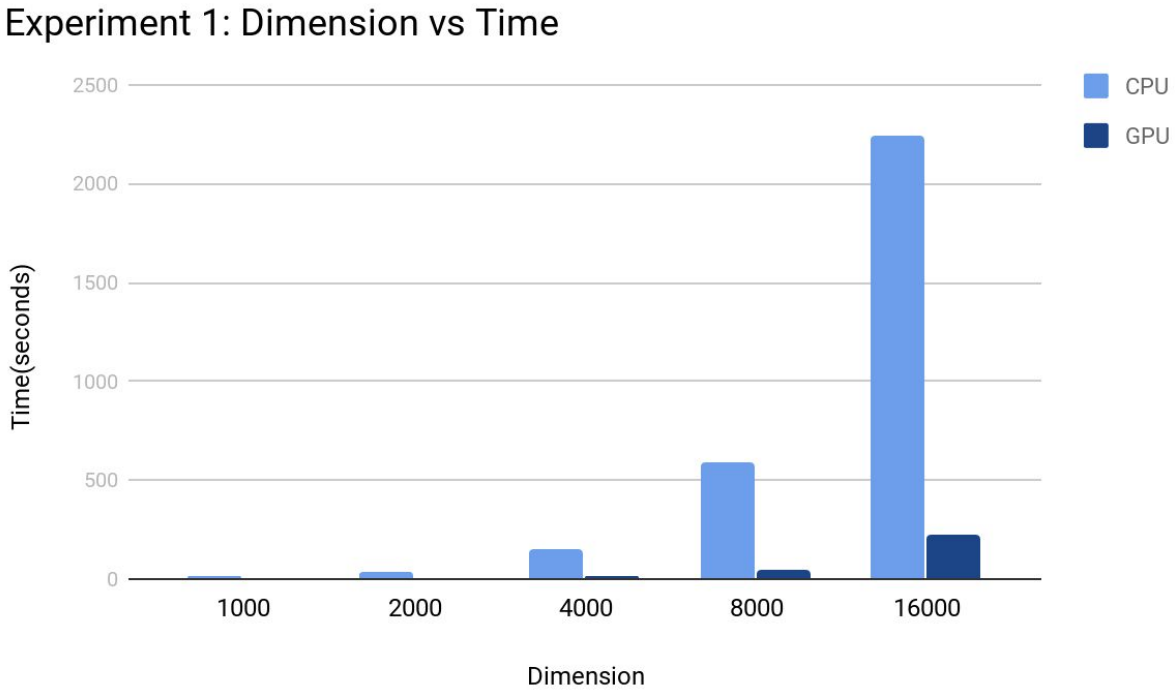
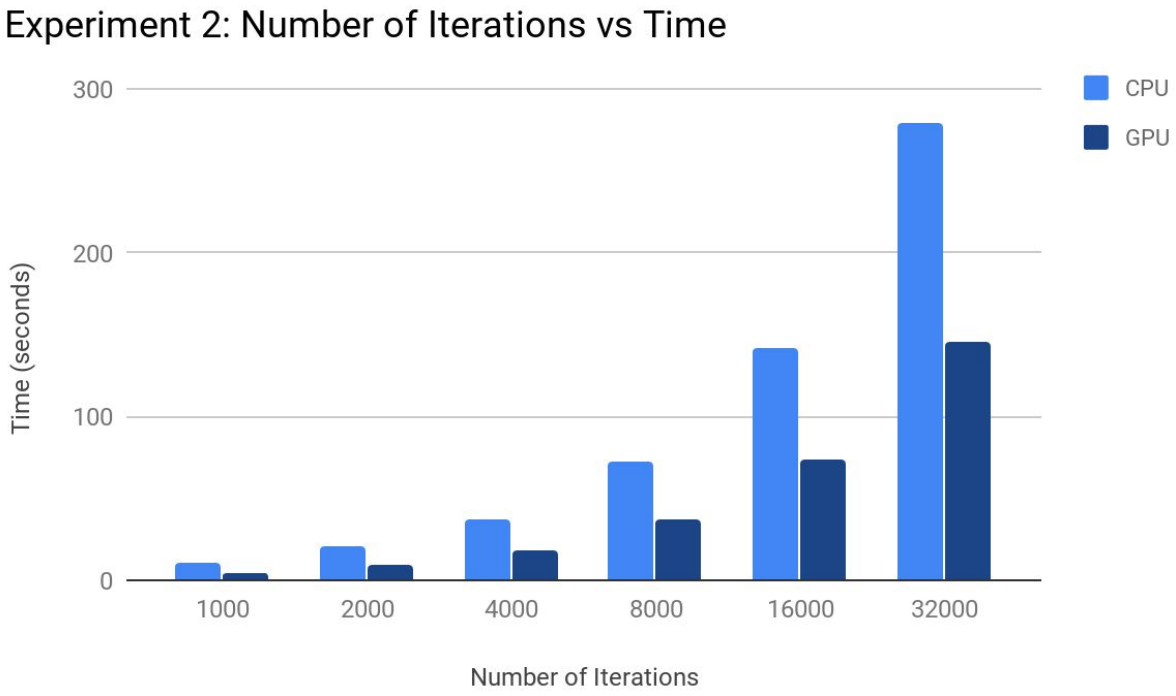


	CPU	GPU
1000	10.63	4.96
2000	40.07	5.64
4000	154.34	12.52
8000	588.56	41.7
16000	2239.27	223.78



	CPU	GPU
1000	10.63	4.96
2000	20.78	9.47
4000	37.32	18.55
8000	72.19	37.15
16000	141.91	73.32
32000	279.34	145.73

Questions

a) When is GPU usage more beneficial (at which n)? and why?
For larger problems at higher n values, such as dimension n=16000, although GPU performs the fastest compared to CPU at dimension n = 8000. The GPU usage is more beneficial at n=8000 and larger dimension n's in general because the parallelization of computing operations on large data sets hides higher latency from memory access. For example, at dimensions n= 1000, accessing and copying data with cudamemcopy takes 57.82% of the GPU's time, which is much less than 95.19% for a dimension n of 8000, but the small amount of data for n=1000 can't make up for the high memory access latency compared to the larger n values.

b) When is the speedup (i.e. time of CPU version / time of CUDA version) at its lowest? And why?
Iterations n=32000 where the GPU is only 1.92 times faster than CPU. This is because the GPU is spending 94.02% of its time launching the kernel, which it does 32,000 times.

c) When is the speedup at its highest? And why?
Speedup is at its highest in Experiment 1 with a dimension of n=8000, where the GPU speedup was 14 times faster than the CPU. The next highest speedups are in Experiment 1 as well for n=4000 (GPU is approximately 12 times faster) and n=16000 (GPU is approximately 10 times faster). This is because the high level of parallelization on larger data sets masks the GPU's slower access to memory. The speedup is most likely the highest at n=8000 because it accesses memory 95.19% of its time while n=4000 accesses memory 98.98% of its time and n=16000 accessed its memory for 99.25% of its time and accessing memory creates a lot of latency.

d) Which has more effect: number of iterations or the problem size? and why?
The problem size has more of an effect on a GPU. This is because iterations call the kernel multiple times for GPU (spending 93.18% of its time for cudaLaunch at iteration = 1600) while with problem size, threads execute in parallel to compute results. The problem size has a big effect on the GPU because if the problem size is big, the level of parallelism makes up for the high latency of memory access (99.25% of the GPU's time

at dimension = 1600 was spent on cudaMemcpy but with an execution time of 74 seconds as opposed to iteration=1600 231 seconds).

Example GPU at iterations = 32000

```
Time taken for GPU is 145.860000
==25120== Profiling application: ./heatdist 1000 32000 1
==25120== Profiling result:
            Type  Time(%)      Time     Calls       Avg       Min       Max  Name
 GPU activities:   99.90%  149.356s     32000  4.6674ms  4.6029ms  6.6827ms  CalculateHeat(float*, float*, unsigned
int)
                    0.10%  146.20ms         1  146.20ms  146.20ms  146.20ms  [CUDA memcpy DtoH]
                    0.00%  3.0407ms         2  1.5203ms  1.4362ms  1.6044ms  [CUDA memcpy HtoD]
      API calls:   94.02%  144.563s     32000  4.5176ms  9.2720us  89.307ms  cudaLaunch
                    3.23%  4.97122s         3  1.65707s  1.9036ms  4.96722s  cudaMemcpy
                    2.61%  4.00608s         2  2.00304s  486.97us  4.00560s  cudaFree
                    0.11%  172.75ms         2  86.374ms  231.82us  172.52ms  cudaMalloc
                    0.01%  20.646ms     96000     215ns     115ns  615.11us  cudaSetupArgument
                    0.01%  12.947ms     32000     404ns     249ns  599.73us  cudaConfigureCall
                    0.00%  3.1162ms       376  8.2870us     151ns  349.48us  cuDeviceGetAttribute
                    0.00%  503.33us         4  125.83us  121.12us  130.63us  cuDeviceTotalMem
                    0.00%  277.76us         4  69.439us  64.550us  81.278us  cuDeviceGetName
                    0.00%  4.8760us         8     609ns     198ns  2.6640us  cuDeviceGet
                    0.00%  2.4620us         3     820ns     210ns  1.7600us  cuDeviceGetCount
```

Example GPU at iterations=16000

```
Time taken for GPU is 74.400000
==23911== Profiling application: ./heatdist 1000 16000 1
==23911== Profiling result:
            Type  Time(%)      Time     Calls       Avg       Min       Max  Name
 GPU activities:  100.00%  74.2294s     16000  4.6393ms  4.6036ms  6.6835ms  CalculateHeat(float*, float*, unsigned
int)
                    0.00%  1.7609ms         2  880.45us  829.94us  930.96us  [CUDA memcpy HtoD]
                    0.00%  1.1928ms         1  1.1928ms  1.1928ms  1.1928ms  [CUDA memcpy DtoH]
      API calls:   93.18%  69.3630s     16000  4.3352ms  8.5510us  76.437ms  cudaLaunch
                    6.44%  4.79240s         3  1.59747s  1.1246ms  4.79012s  cudaMemcpy
                    0.23%  173.55ms         2  86.774ms  214.96us  173.33ms  cudaMalloc
                    0.14%  102.67ms     48000  2.1390us     144ns  74.909ms  cudaSetupArgument
                    0.01%  6.7498ms     16000     421ns     252ns  549.96us  cudaConfigureCall
                    0.00%  3.2024ms       376  8.5160us     151ns  341.85us  cuDeviceGetAttribute
                    0.00%  1.1822ms         2  591.09us  402.67us  779.51us  cudaFree
                    0.00%  563.76us         4  140.94us  127.77us  152.66us  cuDeviceTotalMem
                    0.00%  299.28us         4  74.820us  66.860us  81.259us  cuDeviceGetName
                    0.00%  4.8130us         8     601ns     197ns  2.4920us  cuDeviceGet
                    0.00%  2.1050us         3     701ns     160ns  1.4330us  cuDeviceGetCount
```

Example GPU at dimension=16000

```
Time taken for GPU is 231.950000
```

==23975== Profiling application: ./heatdist 16000 1000 1
==23975== Profiling result:

Type	Time(%)	Time	Calls	Avg	Min	Max	Name
GPU activities:	99.06%	454.179s	1000	454.18ms	438.28ms	486.66ms	CalculateHeat(float*, float*, unsigned int)
	0.82%	3.76721s	1	3.76721s	3.76721s	3.76721s	[CUDA memcpy DtoH]
	0.11%	522.64ms	2	261.32ms	189.59ms	333.05ms	[CUDA memcpy HtoD]
API calls:	**99.25%**	**458.463s**	**3**	**152.821s**	**189.79ms**	**457.940s**	**cudaMemcpy**
	0.71%	3.26518s	2	1.63259s	99.808ms	3.16537s	cudaFree
	0.04%	173.46ms	2	86.730ms	1.4201ms	172.04ms	cudaMalloc
	0.00%	9.7530ms	1000	9.7530us	8.7650us	79.777us	cudaLaunch
	0.00%	3.1408ms	376	8.3530us	145ns	354.67us	cuDeviceGetAttribute
	0.00%	638.89us	3000	212ns	177ns	12.268us	cudaSetupArgument
	0.00%	499.49us	4	124.87us	117.89us	136.06us	cuDeviceTotalMem
	0.00%	266.82us	1000	266ns	248ns	2.8510us	cudaConfigureCall
	0.00%	263.63us	4	65.907us	63.302us	73.199us	cuDeviceGetName
	0.00%	16.954us	8	2.1190us	201ns	14.824us	cuDeviceGet
	0.00%	2.5810us	3	860ns	186ns	1.8340us	cuDeviceGetCount

Example GPU at dimensions = 4000

Time taken for GPU is 12.520000
==24682== Profiling application: ./heatdist 4000 1000 1
==24682== Profiling result:

Type	Time(%)	Time	Calls	Avg	Min	Max	Name
GPU activities:	99.51%	18.7804s	1000	18.780ms	18.541ms	26.774ms	CalculateHeat(float*, float*, unsigned int)
	0.31%	58.495ms	1	58.495ms	58.495ms	58.495ms	[CUDA memcpy DtoH]
	0.18%	33.905ms	2	16.952ms	13.294ms	20.610ms	[CUDA memcpy HtoD]
API calls:	**98.98%**	**18.8655s**	**3**	**6.28851s**	**13.532ms**	**18.8311s**	**cudaMemcpy**
	0.91%	174.19ms	2	87.097ms	277.70us	173.92ms	cudaMalloc
	0.05%	10.244ms	1000	10.244us	9.3420us	68.890us	cudaLaunch
	0.03%	5.1294ms	2	2.5647ms	603.10us	4.5263ms	cudaFree
	0.02%	3.3271ms	376	8.8480us	150ns	383.72us	cuDeviceGetAttribute
	0.00%	537.92us	3000	179ns	131ns	9.2990us	cudaSetupArgument
	0.00%	505.04us	4	126.26us	119.01us	137.10us	cuDeviceTotalMem
	0.00%	268.56us	4	67.139us	62.638us	77.630us	cuDeviceGetName
	0.00%	265.87us	1000	265ns	252ns	2.9080us	cudaConfigureCall
	0.00%	18.910us	8	2.3630us	227ns	15.421us	cuDeviceGet
	0.00%	2.1070us	3	702ns	203ns	1.2840us	cuDeviceGetCount

Example GPU at dimensions = 8000

Time taken for GPU is 41.700000
==24804== Profiling application: ./heatdist 8000 1000 1
==24804== Profiling result:

Type	Time(%)	Time	Calls	Avg	Min	Max	Name
GPU activities:	99.19%	79.1471s	1000	79.147ms	77.864ms	102.96ms	CalculateHeat(float*, float*, unsigned int)
	0.54%	428.38ms	1	428.38ms	428.38ms	428.38ms	[CUDA memcpy DtoH]
	0.27%	219.17ms	2	109.59ms	91.276ms	127.89ms	[CUDA memcpy HtoD]
API calls:	**95.19%**	**79.7881s**	**3**	**26.5960s**	**91.678ms**	**79.5680s**	**cudaMemcpy**

```
                4.59%  3.84550s         2  1.92275s  25.048ms  3.82045s  cudaFree
                0.20%  171.15ms         2  85.573ms  556.27us  170.59ms  cudaMalloc
                0.01%  9.9921ms      1000  9.9920us  8.9480us  70.293us  cudaLaunch
                0.00%  3.2485ms       376  8.6390us     150ns  357.03us  cuDeviceGetAttribute
                0.00%  519.54us         4  129.88us  118.71us  150.65us  cuDeviceTotalMem
                0.00%  496.70us      3000     165ns     128ns  10.341us  cudaSetupArgument
                0.00%  268.85us         4  67.211us  62.358us  77.815us  cuDeviceGetName
                0.00%  247.03us      1000     247ns     228ns  2.6180us  cudaConfigureCall
                0.00%  17.348us         8  2.1680us     234ns  14.819us  cuDeviceGet
                0.00%  2.3260us         3     775ns     179ns  1.5610us  cuDeviceGetCount
```

Example GPU at dimensions = 1000, iterations = 1000

```
Time taken for GPU is 1.720000
==25258== Profiling application: ./heatdist 1000 1000 1
==25258== Profiling result:
            Type  Time(%)      Time     Calls       Avg       Min       Max  Name
 GPU activities:   99.94%  5.07543s      1000  5.0754ms  4.6043ms  6.6821ms  CalculateHeat(float*, float*, unsigned
int)
                  0.04%  1.8547ms         2  927.33us  888.85us  965.81us  [CUDA memcpy HtoD]
                  0.03%  1.3608ms         1  1.3608ms  1.3608ms  1.3608ms  [CUDA memcpy DtoH]
      API calls:   57.82%  5.07097s         3  1.69032s  1.1549ms  5.06860s  cudaMemcpy
                  40.01%  3.50894s         2  1.75447s  477.91us  3.50847s  cudaFree
                  1.99%  174.27ms         2  87.136ms  251.75us  174.02ms  cudaMalloc
                  0.12%  10.593ms      1000  10.592us  9.5710us  50.766us  cudaLaunch
                  0.04%  3.3073ms       376  8.7960us     150ns  398.69us  cuDeviceGetAttribute
                  0.01%  533.71us         4  133.43us  122.12us  156.06us  cuDeviceTotalMem
                  0.01%  501.91us      3000     167ns     122ns  12.388us  cudaSetupArgument
                  0.00%  299.26us      1000     299ns     255ns  12.542us  cudaConfigureCall
                  0.00%  263.66us         4  65.914us  61.866us  74.927us  cuDeviceGetName
                  0.00%  5.1890us         8     648ns     208ns  2.6720us  cuDeviceGet
                  0.00%  2.1030us         3     701ns     174ns  1.3740us  cuDeviceGetCount
```