Assignment 1

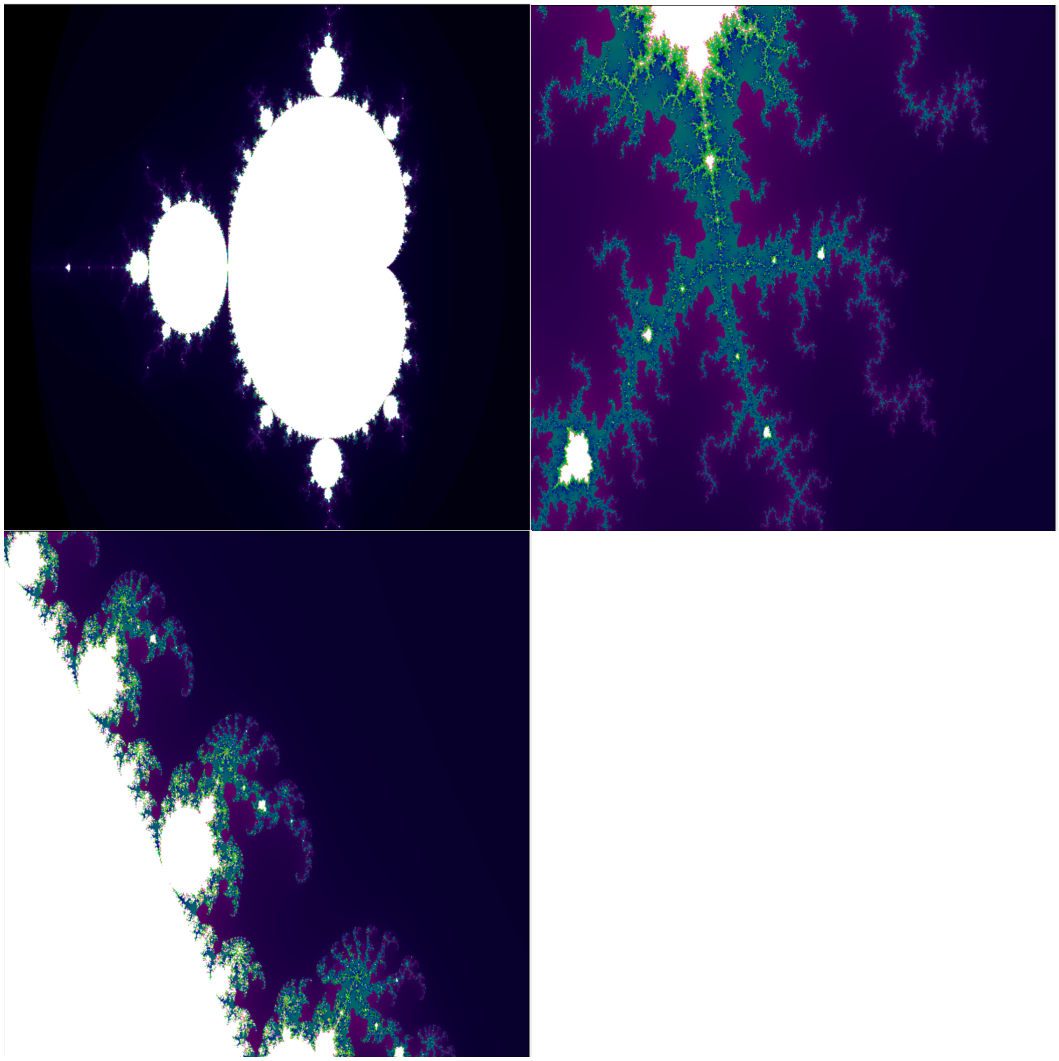
Exploring Multi-Core, Instruction-Level, and SIMD Parallelism

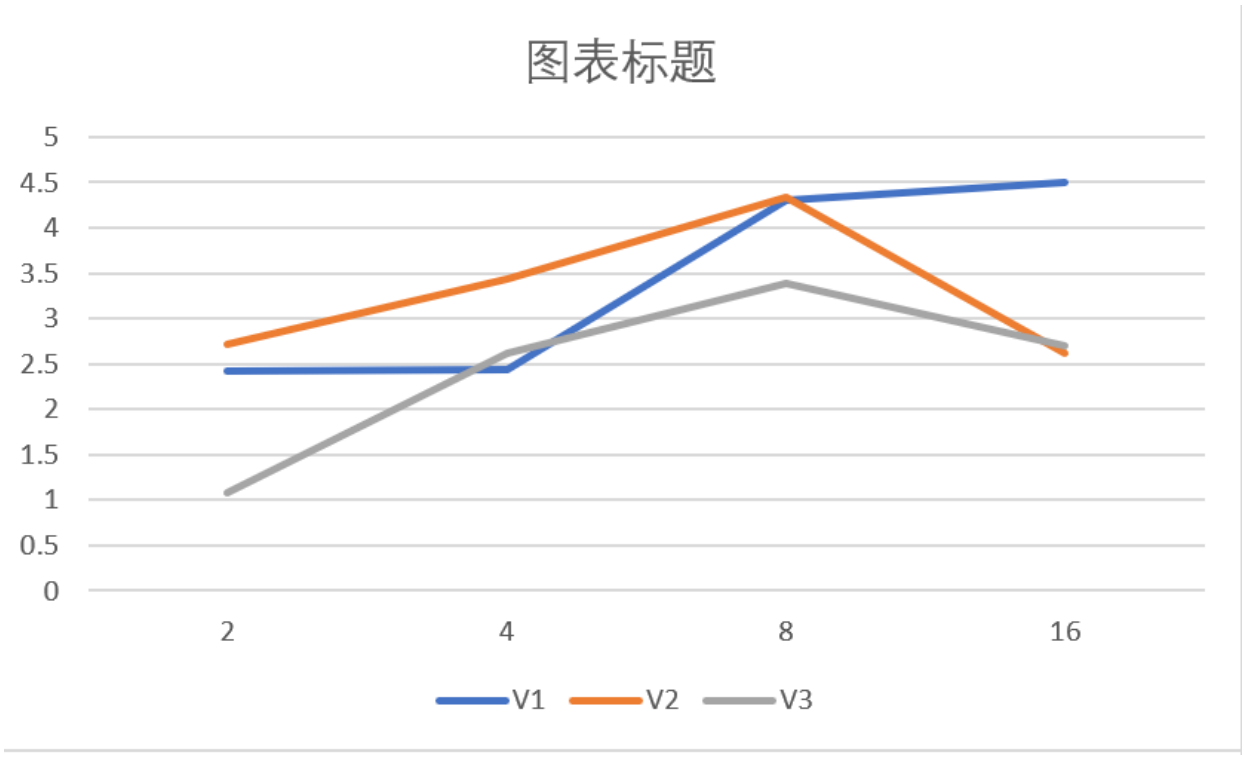
总结：1.如何对负载进行正确的划分，是对特定的数据集进行处理，还是使用泛化的算法?

2.在不同并行编程模型下，对于特定的问题应该采用哪种编程模型，他们对于不同问题来说的优缺点是什么?

一.prob1_mandelbrot_threads

```
consonnm@Consonnm:/mnt/c/Users/consonnm/Desktop/CMU作业/Assignment-1/prob1_mandelbrot_threads$ make
/bin/mkdir -p objs/
g++ -m64 mandelbrot.cpp -I../common -Iobjs/ -O3 -Wall -c -o objs/mandelbrot.o
g++ -m64 -I../common -Iobjs/ -O3 -Wall -o mandelbrot objs/main.o objs/mandelbrot.o objs/ppm.o -lm -lpthread
consonnm@Consonnm:/mnt/c/Users/consonnm/Desktop/CMU作业/Assignment-1/prob1_mandelbrot_threads$ ./mandelbrot -t 2
[mandelbrot serial]:            [217.797] ms
Wrote image file mandelbrot-v0-serial.ppm
[mandelbrot thread]:            [111.132] ms
Wrote image file mandelbrot-v0-thread-2.ppm
++++                            (1.96x speedup from 2 threads)
consonnm@Consonnm:/mnt/c/Users/consonnm/Desktop/CMU作业/Assignment-1/prob1_mandelbrot_threads$ ./mandelbrot -t 4
[mandelbrot serial]:            [206.712] ms
Wrote image file mandelbrot-v0-serial.ppm
[mandelbrot thread]:            [59.582] ms
Wrote image file mandelbrot-v0-thread-4.ppm
++++                            (3.47x speedup from 4 threads)
consonnm@Consonnm:/mnt/c/Users/consonnm/Desktop/CMU作业/Assignment-1/prob1_mandelbrot_threads$ ./mandelbrot -t 8
[mandelbrot serial]:            [211.171] ms
Wrote image file mandelbrot-v0-serial.ppm
[mandelbrot thread]:            [31.792] ms
Wrote image file mandelbrot-v0-thread-8.ppm
++++                            (6.64x speedup from 8 threads)
consonnm@Consonnm:/mnt/c/Users/consonnm/Desktop/CMU作业/Assignment-1/prob1_mandelbrot_threads$ ./mandelbrot -t 16
[mandelbrot serial]:            [218.295] ms
Wrote image file mandelbrot-v0-serial.ppm
[mandelbrot thread]:            [38.901] ms
Wrote image file mandelbrot-v0-thread-16.ppm
++++                            (5.61x speedup from 16 threads)
```



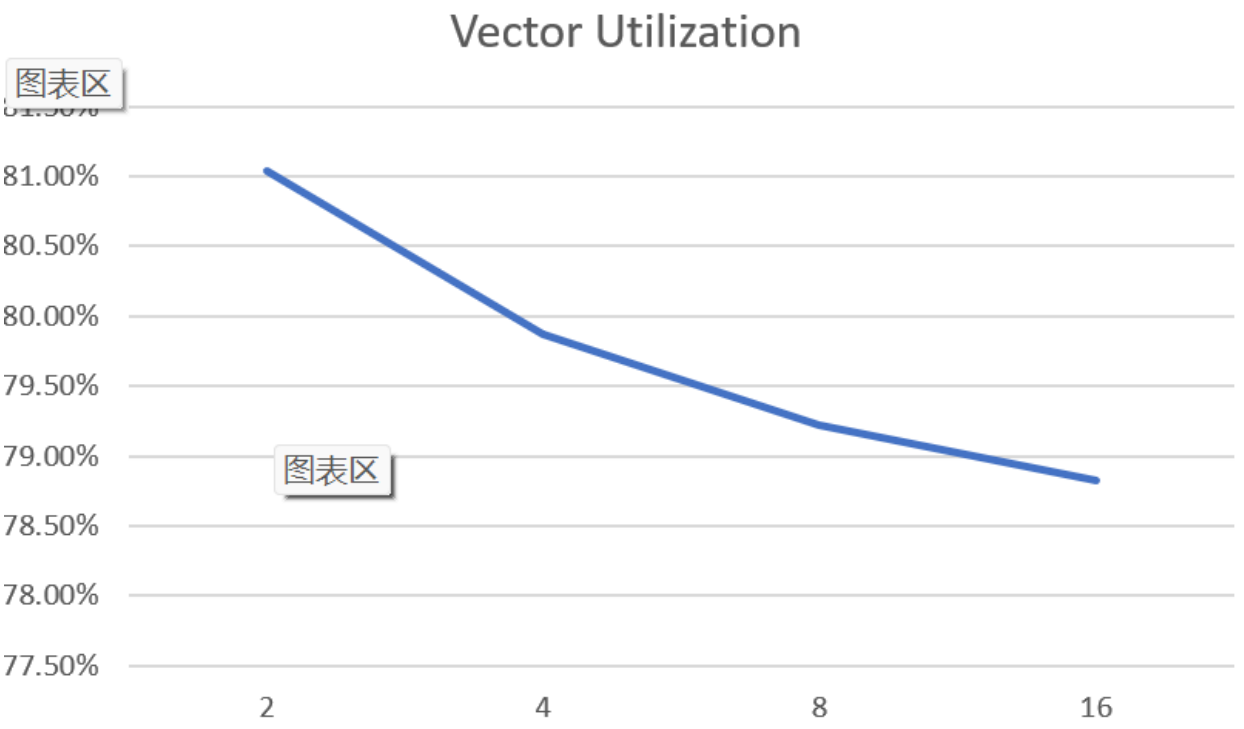


不太准波动大

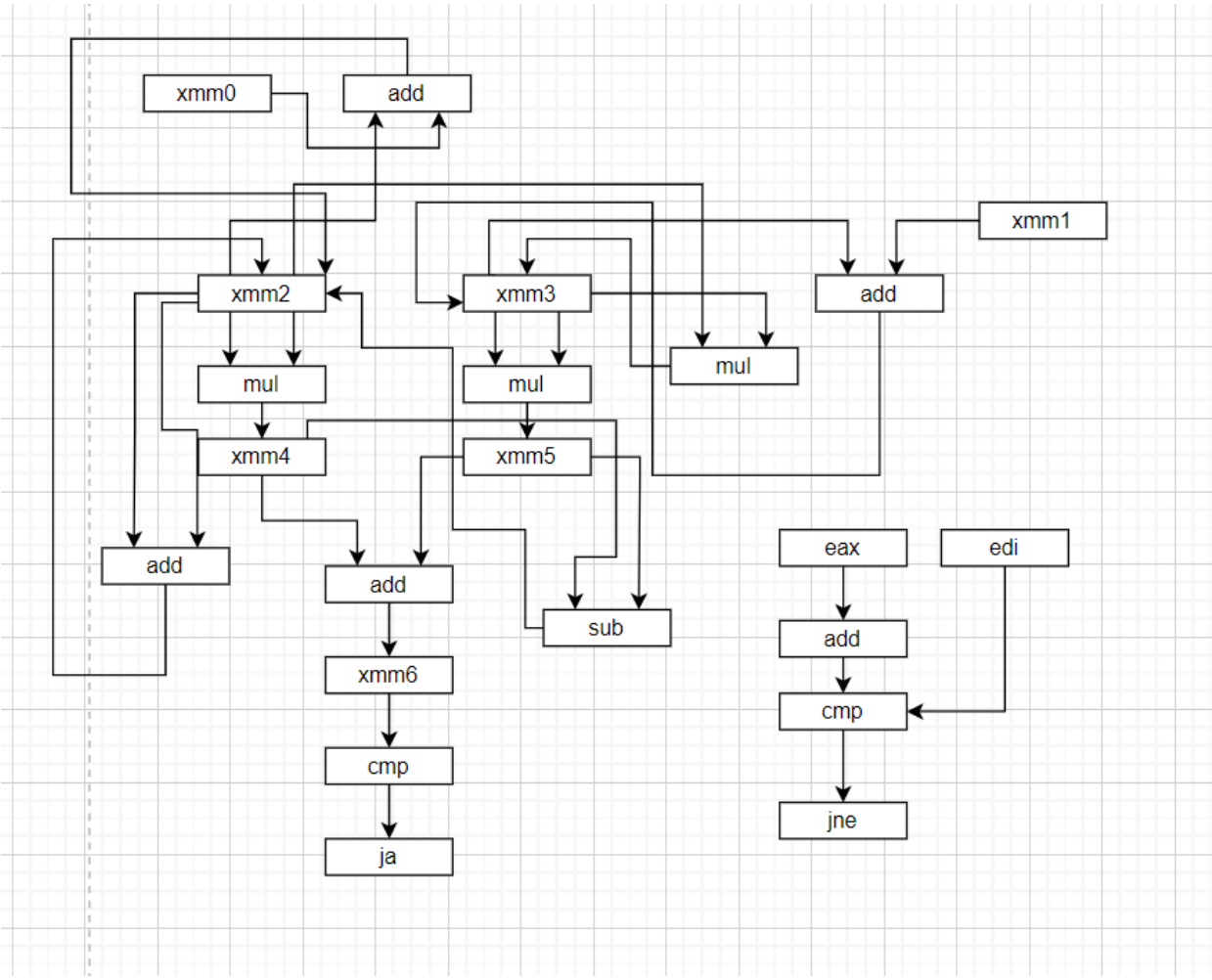
二.prob2_vecintrin

```
consonnm@Consonnm:/mnt/c/Users/consonnm/Desktop/CMU作业/Assignment-1/prob2_vecintrin$ ./vrun -s 1000
CLAMPED EXPONENT (required)
Results matched with answer!
****************** Printing Vector Unit Statistics ******************
Vector Width:              8
Total Vector Instructions: 17160
Vector Utilization:        79.219114%
Utilized Vector Lanes:     108752
Total Vector Lanes:        137280
************************ Result Verification *************************
Passed!!!

ARRAY SUM (bonus)
Passed!!!
```



三.prob3_mandelbrot_ilp1



四.prob4_mandelbrot_ispc

```
consonnm@Consonnm:/mnt/c/Users/consonnm/Desktop/CMU作业/Assignment-1/prob4_mandelbrot_ispc$ ./mandelbrot_ispc -v 0
[mandelbrot serial]:            [836.489] ms
Wrote image file mandelbrot-0-serial.ppm
[mandelbrot ispc]:              [244.842] ms
Wrote image file mandelbrot-0-ispc.ppm
[mandelbrot ispc parallel]:             [225.027] ms
Wrote image file mandelbrot-0-ispc-par.ppm
                                (3.42x speedup from ISPC)
                                (3.72x speedup from ISPC+parallelism)
consonnm@Consonnm:/mnt/c/Users/consonnm/Desktop/CMU作业/Assignment-1/prob4_mandelbrot_ispc$ ./mandelbrot_ispc -v 1
[mandelbrot serial]:            [162.500] ms
Wrote image file mandelbrot-1-serial.ppm
[mandelbrot ispc]:              [44.892] ms
Wrote image file mandelbrot-1-ispc.ppm
[mandelbrot ispc parallel]:             [99.132] ms
Wrote image file mandelbrot-1-ispc-par.ppm
                                (3.62x speedup from ISPC)
                                (1.64x speedup from ISPC+parallelism)
```

- 图一下发双向并行反而会更慢，因为图一是上下对称的，原本划分的方式导致了负载的不均衡，修改后

```
consonnm@Consonnm:/mnt/c/Users/consonnm/Desktop/CMU作业/Assignment-1/prob4_mandelbrot_ispc$ ./mandelbrot_ispc -v 1
[mandelbrot serial]:            [191.430] ms
Wrote image file mandelbrot-1-serial.ppm
[mandelbrot ispc]:              [39.973] ms
Wrote image file mandelbrot-1-ispc.ppm
[mandelbrot ispc parallel]:             [45.760] ms
Wrote image file mandelbrot-1-ispc-par.ppm
                                (4.79x speedup from ISPC)
                                (4.18x speedup from ISPC+parallelism)
consonnm@Consonnm:/mnt/c/Users/consonnm/Desktop/CMU作业/Assignment-1/prob4_mandelbrot_ispc$ ./mandelbrot_ispc -v 0
[mandelbrot serial]:            [771.939] ms
Wrote image file mandelbrot-0-serial.ppm
[mandelbrot ispc]:              [173.043] ms
Wrote image file mandelbrot-0-ispc.ppm
[mandelbrot ispc parallel]:             [174.942] ms
Wrote image file mandelbrot-0-ispc-par.ppm
                                (4.46x speedup from ISPC)
                                (4.41x speedup from ISPC+parallelism)
```

- 修改block size

```
consonnm@Consonnm:/mnt/c/Users/consonnm/Desktop/CMU作业/Assignment-1/prob4_mandelbrot_ispc$ ./mandelbrot_ispc -v 1 -t
[mandelbrot serial]:            [238.495] ms
Wrote image file mandelbrot-1-serial.ppm
[mandelbrot ispc]:              [40.947] ms
Wrote image file mandelbrot-1-ispc.ppm
[mandelbrot ispc parallel]:             [48.509] ms
Wrote image file mandelbrot-1-ispc-par.ppm
[mandelbrot multicore ispc]:    [13.180] ms
Wrote image file mandelbrot-1-task-ispc.ppm
                                (5.82x speedup from ISPC)
                                (4.92x speedup from ISPC+parallelism)
                                (18.10x speedup from task ISPC)
```

- Pthread 抽 象 (在问题 1 中 使用)和 ISPC 任 务 抽 象有什 么区别?

  ISPC任务通常都是数据并行，根据系统的物理资源和当前负载情况来自动分配线程，以保证最佳的性能和资源利用率，竞争和死锁问题更少，竞争和死锁问题更少

五.prob5_cuberoot

```
consonnm@Consonnm:/mnt/c/Users/consonnm/Desktop/CMU作业/Assignment-1/prob5_cuberoot$ ./cuberoot
[cuberoot serial]:              [2932.008] ms
[cuberoot ispc]:                [779.059] ms
[cuberoot task ispc]:   [107.618] ms
                                (3.76x speedup from ISPC)
                                (27.24x speedup from task ISPC)
```

1.单核加速比3.76，多核加速比27.24

2.修改数据观察加速比（观察给出的输入和迭代次数关系图）

- 将所有的数据设置成-1
- 原因：计算时迭代的次数过少，无法忽略，数据初始化，内存分配等花费的时间导致加速比下降

```
consonnm@Consonnm:/mnt/c/Users/consonnm/Desktop/CMU作业/Assignment-1/prob5_cuberoot$ ./cuberoot -d b
[cuberoot serial]:              [57.697] ms
[cuberoot ispc]:                [30.160] ms
[cuberoot task ispc]:   [16.371] ms
                                (1.91x speedup from ISPC)
                                (3.52x speedup from task ISPC)
```

- 数据1.9，-1，交替设置
- 原因：负载不均衡

```
consonnm@Consonnm:/mnt/c/Users/consonnm/Desktop/CMU作业/Assignment-1/prob5_cuberoot$ ./cuberoot -d b
[cuberoot serial]:              [956.572] ms
[cuberoot ispc]:                [394.208] ms
[cuberoot task ispc]:   [70.034] ms
                                (2.43x speedup from ISPC)
                                (13.66x speedup from task ISPC)
```

- 所有的数据设置成1.9
- 原因：负载均衡，迭代次数较大

```
consonnm@Consonnm:/mnt/c/Users/consonnm/Desktop/CMU作业/Assignment-1/prob5_cuberoot$ ./cuberoot -d g
[cuberoot serial]:              [1833.372] ms
[cuberoot ispc]:                [396.767] ms
[cuberoot task ispc]:   [70.385] ms
                                (4.62x speedup from ISPC)
                                (26.05x speedup from task ISPC)
```