Deqing Qu
qud@oregonstate.edu

1. Tell what machine you ran this on

   I ran the code on Flip1.

2. What do you think the actual volume is?

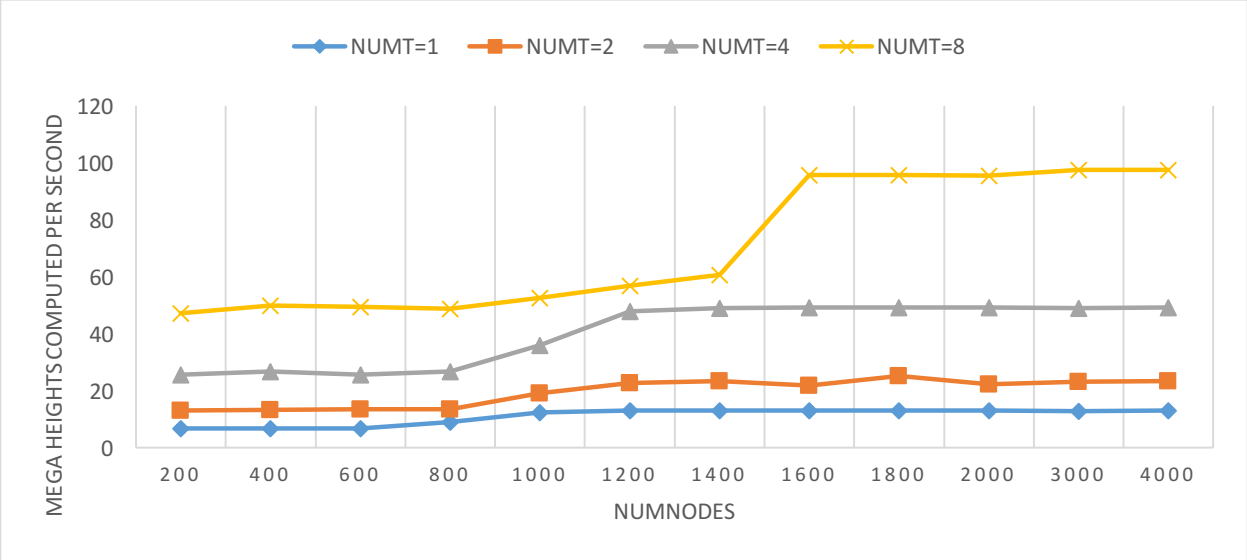
   The actual volume approximately equals to 25.3125

3. Show the performances you achieved in tables and graphs as a function of NUMNODES and NUMT?

   In the table, the first row shows the NUMNODES, the first column shows the NUMT and other cells show the speed using the specified NUMNODES and NUMT. The speed is recorded in the units of 'Mega Height Computed Per Second'.

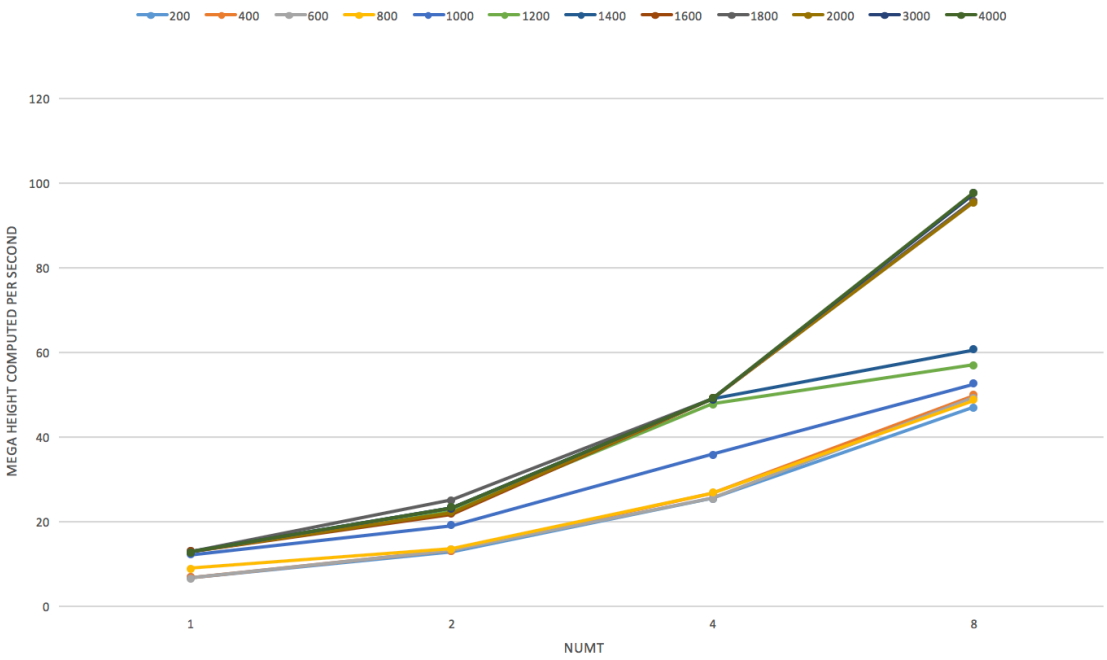
	200	400	600	800	1000	1200	1400	1600	1800	2000	3000	4000
1	6.76	6.76	6.75	9.01	12.19	12.88	12.95	12.93	12.87	12.87	12.81	12.87
2	12.94	13.28	13.36	13.51	19.07	22.62	23.3	21.81	25.05	22.28	23.19	23.35
4	25.53	26.65	25.57	26.74	35.83	47.75	49.04	49.13	49.14	49.06	48.98	49.12
8	47.07	49.88	49.38	48.77	52.6	56.89	60.58	95.74	95.74	95.48	97.49	97.61

In the graph1, the vertical axis indicates the NUMNODES, the horizontal axis indicates the speed using 'Mega Heights Computed Per Second' as a unit and the line indicates the number of threads.



Deqing Qu
qud@oregonstate.edu

In the graph2, the vertical axis indicates the NUMT, the horizontal axis indicates the speed using ‘Mega Heights Computed Per Second’ as a unit and the line indicates the NUMNODES.



4. What patterns are you seeing in the speeds?

(1) As the number of threads increases, the speed will increase. But the speedup of n threads will be less than n.
(2) As the number of nodes increases, the speed will increase. But the speed will not increase without limit and will reach the upper bound.

5. Why do you think it is behaving this way?

(1) In the code, the height calculation process of each node has nothing to do with each other, so the speed will increase using parallel programming. But there is some fraction of the total operation that is inherently sequential and can’t be parallelized. So the speedup of n threads will always be less than n.
(2) As the number of nodes increases, the proportion of overhead in parallel programming is getting smaller and the speed will be faster. But there is still an upper bound of the computing capacity of CPU and the speed will not increase without limit.

6. What is the Parallel Fraction for this application, using the Inverse Amdahl equation?

Deqing Qu
qud@oregonstate.edu

The Parallel Fraction can be calculated by the following equation.

$$\mathrm{F} = \frac{n}{n-1}\left(1 - \frac{1}{Speedup}\right) = \frac{n}{n-1}\left(1 - \frac{MegaHeight_1}{MegaHeight_n}\right)$$

	200	400	600	800	1000	1200
NUMT=2	0.955177743	0.981927711	0.989520958	0.666173205	0.721552176	0.861184792
NUMT=4	0.980284632	0.995121951	0.981358363	0.884068811	0.87970974	0.973682373
NUMT=8	0.978724696	0.987971131	0.986634265	0.93172032	0.878001086	0.884112196
	1400	1600	1800	2000	3000	4000
NUMT=2	0.888412017	0.814305365	0.97245509	0.84470377	0.895213454	0.89764454
NUMT=4	0.981239804	0.982427573	0.984126984	0.983557549	0.984619573	0.983984799
NUMT=8	0.898552092	0.988510549	0.989226775	0.988808427	0.992687895	0.992170006

The average of Parallel Fraction is 0.9327.

7. Given that Parallel Fraction, what is the maximum speed-up you could ever get?
   The maximum speed-up can be calculated by the following equation.

$$\max Speedup = \frac{1}{1 - F_{parallel}}$$

   The maximum speed-up for the application is 14.86