Documentation of Leibniz Formula for Pi(π) Parallel Computation

Release 1.0.0

Harun GÜLEÇ

Muhammed Ali BURSALI

May 20, 2017

Operating Systems (FCS-00017)

harunglec35@gmail.com

muhammedalibursali@gmail.com

Contents

1. Short Explanation

In mathematics, the Leibniz formula for π, named after Gottfried Leibniz, states that

$$1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \cdots = \frac{\pi}{4}.$$

It is also called Madhava–Leibniz series as it is a special case of a more general series expansion for the inverse tangent function, first discovered by the Indian mathematician Madhava of Sangamagrama in the 14th century.

2. Installation

```
$ make clean
$ make serial
$ make pi
$ make install
```

3. Usage

3.1 Serial Computation

```
$./serial numberofiteration
```

e.g. `$./serial 1000`

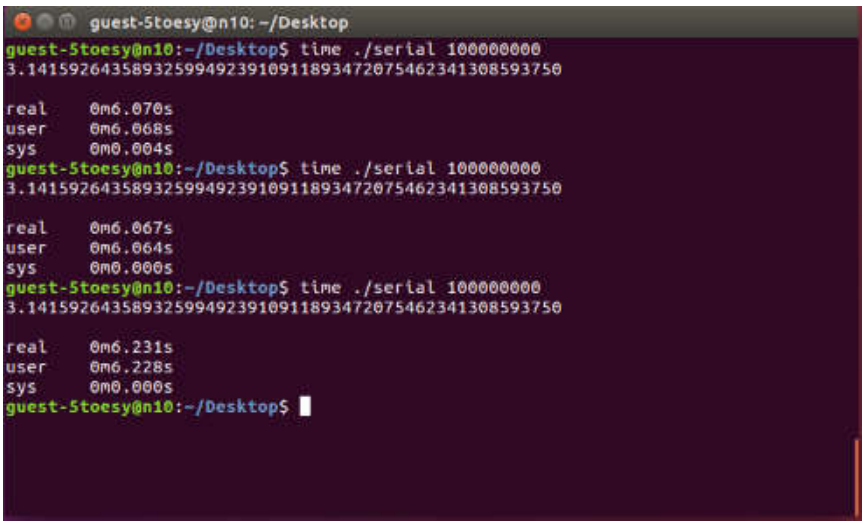
3.2 Parallel Computation

```
$./pi numberofthreads numberofiteration
```

e.g. `$./pi 4 1000`

4. Comparison

In our test laboratory the computer had 8 core processor and ubuntu 16.04 OS. According to these specifications the number of threads can be maximum 8 but we tried serial computation first, then parallel computation increased exponentially starting from 2^1. We used time utilities to check execution time.

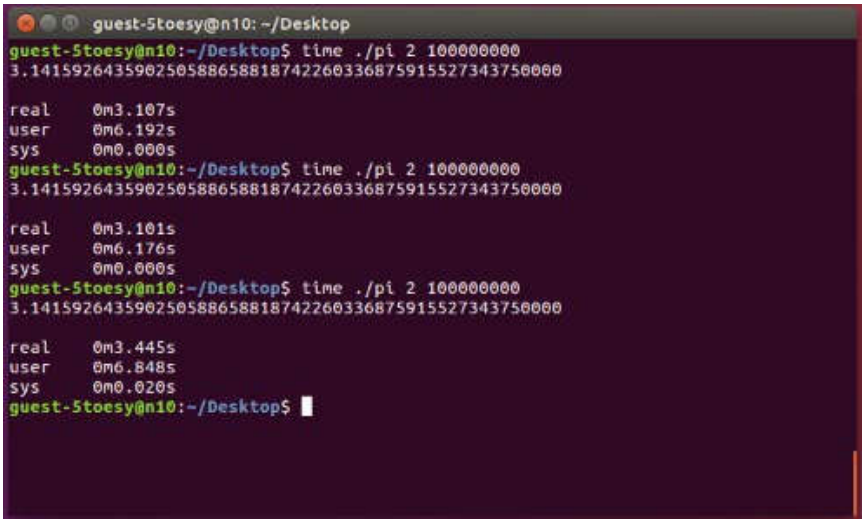


```
guest-5toesy@n10: ~/Desktop
guest-5toesy@n10:~/Desktop$ time ./serial 100000000
3.14159264358932599492391091189347207546234130859375 0

real    0m6.070s
user    0m6.068s
sys     0m0.004s
guest-5toesy@n10:~/Desktop$ time ./serial 100000000
3.14159264358932599492391091189347207546234130859375 0

real    0m6.067s
user    0m6.064s
sys     0m0.000s
guest-5toesy@n10:~/Desktop$ time ./serial 100000000
3.14159264358932599492391091189347207546234130859375 0

real    0m6.231s
user    0m6.228s
sys     0m0.000s
guest-5toesy@n10:~/Desktop$
```



```
guest-5toesy@n10: ~/Desktop
guest-5toesy@n10:~/Desktop$ time ./pi 2 100000000
3.14159264359025058865881874226033687591552734375 0000

real    0m3.107s
user    0m6.192s
sys     0m0.000s
guest-5toesy@n10:~/Desktop$ time ./pi 2 100000000
3.14159264359025058865881874226033687591552734375 0000

real    0m3.101s
user    0m6.176s
sys     0m0.000s
guest-5toesy@n10:~/Desktop$ time ./pi 2 100000000
3.14159264359025058865881874226033687591552734375 0000

real    0m3.445s
user    0m6.848s
sys     0m0.020s
guest-5toesy@n10:~/Desktop$
```

We can see clearly the real time of execution decreased to half. It means that parallel computation with 2 threads x2 faster.

```
guest-5toesy@n10: ~/Desktop
guest-5toesy@n10:~/Desktop$ time ./pi 4 100000000
3.14159264358981715759000508114695549011230468750000

real    0m1.801s
user    0m6.480s
sys     0m0.000s
guest-5toesy@n10:~/Desktop$ time ./pi 4 100000000
3.14159264358981715759000508114695549011230468750000

real    0m1.578s
user    0m6.260s
sys     0m0.000s
guest-5toesy@n10:~/Desktop$ time ./pi 4 100000000
3.14159264358981715759000508114695549011230468750000

real    0m1.583s
user    0m6.268s
sys     0m0.004s
guest-5toesy@n10:~/Desktop$
```

```
guest-5toesy@n10: ~/Desktop
guest-5toesy@n10:~/Desktop$ time ./pi 8 100000000
3.14159264358987977416859393997583538293838500976562 5

real    0m0.901s
user    0m7.124s
sys     0m0.000s
guest-5toesy@n10:~/Desktop$ time ./pi 8 100000000
3.14159264358987933007938408991321921348571777343750 0

real    0m0.907s
user    0m7.120s
sys     0m0.000s
guest-5toesy@n10:~/Desktop$ time ./pi 8 100000000
3.14159264358987977416859393997583538293838500976562 5

real    0m0.909s
user    0m7.124s
sys     0m0.000s
guest-5toesy@n10:~/Desktop$
```

Parallel computation with 4 threads x4 faster and with 8 threads x8 faster.

```
guest-5toesy@n10: ~/Desktop
guest-5toesy@n10:~/Desktop$ time ./pi 16 100000000
3.14159264358989620546935839229263365268707275390625 0

real    0m0.962s
user    0m7.276s
sys     0m0.000s
guest-5toesy@n10:~/Desktop$ time ./pi 16 100000000
3.14159264358989620546935839229263365268707275390625 0

real    0m0.972s
user    0m7.376s
sys     0m0.000s
guest-5toesy@n10:~/Desktop$ time ./pi 16 100000000
3.14159264358989620546935839229263365268707275390625 0

real    0m0.956s
user    0m7.352s
sys     0m0.000s
guest-5toesy@n10:~/Desktop$
```

```
guest-5toesy@n10: ~/Desktop
guest-5toesy@n10:~/Desktop$ time ./pi 32 100000000
3.14159264358966439090181665960699319839477539062500 0

real    0m0.979s
user    0m7.520s
sys     0m0.000s
guest-5toesy@n10:~/Desktop$ time ./pi 32 100000000
3.14159264358966350272339695948176085948944091796875 0

real    0m0.974s
user    0m7.520s
sys     0m0.000s
guest-5toesy@n10:~/Desktop$ time ./pi 32 100000000
3.14159264358966350272339695948176085948944091796875 0

real    0m0.965s
user    0m7.504s
sys     0m0.008s
guest-5toesy@n10:~/Desktop$
```

As you can see above, more than 8 threads don't effect the real time of execution because our test computer has 8 cores. That means time will decreased until 8 threads. So the computer can execute 8 threads at the same time.