

Assignment OpenMP  
Intel i7 10700  
งานเดี่ยว  
ส่งวันที่ 11 ตุลาคม เวลา 13.00 น  
คิดเป็น 10% ของเกรด

จัดตารางการทดลองว่าใครจะทดลองก่อนหลัง

Log in ด้วย secure shell ดังนี้

ssh [hpc@161.246.5.86](mailto:hpc@161.246.5.86) -p 22

password = hpc948323

ตรวจสอบว่ามีผู้ใช้อื่นใช้เครื่องพร้อมกันหรือไม่ โดยใช้คำสั่ง top ดังนี้

```
top - 17:02:32 up 14 days, 6:31, 3 users,
```

หากมีเพียง 3-4 คนให้ทดลองต่อได้

copy Mattson\_OMP\_exercises เป็นโฟลเดอร์ของตนเอง ด้วยคำสั่งต่อไปนี้

```
$ cd Documents
```

```
$ mkdir ชื่อโฟลเดอร์
```

```
$ cp Mattson_OMP_exercises/*.* ชื่อโฟลเดอร์ -r
```

ย้าย current directory ไปยัง ชื่อโฟลเดอร์ ใหม่

```
$ cd ชื่อโฟลเดอร์
```

ทดลองรัน pi ซึ่งเป็น executable ของ pi.c จำนวน 3 รอบโดย

```
$ ./pi
```

หาค่าเฉลี่ยแล้วบันทึกเวลาลงในตารางผลการทดลอง

ทดลองรัน pi\_loop ซึ่งเป็น executable ของ pi\_loop.c จำนวน 3 รอบโดย

```
$ cd solutions
```

```
$ ./pi_loop
```

หาค่าเฉลี่ยแล้วบันทึกเวลาลงในตารางผลการทดลอง

ทดลองรัน pi\_simple ซึ่งเป็น executable ของ pi\_spmd\_simple.c จำนวน 3 รอบโดย

```
$ ./pi_simple
```

หาค่าเฉลี่ยแล้วบันทึกเวลาลงในตารางผลการทดลอง

ทดลองรัน pi\_final ซึ่งเป็น executable ของ pi\_spmd\_final.c จำนวน 3 รอบโดย

```
$ ./pi_final
```

หาค่าเฉลี่ยแล้วบันทึกเวลาลงในตารางผลการทดลอง

ปรับแก้ num\_steps เป็น 1000 ล้าน ใน source file ทุกไฟล์ คือ pi.c, pi\_loop.c, pi\_spmd\_simple.c, pi\_spmd\_final.c แล้วทำการคอมไพล์และทำการทดลองใหม่อีกรอบ แล้วบันทึกผลในตาราง ต่อไปนี้ โปรดสังเกตว่า ไฟล์อยู่กันคนละไดเรกทอรี

```
$ gcc -fopenmp pi.c -o pi
$ gcc -fopenmp pi_loop.c -o pi_loop
$ gcc -fopenmp pi_simple.c -o pi_spmd_simple.c
$ gcc -fopenmp pi_final.c -o pi_spmd_final.c
```

เวลาเฉลี่ย	pi	pi_loop	pi_simple	pi_final
1 Thread num_steps=100 ล้าน	0.19345567	0.23200267	0.30295633	0.209269
8 Thread num_steps=100 ล้าน	-	0.47550867	0.286548	0.04821033
16 Thread num_steps=100 ล้าน	-	0.35684267	0.22	0.025462
1 Thread num_steps=1000 ล้าน	1.93594933	2.31804767	3.02926867	2.09092133
8 Thread num_steps=1000 ล้าน	-	4.59195433	4.295986	0.294676
16 Thread num_steps=1000 ล้าน	-	3.59529467	2.25268933	0.253631
หน่วยเป็นวินาที (sec)				

วิเคราะห์และสรุปผลการทดลอง โดยไม่ลอก นศ สามารถถกเถียงกันได้ว่าเกิดอะไรขึ้น แต่ให้เขียนวิเคราะห์และสรุปผลการทดลองด้วยตัวเอง

pi

```
hpc@i710700:~/Documents/Aekarin_64015172/Mattson_OMP_exercises$ gcc -fopenmp pi.c -o pi
hpc@i710700:~/Documents/Aekarin_64015172/Mattson_OMP_exercises$ ./pi
pi with 100000000 steps is 3.141593 in 0.193491 seconds
hpc@i710700:~/Documents/Aekarin_64015172/Mattson_OMP_exercises$ ./pi
pi with 100000000 steps is 3.141593 in 0.193348 seconds
hpc@i710700:~/Documents/Aekarin_64015172/Mattson_OMP_exercises$ ./pi
pi with 100000000 steps is 3.141593 in 0.193528 seconds

hpc@i710700:~/Documents/Aekarin_64015172/Mattson_OMP_exercises$ ./pi_1000
pi with 1000000000 steps is 3.141593 in 1.935028 seconds
hpc@i710700:~/Documents/Aekarin_64015172/Mattson_OMP_exercises$ ./pi_1000
pi with 1000000000 steps is 3.141593 in 1.936000 seconds
hpc@i710700:~/Documents/Aekarin_64015172/Mattson_OMP_exercises$ ./pi_1000
pi with 1000000000 steps is 3.141593 in 1.936820 seconds
```

(รูปภาพผลการทดลอง pi 100 ล้าน และ 1000 ล้าน)

โปรแกรมนี้คำนวณอินทิกรัลของฟังก์ชัน  $4/(1+x^2)$  จาก 0 ถึง 1 (ค่า pi)

การทำงานของโปรแกรม:

1. เธรดหลักจะเริ่มต้นเวลาโดยใช้ฟังก์ชัน `omp_get_wtime()`
2. เธรดหลักจะวนลูปตั้งแต่ 1 ถึง `num_steps` และคำนวณผลรวมของฟังก์ชัน  $4/(1+x^2)$  โดยใช้ตัวแปร `sum`
3. หลังจากที่ยวนลูปสิ้นสุดลง เธรดหลักจะคำนวณค่าของ pi โดยใช้สูตร  $pi = step * sum$
4. เธรดหลักจะแสดงผลค่าของ pi และเวลาที่ใช้ในการคำนวณโดยใช้ฟังก์ชัน `printf()`

โปรแกรมนี้ไม่มีปัญหาอะไร แต่ประสิทธิภาพของโปรแกรมจะไม่ได้เมื่อใช้กับจำนวนเธรดที่มาก เนื่องจากโปรแกรมนี้ไม่ได้ใช้ประโยชน์จาก OpenMP อย่างเต็มที่

เพื่อปรับปรุงประสิทธิภาพของโปรแกรม สามารถใช้ OpenMP เพื่อแบ่งงานการคำนวณออกเป็นหลายๆ เธรด วิธีนี้จะช่วยให้โปรแกรมสามารถคำนวณอินทิกรัลได้เร็วขึ้นเมื่อใช้กับจำนวนเธรดที่มาก

# pi\_loop

```
hpc@1718700:~/Documents/Aekarin_64015172/Mattson_OHP_exercises/solutions$ ./pi_loop
num_threads = 1
pi is 3.141593 in 0.232716 seconds and 1 threads

num_threads = 2
pi is 3.141593 in 0.386975 seconds and 2 threads

num_threads = 3
pi is 3.141593 in 0.428890 seconds and 3 threads

num_threads = 4
pi is 3.141593 in 0.427185 seconds and 4 threads

num_threads = 5
pi is 3.141593 in 0.449885 seconds and 5 threads

num_threads = 6
pi is 3.141592 in 0.559329 seconds and 6 threads

num_threads = 7
pi is 3.141591 in 0.534814 seconds and 7 threads

num_threads = 8
pi is 3.141590 in 0.517066 seconds and 8 threads

num_threads = 9
pi is 3.141594 in 0.509203 seconds and 9 threads

num_threads = 10
pi is 3.141596 in 0.468992 seconds and 10 threads

num_threads = 11
pi is 3.141595 in 0.455281 seconds and 11 threads

num_threads = 12
pi is 3.141587 in 0.436772 seconds and 12 threads

num_threads = 13
pi is 3.141636 in 0.417105 seconds and 13 threads

num_threads = 14
pi is 3.141597 in 0.405121 seconds and 14 threads

num_threads = 15
pi is 3.141594 in 0.387959 seconds and 15 threads

num_threads = 16
pi is 3.141594 in 0.379027 seconds and 16 threads
```

```
hpc@1718700:~/Documents/Aekarin_64015172/Mattson_OHP_exercises/solutions$ ./pi_loop
num_threads = 1
pi is 3.141593 in 0.231780 seconds and 1 threads

num_threads = 2
pi is 3.141593 in 0.338931 seconds and 2 threads

num_threads = 3
pi is 3.141593 in 0.374745 seconds and 3 threads

num_threads = 4
pi is 3.141594 in 0.418591 seconds and 4 threads

num_threads = 5
pi is 3.141593 in 0.378889 seconds and 5 threads

num_threads = 6
pi is 3.141615 in 0.388565 seconds and 6 threads

num_threads = 7
pi is 3.141586 in 0.411403 seconds and 7 threads

num_threads = 8
pi is 3.141598 in 0.451814 seconds and 8 threads

num_threads = 9
pi is 3.141593 in 0.415780 seconds and 9 threads

num_threads = 10
pi is 3.141595 in 0.417322 seconds and 10 threads

num_threads = 11
pi is 3.141594 in 0.438647 seconds and 11 threads

num_threads = 12
pi is 3.141594 in 0.397982 seconds and 12 threads

num_threads = 13
pi is 3.141620 in 0.394401 seconds and 13 threads

num_threads = 14
pi is 3.141564 in 0.381160 seconds and 14 threads

num_threads = 15
pi is 3.141607 in 0.357211 seconds and 15 threads

num_threads = 16
pi is 3.141639 in 0.345080 seconds and 16 threads
```

```
hpc@1718700:~/Documents/Aekarin_64015172/Mattson_OHP_exercises/solutions$ ./pi_loop
num_threads = 1
pi is 3.141593 in 0.231512 seconds and 1 threads

num_threads = 2
pi is 3.141593 in 0.325938 seconds and 2 threads

num_threads = 3
pi is 3.141592 in 0.373997 seconds and 3 threads

num_threads = 4
pi is 3.141593 in 0.375930 seconds and 4 threads

num_threads = 5
pi is 3.141593 in 0.346842 seconds and 5 threads

num_threads = 6
pi is 3.141623 in 0.366594 seconds and 6 threads

num_threads = 7
pi is 3.141592 in 0.450801 seconds and 7 threads

num_threads = 8
pi is 3.141604 in 0.457646 seconds and 8 threads

num_threads = 9
pi is 3.141593 in 0.440344 seconds and 9 threads

num_threads = 10
pi is 3.141582 in 0.446288 seconds and 10 threads

num_threads = 11
pi is 3.141671 in 0.432499 seconds and 11 threads

num_threads = 12
pi is 3.141807 in 0.413070 seconds and 12 threads

num_threads = 13
pi is 3.141679 in 0.390632 seconds and 13 threads

num_threads = 14
pi is 3.141654 in 0.375834 seconds and 14 threads

num_threads = 15
pi is 3.141597 in 0.361694 seconds and 15 threads

num_threads = 16
pi is 3.141703 in 0.346421 seconds and 16 threads
```

```
hpc@1718700:~/Documents/Aekarin_64015172/Mattson_OHP_exercises/solutions$ ./pi_loop_1000
num_threads = 1
pi is 3.141593 in 2.316029 seconds and 1 threads

num_threads = 2
pi is 3.141593 in 2.514365 seconds and 2 threads

num_threads = 3
pi is 3.141593 in 3.439130 seconds and 3 threads

num_threads = 4
pi is 3.141593 in 3.815319 seconds and 4 threads

num_threads = 5
pi is 3.141593 in 4.080334 seconds and 5 threads

num_threads = 6
pi is 3.141597 in 4.288223 seconds and 6 threads

num_threads = 7
pi is 3.141591 in 4.452287 seconds and 7 threads

num_threads = 8
pi is 3.141592 in 4.959945 seconds and 8 threads

num_threads = 9
pi is 3.141594 in 4.525663 seconds and 9 threads

num_threads = 10
pi is 3.141593 in 4.171147 seconds and 10 threads

num_threads = 11
pi is 3.141541 in 4.113266 seconds and 11 threads

num_threads = 12
pi is 3.141767 in 4.093367 seconds and 12 threads

num_threads = 13
pi is 3.141709 in 3.787243 seconds and 13 threads

num_threads = 14
pi is 3.141683 in 3.748866 seconds and 14 threads

num_threads = 15
pi is 3.141644 in 3.705949 seconds and 15 threads

num_threads = 16
pi is 3.141629 in 3.558420 seconds and 16 threads
```

```
hpc@1718700:~/Documents/Aekarin_64015172/Mattson_OHP_exercises/solutions$ ./pi_loop_1000
num_threads = 1
pi is 3.141593 in 2.319224 seconds and 1 threads

num_threads = 2
pi is 3.141593 in 2.648889 seconds and 2 threads

num_threads = 3
pi is 3.141593 in 3.237181 seconds and 3 threads

num_threads = 4
pi is 3.141593 in 3.242517 seconds and 4 threads

num_threads = 5
pi is 3.141598 in 3.148239 seconds and 5 threads

num_threads = 6
pi is 3.141593 in 3.197217 seconds and 6 threads

num_threads = 7
pi is 3.141590 in 3.244097 seconds and 7 threads

num_threads = 8
pi is 3.141594 in 3.189899 seconds and 8 threads

num_threads = 9
pi is 3.141604 in 3.226333 seconds and 9 threads

num_threads = 10
pi is 3.141778 in 3.323354 seconds and 10 threads

num_threads = 11
pi is 3.141729 in 3.324698 seconds and 11 threads

num_threads = 12
pi is 3.141684 in 3.203371 seconds and 12 threads

num_threads = 13
pi is 3.141697 in 3.160530 seconds and 13 threads

num_threads = 14
pi is 3.141625 in 3.267700 seconds and 14 threads

num_threads = 15
pi is 3.141610 in 3.228204 seconds and 15 threads

num_threads = 16
pi is 3.141611 in 3.222235 seconds and 16 threads
```

```
hpc@1718700:~/Documents/Aekarin_64015172/Mattson_OHP_exercises/solutions$ ./pi_loop_1000
num_threads = 1
pi is 3.141593 in 2.318890 seconds and 1 threads

num_threads = 2
pi is 3.141593 in 3.772390 seconds and 2 threads

num_threads = 3
pi is 3.141593 in 4.656778 seconds and 3 threads

num_threads = 4
pi is 3.141593 in 4.635795 seconds and 4 threads

num_threads = 5
pi is 3.141593 in 4.609663 seconds and 5 threads

num_threads = 6
pi is 3.141592 in 5.408820 seconds and 6 threads

num_threads = 7
pi is 3.141592 in 5.687257 seconds and 7 threads

num_threads = 8
pi is 3.141594 in 5.626819 seconds and 8 threads

num_threads = 9
pi is 3.141598 in 5.284597 seconds and 9 threads

num_threads = 10
pi is 3.141598 in 5.198877 seconds and 10 threads

num_threads = 11
pi is 3.141598 in 4.819721 seconds and 11 threads

num_threads = 12
pi is 3.141595 in 4.576362 seconds and 12 threads

num_threads = 13
pi is 3.141595 in 4.576252 seconds and 13 threads

num_threads = 14
pi is 3.141595 in 4.345220 seconds and 14 threads

num_threads = 15
pi is 3.141595 in 4.129740 seconds and 15 threads

num_threads = 16
pi is 3.141594 in 4.005229 seconds and 16 threads
```

## รูปภาพผลการทดลอง pi\_loop 100 ล้าน และ 1000 ล้าน

โปรแกรมนี้ทำงานโดยแบ่งการวนซ้ำของลูป for ออกเป็นหลายๆ เธรด แต่ละเธรดจะคำนวณผลรวมของฟังก์ชัน  $4/(1+x^2)$  สำหรับจำนวนเต็ม  $i$  ที่แตกต่างกัน หลังจากทีเธรดแต่ละตัวคำนวณผลรวมของส่วนเสร็จแล้ว เธรดนั้นจะรวมผลรวมของส่วนเข้ากับผลรวมของเธรดอื่นๆ เพื่อให้ได้ผลรวมสุดท้าย

การทำงานของเธรด:

1. เธรดหลักจะสร้างทีมเธรดขึ้นมาโดยใช้คำสั่ง `#pragma omp parallel`.
2. เธรดแต่ละตัวในทีมจะคำนวณผลรวมของฟังก์ชัน  $4/(1+x^2)$  สำหรับจำนวนเต็ม  $i$  ที่แตกต่างกัน โดยใช้คำสั่ง `#pragma omp for reduction(+:sum)`.
3. หลังจากทีเธรดแต่ละตัวคำนวณผลรวมของส่วนเสร็จแล้ว เธรดนั้นจะรวมผลรวมของส่วนเข้ากับผลรวมของเธรดอื่นๆ โดยใช้คำสั่ง `reduction(+:sum)`.
4. เธรดหลักจะรวมผลรวมของเธรดทั้งหมดเข้าด้วยกันเพื่อให้ได้ผลรวมสุดท้าย

**ปัญหาในโปรแกรมนี้คือ**เธรดแต่ละตัวจะคำนวณค่าของตัวแปร `sum` แยกจากกันโดยไม่มีการใช้โครโนซ์ สิ่งนี้อาจทำให้เกิดปัญหาการแข่งขันข้อมูล (race condition) ซึ่งหมายความว่าค่าของตัวแปร `sum` อาจไม่ถูกต้อง

# Pi\_simple

```
root@187080:~/Documents/Aekarin_64015172/Hattson_OMP_exercises/solutions$ gcc -fopenmp -std=c11mpi8080/Documents/Aekarin_64015172/Hattson_OMP_exercises/solutions$ ./pi_simple
num_threads = 1
pi is 3.141593 in 0.302973 seconds 1 thrs
num_threads = 2
pi is 3.141593 in 0.318964 seconds 2 thrs
num_threads = 3
pi is 3.141593 in 0.394165 seconds 3 thrs
num_threads = 4
pi is 3.141593 in 0.417861 seconds 4 thrs
num_threads = 5
pi is 3.141593 in 0.336622 seconds 5 thrs
num_threads = 6
pi is 3.141593 in 0.341891 seconds 6 thrs
num_threads = 7
pi is 3.141593 in 0.303948 seconds 7 thrs
num_threads = 8
pi is 3.141593 in 0.313467 seconds 8 thrs
num_threads = 9
pi is 3.141593 in 0.269915 seconds 9 thrs
num_threads = 10
pi is 3.141593 in 0.249448 seconds 10 thrs
num_threads = 11
pi is 3.141593 in 0.237567 seconds 11 thrs
num_threads = 12
pi is 3.141593 in 0.228180 seconds 12 thrs
num_threads = 13
pi is 3.141593 in 0.233066 seconds 13 thrs
num_threads = 14
pi is 3.141593 in 0.228513 seconds 14 thrs
num_threads = 15
pi is 3.141593 in 0.217557 seconds 15 thrs
num_threads = 16
pi is 3.141593 in 0.201162 seconds 16 thrs
```

```
root@1710700:~/Documents/Aekarin_64015172/Hattson_OMP_exercises/solutions$ ./pi_simple
num_threads = 1
pi is 3.141593 in 0.302955 seconds 1 thrs
num_threads = 2
pi is 3.141593 in 0.230476 seconds 2 thrs
num_threads = 3
pi is 3.141593 in 0.318885 seconds 3 thrs
num_threads = 4
pi is 3.141593 in 0.275875 seconds 4 thrs
num_threads = 5
pi is 3.141593 in 0.325899 seconds 5 thrs
num_threads = 6
pi is 3.141593 in 0.344469 seconds 6 thrs
num_threads = 7
pi is 3.141593 in 0.290489 seconds 7 thrs
num_threads = 8
pi is 3.141593 in 0.275176 seconds 8 thrs
num_threads = 9
pi is 3.141593 in 0.254862 seconds 9 thrs
num_threads = 10
pi is 3.141593 in 0.233139 seconds 10 thrs
num_threads = 11
pi is 3.141593 in 0.265293 seconds 11 thrs
num_threads = 12
pi is 3.141593 in 0.260340 seconds 12 thrs
num_threads = 13
pi is 3.141593 in 0.305958 seconds 13 thrs
num_threads = 14
pi is 3.141593 in 0.287645 seconds 14 thrs
num_threads = 15
pi is 3.141593 in 0.281059 seconds 15 thrs
num_threads = 16
pi is 3.141593 in 0.281834 seconds 16 thrs
```

```
root@1710700:~/Documents/Aekarin_64015172/Hattson_OMP_exercises/solutions$ ./pi_simple
num_threads = 1
pi is 3.141593 in 0.302941 seconds 1 thrs
num_threads = 2
pi is 3.141593 in 0.294353 seconds 2 thrs
num_threads = 3
pi is 3.141593 in 0.289594 seconds 3 thrs
num_threads = 4
pi is 3.141593 in 0.291581 seconds 4 thrs
num_threads = 5
pi is 3.141593 in 0.307029 seconds 5 thrs
num_threads = 6
pi is 3.141593 in 0.348697 seconds 6 thrs
num_threads = 7
pi is 3.141593 in 0.304601 seconds 7 thrs
num_threads = 8
pi is 3.141593 in 0.271801 seconds 8 thrs
num_threads = 9
pi is 3.141593 in 0.251882 seconds 9 thrs
num_threads = 10
pi is 3.141593 in 0.233844 seconds 10 thrs
num_threads = 11
pi is 3.141593 in 0.223646 seconds 11 thrs
num_threads = 12
pi is 3.141593 in 0.214183 seconds 12 thrs
num_threads = 13
pi is 3.141593 in 0.212653 seconds 13 thrs
num_threads = 14
pi is 3.141593 in 0.197427 seconds 14 thrs
num_threads = 15
pi is 3.141593 in 0.186758 seconds 15 thrs
num_threads = 16
pi is 3.141593 in 0.177004 seconds 16 thrs
```

```
root@1710700:~/Documents/Aekarin_64015172/Hattson_OMP_exercises/solutions$ ./pi_simple_1000
num_threads = 1
pi is 3.141593 in 3.028829 seconds 1 thrs
num_threads = 2
pi is 3.141593 in 3.750945 seconds 2 thrs
num_threads = 3
pi is 3.141593 in 4.117874 seconds 3 thrs
num_threads = 4
pi is 3.141593 in 4.388932 seconds 4 thrs
num_threads = 5
pi is 3.141593 in 4.783999 seconds 5 thrs
num_threads = 6
pi is 3.141593 in 5.133961 seconds 6 thrs
num_threads = 7
pi is 3.141593 in 5.258390 seconds 7 thrs
num_threads = 8
pi is 3.141593 in 5.665441 seconds 8 thrs
num_threads = 9
pi is 3.141593 in 5.185722 seconds 9 thrs
num_threads = 10
pi is 3.141593 in 4.590706 seconds 10 thrs
num_threads = 11
pi is 3.141593 in 4.128230 seconds 11 thrs
num_threads = 12
pi is 3.141593 in 3.856188 seconds 12 thrs
num_threads = 13
pi is 3.141593 in 3.544361 seconds 13 thrs
num_threads = 14
pi is 3.141593 in 3.105355 seconds 14 thrs
num_threads = 15
pi is 3.141593 in 3.013557 seconds 15 thrs
num_threads = 16
pi is 3.141593 in 2.613390 seconds 16 thrs
```

```
root@1710700:~/Documents/Aekarin_64015172/Hattson_OMP_exercises/solutions$ ./pi_simple_10000
num_threads = 1
pi is 3.141593 in 3.029789 seconds 1 thrs
num_threads = 2
pi is 3.141593 in 6.003995 seconds 2 thrs
num_threads = 3
pi is 3.141593 in 4.936087 seconds 3 thrs
num_threads = 4
pi is 3.141593 in 4.349717 seconds 4 thrs
num_threads = 5
pi is 3.141593 in 4.056185 seconds 5 thrs
num_threads = 6
pi is 3.141593 in 3.766877 seconds 6 thrs
num_threads = 7
pi is 3.141593 in 3.086378 seconds 7 thrs
num_threads = 8
pi is 3.141593 in 4.117615 seconds 8 thrs
num_threads = 9
pi is 3.141593 in 4.291119 seconds 9 thrs
num_threads = 10
pi is 3.141593 in 4.191127 seconds 10 thrs
num_threads = 11
pi is 3.141593 in 3.808101 seconds 11 thrs
num_threads = 12
pi is 3.141593 in 3.387534 seconds 12 thrs
num_threads = 13
pi is 3.141593 in 3.273856 seconds 13 thrs
num_threads = 14
pi is 3.141593 in 3.054848 seconds 14 thrs
num_threads = 15
pi is 3.141593 in 2.921398 seconds 15 thrs
num_threads = 16
pi is 3.141593 in 2.828842 seconds 16 thrs
```

```
root@1710700:~/Documents/Aekarin_64015172/Hattson_OMP_exercises/solutions$ ./pi_simple_10000
num_threads = 1
pi is 3.141593 in 3.029188 seconds 1 thrs
num_threads = 2
pi is 3.141593 in 3.934257 seconds 2 thrs
num_threads = 3
pi is 3.141593 in 4.340859 seconds 3 thrs
num_threads = 4
pi is 3.141593 in 4.901620 seconds 4 thrs
num_threads = 5
pi is 3.141593 in 4.177503 seconds 5 thrs
num_threads = 6
pi is 3.141593 in 3.705297 seconds 6 thrs
num_threads = 7
pi is 3.141593 in 3.405839 seconds 7 thrs
num_threads = 8
pi is 3.141593 in 3.104981 seconds 8 thrs
num_threads = 9
pi is 3.141593 in 2.817822 seconds 9 thrs
num_threads = 10
pi is 3.141593 in 2.698444 seconds 10 thrs
num_threads = 11
pi is 3.141593 in 2.512855 seconds 11 thrs
num_threads = 12
pi is 3.141593 in 2.383126 seconds 12 thrs
num_threads = 13
pi is 3.141593 in 2.250497 seconds 13 thrs
num_threads = 14
pi is 3.141593 in 2.064081 seconds 14 thrs
num_threads = 15
pi is 3.141593 in 1.948803 seconds 15 thrs
num_threads = 16
pi is 3.141593 in 1.767887 seconds 16 thrs
```

รูปภาพผลการทดลอง pi\_simple 100 ล้าน และ 1000 ล้าน

SPMD (Single Program Multiple Data) เป็นรูปแบบการเขียนโปรแกรมแบบคู่ขนานในรูปแบบนี้ เธรดแต่ละตัวจะรันโปรแกรมเดียวกัน แต่จะมีข้อมูลที่แตกต่างกัน

การทำงานของเธรด:

1. เธรดหลักจะสร้างทีมเธรดขึ้นมาโดยใช้คำสั่ง `#pragma omp parallel`.
2. เธรดแต่ละตัวในทีมจะคำนวณผลรวมของฟังก์ชัน  $4/(1+x^2)$  สำหรับจำนวนเต็ม  $i$  ที่แตกต่างกัน โดยใช้คำสั่ง `for`.
3. หลังจากทีเธรดแต่ละตัวคำนวณผลรวมของตนเองเสร็จแล้ว เธรดนั้นจะรวมผลรวมของตนเองเข้ากับผลรวมของเธรดอื่นๆ โดยใช้คำสั่ง `sum[id]`.
4. เธรดหลักจะรวมผลรวมของเธรดทั้งหมดเข้าด้วยกันเพื่อให้ได้ผลรวมสุดท้าย

**ปัญหาในโปรแกรมนี้คือ**เธรดแต่ละตัวจะคำนวณค่าของตัวแปร `sum[id]` แยกจากกันโดยไม่มีการซิงโครไนซ์ สิ่งนี้อาจทำให้เกิดปัญหาการแข่งขันข้อมูล (race condition) ซึ่งหมายความว่าค่าของตัวแปร `sum[id]` อาจไม่ถูกต้อง



# Pi\_final

```
hp@1718700:~/Documents/Aekarin_64015172/Mattson_OMP_exercises/solutions$ ./pi_final
num_threads = 1
pi is 3.141593 in 0.209189 seconds 1 threads
num_threads = 2
pi is 3.141593 in 0.104670 seconds 2 threads
num_threads = 3
pi is 3.141593 in 0.069792 seconds 3 threads
num_threads = 4
pi is 3.141593 in 0.052776 seconds 4 threads
num_threads = 5
pi is 3.141593 in 0.042794 seconds 5 threads
num_threads = 6
pi is 3.141593 in 0.035668 seconds 6 threads
num_threads = 7
pi is 3.141593 in 0.030574 seconds 7 threads
num_threads = 8
pi is 3.141593 in 0.048213 seconds 8 threads
num_threads = 9
pi is 3.141593 in 0.042928 seconds 9 threads
num_threads = 10
pi is 3.141593 in 0.040381 seconds 10 threads
num_threads = 11
pi is 3.141593 in 0.036737 seconds 11 threads
num_threads = 12
pi is 3.141593 in 0.033762 seconds 12 threads
num_threads = 13
pi is 3.141593 in 0.031200 seconds 13 threads
num_threads = 14
pi is 3.141593 in 0.028961 seconds 14 threads
num_threads = 15
pi is 3.141593 in 0.027035 seconds 15 threads
num_threads = 16
pi is 3.141593 in 0.025383 seconds 16 threads
```

```
hp@1718700:~/Documents/Aekarin_64015172/Mattson_OMP_exercises/solutions$ ./pi_final
num_threads = 1
pi is 3.141593 in 0.209235 seconds 1 threads
num_threads = 2
pi is 3.141593 in 0.104614 seconds 2 threads
num_threads = 3
pi is 3.141593 in 0.069919 seconds 3 threads
num_threads = 4
pi is 3.141593 in 0.052832 seconds 4 threads
num_threads = 5
pi is 3.141593 in 0.042816 seconds 5 threads
num_threads = 6
pi is 3.141593 in 0.035705 seconds 6 threads
num_threads = 7
pi is 3.141593 in 0.030584 seconds 7 threads
num_threads = 8
pi is 3.141593 in 0.048210 seconds 8 threads
num_threads = 9
pi is 3.141593 in 0.042927 seconds 9 threads
num_threads = 10
pi is 3.141593 in 0.040374 seconds 10 threads
num_threads = 11
pi is 3.141593 in 0.036774 seconds 11 threads
num_threads = 12
pi is 3.141593 in 0.033763 seconds 12 threads
num_threads = 13
pi is 3.141593 in 0.031200 seconds 13 threads
num_threads = 14
pi is 3.141593 in 0.028969 seconds 14 threads
num_threads = 15
pi is 3.141593 in 0.027045 seconds 15 threads
num_threads = 16
pi is 3.141593 in 0.025378 seconds 16 threads
```

```
hp@1718700:~/Documents/Aekarin_64015172/Mattson_OMP_exercises/solutions$ ./pi_final
num_threads = 1
pi is 3.141593 in 0.209383 seconds 1 threads
num_threads = 2
pi is 3.141593 in 0.104724 seconds 2 threads
num_threads = 3
pi is 3.141593 in 0.069982 seconds 3 threads
num_threads = 4
pi is 3.141593 in 0.052765 seconds 4 threads
num_threads = 5
pi is 3.141593 in 0.042788 seconds 5 threads
num_threads = 6
pi is 3.141593 in 0.035678 seconds 6 threads
num_threads = 7
pi is 3.141593 in 0.030591 seconds 7 threads
num_threads = 8
pi is 3.141593 in 0.048280 seconds 8 threads
num_threads = 9
pi is 3.141593 in 0.042924 seconds 9 threads
num_threads = 10
pi is 3.141593 in 0.040367 seconds 10 threads
num_threads = 11
pi is 3.141593 in 0.036734 seconds 11 threads
num_threads = 12
pi is 3.141593 in 0.033762 seconds 12 threads
num_threads = 13
pi is 3.141593 in 0.031183 seconds 13 threads
num_threads = 14
pi is 3.141593 in 0.028969 seconds 14 threads
num_threads = 15
pi is 3.141593 in 0.027028 seconds 15 threads
num_threads = 16
pi is 3.141593 in 0.025625 seconds 16 threads
```

```
hp@1718700:~/Documents/Aekarin_64015172/Mattson_OMP_exercises/solutions$ ./pi_final_1000
num_threads = 1
pi is 3.141593 in 2.091232 seconds 1 threads
num_threads = 2
pi is 3.141593 in 1.045871 seconds 2 threads
num_threads = 3
pi is 3.141593 in 0.698489 seconds 3 threads
num_threads = 4
pi is 3.141593 in 0.527321 seconds 4 threads
num_threads = 5
pi is 3.141593 in 0.427279 seconds 5 threads
num_threads = 6
pi is 3.141593 in 0.356073 seconds 6 threads
num_threads = 7
pi is 3.141593 in 0.305315 seconds 7 threads
num_threads = 8
pi is 3.141593 in 0.267432 seconds 8 threads
num_threads = 9
pi is 3.141593 in 0.385177 seconds 9 threads
num_threads = 10
pi is 3.141593 in 0.397711 seconds 10 threads
num_threads = 11
pi is 3.141593 in 0.365158 seconds 11 threads
num_threads = 12
pi is 3.141593 in 0.336273 seconds 12 threads
num_threads = 13
pi is 3.141593 in 0.311277 seconds 13 threads
num_threads = 14
pi is 3.141593 in 0.289198 seconds 14 threads
num_threads = 15
pi is 3.141593 in 0.278033 seconds 15 threads
num_threads = 16
pi is 3.141593 in 0.253078 seconds 16 threads
```

```
hp@1718700:~/Documents/Aekarin_64015172/Mattson_OMP_exercises/solutions$ ./pi_final_1000
num_threads = 1
pi is 3.141593 in 2.090840 seconds 1 threads
num_threads = 2
pi is 3.141593 in 1.046347 seconds 2 threads
num_threads = 3
pi is 3.141593 in 0.698939 seconds 3 threads
num_threads = 4
pi is 3.141593 in 0.527353 seconds 4 threads
num_threads = 5
pi is 3.141593 in 0.427273 seconds 5 threads
num_threads = 6
pi is 3.141593 in 0.356117 seconds 6 threads
num_threads = 7
pi is 3.141593 in 0.305369 seconds 7 threads
num_threads = 8
pi is 3.141593 in 0.312273 seconds 8 threads
num_threads = 9
pi is 3.141593 in 0.401052 seconds 9 threads
num_threads = 10
pi is 3.141593 in 0.354034 seconds 10 threads
num_threads = 11
pi is 3.141593 in 0.326263 seconds 11 threads
num_threads = 12
pi is 3.141593 in 0.336080 seconds 12 threads
num_threads = 13
pi is 3.141593 in 0.311366 seconds 13 threads
num_threads = 14
pi is 3.141593 in 0.289254 seconds 14 threads
num_threads = 15
pi is 3.141593 in 0.269937 seconds 15 threads
num_threads = 16
pi is 3.141593 in 0.254458 seconds 16 threads
```

```
hp@1718700:~/Documents/Aekarin_64015172/Mattson_OMP_exercises/solutions$ ./pi_final_1000
num_threads = 1
pi is 3.141593 in 2.090692 seconds 1 threads
num_threads = 2
pi is 3.141593 in 1.046278 seconds 2 threads
num_threads = 3
pi is 3.141593 in 0.698430 seconds 3 threads
num_threads = 4
pi is 3.141593 in 0.527126 seconds 4 threads
num_threads = 5
pi is 3.141593 in 0.427316 seconds 5 threads
num_threads = 6
pi is 3.141593 in 0.356098 seconds 6 threads
num_threads = 7
pi is 3.141593 in 0.305372 seconds 7 threads
num_threads = 8
pi is 3.141593 in 0.304323 seconds 8 threads
num_threads = 9
pi is 3.141593 in 0.369605 seconds 9 threads
num_threads = 10
pi is 3.141593 in 0.384222 seconds 10 threads
num_threads = 11
pi is 3.141593 in 0.365092 seconds 11 threads
num_threads = 12
pi is 3.141593 in 0.338766 seconds 12 threads
num_threads = 13
pi is 3.141593 in 0.311282 seconds 13 threads
num_threads = 14
pi is 3.141593 in 0.289200 seconds 14 threads
num_threads = 15
pi is 3.141593 in 0.269927 seconds 15 threads
num_threads = 16
pi is 3.141593 in 0.253357 seconds 16 threads
```

รูปภาพผลการทดลอง pi\_final 100 ล้าน และ 1000 ล้าน

โปรแกรมนี้ทำงานได้ถูกต้องและมีประสิทธิภาพมากกว่าโปรแกรมต้นฉบับ ( pi.c )

การทำงานของเธรด:

1. เธรดหลักจะสร้างทีมเธรดขึ้นมาโดยใช้คำสั่ง `#pragma omp parallel`.
2. เธรดแต่ละตัวในทีมจะคำนวณผลรวมของฟังก์ชัน  $4/(1+x^2)$  สำหรับจำนวนเต็ม  $i$  ที่แตกต่างกัน โดยใช้คำสั่ง `for`.
3. หลังจากทีเธรดแต่ละตัวคำนวณผลรวมของตนเสร็จแล้ว เธรดนั้นจะรวมผลรวมของตนเข้ากับผลรวมของเธรดอื่นๆ โดยใช้คำสั่ง `critical`.
4. เธรดหลักจะรวมผลรวมของเธรดทั้งหมดเข้าด้วยกันเพื่อให้ได้ผลรวมสุดท้าย

โปรแกรมนี้หลีกเลี่ยงปัญหา false sharing โดยการเก็บผลรวมบางส่วนไว้ในตัวแปร private ช่วยให้แน่ใจว่าเธรดแต่ละตัวจะไม่เขียนทับข้อมูลของเธรดอื่นๆ

# Pi\_God (ทดลองแก้ไข pi\_final ให้เร็วขึ้น)

```
hpc@1710700:~/Documents/Akarin_64015172/Matton_OMP_exercises/solutions$ ./pi_god
pi is 3.141593 in 0.189933 seconds 1 threads
pi is 3.141593 in 0.095227 seconds 2 threads
pi is 3.141593 in 0.064297 seconds 3 threads
pi is 3.141593 in 0.048627 seconds 4 threads
pi is 3.141593 in 0.039386 seconds 5 threads
pi is 3.141593 in 0.032835 seconds 6 threads
pi is 3.141593 in 0.028143 seconds 7 threads
pi is 3.141593 in 0.041193 seconds 8 threads
pi is 3.141593 in 0.036665 seconds 9 threads
pi is 3.141593 in 0.037419 seconds 10 threads
pi is 3.141593 in 0.034119 seconds 11 threads
pi is 3.141593 in 0.031249 seconds 12 threads
pi is 3.141593 in 0.028901 seconds 13 threads
pi is 3.141593 in 0.026835 seconds 14 threads
pi is 3.141593 in 0.025035 seconds 15 threads
pi is 3.141593 in 0.023479 seconds 16 threads
```

```
hpc@1710700:~/Documents/Akarin_64015172/Matton_OMP_exercises/solutions$ ./pi_god
pi is 3.141593 in 0.192469 seconds 1 threads
pi is 3.141593 in 0.096251 seconds 2 threads
pi is 3.141593 in 0.064379 seconds 3 threads
pi is 3.141593 in 0.048631 seconds 4 threads
pi is 3.141593 in 0.039393 seconds 5 threads
pi is 3.141593 in 0.032833 seconds 6 threads
pi is 3.141593 in 0.028173 seconds 7 threads
pi is 3.141593 in 0.041252 seconds 8 threads
pi is 3.141593 in 0.036735 seconds 9 threads
pi is 3.141593 in 0.037471 seconds 10 threads
pi is 3.141593 in 0.034033 seconds 11 threads
pi is 3.141593 in 0.031253 seconds 12 threads
pi is 3.141593 in 0.028999 seconds 13 threads
pi is 3.141593 in 0.026843 seconds 14 threads
pi is 3.141593 in 0.025035 seconds 15 threads
pi is 3.141593 in 0.023534 seconds 16 threads
```

```
hpc@1710700:~/Documents/Akarin_64015172/Matton_OMP_exercises/solutions$ ./pi_god
pi is 3.141593 in 0.192397 seconds 1 threads
pi is 3.141593 in 0.096256 seconds 2 threads
pi is 3.141593 in 0.064340 seconds 3 threads
pi is 3.141593 in 0.048712 seconds 4 threads
pi is 3.141593 in 0.039387 seconds 5 threads
pi is 3.141593 in 0.032825 seconds 6 threads
pi is 3.141593 in 0.028162 seconds 7 threads
pi is 3.141593 in 0.041251 seconds 8 threads
pi is 3.141593 in 0.036735 seconds 9 threads
pi is 3.141593 in 0.037412 seconds 10 threads
pi is 3.141593 in 0.034117 seconds 11 threads
pi is 3.141593 in 0.031296 seconds 12 threads
pi is 3.141593 in 0.028900 seconds 13 threads
pi is 3.141593 in 0.026826 seconds 14 threads
pi is 3.141593 in 0.025038 seconds 15 threads
pi is 3.141593 in 0.023454 seconds 16 threads
```

```
hpc@1710700:~/Documents/Akarin_64015172/Matton_OMP_exercises/solutions$ ./pi_god_1000
pi is 3.141593 in 1.924176 seconds 1 threads
pi is 3.141593 in 0.962112 seconds 2 threads
pi is 3.141593 in 0.641592 seconds 3 threads
pi is 3.141593 in 0.485162 seconds 4 threads
pi is 3.141593 in 0.393991 seconds 5 threads
pi is 3.141593 in 0.328313 seconds 6 threads
pi is 3.141593 in 0.281024 seconds 7 threads
pi is 3.141593 in 0.284192 seconds 8 threads
pi is 3.141593 in 0.344623 seconds 9 threads
pi is 3.141593 in 0.365559 seconds 10 threads
pi is 3.141593 in 0.337375 seconds 11 threads
pi is 3.141593 in 0.310541 seconds 12 threads
pi is 3.141593 in 0.280895 seconds 13 threads
pi is 3.141593 in 0.267639 seconds 14 threads
pi is 3.141593 in 0.250282 seconds 15 threads
pi is 3.141593 in 0.234429 seconds 16 threads
```

```
hpc@1710700:~/Documents/Akarin_64015172/Matton_OMP_exercises/solutions$ ./pi_god_1000
pi is 3.141593 in 1.924296 seconds 1 threads
pi is 3.141593 in 0.965963 seconds 2 threads
pi is 3.141593 in 0.642371 seconds 3 threads
pi is 3.141593 in 0.485217 seconds 4 threads
pi is 3.141593 in 0.393214 seconds 5 threads
pi is 3.141593 in 0.327684 seconds 6 threads
pi is 3.141593 in 0.280974 seconds 7 threads
pi is 3.141593 in 0.315563 seconds 8 threads
pi is 3.141593 in 0.339644 seconds 9 threads
pi is 3.141593 in 0.335413 seconds 10 threads
pi is 3.141593 in 0.337793 seconds 11 threads
pi is 3.141593 in 0.318562 seconds 12 threads
pi is 3.141593 in 0.287851 seconds 13 threads
pi is 3.141593 in 0.267671 seconds 14 threads
pi is 3.141593 in 0.252526 seconds 15 threads
pi is 3.141593 in 0.234232 seconds 16 threads
```

```
hpc@1710700:~/Documents/Akarin_64015172/Matton_OMP_exercises/solutions$ ./pi_god_1000
pi is 3.141593 in 1.923994 seconds 1 threads
pi is 3.141593 in 0.962323 seconds 2 threads
pi is 3.141593 in 0.642115 seconds 3 threads
pi is 3.141593 in 0.485310 seconds 4 threads
pi is 3.141593 in 0.393510 seconds 5 threads
pi is 3.141593 in 0.327699 seconds 6 threads
pi is 3.141593 in 0.280988 seconds 7 threads
pi is 3.141593 in 0.279924 seconds 8 threads
pi is 3.141593 in 0.344542 seconds 9 threads
pi is 3.141593 in 0.371493 seconds 10 threads
pi is 3.141593 in 0.337699 seconds 11 threads
pi is 3.141593 in 0.311031 seconds 12 threads
pi is 3.141593 in 0.287925 seconds 13 threads
pi is 3.141593 in 0.267670 seconds 14 threads
pi is 3.141593 in 0.249880 seconds 15 threads
pi is 3.141593 in 0.234246 seconds 16 threads
```

## (รูปภาพผลการทดลอง pi\_God 100 ล้าน และ 1000 ล้าน)

ตารางเปรียบเทียบ pi_final กับ pi_God		
เวลาเฉลี่ย	Pi_final	Pi_God
1 Thread 100 ล้าน	0.209269	0.19159967
8 Thread 100 ล้าน	0.04821033	0.041232
16 Thread 100 ล้าน	0.025462	0.023489
1 Thread 1000 ล้าน	2.09092133	1.92415533
8 Thread 1000 ล้าน	0.294676	0.29322633
16 Thread 1000 ล้าน	0.253631	0.23430233

ปัจจัยที่จะทำให้โปรแกรมความเร็วต่างกัน (สมมติฐาน):

1. การใช้ `#pragma omp for` ในลูป ทำให้ OpenMP รับผิดชอบในการแบ่งงานอัตโนมัติให้แต่ละ Thread โดยที่ไม่ต้องระบุ `id` และ `numthreads` แต่สิ่งที่ต้องทำคือใช้ `#pragma omp atomic` เพื่อรวมผลลัพธ์ระหว่าง Threads โดยปลอดภัยจาก `race condition` ในการเขียนผลลัพธ์ `full_sum`.
2. กำหนด `partial_sum` เป็น 0.0 เพื่อแก้ปัญหา `false sharing` โดยเริ่มต้นค่า `partial_sum` ในแต่ละ Thread ในแต่ละรอบของการคำนวณ.
3. การใช้ `#pragma omp for` แทนการแบ่งงานด้วยตัวแปร `i` ทำให้เราไม่ต้องจัดการกับการคำนวณที่จำนวน Thread หรือ `num_steps` ไม่ลงตัว

#สรุปได้ว่า โค้ดของ pi\_god เร็วกว่าเล็กน้อย เนื่องจากสามารถทำงานได้หลายขั้นตอนพร้อมกัน และไม่ต้องจัดการกับการคำนวณที่จำนวน Thread หรือ `num_steps` ไม่ลงตัว