 1
0 1 0 0 0 0 0 0 2
3 0 0 0 0 4 0 0 0
0 0 0 0 5 0 0 0 0
0 0 0 0 0 3 0 7 0
0 0 0 0 4 0 5 0 0
0 2 0 7 0 0 0 0 0
0 0 5 0 2 0 0 0 0
0 0 0 8 0 0 0 0 3
7 0 0 0 0 0 0 4 0

At 5 5 change to 1, 8 4 change to 9
At 6 4 change to 4, 1 9 change to 9
At 6 2 change to 9, 1 2 change to 6
At 3 7 change to 7, 6 4 change to 6
At 3 5 change to 8, 5 7 change to 8
At 7 3 change to 8, 5 3 change to 5
At 7 5 change to 5, 7 3 change to 2
At 4 8 change to 2, 3 7 change to 2
At 2 6 change to 7, 5 5 change to 7
0 0 0 0 0 0 0 0 9
3 0 0 0 0 7 0 0 0
0 0 0 0 8 0 2 0 0
0 0 0 0 0 3 0 2 0
0 0 5 0 7 0 8 0 0
0 9 0 0 0 0 0 0 0
0 0 2 0 5 0 0 0 0
0 0 0 9 0 0 0 0 3
7 0 0 0 0 0 0 4 0

At 5 3 change to 4, 3 7 change to 5
At 3 7 change to 1, 9 8 change to 8
At 8 4 change to 1, 7 5 change to 9
At 8 9 change to 6, 8 4 change to 4
At 7 3 change to 1, 4 6 change to 1
At 4 6 change to 3, 2 1 change to 1
At 4 8 change to 1, 2 1 change to 2
0 0 0 0 0 0 0 0 9
2 0 0 0 0 7 0 0 0
0 0 0 0 8 0 1 0 0
0 0 0 0 0 3 0 1 0
0 0 4 0 7 0 8 0 0
0 9 0 0 0 0 0 0 0
0 0 1 0 9 0 0 0 0
0 0 0 4 0 0 0 0 0
7 0 0 0 0 0 0 8 0

At 3 7 change to 3, 7 3 change to 3
At 8 4 change to 2, 4 6 change to 4
At 7 3 change to 5, 4 8 change to 5
At 4 8 change to 3, 3 7 change to 4
0 0 0 0 0 0 0 0 9
2 0 0 0 0 7 0 0 0
0 0 0 0 8 0 4 0 0
0 0 0 0 0 4 0 3 0
0 0 4 0 7 0 8 0 0
0 9 0 0 0 0 0 0 0
0 0 5 0 9 0 0 0 0
0 0 0 2 0 0 0 0 0
7 0 0 0 0 0 0 8 0

Test
0 0 0 0 0 0 0 0 9
2 0 0 0 0 7 0 0 0
0 0 0 0 8 0 4 0 0
0 0 0 0 0 4 0 3 0
0 0 4 0 7 0 8 0 0
0 9 0 0 0 0 0 0 0
0 0 5 0 9 0 0 0 0
0 0 0 2 0 0 0 0 0
7 0 0 0 0 0 0 8 0

Puzzle count: changes: 1438
Elapsed time: 159.17
[ perf record: Woken up 79 times to write data ]
[ perf record: Captured and wrote 28,874 MB perf.data (147138 samples) ]
lv@mitsumi-two:~/working_test/exchange2_r/src$
```

Рисунок 10. Результат работы теста exchange2\_r

На рисунке 11 изображен отчёт, полученный в результате работы утилиты perf. Исходя из отчёта можно сделать вывод, что чаще всего cache-misses встречается в `__brute_force_MOD_digits_2`. Посмотрим аннотированный код и найдем соответствующие участки кода в исходных файлах.

[illegible]

Рисунок 11. Отчёт perf.

В данном тесте нет серьезных cache-misses, исходя из аннотированного кода (рисунок 12). Небольшие cache-misses встречаются в модуле brute\_force в файле exchange2.F90



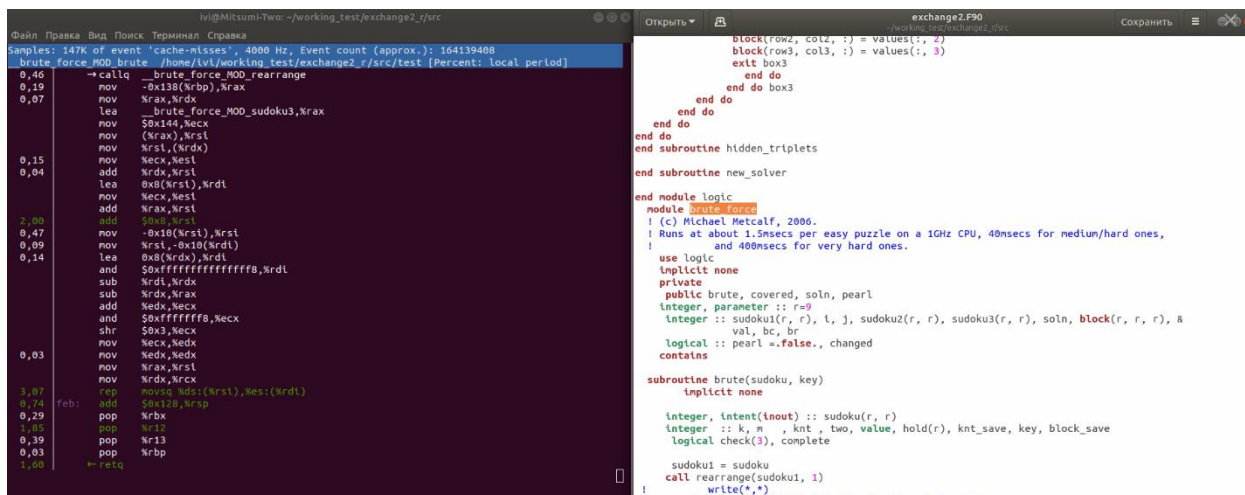


Рисунок 12. Аннотированный код и модуль brute\_force.

Следующим был скомпилирован тест:

- **cactuBSSN**. Компиляция выполнялась утилитой `simple-build-cactuBSSN_s-607.sh`. Запускающий файл `cactuBSSN_s`. При запуске, в качестве аргумента указываем файл `спес_test.par`. На рисунке 13 изображен результат работы данного теста.

Далее запустим утилиту `perf report`, чтобы увидеть где есть `cache-misses`. На рисунке 14 изображен `report` сформированный утилитой `perf`.

Исходя из отчёта в тесте чаще всего встречаются `cache-misses` в `or_real_update_2._omp_fn.6` и в `GOMP_parallel`, но `GOMP_parallel` является сторонней библиотекой (GNU libgomp) и поэтому её рассматривать не будет. Посмотрим на аннотированный код `or_real_update_2._omp_fn.6` рисунок 15.

Найдем файл, в котором присутствуют `cache-misses`. Для этого будем использовать утилиту `grep` с аргументом рекурсивного поиска по каталогу исходных файлов теста. Файл в котором присутствует проблема `Operators.c`. Промахи кеша случаются при вычислении значения по формуле  $scale * varptr[i] + fact0 * srcptr[i] + fact1 * srcptr[i]$  и записи этого значения в массив `varptr[]` (Рисунок 15).

```
ivi@Mitsumi-Two: ~/working_test/cactuBSSN/src
Файл Правка Вид Поиск Терминал Справка
1 0101 *****
01 1010 10 The Cactus Code V4.2.1
1010 1101 011 www.cactuscode.org
1001 100101 *****
00010101
100011 (c) Copyright The Authors
0100 GNU Licensed. No Warranty
0101
-----
Cactus version: 4.2.1
Parameter File: spec_test.par
-----
Activating thorn Cactus...Success -> active implementation Cactus
Activation requested for
-->PUGH PUGHInterp PUGHReduce PUGSlab Boundary CartGrid3D CoordBase LocalReduce SymBase ADMBase ADMCoupling ADMMacros CoordGauge StaticConformal Time Mol ML_BSSN ML_BSSN_Helper GaugeWave IOBasic IOASCII IOUtil <---
Thorn GaugeWave requests automatic activation of GenericFO
Activating thorn ADMBase...Success -> active implementation ADMBase
Activating thorn ADMCoupling...Success -> active implementation ADMCoupling
Activating thorn ADMMacros...Success -> active implementation ADMMacros
Activating thorn Boundary...Success -> active implementation boundary
Activating thorn CartGrid3D...Success -> active implementation grid
Activating thorn CoordBase...Success -> active implementation CoordBase
Activating thorn CoordGauge...Success -> active implementation CoordGauge
Activating thorn GaugeWave...Success -> active implementation GaugeWave
Activating thorn GenericFO...Success -> active implementation GenericFO
Activating thorn IOASCII...Success -> active implementation IOASCII
Activating thorn IOBasic...Success -> active implementation IOBasic
Activating thorn IOUtil...Success -> active implementation IOUtil
Activating thorn LocalReduce...Success -> active implementation LocalReduce
Activating thorn ML_BSSN...Success -> active implementation ML_BSSN
Activating thorn ML_BSSN_Helper...Success -> active implementation ML_BSSN_Helper
Activating thorn Mol...Success -> active implementation MethodOfLines
Activating thorn PUGH...Success -> active implementation Driver
Activating thorn PUGHInterp...Success -> active implementation Interp
Activating thorn PUGHReduce...Success -> active implementation Reduce
Activating thorn PUGSlab...Success -> active implementation HyperSlab
Activating thorn StaticConformal...Success -> active implementation StaticConformal
Activating thorn SymBase...Success -> active implementation SymBase
Activating thorn Time...Success -> active implementation time
INFO (PUGH): Using physical to logical mappings: direct
INFO (PUGH): Using topology generator: automatic
-----
GaugeWave
-----
Driver provided by PUGH
-----
ML_BSSN
-----
Mol: Generalized time integration.
-----
INFO (PUGH): Not setting up a topology for 1 dimensions
INFO (PUGH): Not setting up a topology for 2 dimensions
INFO (PUGH): Setting up a topology for 3 dimensions
INFO (IOASCII): I/O Method 'IOASCII_1D' registered: output of 1D lines of grid functions/arrays to ASCII files
INFO (IOASCII): Periodic 1D output every 1 iterations
INFO (IOASCII): Periodic 1D output requested for 'ADMBase::gpx', 'ADMBase::gky'
INFO (IOASCII): I/O Method 'IOASCII_2D' registered: output of 2D planes of grid functions/arrays to ASCII files
INFO (IOASCII): Periodic 2D output turned off
INFO (IOASCII): I/O Method 'IOASCII_3D' registered: output of 3D grid functions/arrays to ASCII files
INFO (IOASCII): Periodic 3D output turned off
INFO (IOBasic): I/O Method 'Scalar' registered: output of scalar quantities (grid scalars, reductions) to ASCII files
INFO (IOBasic): I/O Method 'Info' registered: output of scalar quantities (grid scalars, reductions) to screen
INFO (Mol): Using Runge-Kutta 3 as the time integrator.
INFO (Mol): The maximum number of evolved variables is 25. 25 are registered.
INFO (Mol): The maximum number of constrained variables is 20. 20 are registered.
INFO (Mol): The maximum number of SandR variables is 0. 0 are registered.
INFO (Mol): The maximum number of evolved complex variables is 0. 0 are registered.
INFO (Mol): The maximum number of constrained complex variables is 0. 0 are registered.
INFO (Mol): The maximum number of SandR complex variables is 0. 0 are registered.
INFO (Mol): The maximum number of evolved array variables is 0. 0 are registered.
INFO (Mol): The maximum number of constrained array variables is 0. 0 are registered.
INFO (Mol): The maximum number of SandR array variables is 0. 0 are registered.
INFO (Mol): The maximum number of evolved complex array variables is 0. 0 are registered.
INFO (Mol): The maximum number of constrained complex array variables is 0. 0 are registered.
INFO (Mol): The maximum size of any array variables is 0.
INFO (ADMMacros): Spatial finite differencing order: 2
INFO (CartGrid3D): Grid Settings:
INFO (CartGrid3D): dx=>1.00000000e-01 dy=>1.00000000e-01 dz=>1.00000000e-01
INFO (CartGrid3D): Computational Coordinates:
INFO (CartGrid3D): x=[-0.200, 3.900] y=[-0.200, 3.900] z=[-0.200, 3.900]
INFO (CartGrid3D): Indices of Physical Coordinates:
INFO (CartGrid3D): x=[0.41] y=[0.41] z=[0.41]
INFO (PUGH): Single processor evolution
INFO (PUGH): 3-dimensional grid functions
INFO (PUGH): Size: 42 42 42
INFO (Time): Timestep set to 0.025 (courant_static)
INFO (IOBasic): Periodic info output requested for 'ADMBase::gpx', 'ADMBase::gky'
-----
| t | | ADMBase::gpx | | ADMBase::gky |
| | | minimum | maximum | minimum | maximum |
-----
0 | 0.000 | 0.90829812 | 1.03170188 | -0.03170188 | 0.03170188 |
1 | 0.025 | 0.90581902 | 1.03402338 | -0.03354010 | 0.03342008 |
2 | 0.050 | 0.90403574 | 1.03499120 | -0.03454845 | 0.03409070 |
3 | 0.075 | 0.90189543 | 1.04025080 | -0.04273450 | 0.04115023 |
4 | 0.100 | 0.95035044 | 1.04479427 | -0.05083347 | 0.04823882 |
5 | 0.125 | 0.94939011 | 1.05313010 | -0.05740441 | 0.05438592 |
6 | 0.150 | 0.92411415 | 1.07117441 | -0.06090247 | 0.05251598 |
7 | 0.175 | 0.92115474 | 1.06939575 | -0.06041808 | 0.05847100 |
8 | 0.200 | 0.89810333 | 1.10571741 | -0.08037250 | 0.06099347 |
9 | 0.225 | 0.88509800 | 1.13039324 | -0.09951101 | 0.07805237 |
10 | 0.250 | 0.80009001 | 1.14205740 | -0.10022732 | 0.09204240 |
-----
Done.
[ perf record: Woken up 49 times to write data ]
[ perf record: Captured and wrote 9,468 MB perf.data (102290 samples) ]
ivi@Mitsumi-Two:~/working_test/cactuBSSN/src$
```

Рисунок 13. Результат работы теста cactuBSSN

ivi@Mitsumi-Two: ~/working_test/cactuBSSN/src					
Файл Правка Вид Поиск Терминал Справка					
Samples: 102K of event 'cache-misses', Event count (approx.): 348847239					
Children	Self	Command	Shared Object	Symbol	
+ 74,75% 0,00% cactuBSSN_s [unknown] [.] 0000000000000000					
+ 59,48% 0,18% cactuBSSN_s libgomp.so.1.0.0 [.] omp_in_final					
- 22,03% 22,51% cactuBSSN_s cactuBSSN_s [.] ML_BSSN_Advect_Body					
- 16,88% 0					
omp_in_final					
ML_BSSN_Advect_Body					
- 5,63% 0x2a					
+ 2,27% 0x7f64544b1010					
+ 1,69% 0x7f64545d3010					
+ 1,67% 0x7f6454542010					
COMP_parallel					
ML_BSSN_Advect_Body					
+ 21,14% 0,02% cactuBSSN_s libgomp.so.1.0.0 [.] COMP_parallel					
+ 10,40% 0,00% cactuBSSN_s [unknown] [k] 0x00007f64544b1010					
+ 15,02% 15,14% cactuBSSN_s cactuBSSN_s [.] ML_BSSN_RHS_Body					
- 11,34% 0					
omp_in_final					
ML_BSSN_RHS_Body					
- 3,80% 0x2a					
- 1,51% 0x7f64544b1010					
COMP_parallel					
ML_BSSN_RHS_Body					
- 1,15% 0x7f64545d3010					
COMP_parallel					
ML_BSSN_RHS_Body					
+ 1,14% 0x7f6454542010					
+ 15,39% 0,67% cactuBSSN_s libgomp.so.1.0.0 [.] omp_get_num_procs					
+ 12,93% 0,44% cactuBSSN_s [kernel.kallsyms] [k] apic_timer_interrupt					
+ 12,37% 0,36% cactuBSSN_s [kernel.kallsyms] [k] smp_apic_timer_interrupt					
+ 10,14% 0,62% cactuBSSN_s [kernel.kallsyms] [k] hrtimer_interrupt					
+ 8,82% 0,47% cactuBSSN_s [kernel.kallsyms] [k] __hrtimer_run_queues					
+ 8,03% 7,83% cactuBSSN_s cactuBSSN_s [.] ML_BSSN_convertToADMBaseDtLapseShift_Bod					
+ 7,45% 0,11% cactuBSSN_s [kernel.kallsyms] [k] tick_sched_timer					
+ 6,65% 0,60% cactuBSSN_s cactuBSSN_s [.] op_real_update_2_omp_fn.6					
- 5,63% 0					
- 4,96% omp_in_final					
op_real_update_2_omp_fn.6					
+ 6,18% 0,02% cactuBSSN_s [kernel.kallsyms] [k] tick_sched_handle					
+ 6,08% 0,02% cactuBSSN_s [kernel.kallsyms] [k] update_process_times					
+ 5,35% 0,00% cactuBSSN_s [unknown] [.] 0x00007f64544b1010					
+ 4,53% 4,38% cactuBSSN_s cactuBSSN_s [.] ML_BSSN_constraints_Body					
+ 4,20% 0,00% cactuBSSN_s [unknown] [k] 0x00007f64545d3010					
+ 4,14% 0,24% cactuBSSN_s [kernel.kallsyms] [k] scheduler_tick					
+ 4,12% 4,05% cactuBSSN_s cactuBSSN_s [.] ML_BSSN_convertToADMBaseDtLapseShiftBoun					
+ 4,10% 0,00% cactuBSSN_s [unknown] [.] 0xc4a6258d4c544155					
+ 4,10% 0,00% cactuBSSN_s libc-2.27.so [.] __libc_start_main					
+ 4,10% 0,00% cactuBSSN_s cactuBSSN_s [.] main					
+ 3,97% 0,00% cactuBSSN_s [unknown] [k] 0x00007f6454542010					
+ 3,78% 0,00% cactuBSSN_s cactuBSSN_s [.] PUGH_Evolve					
+ 3,75% 0,00% cactuBSSN_s cactuBSSN_s [.] CCTK_Traverse					
+ 3,75% 3,68% cactuBSSN_s cactuBSSN_s [.] op_real_set_0_omp_fn.0					
+ 2,92% 0,00% cactuBSSN_s cactuBSSN_s [.] CCTKi_ScheduleTraverseGHExtensions					

Рисунок 14. report сформированной утилитой perf

ivi@Mitsumi-Two: ~/working_test/cactuBSSN/src	
Файл Правка Вид Поиск Терминал Справка	
Samples: 102K of event 'cache-misses', 1000 Hz, Event count (approx.): 348847239	
op_real_update_2_omp_fn.6: /home/ivi/working_test/cactuBSSN/src/cactuBSSN_s (Percent: local period)	
130	mov rax, 0x00000000
131	mov rax, 0x00000000
132	mov rax, 0x00000000
133	mov rax, 0x00000000
134	mov rax, 0x00000000
135	mov rax, 0x00000000
136	mov rax, 0x00000000
137	mov rax, 0x00000000
138	mov rax, 0x00000000
139	mov rax, 0x00000000
140	mov rax, 0x00000000
141	mov rax, 0x00000000
142	mov rax, 0x00000000
143	mov rax, 0x00000000
144	mov rax, 0x00000000
145	mov rax, 0x00000000
146	mov rax, 0x00000000
147	mov rax, 0x00000000
148	mov rax, 0x00000000
149	mov rax, 0x00000000
150	mov rax, 0x00000000
151	mov rax, 0x00000000
152	mov rax, 0x00000000
153	mov rax, 0x00000000
154	mov rax, 0x00000000
155	mov rax, 0x00000000
156	mov rax, 0x00000000
157	mov rax, 0x00000000
158	mov rax, 0x00000000
159	mov rax, 0x00000000
160	mov rax, 0x00000000
161	mov rax, 0x00000000
162	mov rax, 0x00000000
163	mov rax, 0x00000000
164	mov rax, 0x00000000
165	mov rax, 0x00000000
166	mov rax, 0x00000000
167	mov rax, 0x00000000
168	mov rax, 0x00000000
169	mov rax, 0x00000000
170	mov rax, 0x00000000
171	mov rax, 0x00000000
172	mov rax, 0x00000000
173	mov rax, 0x00000000
174	mov rax, 0x00000000
175	mov rax, 0x00000000
176	mov rax, 0x00000000
177	mov rax, 0x00000000
178	mov rax, 0x00000000
179	mov rax, 0x00000000
180	mov rax, 0x00000000
181	mov rax, 0x00000000
182	mov rax, 0x00000000
183	mov rax, 0x00000000
184	mov rax, 0x00000000
185	mov rax, 0x00000000
186	mov rax, 0x00000000
187	mov rax, 0x00000000
188	mov rax, 0x00000000
189	mov rax, 0x00000000
190	mov rax, 0x00000000
191	mov rax, 0x00000000
192	mov rax, 0x00000000
193	mov rax, 0x00000000
194	mov rax, 0x00000000
195	mov rax, 0x00000000
196	mov rax, 0x00000000
197	mov rax, 0x00000000
198	mov rax, 0x00000000
199	mov rax, 0x00000000
200	mov rax, 0x00000000
201	mov rax, 0x00000000
202	mov rax, 0x00000000
203	mov rax, 0x00000000
204	mov rax, 0x00000000
205	mov rax, 0x00000000
206	mov rax, 0x00000000
207	mov rax, 0x00000000
208	mov rax, 0x00000000
209	mov rax, 0x00000000
210	mov rax, 0x00000000
211	mov rax, 0x00000000
212	mov rax, 0x00000000
213	mov rax, 0x00000000
214	mov rax, 0x00000000
215	mov rax, 0x00000000
216	mov rax, 0x00000000
217	mov rax, 0x00000000
218	mov rax, 0x00000000
219	mov rax, 0x00000000
220	mov rax, 0x00000000
221	mov rax, 0x00000000
222	mov rax, 0x00000000
223	mov rax, 0x00000000
224	mov rax, 0x00000000
225	mov rax, 0x00000000
226	mov rax, 0x00000000
227	mov rax, 0x00000000
228	mov rax, 0x00000000
229	mov rax, 0x00000000
230	mov rax, 0x00000000
231	mov rax, 0x00000000
232	mov rax, 0x00000000
233	mov rax, 0x00000000
234	mov rax, 0x00000000
235	mov rax, 0x00000000
236	mov rax, 0x00000000
237	mov rax, 0x00000000
238	mov rax, 0x00000000
239	mov rax, 0x00000000
240	mov rax, 0x00000000
241	mov rax, 0x00000000
242	mov rax, 0x00000000
243	mov rax, 0x00000000
244	mov rax, 0x00000000
245	mov rax, 0x00000000
246	mov rax, 0x00000000
247	mov rax, 0x00000000
248	mov rax, 0x00000000
249	mov rax, 0x00000000
250	mov rax, 0x00000000
251	mov rax, 0x00000000
252	mov rax, 0x00000000
253	mov rax, 0x00000000
254	mov rax, 0x00000000
255	mov rax, 0x00000000
256	mov rax, 0x00000000
257	mov rax, 0x00000000
258	mov rax, 0x00000000
259	mov rax, 0x00000000
260	mov rax, 0x00000000
261	mov rax, 0x00000000
262	mov rax, 0x00000000
263	mov rax, 0x00000000
264	mov rax, 0x00000000
265	mov rax, 0x00000000
266	mov rax, 0x00000000
267	mov rax, 0x00000000
268	mov rax, 0x00000000
269	mov rax, 0x00000000
270	mov rax, 0x00000000
271	mov rax, 0x00000000
272	mov rax, 0x00000000
273	mov rax, 0x00000000
274	mov rax, 0x00000000
275	mov rax, 0x00000000
276	mov rax, 0x00000000
277	mov rax, 0x00000000
278	mov rax, 0x00000000
279	mov rax, 0x00000000
280	mov rax, 0x00000000
281	mov rax, 0x00000000
282	mov rax, 0x00000000
283	mov rax, 0x00000000
284	mov rax, 0x00000000
285	mov rax, 0x00000000
286	mov rax, 0x00000000
287	mov rax, 0x00000000
288	mov rax, 0x00000000
289	mov rax, 0x00000000
290	mov rax, 0x00000000
291	mov rax, 0x00000000
292	mov rax, 0x00000000
293	mov rax, 0x00000000
294	mov rax, 0x00000000
295	mov rax, 0x00000000
296	mov rax, 0x00000000
297	mov rax, 0x00000000
298	mov rax, 0x00000000
299	mov rax, 0x00000000
300	mov rax, 0x00000000
301	mov rax, 0x00000000
302	mov rax, 0x00000000
303	mov rax, 0x00000000
304	mov rax, 0x00000000
305	mov rax, 0x00000000
306	mov rax, 0x00000000
307	mov rax, 0x00000000
308	mov rax, 0x00000000
309	mov rax, 0x00000000
310	mov rax, 0x00000000
311	mov rax, 0x00000000
312	mov rax, 0x00000000
313	mov rax, 0x00000000
314	mov rax, 0x00000000
315	mov rax, 0x00000000
316	mov rax, 0x00000000
317	mov rax, 0x00000000
318	mov rax, 0x00000000
319	mov rax, 0x00000000
320	mov rax, 0x00000000
321	mov rax, 0x00000000
322	mov rax, 0x00000000
323	mov rax, 0x00000000
324	mov rax, 0x00000000
325	mov rax, 0x00000000
326	mov rax, 0x00000000
327	mov rax, 0x00000000
328	mov rax, 0x00000000
329	mov rax, 0x00000000
330	mov rax, 0x00000000
331	mov rax, 0x00000000
332	mov rax, 0x00000000
333	mov rax, 0x00000000
334	mov rax, 0x00000000
335	mov rax, 0x00000000
336	mov rax, 0x00000000
337	mov rax, 0x00000000
338	mov rax, 0x00000000
339	mov rax, 0x00000000
340	mov rax, 0x00000000
341	mov rax, 0x00000000
342	mov rax, 0x00000000
343	mov rax, 0x00000000
344	mov rax, 0x00000000
345	mov rax, 0x00000000
346	mov rax, 0x00000000
347	mov rax, 0x00000000
348	mov rax, 0x00000000
349	mov rax, 0x00000000
350	mov rax, 0x00000000
351	mov rax, 0x0