PROFILING ON RSA ENCRYPTION

High Performance Computing Project Report

Problem Statement: Encryption and Decryption of Very Large Numbers using the RSA Algorithm

Paul Babu Kadali - CED19I002

Profiling:

In a performance engineering context performance profiling means to relate performance metric measurements to source code execution. Data sources are typically either operating systems, execution environments or measurement facilities in the hardware

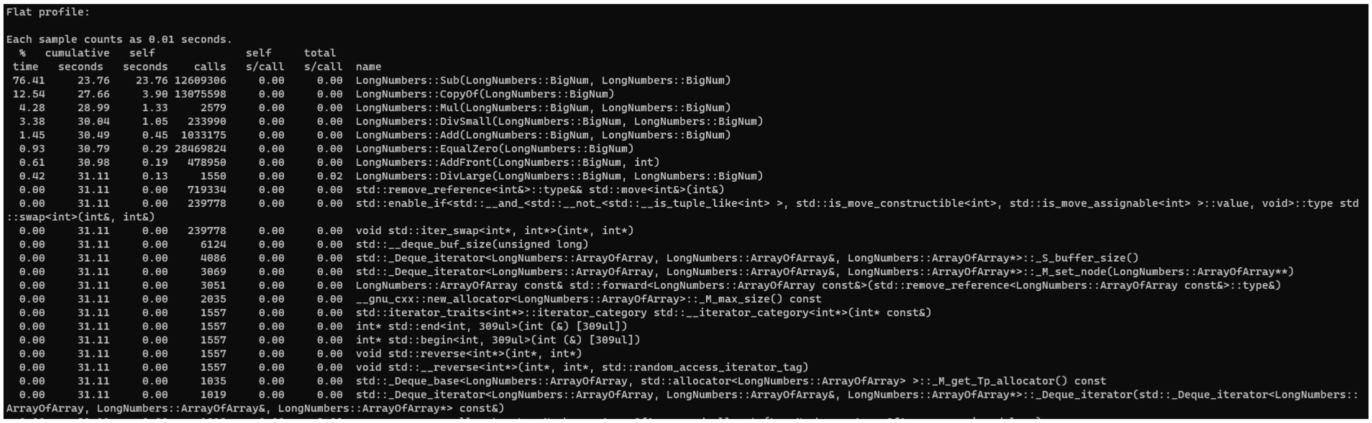
Tools used for Profiling:

1. Gprof - Function-based Profiling
2. Gconv - Line-based Profiling
3. Likwid - HArdware-Based Profiling

Gprof:

Gprof is used to get the frequency of each function calls, to determine the computation intensive function

```
g++ -fopenmp -pg -g -O0 rsa_serial1.cpp && ./a.out
gprof -b a.out > gprof.out
cat gprof.out
```



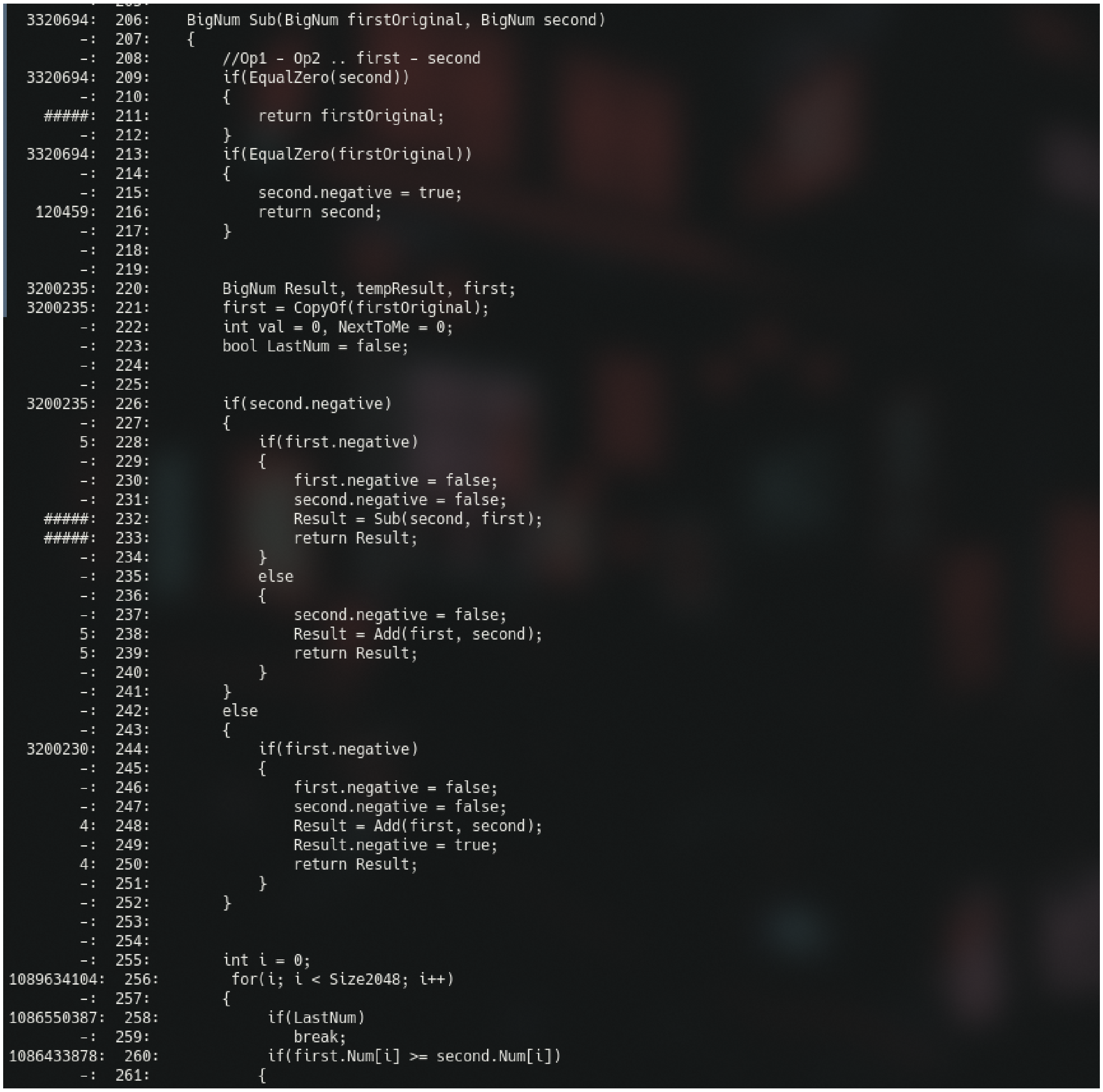
```
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls   s/call   s/call  name
 76.41     23.76    23.76 12609306     0.00     0.00  LongNumbers::Sub(LongNumbers::BigNum, LongNumbers::BigNum)
 12.54     27.66     3.90 13075598     0.00     0.00  LongNumbers::CopyOf(LongNumbers::BigNum)
  4.28     28.99     1.33     2579     0.00     0.00  LongNumbers::Mul(LongNumbers::BigNum, LongNumbers::BigNum)
  3.38     30.04     1.05   233990     0.00     0.00  LongNumbers::DivSmall(LongNumbers::BigNum, LongNumbers::BigNum)
  1.45     30.49     0.45  1033175     0.00     0.00  LongNumbers::Add(LongNumbers::BigNum, LongNumbers::BigNum)
  0.93     30.79     0.29 28469824     0.00     0.00  LongNumbers::EqualZero(LongNumbers::BigNum)
  0.61     30.98     0.19   478950     0.00     0.00  LongNumbers::AddFront(LongNumbers::BigNum, int)
  0.42     31.11     0.13     1550     0.00     0.02  LongNumbers::DivLarge(LongNumbers::BigNum, LongNumbers::BigNum)
  0.00     31.11     0.00   719334     0.00     0.00  std::remove_reference<int&>::type&& std::move<int&>(int&)
  0.00     31.11     0.00   239778     0.00     0.00  std::enable_if<std::__and_<std::__not_<std::__is_tuple_like<int> >, std::is_move_constructible<int>, std::is_move_assignable<int> >::value, void>::type std
::swap<int>(int&, int&)
  0.00     31.11     0.00   239778     0.00     0.00  void std::iter_swap<int*, int*>(int*, int*)
  0.00     31.11     0.00     6124     0.00     0.00  std::__deque_buf_size(unsigned long)
  0.00     31.11     0.00     4086     0.00     0.00  std::_Deque_iterator<LongNumbers::ArrayOfArray, LongNumbers::ArrayOfArray&, LongNumbers::ArrayOfArray*>::_S_buffer_size()
  0.00     31.11     0.00     3069     0.00     0.00  std::_Deque_iterator<LongNumbers::ArrayOfArray, LongNumbers::ArrayOfArray&, LongNumbers::ArrayOfArray*>::_M_set_node(LongNumbers::ArrayOfArray**)
  0.00     31.11     0.00     3051     0.00     0.00  LongNumbers::ArrayOfArray const& std::forward<LongNumbers::ArrayOfArray const&>(std::remove_reference<LongNumbers::ArrayOfArray const&>::type&)
  0.00     31.11     0.00     2035     0.00     0.00  __gnu_cxx::new_allocator<LongNumbers::ArrayOfArray>::_M_max_size() const
  0.00     31.11     0.00     1557     0.00     0.00  std::iterator_traits<int*>::iterator_category std::__iterator_category<int*>(int* const&)
  0.00     31.11     0.00     1557     0.00     0.00  int* std::end<int, 309ul>(int (&) [309ul])
  0.00     31.11     0.00     1557     0.00     0.00  int* std::begin<int, 309ul>(int (&) [309ul])
  0.00     31.11     0.00     1557     0.00     0.00  void std::reverse<int*>(int*, int*)
  0.00     31.11     0.00     1557     0.00     0.00  void std::__reverse<int*>(int*, int*, std::random_access_iterator_tag)
  0.00     31.11     0.00     1035     0.00     0.00  std::_Deque_base<LongNumbers::ArrayOfArray, std::allocator<LongNumbers::ArrayOfArray> >::_M_get_Tp_allocator() const
  0.00     31.11     0.00     1019     0.00     0.00  std::_Deque_iterator<LongNumbers::ArrayOfArray, LongNumbers::ArrayOfArray&, LongNumbers::ArrayOfArray*>::_Deque_iterator(std::_Deque_iterator<LongNumbers::
ArrayOfArray, LongNumbers::ArrayOfArray&, LongNumbers::ArrayOfArray*> const&)
```

As shown in the outputs, The most called functions are Subtraction and Multiplication of the Long Numbers performed during the Encryption.

Gcov:

Gcov is used to get the frequency of each line, to determine the computation intensive section

```
g++ --coverage -fopenmp -fprofile-arcs -ftest-coverage -O rsa_serial1.cpp -lgcov
&& ./a.out
gcov rsa_serial1.cpp
cat rsa_serial1.cpp.gcov
```

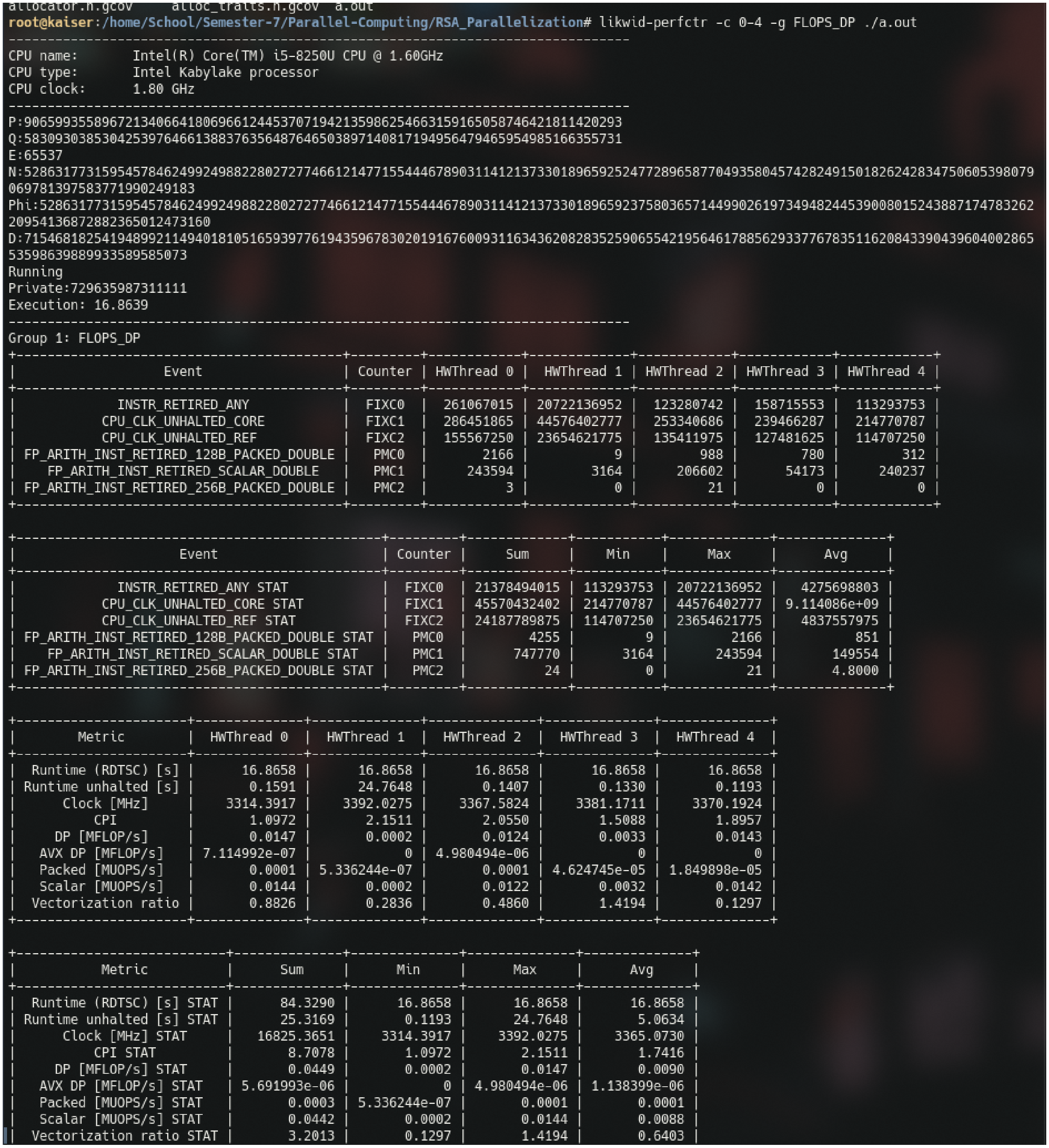


```
  3320694:  206:    BigNum Sub(BigNum firstOriginal, BigNum second)
        -:  207:    {
        -:  208:        //Op1 - Op2 .. first - second
  3320694:  209:        if(EqualZero(second))
        -:  210:        {
    #####:  211:            return firstOriginal;
        -:  212:        }
  3320694:  213:        if(EqualZero(firstOriginal))
        -:  214:        {
        -:  215:            second.negative = true;
   120459:  216:            return second;
        -:  217:        }
        -:  218:
        -:  219:
  3200235:  220:        BigNum Result, tempResult, first;
  3200235:  221:        first = CopyOf(firstOriginal);
        -:  222:        int val = 0, NextToMe = 0;
        -:  223:        bool LastNum = false;
        -:  224:
        -:  225:
  3200235:  226:        if(second.negative)
        -:  227:        {
        5:  228:            if(first.negative)
        -:  229:            {
        -:  230:                first.negative = false;
        -:  231:                second.negative = false;
    #####:  232:                Result = Sub(second, first);
    #####:  233:                return Result;
        -:  234:            }
        -:  235:            else
        -:  236:            {
        -:  237:                second.negative = false;
        5:  238:                Result = Add(first, second);
        5:  239:                return Result;
        -:  240:            }
        -:  241:        }
        -:  242:        else
        -:  243:        {
  3200230:  244:            if(first.negative)
        -:  245:            {
        -:  246:                first.negative = false;
        -:  247:                second.negative = false;
        4:  248:                Result = Add(first, second);
        -:  249:                Result.negative = true;
        4:  250:                return Result;
        -:  251:            }
        -:  252:        }
        -:  253:
        -:  254:
        -:  255:        int i = 0;
1089634104:  256:         for(i; i < Size2048; i++)
        -:  257:        {
1086550387:  258:             if(LastNum)
        -:  259:                break;
1086433878:  260:             if(first.Num[i] >= second.Num[i])
        -:  261:            {
```

Likwid:

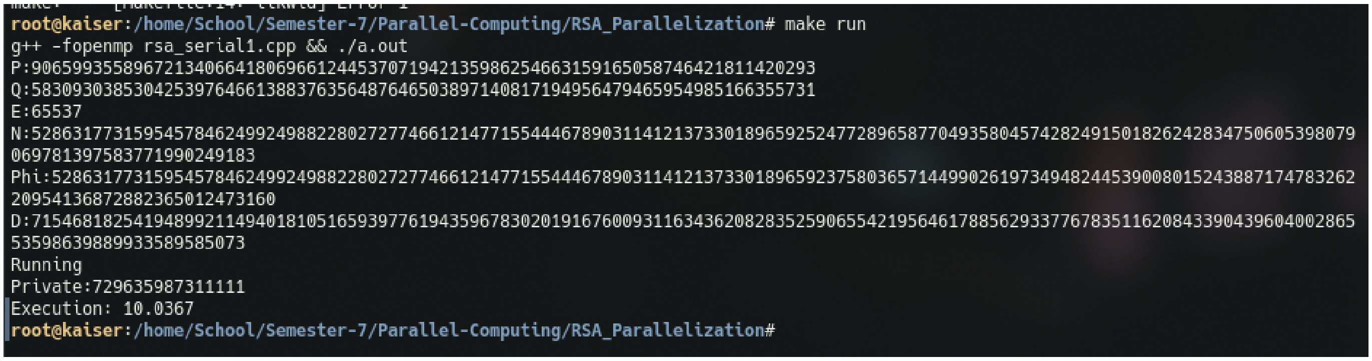
As the program is not currently parallelized, the program just runs in the core 0.

```
likwid-perfctr -c 0-4 -g FLOPS_DP ./a.out
```



```
allocator.h.gcov     alloc_traits.h.gcov  a.out
root@kaiser:/home/School/Semester-7/Parallel-Computing/RSA_Parallelization# likwid-perfctr -c 0-4 -g FLOPS_DP ./a.out
--------------------------------------------------------------------------------
CPU name:       Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz
CPU type:       Intel Kabylake processor
CPU clock:      1.80 GHz
--------------------------------------------------------------------------------
P:90659935589672134066418069661244537071942135986254663159165058746421811420293
Q:58309303853042539764661388376356487646503897140817194956479465954985166355731
E:65537
N:52863177315954578462499249882280272774661214771554446789031141213733018965925247728965877049358045742824915018262428347506053980790697813975837719902491 83
Phi:52863177315954578462499249882280272774661214771554446789031141213733018965923758036571449902619734948244539008015243887174783262209541368728823650124731 60
D:71546818254194899211494018105165939776194359678302019167600931163436208283525906554219564617885629337767835116208433904396040028655359863988993358958507 3
Running
Private:729635987311111
Execution: 16.8639
--------------------------------------------------------------------------------
Group 1: FLOPS_DP
+------------------------------------------+---------+-------------+-------------+-------------+-------------+-------------+
|                   Event                  | Counter | HWThread 0  |  HWThread 1 | HWThread 2  | HWThread 3  | HWThread 4  |
+------------------------------------------+---------+-------------+-------------+-------------+-------------+-------------+
|             INSTR_RETIRED_ANY            |  FIXC0  |   261067015 | 20722136952 |   123280742 |   158715553 |   113293753 |
|           CPU_CLK_UNHALTED_CORE          |  FIXC1  |   286451865 | 44576402777 |   253340686 |   239466287 |   214770787 |
|           CPU_CLK_UNHALTED_REF           |  FIXC2  |   155567250 | 23654621775 |   135411975 |   127481625 |   114707250 |
| FP_ARITH_INST_RETIRED_128B_PACKED_DOUBLE |   PMC0  |        2166 |           9 |         988 |         780 |         312 |
|    FP_ARITH_INST_RETIRED_SCALAR_DOUBLE   |   PMC1  |      243594 |        3164 |      206602 |       54173 |      240237 |
| FP_ARITH_INST_RETIRED_256B_PACKED_DOUBLE |   PMC2  |           3 |           0 |          21 |           0 |           0 |
+------------------------------------------+---------+-------------+-------------+-------------+-------------+-------------+

+-----------------------------------------------+---------+-------------+------------+-------------+--------------+
|                     Event                     | Counter |     Sum     |     Min    |     Max     |      Avg     |
+-----------------------------------------------+---------+-------------+------------+-------------+--------------+
|             INSTR_RETIRED_ANY STAT            |  FIXC0  | 21378494015 |  113293753 | 20722136952 |   4275698803 |
|           CPU_CLK_UNHALTED_CORE STAT          |  FIXC1  | 45570432402 |  214770787 | 44576402777 | 9.114086e+09 |
|           CPU_CLK_UNHALTED_REF STAT           |  FIXC2  | 24187789875 |  114707250 | 23654621775 |   4837557975 |
| FP_ARITH_INST_RETIRED_128B_PACKED_DOUBLE STAT |   PMC0  |        4255 |          9 |        2166 |          851 |
|    FP_ARITH_INST_RETIRED_SCALAR_DOUBLE STAT   |   PMC1  |      747770 |       3164 |      243594 |       149554 |
| FP_ARITH_INST_RETIRED_256B_PACKED_DOUBLE STAT |   PMC2  |          24 |          0 |          21 |       4.8000 |
+-----------------------------------------------+---------+-------------+------------+-------------+--------------+

+----------------------+--------------+--------------+--------------+--------------+--------------+
|        Metric        |  HWThread 0  |  HWThread 1  |  HWThread 2  |  HWThread 3  |  HWThread 4  |
+----------------------+--------------+--------------+--------------+--------------+--------------+
|  Runtime (RDTSC) [s] |      16.8658 |      16.8658 |      16.8658 |      16.8658 |      16.8658 |
| Runtime unhalted [s] |       0.1591 |      24.7648 |       0.1407 |       0.1330 |       0.1193 |
|      Clock [MHz]     |    3314.3917 |    3392.0275 |    3367.5824 |    3381.1711 |    3370.1924 |
|          CPI         |       1.0972 |       2.1511 |       2.0550 |       1.5088 |       1.8957 |
|     DP [MFLOP/s]     |       0.0147 |       0.0002 |       0.0124 |       0.0033 |       0.0143 |
|   AVX DP [MFLOP/s]   | 7.114992e-07 |            0 | 4.980494e-06 |            0 |            0 |
|   Packed [MUOPS/s]   |       0.0001 | 5.336244e-07 |       0.0001 | 4.624745e-05 | 1.849898e-05 |
|   Scalar [MUOPS/s]   |       0.0144 |       0.0002 |       0.0122 |       0.0032 |       0.0142 |
|  Vectorization ratio |       0.8826 |       0.2836 |       0.4860 |       1.4194 |       0.1297 |
+----------------------+--------------+--------------+--------------+--------------+--------------+

+---------------------------+--------------+--------------+--------------+--------------+
|           Metric          |      Sum     |      Min     |      Max     |      Avg     |
+---------------------------+--------------+--------------+--------------+--------------+
|  Runtime (RDTSC) [s] STAT |      84.3290 |      16.8658 |      16.8658 |      16.8658 |
| Runtime unhalted [s] STAT |      25.3169 |       0.1193 |      24.7648 |       5.0634 |
|      Clock [MHz] STAT     |   16825.3651 |    3314.3917 |    3392.0275 |    3365.0730 |
|          CPI STAT         |       8.7078 |       1.0972 |       2.1511 |       1.7416 |
|     DP [MFLOP/s] STAT     |       0.0449 |       0.0002 |       0.0147 |       0.0090 |
|   AVX DP [MFLOP/s] STAT   | 5.691993e-06 |            0 | 4.980494e-06 | 1.138399e-06 |
|   Packed [MUOPS/s] STAT   |       0.0003 | 5.336244e-07 |       0.0001 |       0.0001 |
|   Scalar [MUOPS/s] STAT   |       0.0442 |       0.0002 |       0.0144 |       0.0088 |
|  Vectorization ratio STAT |       3.2013 |       0.1297 |       1.4194 |       0.6403 |
```

Output



```
make: *** [Makefile:14: likwid] Error 1
root@kaiser:/home/School/Semester-7/Parallel-Computing/RSA_Parallelization# make run
g++ -fopenmp rsa_serial1.cpp && ./a.out
P:906599355896721340664180696612445370719421359862546631591650587464218114202930
Q:58309303853042539764661388376356487646503897140817194956479465954985166355731
E:65537
N:5286317731595457846249924988228027277466121477155444678903114121373301896592524772896587704935804574282491501826242834750605398079
06978139758377199024918 3
Phi:5286317731595457846249924988228027277466121477155444678903114121373301896592375803657144990261973494824453900801524388717478326 2
20954136872882365012473160
D:7154681825419489921149401810516593977619435967830201916760093116343620828352590655421956461788562933776783511620843390439604002865
535986398899335895850 73
Running
Private:729635987311111
Execution: 10.0367
root@kaiser:/home/School/Semester-7/Parallel-Computing/RSA_Parallelization#
```

Observations:

On application of the mentioned profiling tools, namely, GPROF, GCOV, LIKWID, the hot spot of the program is determined.

Gprof: The most called functions are Subtraction and Multiplication of the Long Numbers performed during the Encryption.

Gconv: The most commonly run lines are the ones within the loop that is Subtracting the Long Numbers and Copyof Long Numbers which can potentially be parallelizable.

Likwid: No inference was drawn from these outputs as the code is not yet parallelized and likwid-perfctr does not support the student's processor.