Instituto Tecnológico de Costa Rica

Área Académica Ingeniería en Computadores

Curso: Arquitectura de Computadores I

Profesor: Ronald García Fernández

Proyecto Individual: Diseño e Implementación de un sistema de encriptación y desencriptación RSA empleando lenguaje ensamblador

Pruebas de desempeño de la aplicación

Estudiante: Roberto Sánchez Gutiérrez

I Semestre 2019

1. Pruebas

Las pruebas de rendimiento a las que se somete la aplicación de RSA se basarán en el uso de distinta cantidad de bits para realizar el proceso de encriptación o desencriptación. Los resultados serán basados en el análisis contra los branches y el uso de la memoria caché, es decir los misses de ambos, tanto de prediction o hit, dependiendo también si se cambian los parámetros de caché. Todo este análisis será utilizando las herramientas de valgrind.

El ambiente de desarrollo de las pruebas es el siguiente:
Sistema operativo Ubuntu 18.04 LTS, con la siguiente información del procesador utilizado al correr el comando “lscpu” en la consola:

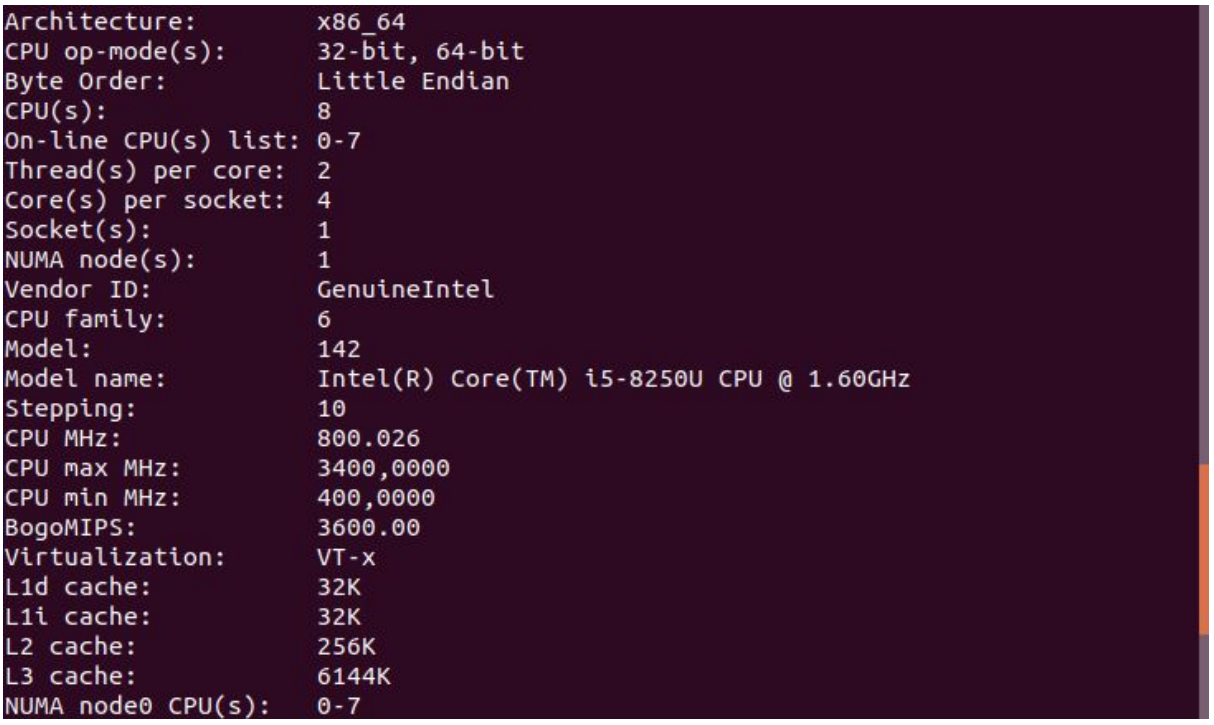

Figura 1. Información del cpu utilizado en las pruebas de desempeño

Las bibliotecas necesarias para correr la aplicación en consola y ver el desempeño son las siguientes:

Biblioteca	**Instalación**
build-essential	sudo apt install build-essential
nasm	sudo apt install nasm
gdb	sudo apt install gdb
valgrind	sudo apt-get install valgrind kcachegrind graphviz

2. Resultados

A continuación se muestran los resultados de las pruebas a partir de la salida de los códigos ejecutados

2.1. Encripción para primos de 256 bits con valores default

Para números primos de 256 bits, llaves de 512 bits se obtienen los siguientes resultados

```
Starting encryption
********************************
==24701==
==24701== I   refs:      13,683,746,777
==24701== I1  misses:              812
==24701== LLi misses:              805
==24701== I1  miss rate:          0.00%
==24701== LLi miss rate:          0.00%
==24701==
==24701== D   refs:       8,919,450,499  (5,882,426,812 rd   + 3,037,023,687 wr)
==24701== D1  misses:             1,819  (        1,269 rd   +           550 wr)
==24701== LLd misses:             1,657  (        1,128 rd   +           529 wr)
==24701== D1  miss rate:            0.0% (          0.0%     +           0.0%  )
==24701== LLd miss rate:            0.0% (          0.0%     +           0.0%  )
==24701==
==24701== LL refs:                2,631  (        2,081 rd   +           550 wr)
==24701== LL misses:              2,462  (        1,933 rd   +           529 wr)
==24701== LL miss rate:             0.0% (          0.0%     +           0.0%  )
==24701==
==24701== Branches:         966,935,755  (  966,935,512 cond +           243 ind)
==24701== Mispredicts:       28,558,382  (   28,558,294 cond +            88 ind)
==24701== Mispred rate:             3.0% (          3.0%     +          36.2%   )
```

Figura 2. Resultados prueba primos de 256 bits

2.2. Encripción para primos de 512 bits con valores default

Para números primos de 512 bits, llaves de 1024 bits se obtienen los siguientes resultados

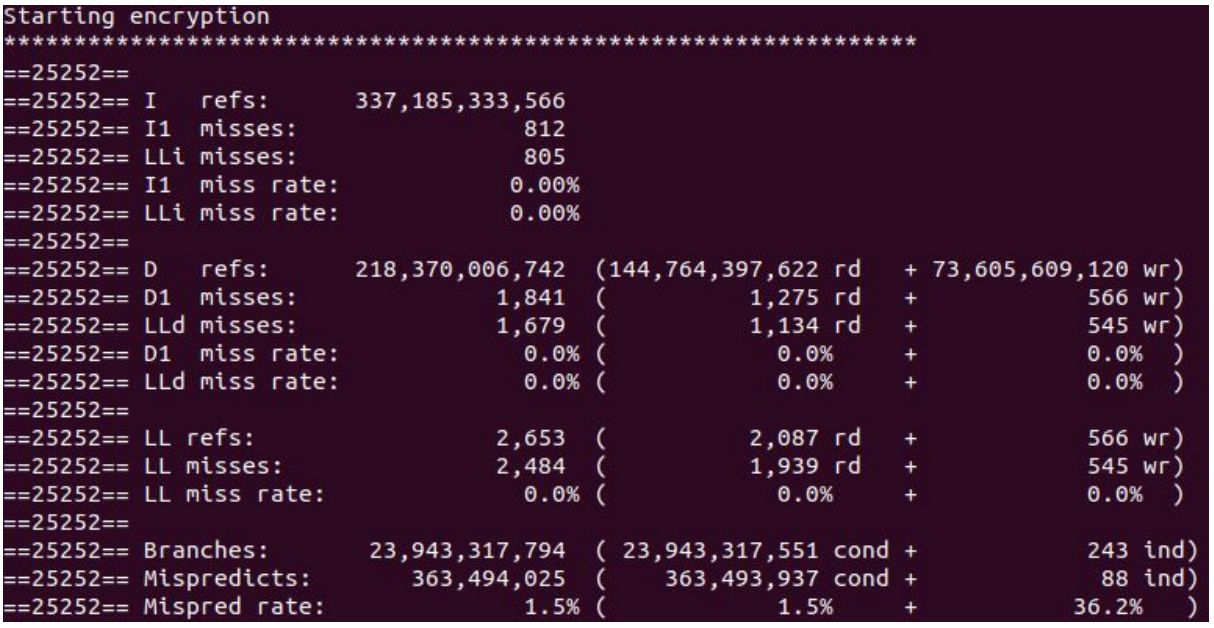

Figura 3. Resultados prueba primos de 512 bits

2.3. Encripción para primos de 256 bits cambiando el tamaño de caché a 64 bytes, y la línea de caché a 32 bytes con full asociatividad

Para números primos de 256 bits, llaves de 512 bits se obtienen los siguientes resultados, en donde el caché solamente tiene dos líneas

```
Starting encryption
********************************
==30458==
==30458== I   refs:      12,624,186,865
==30458== I1  misses:              812
==30458== LLi misses:              805
==30458== I1  miss rate:          0.00%
==30458== LLi miss rate:          0.00%
==30458==
==30458== D   refs:       8,228,791,778  (5,426,935,270 rd   + 2,801,856,508 wr)
==30458== D1  misses:     4,915,794,585  (3,524,734,989 rd   + 1,391,059,596 wr)
==30458== LLd misses:             1,657  (        1,128 rd   +           529 wr)
==30458== D1  miss rate:           59.7% (         64.9%     +          49.6%  )
==30458== LLd miss rate:            0.0% (          0.0%     +           0.0%  )
==30458==
==30458== LL refs:        4,915,795,397  (3,524,735,801 rd   + 1,391,059,596 wr)
==30458== LL misses:              2,462  (        1,933 rd   +           529 wr)
==30458== LL miss rate:             0.0% (          0.0%     +           0.0%  )
==30458==
==30458== Branches:         892,069,008  (  892,068,765 cond +           243 ind)
==30458== Mispredicts:       26,347,695  (   26,347,607 cond +            88 ind)
==30458== Mispred rate:             3.0% (          3.0%     +          36.2%   )
```

Figura 4. Resultados prueba primos de 256 bits, caso 2

2.4. Encripción para primos de 256 bits cambiando el tamaño de caché a 512 bytes, y la línea de caché a 256 bytes con full asociatividad

Para números primos de 256 bits, llaves de 512 bits se obtienen los siguientes resultados, en donde el caché solamente tiene dos líneas, es igual que el caso anterior pero más grande

```
Starting encryption
********************************
==30728==
==30728== I   refs:      12,624,186,865
==30728== I1  misses:              812
==30728== LLi misses:              805
==30728== I1  miss rate:          0.00%
==30728== LLi miss rate:          0.00%
==30728==
==30728== D   refs:       8,228,791,778  (5,426,935,270 rd   + 2,801,856,508 wr)
==30728== D1  misses:     3,827,329,590  (2,399,015,379 rd   + 1,428,314,211 wr)
==30728== LLd misses:             1,379  (          963 rd   +           416 wr)
==30728== D1  miss rate:           46.5% (         44.2%     +          51.0%  )
==30728== LLd miss rate:            0.0% (          0.0%     +           0.0%  )
==30728==
==30728== LL refs:        3,827,330,402  (2,399,016,191 rd   + 1,428,314,211 wr)
==30728== LL misses:              2,184  (        1,768 rd   +           416 wr)
==30728== LL miss rate:             0.0% (          0.0%     +           0.0%  )
==30728==
==30728== Branches:         892,069,008  (  892,068,765 cond +           243 ind)
==30728== Mispredicts:       26,347,695  (   26,347,607 cond +            88 ind)
==30728== Mispred rate:             3.0% (          3.0%     +          36.2%   )
```

3. Figura 4. Resultados prueba primos de 256 bits, caso 3

3. Análisis de resultados

Sintetizando los resultados mostrados anteriormente, se observa entre las figuras 2 y 3, que el hecho de aumentar la cantidad de bits de los primos de 256 a 512, es decir aumentar el tamaño de las llaves de 512 a 1024 bits, hace que la cantidad de veces que se intenta pedir un dato a la memoria caché de datos 1, aumente de 8 919 450 499 a 218 370 006 742, es decir aumenta alrededor de 24.48 veces. Se puede notar también que el miss rate no aumentó considerablemente, relativamente nada, por lo que sin importar la cantidad de bits y el largo de los números, la cantidad de misses se va a mantener constante, debido a la localidad espacial.
De igual forma entre las figuras 2 y 3 se observa lo siguiente respecto a los branches:

- Aumento de 966 935 755 a 23 943 317 794, en la cantidad de branches
- Aumento de 38 558 382 a 363 494 025, en la cantidad de mispredictions
- Disminución de 3.0% a 1.5%, en el porcentaje de misspredictions

De esta manera se nota un aumento de 24.76 veces en branches, 9.427 veces en misspredictions. Cabe notar que a pesar que aumentan los branches, aumenta en menos la cantidad de misspredictions.

Para la figura 4, se observa que se fuerza un miss en el cache debido a que se tienen tamaños de líneas de 32 bytes, y solamente 64 bytes para todo el tamaño de caché. Esto provoca que como los números son de 512 bits, pero 1024 de operaciones, tengan cada uno 128 bytes y aún no quepan completos en una línea de caché. Similar sucede con la figura 5, pero en esta si caben dos números utilizados, por lo que baja la cantidad de misses en el caché de nivel uno de datos D1.

4. Referencias

Valgrind.org. (2019). *Valgrind*. [online] Available at: http://valgrind.org/docs/manual/cg-manual.html#cg-manual.cgopts [Accessed 25 May 2019].