TCCM lectures – Advanced Computational Techniques

Peter Reinhardt

Laboratoire de Chimie Théorique, Sorbonne Université, 75252 Paris CEDEX 05,

Peter.Reinhardt@upmc.fr

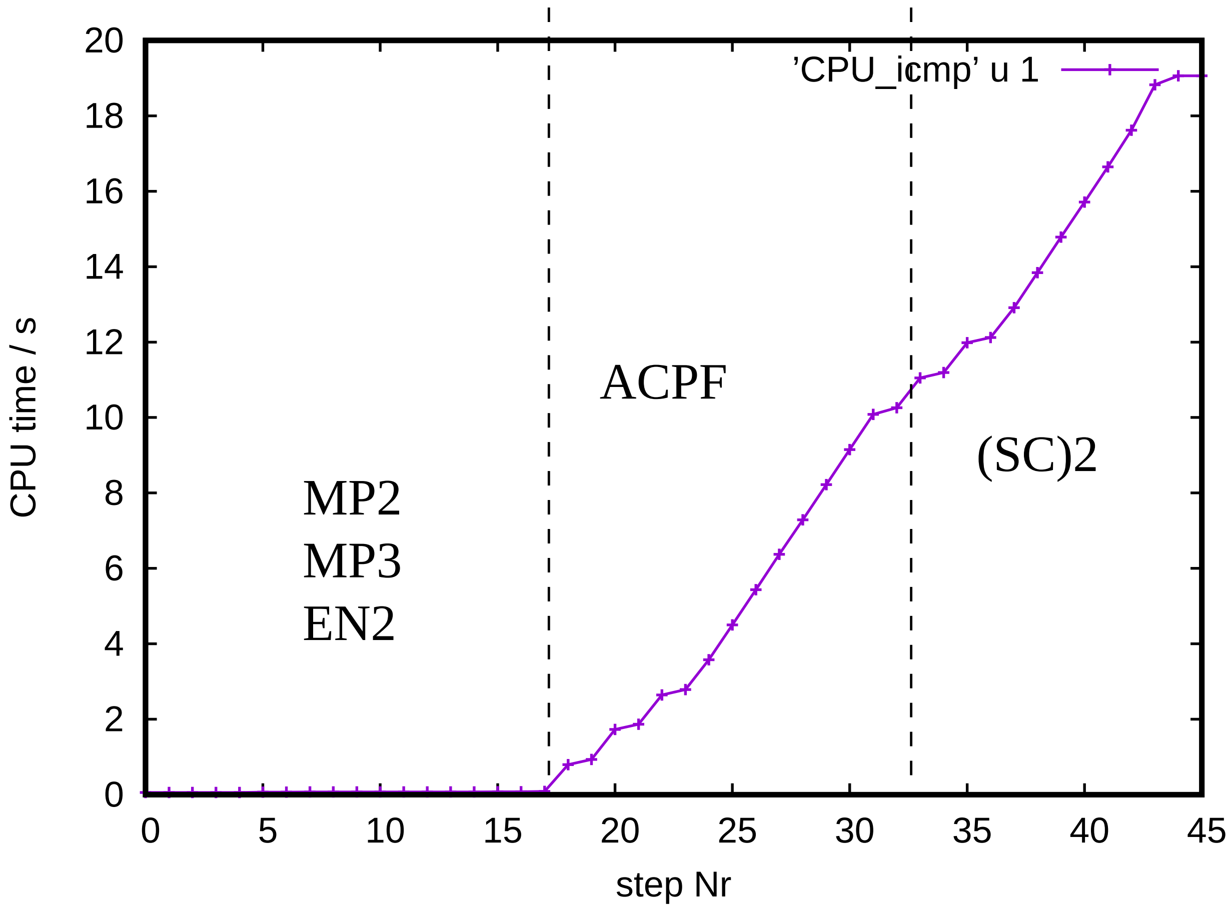
Wednesday morning — I

- Profiling of code

- Example of a water molecule in a DZP basis set (41 basis functions)

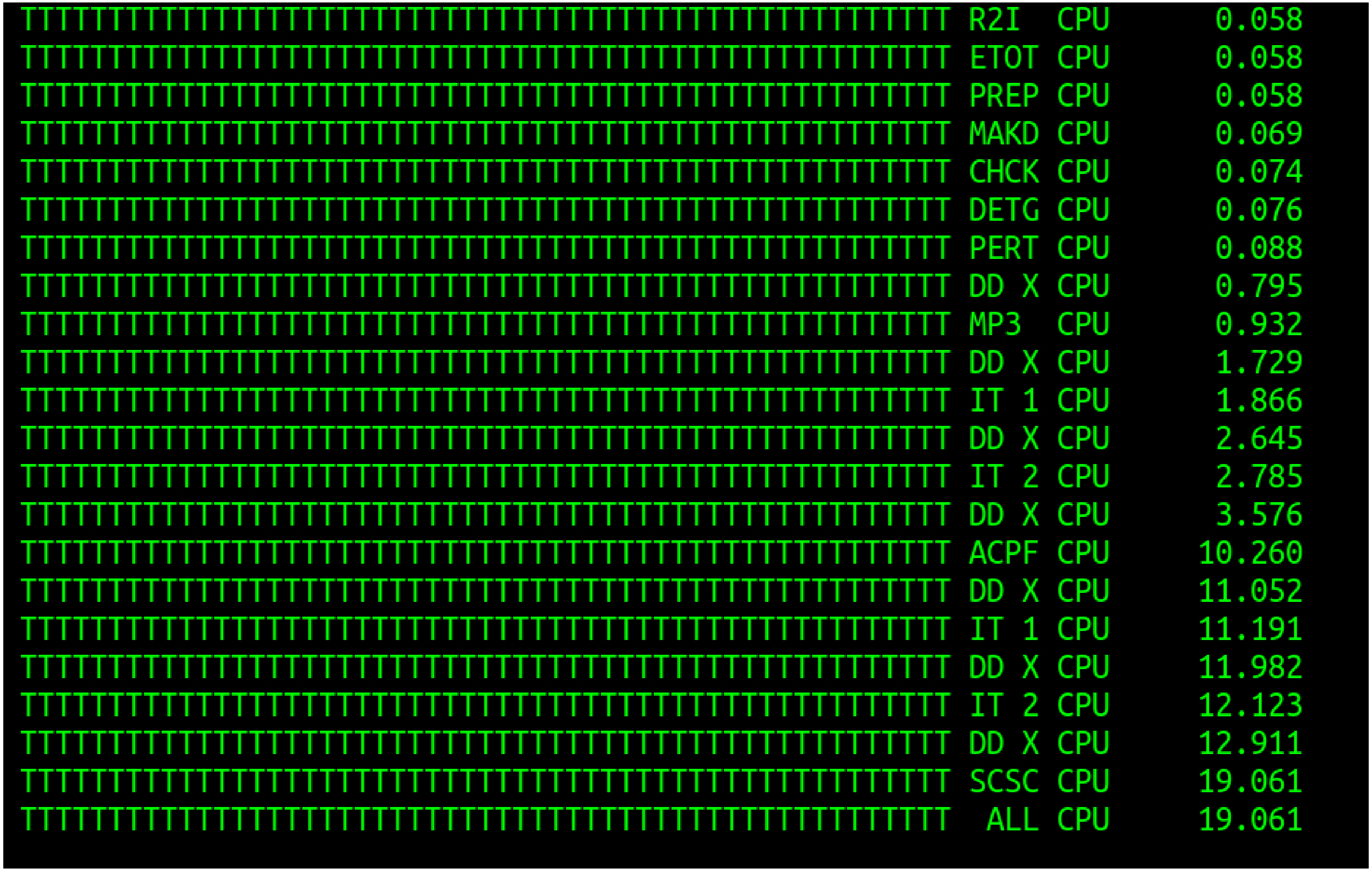
Profiling

Look where your program consumes ressources



Timer breaks

Profiling

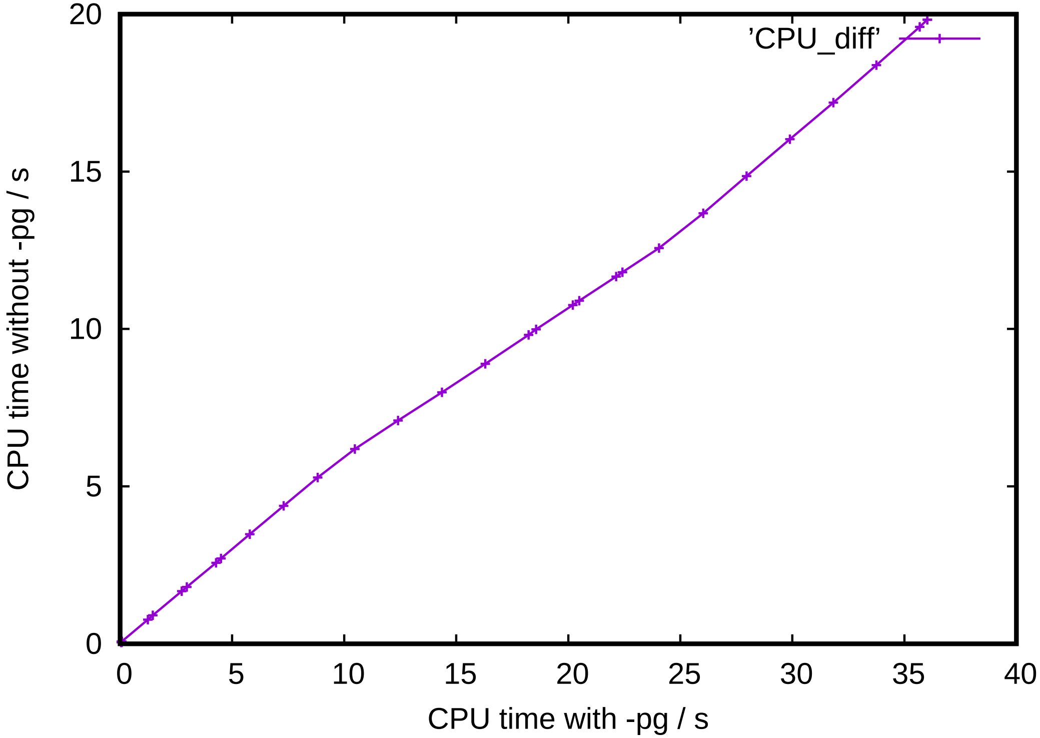


```
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT R2I  CPU      0.058
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT ETOT CPU      0.058
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT PREP CPU      0.058
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT MAKD CPU      0.069
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT CHCK CPU      0.074
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT DETG CPU      0.076
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT PERT CPU      0.088
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT DD X CPU      0.795
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT MP3  CPU      0.932
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT DD X CPU      1.729
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT IT 1 CPU      1.866
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT DD X CPU      2.645
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT IT 2 CPU      2.785
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT DD X CPU      3.576
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT ACPF CPU     10.260
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT DD X CPU     11.052
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT IT 1 CPU     11.191
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT DD X CPU     11.982
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT IT 2 CPU     12.123
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT DD X CPU     12.911
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT SCSC CPU     19.061
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT  ALL CPU     19.061
```

Profiling

But where is the effort spent?

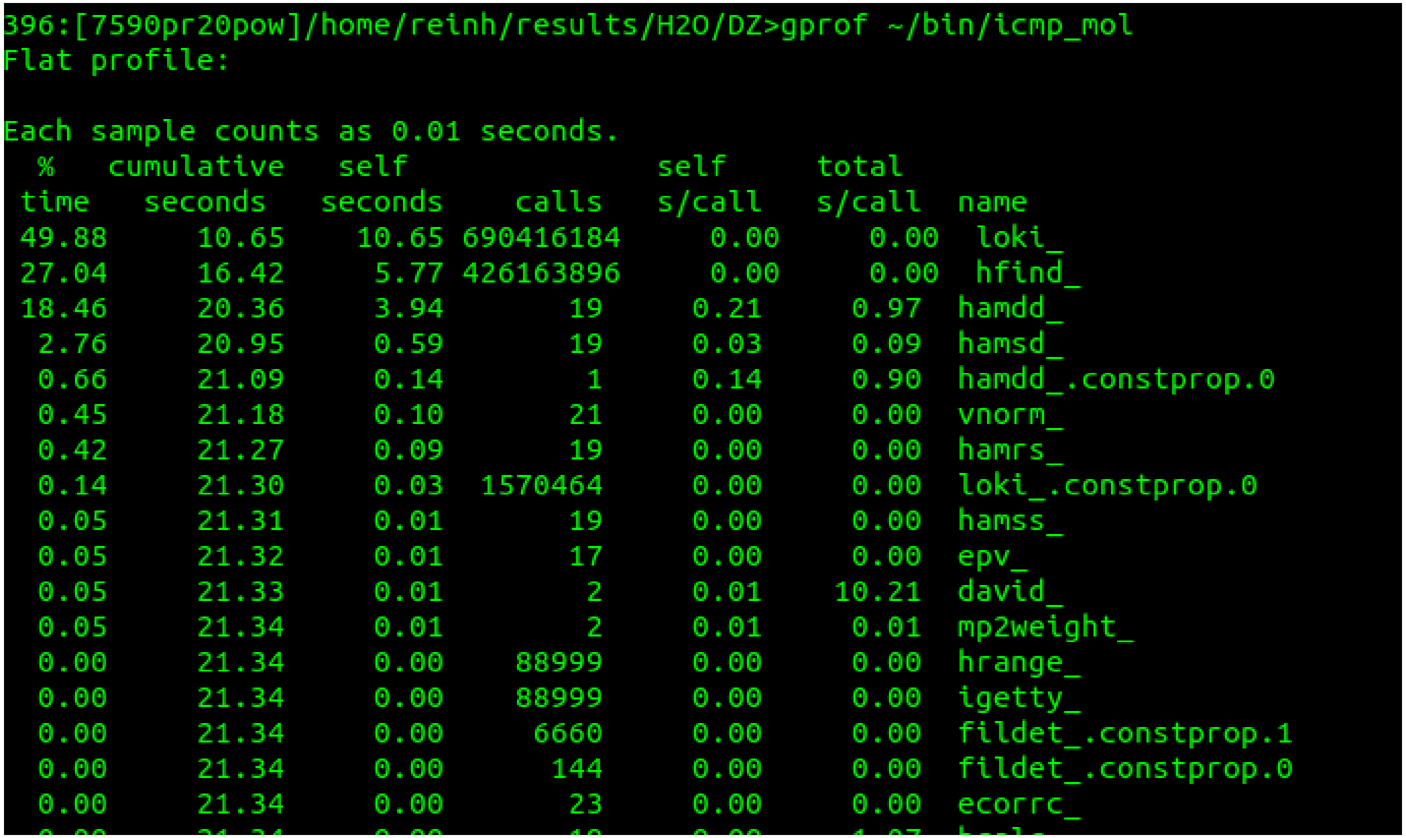
- Compile source with `-g` option for inserting debugging labels
- Compile source with `-pg` option for inserting profiling labels
- run your program
- slows code down significantly



Profiling

But where is the effort spent?

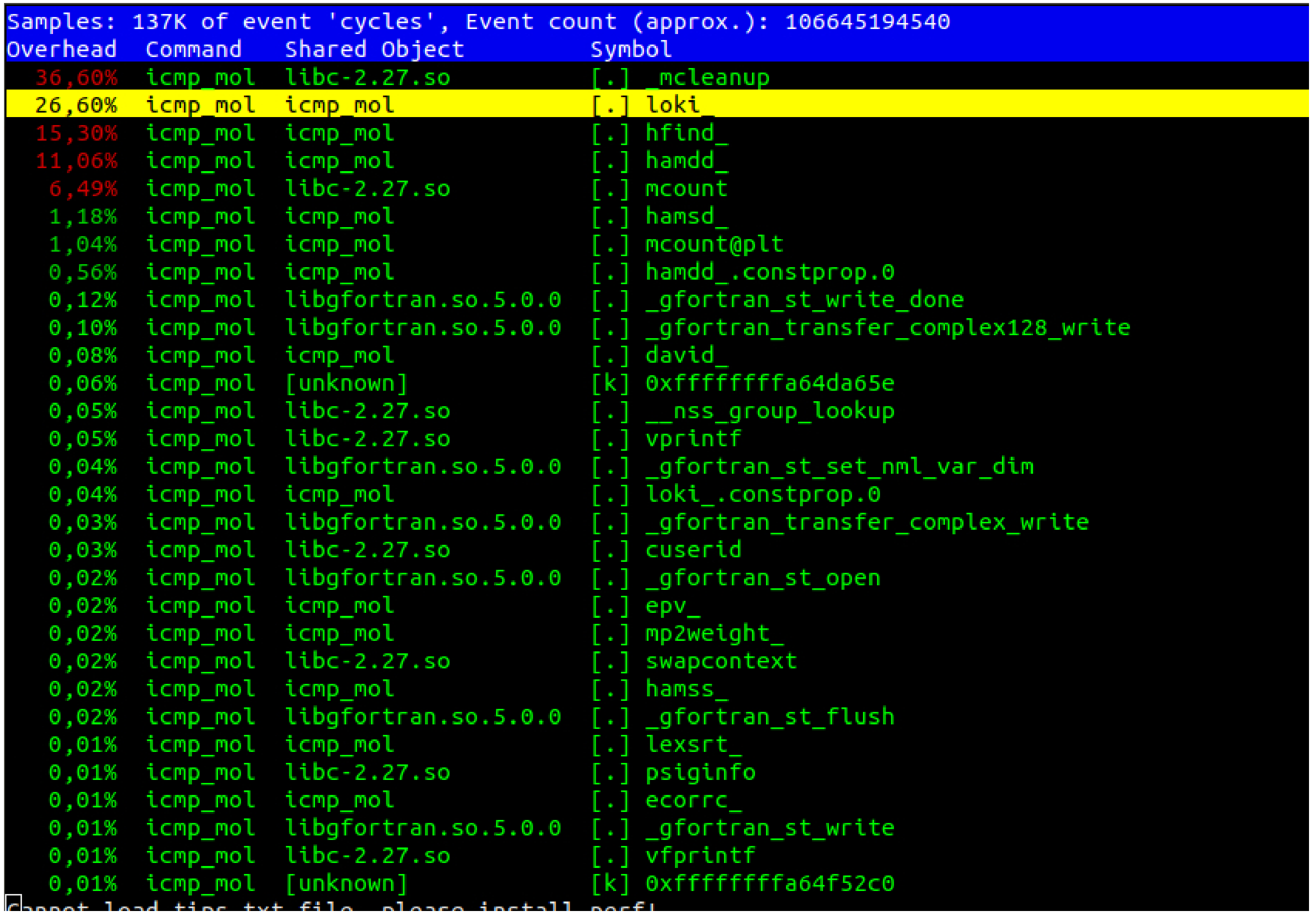
- program run produces a `gmon.out` file
- Analyze the `gmon.out` file with `gprof`



```
396:[7590pr20pow]/home/reinh/results/H2O/DZ>gprof ~/bin/icmp_mol
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls   s/call   s/call  name
 49.88     10.65    10.65 690416184     0.00     0.00  loki_
 27.04     16.42     5.77 426163896     0.00     0.00  hfind_
 18.46     20.36     3.94       19     0.21     0.97  hamdd_
  2.76     20.95     0.59       19     0.03     0.09  hamsd_
  0.66     21.09     0.14        1     0.14     0.90  hamdd_.constprop.0
  0.45     21.18     0.10       21     0.00     0.00  vnorm_
  0.42     21.27     0.09       19     0.00     0.00  hamrs_
  0.14     21.30     0.03  1570464     0.00     0.00  loki_.constprop.0
  0.05     21.31     0.01       19     0.00     0.00  hamss_
  0.05     21.32     0.01       17     0.00     0.00  epv_
  0.05     21.33     0.01        2     0.01    10.21  david_
  0.05     21.34     0.01        2     0.01     0.01  mp2weight_
  0.00     21.34     0.00    88999     0.00     0.00  hrange_
  0.00     21.34     0.00    88999     0.00     0.00  igetty_
  0.00     21.34     0.00     6660     0.00     0.00  fildet_.constprop.1
  0.00     21.34     0.00      144     0.00     0.00  fildet_.constprop.0
  0.00     21.34     0.00       23     0.00     0.00  ecorrc_
```

Profiling

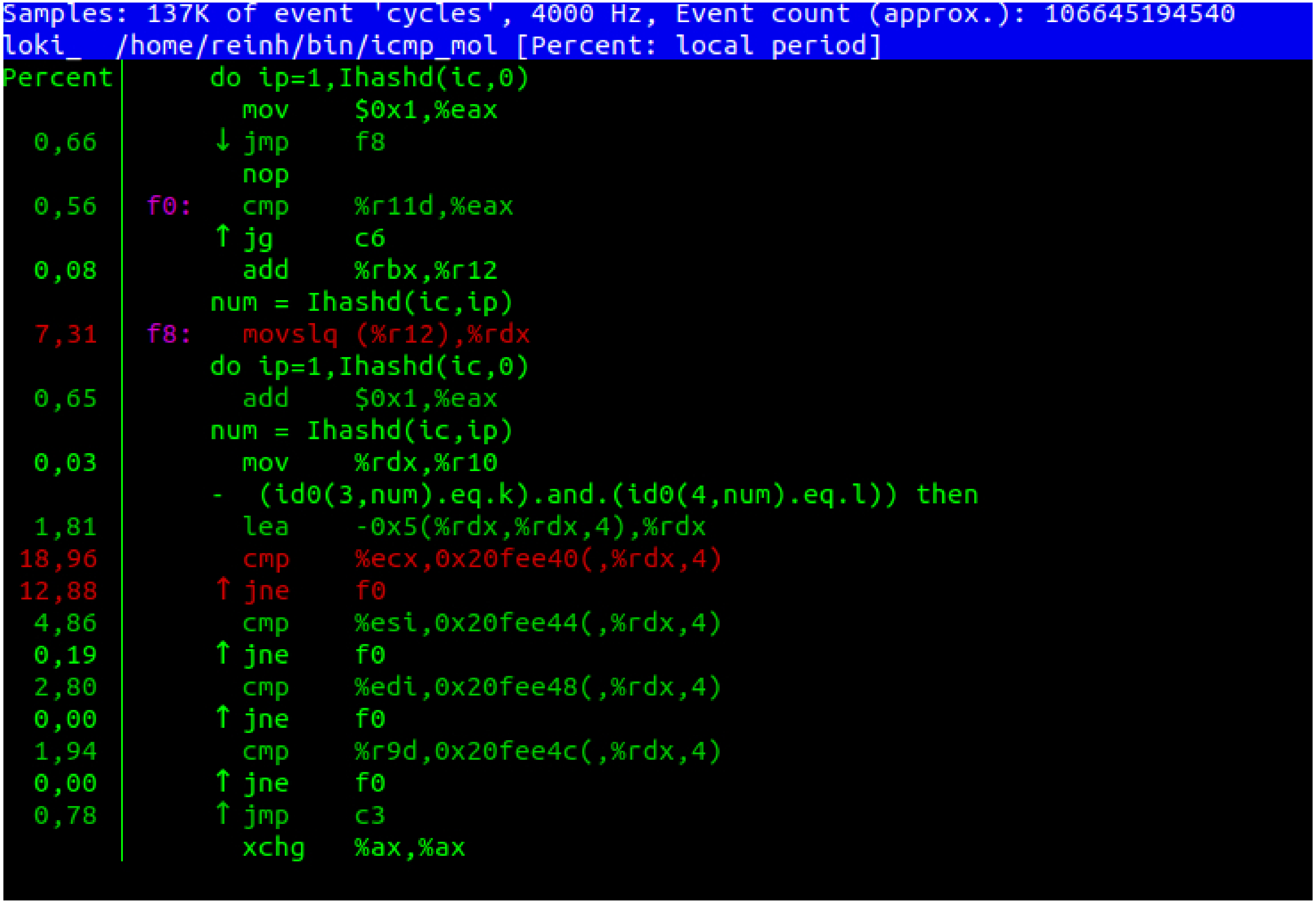
- Alternative: `perf record your_program` does the same job without specific compiler options
- Analyze result with `perf report`



```
Samples: 137K of event 'cycles', Event count (approx.): 106645194540
Overhead  Command    Shared Object          Symbol
  36,60%  icmp_mol   libc-2.27.so           [.] _mcleanup
  26,60%  icmp_mol   icmp_mol               [.] loki_
  15,30%  icmp_mol   icmp_mol               [.] hfind_
  11,06%  icmp_mol   icmp_mol               [.] hamdd_
   6,49%  icmp_mol   libc-2.27.so           [.] mcount
   1,18%  icmp_mol   icmp_mol               [.] hamsd_
   1,04%  icmp_mol   icmp_mol               [.] mcount@plt
   0,56%  icmp_mol   icmp_mol               [.] hamdd_.constprop.0
   0,12%  icmp_mol   libgfortran.so.5.0.0   [.] _gfortran_st_write_done
   0,10%  icmp_mol   libgfortran.so.5.0.0   [.] _gfortran_transfer_complex128_write
   0,08%  icmp_mol   icmp_mol               [.] david_
   0,06%  icmp_mol   [unknown]              [k] 0xffffffffa64da65e
   0,05%  icmp_mol   libc-2.27.so           [.] __nss_group_lookup
   0,05%  icmp_mol   libc-2.27.so           [.] vprintf
   0,04%  icmp_mol   libgfortran.so.5.0.0   [.] _gfortran_st_set_nml_var_dim
   0,04%  icmp_mol   icmp_mol               [.] loki_.constprop.0
   0,03%  icmp_mol   libgfortran.so.5.0.0   [.] _gfortran_transfer_complex_write
   0,03%  icmp_mol   libc-2.27.so           [.] cuserid
   0,02%  icmp_mol   libgfortran.so.5.0.0   [.] _gfortran_st_open
   0,02%  icmp_mol   icmp_mol               [.] epv_
   0,02%  icmp_mol   icmp_mol               [.] mp2weight_
   0,02%  icmp_mol   libc-2.27.so           [.] swapcontext
   0,02%  icmp_mol   icmp_mol               [.] hamss_
   0,02%  icmp_mol   libgfortran.so.5.0.0   [.] _gfortran_st_flush
   0,01%  icmp_mol   icmp_mol               [.] lexsrt_
   0,01%  icmp_mol   libc-2.27.so           [.] psiginfo
   0,01%  icmp_mol   icmp_mol               [.] ecorrc_
   0,01%  icmp_mol   libgfortran.so.5.0.0   [.] _gfortran_st_write
   0,01%  icmp_mol   libc-2.27.so           [.] vfprintf
   0,01%  icmp_mol   [unknown]              [k] 0xffffffffa64f52c0
Cannot load tips.txt file, please install perf!
```

Profiling

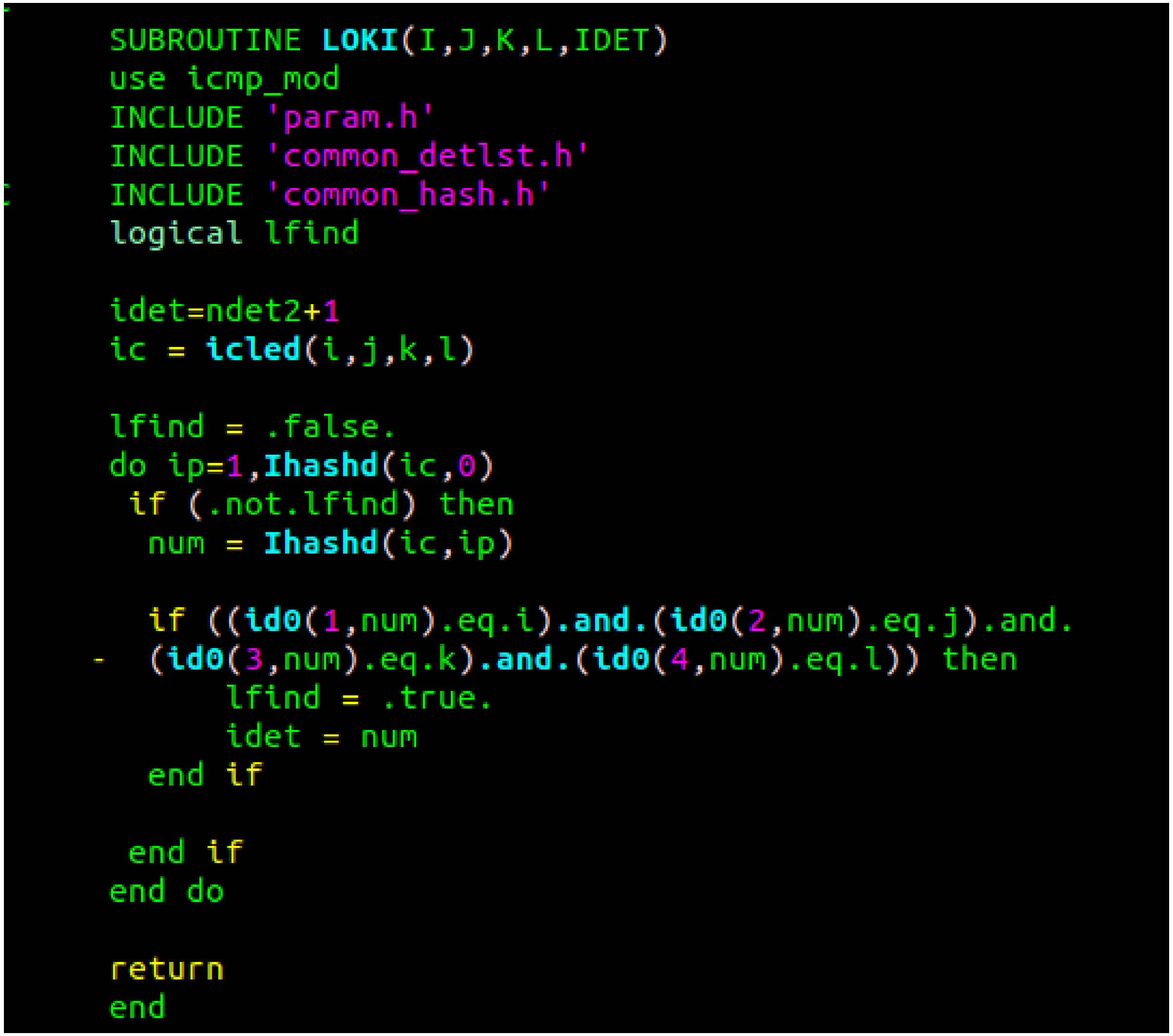
Look into the source of the program



```
Samples: 137K of event 'cycles', 4000 Hz, Event count (approx.): 106645194540
loki_  /home/reinh/bin/icmp_mol [Percent: local period]
Percent│      do ip=1,Ihashd(ic,0)
       │        mov    $0x1,%eax
  0,66 │      ↓ jmp    f8
       │        nop
  0,56 │ f0:    cmp    %r11d,%eax
       │      ↑ jg     c6
  0,08 │        add    %rbx,%r12
       │      num = Ihashd(ic,ip)
  7,31 │ f8:    movslq (%r12),%rdx
       │      do ip=1,Ihashd(ic,0)
  0,65 │        add    $0x1,%eax
       │      num = Ihashd(ic,ip)
  0,03 │        mov    %rdx,%r10
       │      -  (id0(3,num).eq.k).and.(id0(4,num).eq.l)) then
  1,81 │        lea    -0x5(%rdx,%rdx,4),%rdx
 18,96 │        cmp    %ecx,0x20fee40(,%rdx,4)
 12,88 │      ↑ jne    f0
  4,86 │        cmp    %esi,0x20fee44(,%rdx,4)
  0,19 │      ↑ jne    f0
  2,80 │        cmp    %edi,0x20fee48(,%rdx,4)
  0,00 │      ↑ jne    f0
  1,94 │        cmp    %r9d,0x20fee4c(,%rdx,4)
  0,00 │      ↑ jne    f0
  0,78 │      ↑ jmp    c3
       │        xchg   %ax,%ax
```

Profiling

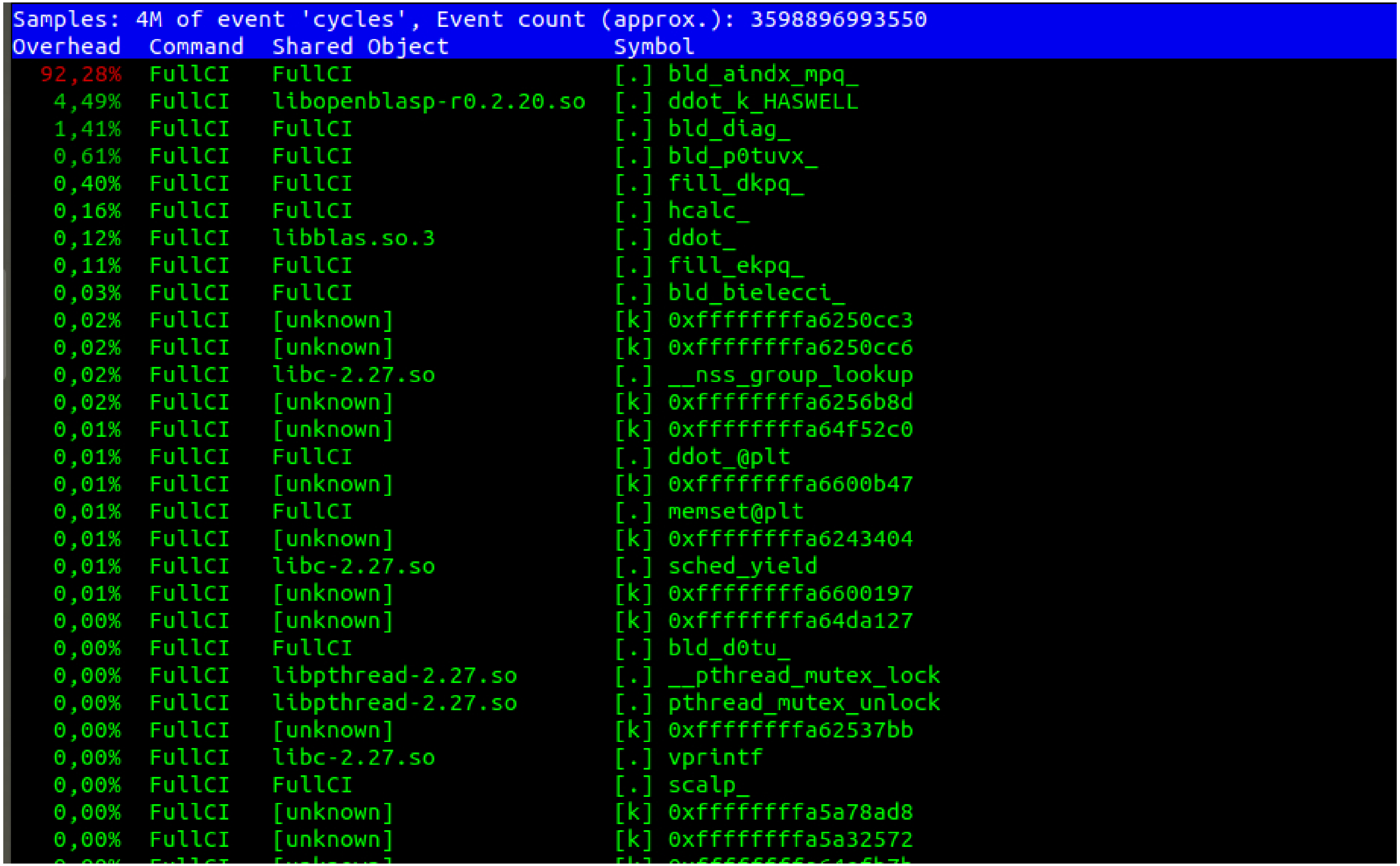
The source code



```
      SUBROUTINE LOKI(I,J,K,L,IDET)
      use icmp_mod
      INCLUDE 'param.h'
      INCLUDE 'common_detlst.h'
      INCLUDE 'common_hash.h'
      logical lfind

      idet=ndet2+1
      ic = icled(i,j,k,l)

      lfind = .false.
      do ip=1,Ihashd(ic,0)
       if (.not.lfind) then
        num = Ihashd(ic,ip)

        if ((id0(1,num).eq.i).and.(id0(2,num).eq.j).and.
     -  (id0(3,num).eq.k).and.(id0(4,num).eq.l)) then
            lfind = .true.
            idet = num
        end if

       end if
      end do

      return
      end
```

Profiling

Another one – a Full CI code

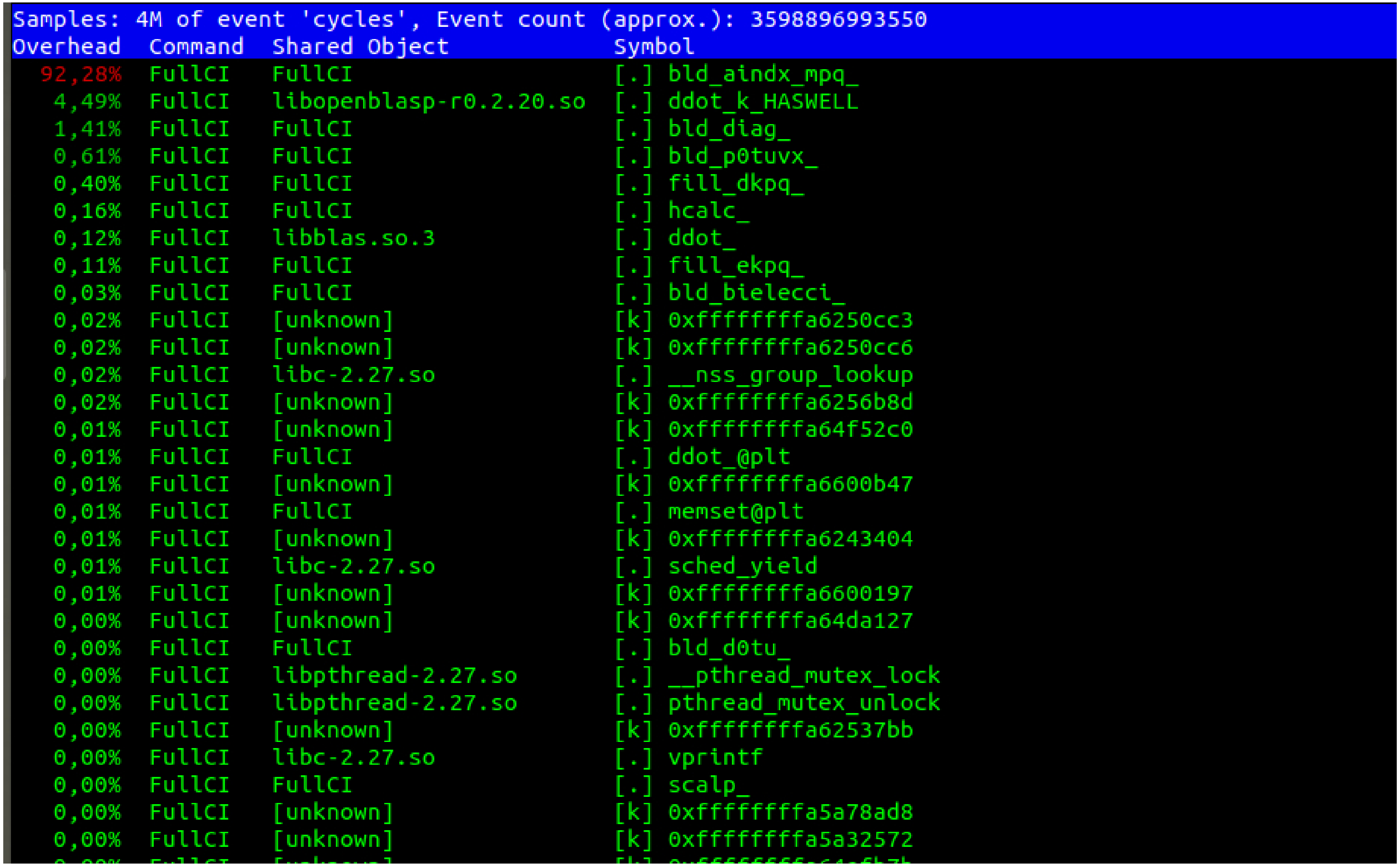


```
Samples: 4M of event 'cycles', Event count (approx.): 3598896993550
Overhead  Command  Shared Object             Symbol
  92,28%  FullCI   FullCI                    [.] bld_aindx_mpq_
   4,49%  FullCI   libopenblasp-r0.2.20.so   [.] ddot_k_HASWELL
   1,41%  FullCI   FullCI                    [.] bld_diag_
   0,61%  FullCI   FullCI                    [.] bld_p0tuvx_
   0,40%  FullCI   FullCI                    [.] fill_dkpq_
   0,16%  FullCI   FullCI                    [.] hcalc_
   0,12%  FullCI   libblas.so.3              [.] ddot_
   0,11%  FullCI   FullCI                    [.] fill_ekpq_
   0,03%  FullCI   FullCI                    [.] bld_bielecci_
   0,02%  FullCI   [unknown]                 [k] 0xffffffffa6250cc3
   0,02%  FullCI   [unknown]                 [k] 0xffffffffa6250cc6
   0,02%  FullCI   libc-2.27.so              [.] __nss_group_lookup
   0,02%  FullCI   [unknown]                 [k] 0xffffffffa6256b8d
   0,01%  FullCI   [unknown]                 [k] 0xffffffffa64f52c0
   0,01%  FullCI   FullCI                    [.] ddot_@plt
   0,01%  FullCI   [unknown]                 [k] 0xffffffffa6600b47
   0,01%  FullCI   FullCI                    [.] memset@plt
   0,01%  FullCI   [unknown]                 [k] 0xffffffffa6243404
   0,01%  FullCI   libc-2.27.so              [.] sched_yield
   0,01%  FullCI   [unknown]                 [k] 0xffffffffa6600197
   0,00%  FullCI   [unknown]                 [k] 0xffffffffa64da127
   0,00%  FullCI   FullCI                    [.] bld_d0tu_
   0,00%  FullCI   libpthread-2.27.so        [.] __pthread_mutex_lock
   0,00%  FullCI   libpthread-2.27.so        [.] pthread_mutex_unlock
   0,00%  FullCI   [unknown]                 [k] 0xffffffffa62537bb
   0,00%  FullCI   libc-2.27.so              [.] vprintf
   0,00%  FullCI   FullCI                    [.] scalp_
   0,00%  FullCI   [unknown]                 [k] 0xffffffffa5a78ad8
   0,00%  FullCI   [unknown]                 [k] 0xffffffffa5a32572
   0,00%  FullCI   [unknown]                 [k] 0xffffffffa64afb7b
```

The problem is in the routine `bld_aindx_mpq`

Profiling

Another one – a Full CI code

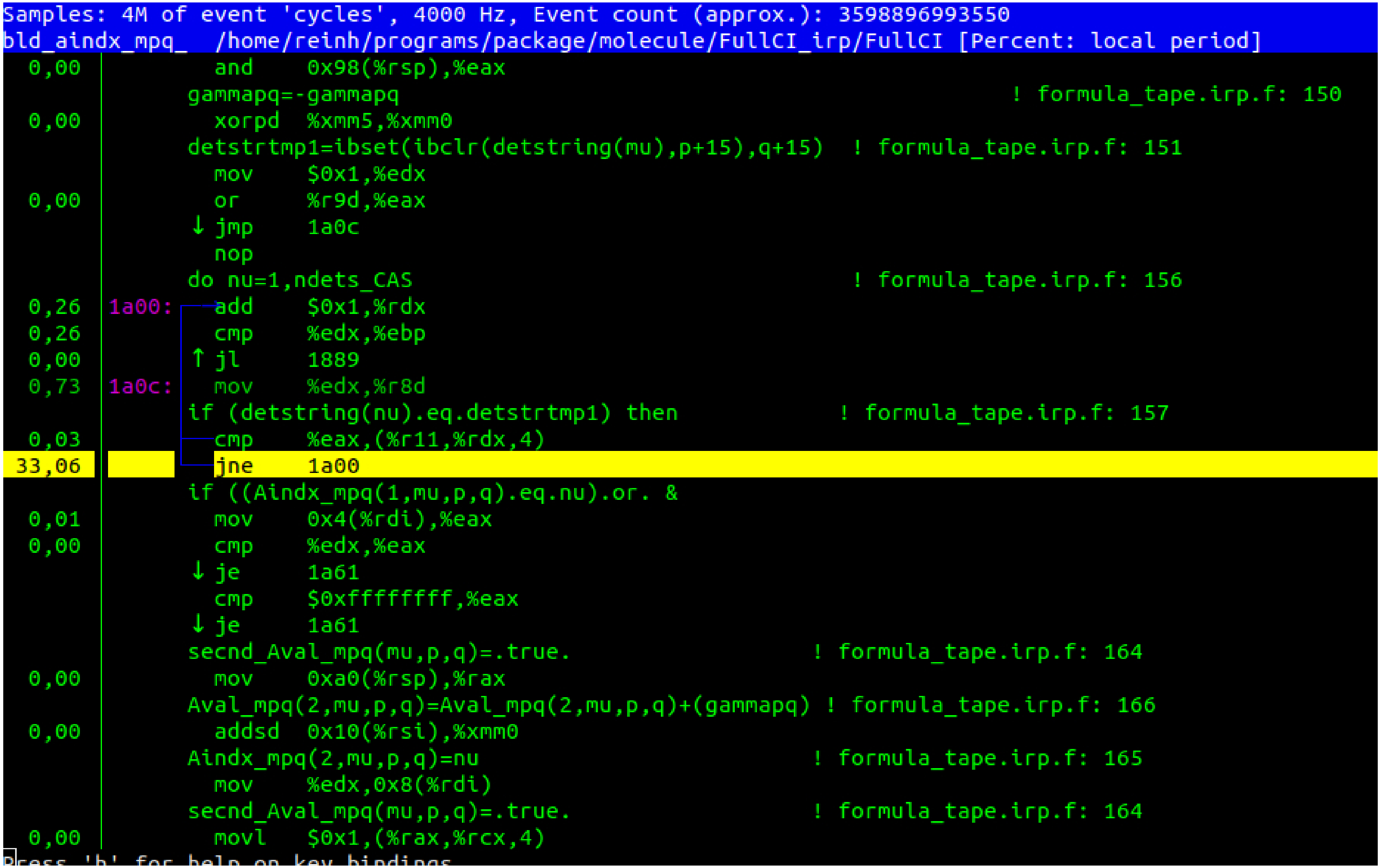


```
Samples: 4M of event 'cycles', Event count (approx.): 3598896993550
Overhead  Command  Shared Object             Symbol
  92,28%  FullCI   FullCI                    [.] bld_aindx_mpq_
   4,49%  FullCI   libopenblasp-r0.2.20.so   [.] ddot_k_HASWELL
   1,41%  FullCI   FullCI                    [.] bld_diag_
   0,61%  FullCI   FullCI                    [.] bld_p0tuvx_
   0,40%  FullCI   FullCI                    [.] fill_dkpq_
   0,16%  FullCI   FullCI                    [.] hcalc_
   0,12%  FullCI   libblas.so.3              [.] ddot_
   0,11%  FullCI   FullCI                    [.] fill_ekpq_
   0,03%  FullCI   FullCI                    [.] bld_bielecci_
   0,02%  FullCI   [unknown]                 [k] 0xffffffffa6250cc3
   0,02%  FullCI   [unknown]                 [k] 0xffffffffa6250cc6
   0,02%  FullCI   libc-2.27.so              [.] __nss_group_lookup
   0,02%  FullCI   [unknown]                 [k] 0xffffffffa6256b8d
   0,01%  FullCI   [unknown]                 [k] 0xffffffffa64f52c0
   0,01%  FullCI   FullCI                    [.] ddot_@plt
   0,01%  FullCI   [unknown]                 [k] 0xffffffffa6600b47
   0,01%  FullCI   FullCI                    [.] memset@plt
   0,01%  FullCI   [unknown]                 [k] 0xffffffffa6243404
   0,01%  FullCI   libc-2.27.so              [.] sched_yield
   0,01%  FullCI   [unknown]                 [k] 0xffffffffa6600197
   0,00%  FullCI   [unknown]                 [k] 0xffffffffa64da127
   0,00%  FullCI   FullCI                    [.] bld_d0tu_
   0,00%  FullCI   libpthread-2.27.so        [.] __pthread_mutex_lock
   0,00%  FullCI   libpthread-2.27.so        [.] pthread_mutex_unlock
   0,00%  FullCI   [unknown]                 [k] 0xffffffffa62537bb
   0,00%  FullCI   libc-2.27.so              [.] vprintf
   0,00%  FullCI   FullCI                    [.] scalp_
   0,00%  FullCI   [unknown]                 [k] 0xffffffffa5a78ad8
   0,00%  FullCI   [unknown]                 [k] 0xffffffffa5a32572
   0,00%  FullCI   [unknown]                 [k] 0xffffffffa64afb7b
```

The problem is in the routine `bld_aindx_mpq`

Profiling

Another one – a Full CI code

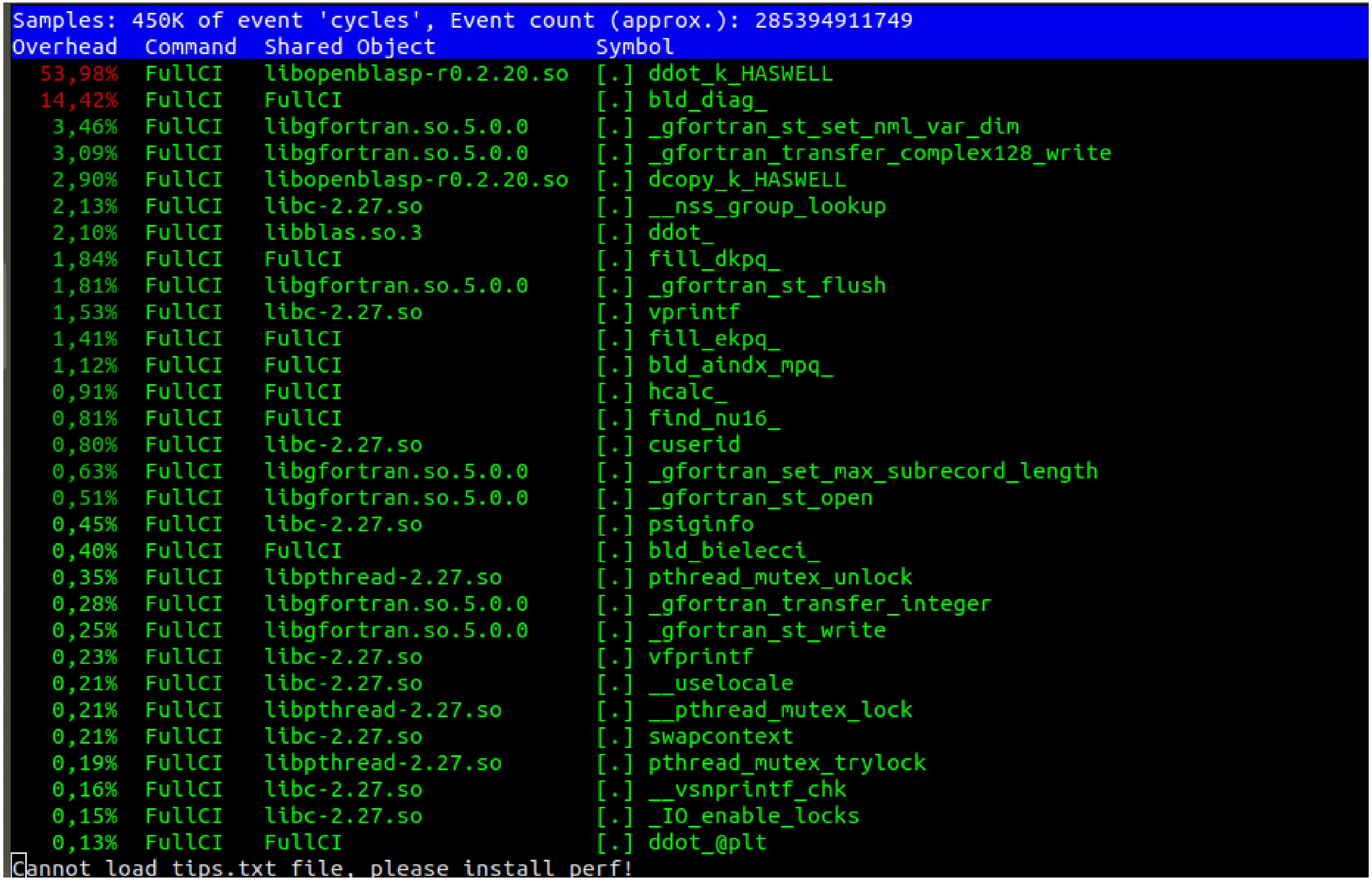


```
Samples: 4M of event 'cycles', 4000 Hz, Event count (approx.): 3598896993550
bld_aindx_mpq_  /home/reinh/programs/package/molecule/FullCI_irp/FullCI [Percent: local period]
  0,00 │          and    0x98(%rsp),%eax
       │        gammapq=-gammapq                                         ! formula_tape.irp.f: 150
  0,00 │          xorpd  %xmm5,%xmm0
       │        detstrtmp1=ibset(ibclr(detstring(mu),p+15),q+15)  ! formula_tape.irp.f: 151
       │          mov    $0x1,%edx
  0,00 │          or     %r9d,%eax
       │        ↓ jmp    1a0c
       │          nop
       │        do nu=1,ndets_CAS                                 ! formula_tape.irp.f: 156
  0,26 │1a00:     add    $0x1,%rdx
  0,26 │          cmp    %edx,%ebp
  0,00 │        ↑ jl     1889
  0,73 │1a0c:     mov    %edx,%r8d
       │        if (detstring(nu).eq.detstrtmp1) then            ! formula_tape.irp.f: 157
  0,03 │          cmp    %eax,(%r11,%rdx,4)
 33,06 │          jne    1a00
       │        if ((Aindx_mpq(1,mu,p,q).eq.nu).or. &
  0,01 │          mov    0x4(%rdi),%eax
  0,00 │          cmp    %edx,%eax
       │        ↓ je     1a61
       │          cmp    $0xffffffff,%eax
       │        ↓ je     1a61
       │        secnd_Aval_mpq(mu,p,q)=.true.                   ! formula_tape.irp.f: 164
  0,00 │          mov    0xa0(%rsp),%rax
       │        Aval_mpq(2,mu,p,q)=Aval_mpq(2,mu,p,q)+(gammapq) ! formula_tape.irp.f: 166
  0,00 │          addsd  0x10(%rsi),%xmm0
       │        Aindx_mpq(2,mu,p,q)=nu                          ! formula_tape.irp.f: 165
       │          mov    %edx,0x8(%rdi)
       │        secnd_Aval_mpq(mu,p,q)=.true.                   ! formula_tape.irp.f: 164
  0,00 │          movl   $0x1,(%rax,%rcx,4)
Press 'h' for help on key bindings
```

What is going on here? We search in a list step by step!

Profiling

Another one – a Full CI code



```
Samples: 450K of event 'cycles', Event count (approx.): 285394911749
Overhead  Command  Shared Object             Symbol
  53,98%  FullCI   libopenblasp-r0.2.20.so   [.] ddot_k_HASWELL
  14,42%  FullCI   FullCI                    [.] bld_diag_
   3,46%  FullCI   libgfortran.so.5.0.0      [.] _gfortran_st_set_nml_var_dim
   3,09%  FullCI   libgfortran.so.5.0.0      [.] _gfortran_transfer_complex128_write
   2,90%  FullCI   libopenblasp-r0.2.20.so   [.] dcopy_k_HASWELL
   2,13%  FullCI   libc-2.27.so              [.] __nss_group_lookup
   2,10%  FullCI   libblas.so.3              [.] ddot_
   1,84%  FullCI   FullCI                    [.] fill_dkpq_
   1,81%  FullCI   libgfortran.so.5.0.0      [.] _gfortran_st_flush
   1,53%  FullCI   libc-2.27.so              [.] vprintf
   1,41%  FullCI   FullCI                    [.] fill_ekpq_
   1,12%  FullCI   FullCI                    [.] bld_aindx_mpq_
   0,91%  FullCI   FullCI                    [.] hcalc_
   0,81%  FullCI   FullCI                    [.] find_nu16_
   0,80%  FullCI   libc-2.27.so              [.] cuserid
   0,63%  FullCI   libgfortran.so.5.0.0      [.] _gfortran_set_max_subrecord_length
   0,51%  FullCI   libgfortran.so.5.0.0      [.] _gfortran_st_open
   0,45%  FullCI   libc-2.27.so              [.] psiginfo
   0,40%  FullCI   FullCI                    [.] bld_bielecci_
   0,35%  FullCI   libpthread-2.27.so        [.] pthread_mutex_unlock
   0,28%  FullCI   libgfortran.so.5.0.0      [.] _gfortran_transfer_integer
   0,25%  FullCI   libgfortran.so.5.0.0      [.] _gfortran_st_write
   0,23%  FullCI   libc-2.27.so              [.] vfprintf
   0,21%  FullCI   libc-2.27.so              [.] __uselocale
   0,21%  FullCI   libpthread-2.27.so        [.] __pthread_mutex_lock
   0,21%  FullCI   libc-2.27.so              [.] swapcontext
   0,19%  FullCI   libpthread-2.27.so        [.] pthread_mutex_trylock
   0,16%  FullCI   libc-2.27.so              [.] __vsnprintf_chk
   0,15%  FullCI   libc-2.27.so              [.] _IO_enable_locks
   0,13%  FullCI   FullCI                    [.] ddot_@plt
Cannot load tips.txt file, please install perf!
```

Replaced by a bisection