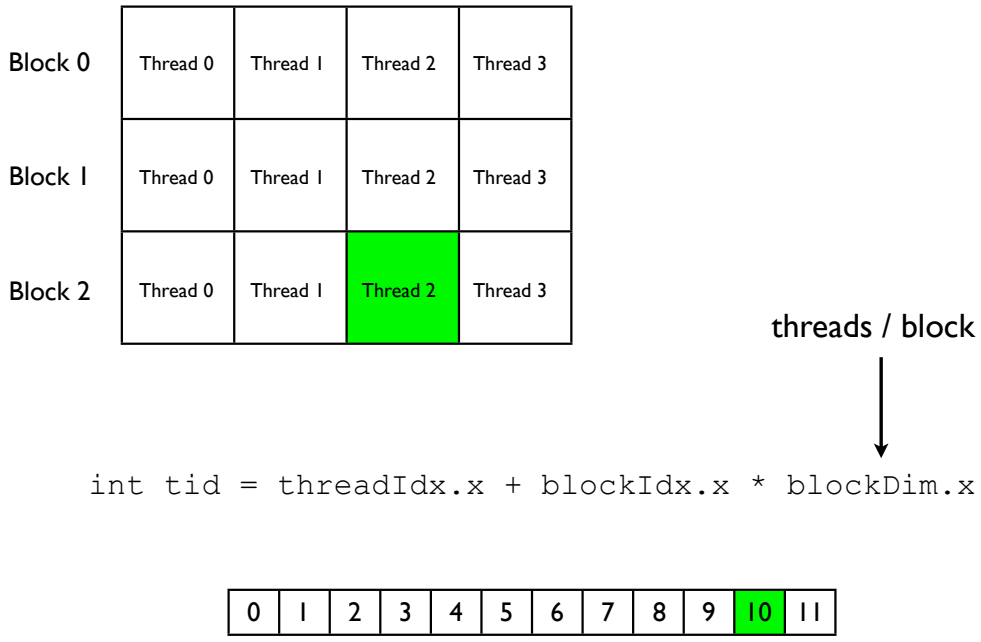
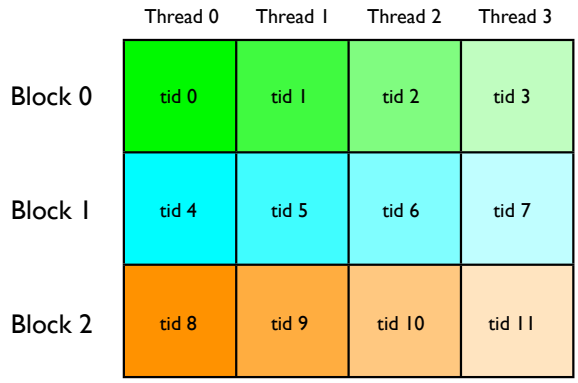
Reductions

OSU Stat GPU Group
February 24, 2012

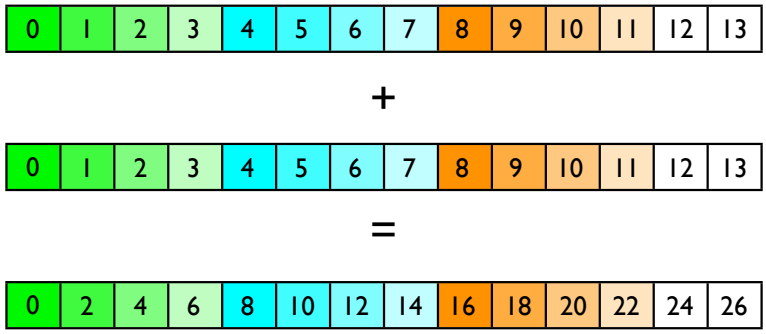
with examples from

CUDA by Example
Sanders & Kandrot

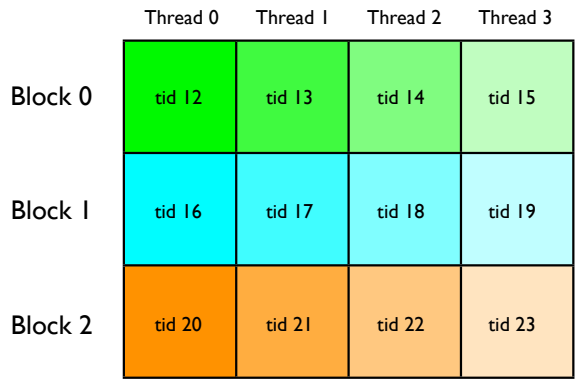




Initial id's for each thread



3



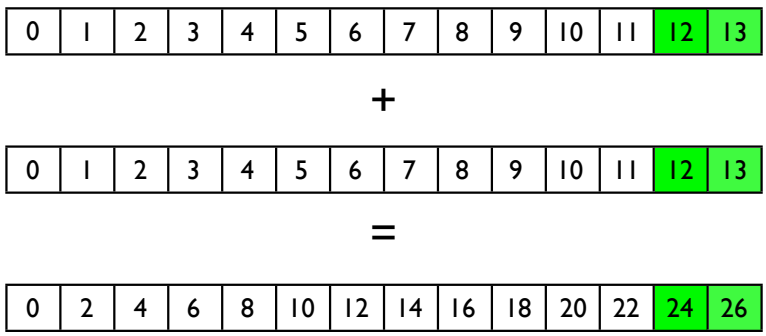
Now increment

```
tid += blockDim.x * gridDim.x
```

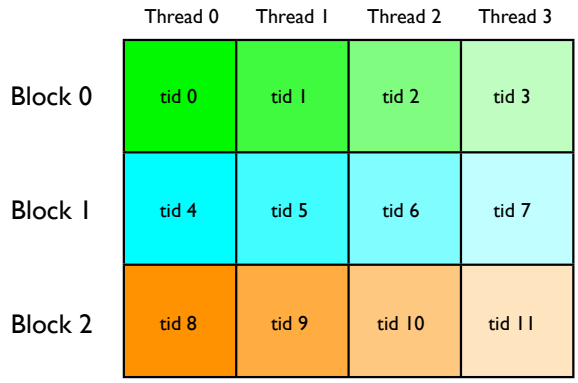
threads / block

blocks

total threads = 12

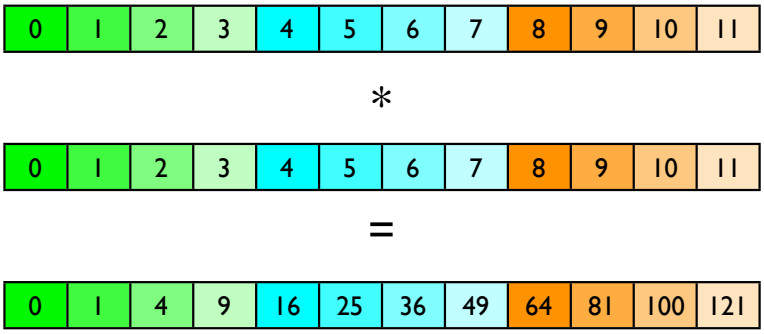


4



Dot product

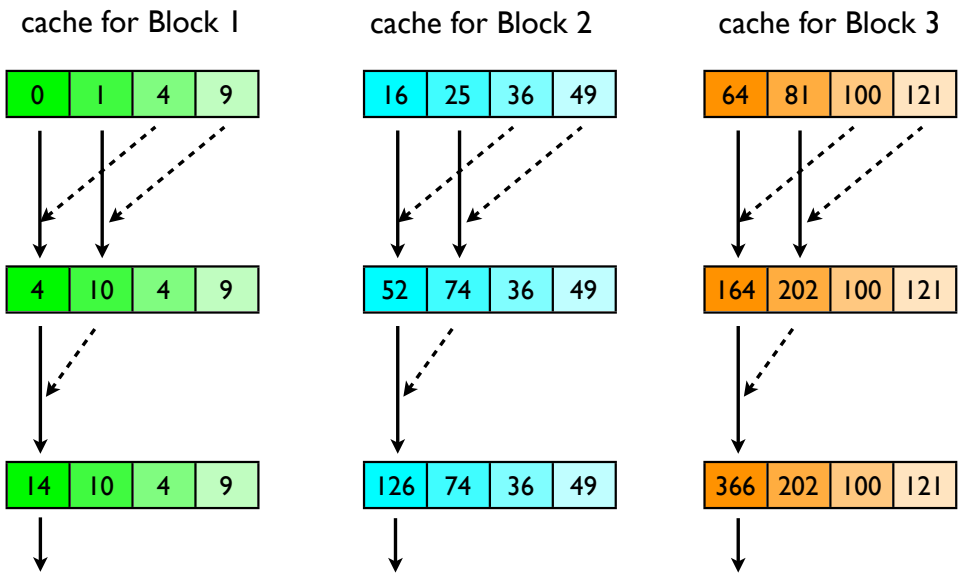
Can do addition on CPU
OR
partially in parallel on GPU



5

Use *shared memory* within each block

```
__shared__ float cache[threadsPerBlock]
```



6

What are consequences of block/thread design?

Too many blocks

The CPU ends up summing elements from a long vector.
Should be doing some of that work in parallel on GPU.

Too few blocks / too many threads per block

Most of the threads are idle during the last few reduction steps.
CPU could be doing the sums while GPU does other work.

Example: *Estimating a normalizing constant*

$$\pi(\gamma \mid y) = \frac{m(y \mid \gamma)\pi(\gamma)}{\sum_{\gamma \in \Gamma} m(y \mid \gamma)\pi(\gamma)}$$

In this example the model probabilities are uniform and so we have

$$\pi(\gamma \mid y) = Cm(y \mid \gamma) \qquad \text{where} \qquad C^{-1} = \sum_{\gamma \in \Gamma} m(y \mid \gamma)$$

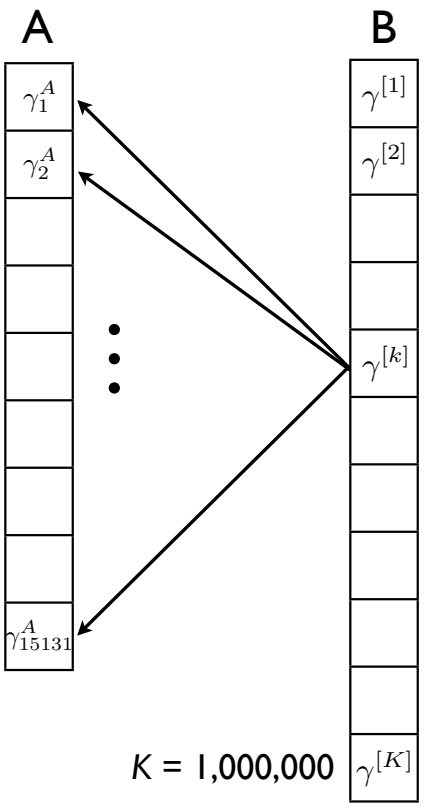
2^p total models (p = 88)

$$C^{-1} = \sum_{\gamma \in \Gamma} m(y \mid \gamma)$$

MCMC: $\gamma^{[1]}, \gamma^{[2]}, \ldots, \gamma^{[K]}$

$A \subset \Gamma$

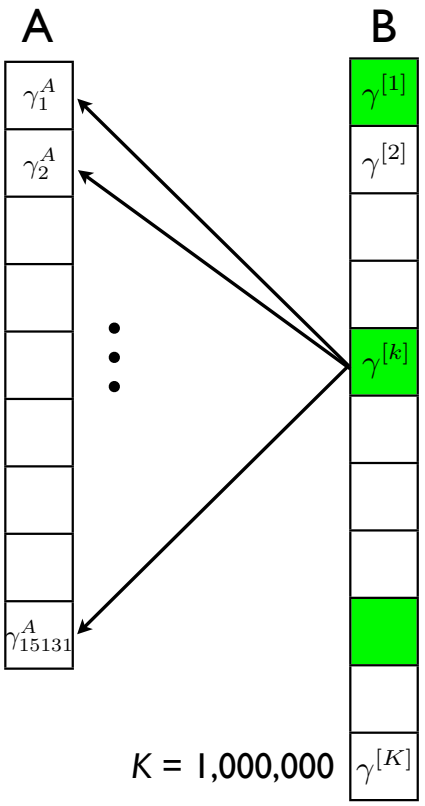
$$\hat{C} = \frac{\frac{1}{K}\sum_{k=1}^{K} I_A(\gamma^{[k]})}{\sum_{\gamma \in A} m(y \mid \gamma)}$$



Block 0	Thread 0	Thread 1	Thread 2	Thread 3
Block 1	Thread 0	Thread 1	Thread 2	Thread 3
Block 2	Thread 0	Thread 1	Thread 2	Thread 3

Each thread is responsible for a certain number of MCMC samples and counts the number of those samples that appear in A.

A reduction is then applied within each block and the CPU completes the sum to find the proportion of entries of B that are also entries in A.



Timing Results

CPU average: `16587.7 milliseconds`

milliseconds

16505.9	16638.2	16770.9	16625.5	16498.4	16487.5

GPU results:

Threads / Block	128 Blocks						Avg.
128	368.5	368.2	368.4	368.5	369.0	369.0	**368.6**
256	318.0	316.3	314.9	314.9	315.8	316.1	**316**
512	314.2	312.2	314.2	314.0	319.0	317.0	**315.1**
1024	342.0	343.5	343.6	345.5	341.1	341.2	**342.82**

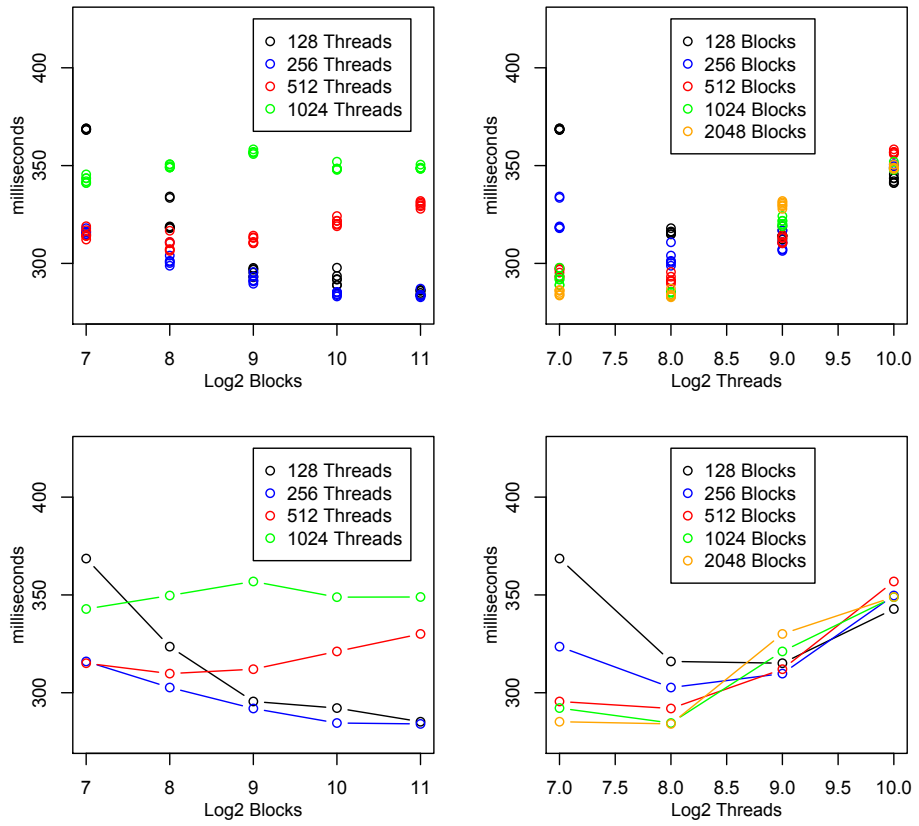
Threads / Block	256 Blocks						Avg.
128	318.3	318.0	333.5	334.2	318.5	318.9	**323.57**
256	298.8	300.2	310.8	304.0	301.3	300.9	**302.67**
512	306.4	307.3	316.7	310.3	311.0	307.0	**309.78**
1024	350.6	349.1	349.7	349.0	349.1	350.7	**349.7**

GPU results:

Threads / Block	512 Blocks						Avg.
128	296.8	297.1	292.9	293.4	297.7	295.3	**295.53**
256	295.6	291.2	293.2	290.8	289.6	291.3	**291.95**
512	313.3	310.4	313.0	310.3	314.2	311.0	**312.03**
1024	357.0	356.8	356.6	355.9	356.7	358.3	**356.88**

Threads / Block	1024 Blocks						Avg.
128	289.0	292.0	291.7	293.6	297.8	288.9	**292.17**
256	283.1	285.2	284.3	283.8	285.3	285.4	**284.52**
512	324.2	321.7	320.2	321.8	319.0	319.8	**321.12**
1024	348.3	352.0	348.4	348.5	347.8	348.2	**348.87**

Threads / Block	2048 Blocks						Avg.
128	286.2	283.6	284.0	284.9	286.0	286.4	**285.18**
256	287.1	283.3	283.8	283.9	283.6	282.7	**284.07**
512	327.9	329.7	330.5	331.9	331.2	329.2	**330.07**
1024	348.9	348.5	350.5	348.9	348.6	348.3	**348.95**



13

Timing Results – Worst Case Scenario

CPU average: `47778.7 milliseconds`

milliseconds					
47332.2	48484.4	47666.4	47715.0	47622.8	47851.2

GPU results:

Threads / Block	128 Blocks						Avg.
128	967.0	964.0	963.2	964.3	963.4	966.5	**964.73**
256	641.9	641.5	639.3	641.9	641.2	643.2	**641.5**
512	637.4	637.7	637.7	637.9	638.4	640.2	**638.22**
1024	634.1	633.7	636.0	633.8	633.8	635.5	**634.48**

Threads / Block	256 Blocks						Avg.
128	737.3	737.0	737.6	737.3	767.4	738.1	**742.45**
256	610.9	611.6	613.1	610.5	610.4	610.3	**611.13**
512	629.4	629.3	630.1	631.9	629.9	630.7	**630.22**
1024	608.2	608.1	608.6	608.9	608.0	608.4	**608.37**

14

Timing Results – Worst Case Scenario

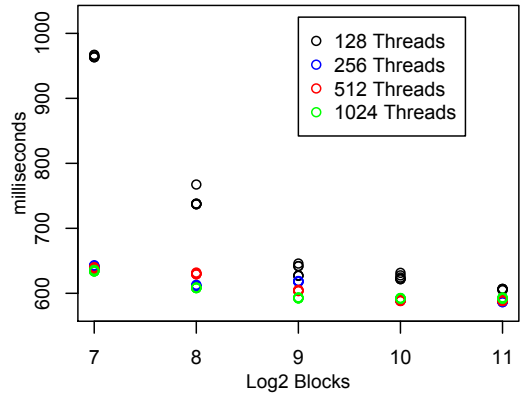
GPU results:

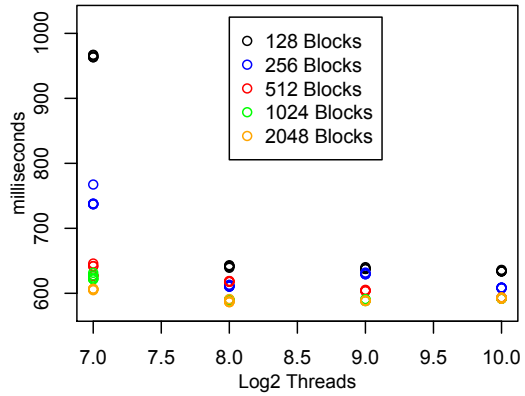
Threads / Block	512 Blocks						Avg.
128	627.4	641.7	641.9	627.3	626.9	645.8	**635.17**
256	617.8	618.0	617.8	618.3	618.9	618.5	**618.22**
512	603.9	603.5	603.6	604.5	604.5	605.3	**604.22**
1024	592.7	593.1	592.2	592.2	593.7	592.9	**592.8**

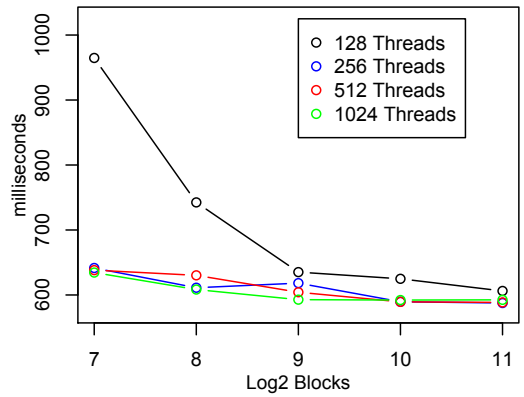
Threads / Block	1024 Blocks						Avg.
128	624.3	622.3	621.7	631.5	627.6	621.9	**624.88**
256	589.6	590.3	589.6	590.3	588.8	588.9	**589.58**
512	590.7	591.0	589.0	588.4	589.2	588.1	**589.4**
1024	591.9	591.8	591.9	592.7	592.6	592.1	**592.17**

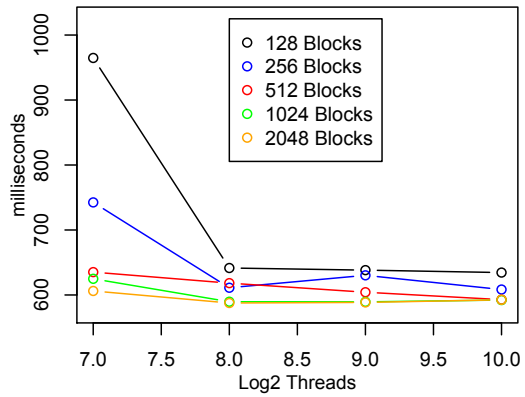
Threads / Block	2048 Blocks						Avg.
128	606.5	606.8	606.5	604.8	605.9	606.5	**606.17**
256	586.6	587.7	591.1	586.3	587.1	587.1	**587.65**
512	588.7	588.4	589.2	588.3	588.2	588.5	**588.55**
1024	592.8	592.2	592.7	591.9	593.7	591.8	**592.52**

15









16