CS475: Project 1

Trevor Bramwell

April 21, 2015

Numerical Integration

This assignment involved computing the volume under a Bezier surface using the OpenMP library. The program was compiled using *gcc* and ran 10 times on *rabbit.engr.oregonstate.edu*, a 32 CPU machine, using the following combination of parameters:

Subdivisions:

- 1000
- 2000
- 4000
- 8000
- 16000
- 32000
- 46340

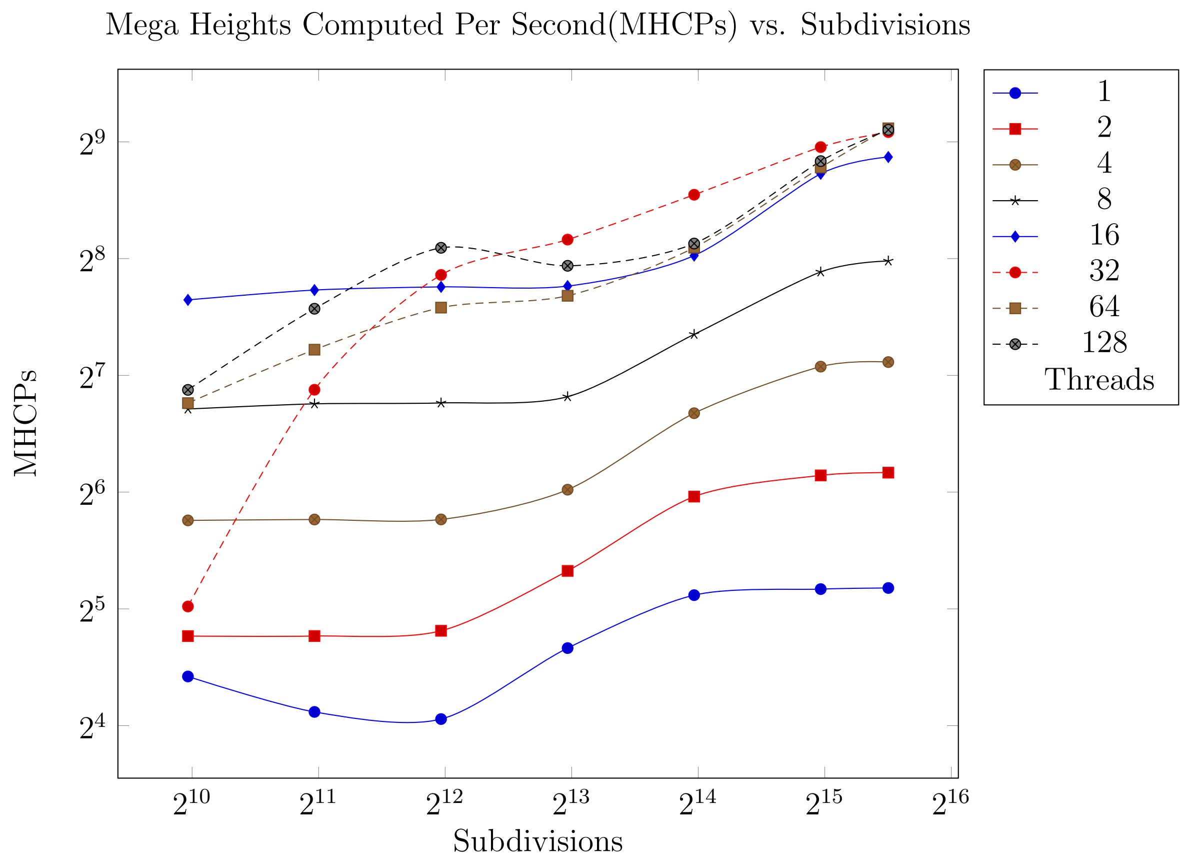
Threads:

- 1
- 2
- 4
- 8
- 16
- 32
- 64
- 128

Each run output the number of subdivision of the problem, the number of threads used, Mega Heights Computer Per Second (MHCPs), and total calculated volume. After all 10 runs the average MHCPs was computed

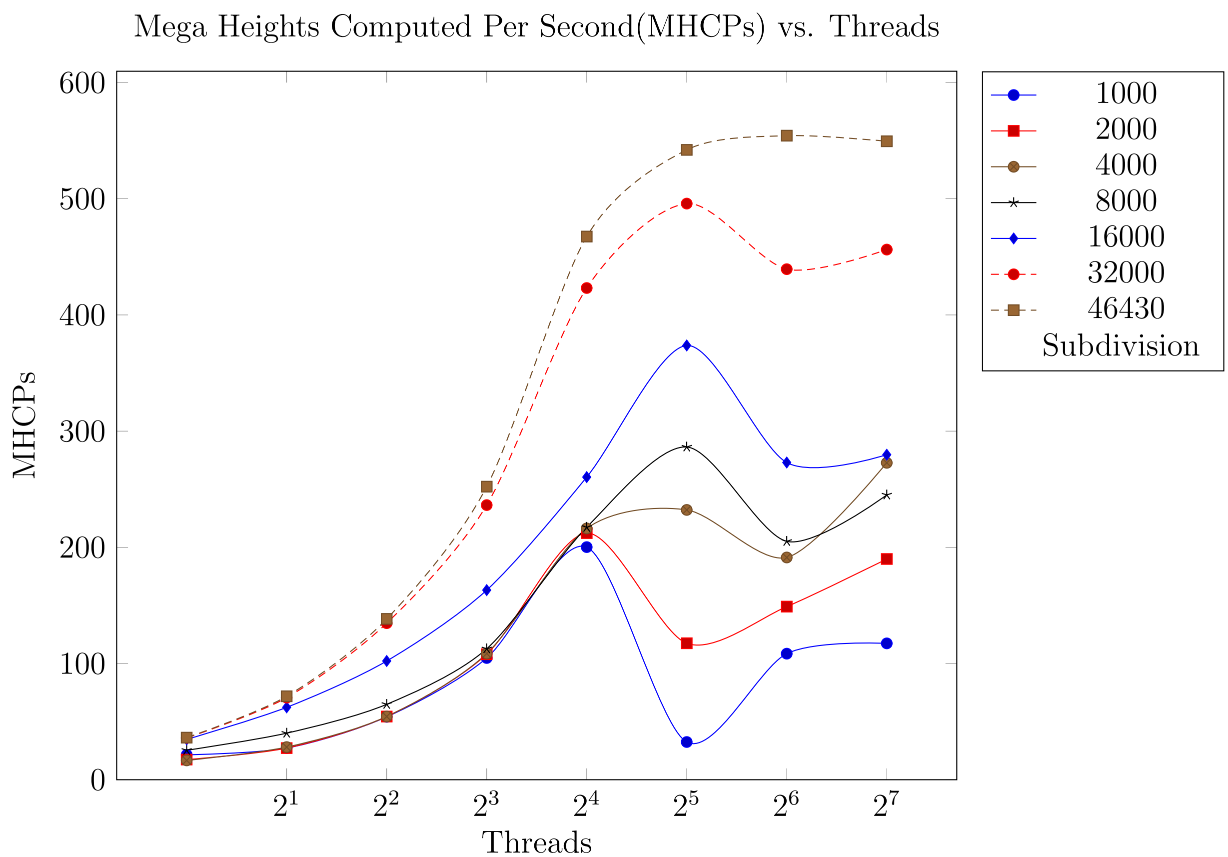
for each group of subdivisions and threads. These results can be seen in the Tables section at the end of this report. A LibreOffice Calc document containing all the data is included in the *data* directory of the project.
The results after 10 runs agree on the volume under the Bezier surface equalling: **14.0625**.

Graphs



It can be seen from this graph that adding more threads consistently increased the MHCPs. Starting at 32 threads, MHCPs dropped before increasing at a faster rate than lower numbers of threads. This is not very surprising because rabbit only has 32 CPUs, and is most likely due to the overhead incurred through process scheduling.

Another interesting point is that MHCPs didn't increase until the subdivisions reached 8192. Given the number of CPUs on rabbit is 32, we can see: $8192/32 = 512 = (2^{13})$. I'm unsure as to why 512 is a special number in this instance.



In this graph of MHCPs vs Threads, it is clear to see the scheduling overhead reduce MHCPs when threads surpass the number of CPUs available.

Speedup

Since these computations were run on a 32 CPU machine, the speedup for each can be determined by dividing the MHCPs by 32. Using this speedup we can determine the fraction of the program that is parallelizable. The largest

MHCPs (554.28) was computed using 64 threads and 46,340 subdivisions.

Using Amdahl's Inverse Law, we have:

$$
\begin{aligned}
F_{parallel} &= \frac{n}{n-1}\left(1 - \frac{1}{Speedup}\right) \\
&= \frac{32}{31}\left(1 - \frac{1}{(554.28/32)}\right) \\
&= 0.97266
\end{aligned}
$$

Given that 0.97266 (97.26%) of the program is parallelizable, only 0.02734 (2.7%) is sequential. Using Amdahl's Law, we can see the max speedup this program could ever achive is:

$$
\begin{aligned}
maxSpeedup &= \frac{1}{1 - F_{parallel}} \\
&= \frac{1}{1 - 0.97266} \\
&= 36.57644
\end{aligned}
$$

Tables

NUMS	NUMT	MHCPS		NUMS	NUMT	MHCPS		NUMS	NUMT	MHCPS
1,000	1	21.41		1,000	2	27.22		1,000	4	54.08
2,000	1	17.35		2,000	2	27.22		2,000	4	54.37
4,000	1	16.63	—	4,000	2	28.08	—	4,000	4	54.39
8,000	1	25.34		8,000	2	40.07		8,000	4	64.9
16,000	1	34.7		16,000	2	62.29		16,000	4	102.18
32,000	1	35.97		32,000	2	70.65		32,000	4	134.72
46,340	1	36.2		46,340	2	71.81		46,340	4	138.4
NUMS	NUMT	MHCPS		NUMS	NUMT	MHCPS		NUMS	NUMT	MHCPS
1,000	8	104.75		1,000	16	200.18		1,000	32	32.45
2,000	8	108.02		2,000	16	212.25		2,000	32	117.45
4,000	8	108.6	—	4,000	16	216.27	—	4,000	32	232.19
8,000	8	112.71		8,000	16	217.27		8,000	32	286.35
16,000	8	163.18		16,000	16	260.44		16,000	32	373.72
32,000	8	236.34		32,000	16	423.17		32,000	32	495.82
46,340	8	252.21		46,340	16	467.46		46,340	32	542.04

NUMS	NUMT	MHCPS		NUMS	NUMT	MHCPS
1,000	64	108.48		1,000	128	117.34
2,000	64	148.98		2,000	128	189.96
4,000	64	191.27		4,000	128	272.72
8,000	64	205.16	—	8,000	128	245.09
16,000	64	272.99		16,000	128	279.71
32,000	64	439.38		32,000	128	456.24
46,340	64	554.28		46,340	128	549.47