Finale

Chris Kauffman

Last Updated:
Thu Apr 27 01:53:46 PM CDT 2023

Further Coursework

- **EE 5351 Applied Parallel Programming** and
  EE 5355 Algorithmic Techniques for Scalable Many-core Computing
  These two hit CUDA much harder and would build on your understanding of that parallel technology. Based on 5451, possible you would want to go immediately into EE 5355 though discussing with the instructor beforehand would be wise.
- **CSCI 8205 Parallel Computer Organization**: Study hardware issues associated with parallel / multicore machines. Requires significant Hardware background.
- **CSCI 5304 Computational Aspects of Matrix Theory**: Deep dive into using matrices and linear algebra in computing. Essential stuff for those in scientific computing with any self-respect.
- **CSCI 8314 Sparse Matrix Computations**: Focused on sparse operations, offered periodically.
- **Application Areas:** Have seen that machine learning, graphics, cryptography, physical simulations, etc. all benefit from parallel computing. Find a serial algorithm in your domain and parallelize it!

Review Topic: Shared Memory Coordination

- Nearby is a pair of routines which sum random numbers in a sequence
- On running the code and timing with increasing numbers of threads, the following are observed

#threads	niters	time(s)	correct?
serial	10mil	8.05	Yes
omp 1	10mil	8.07	Yes
omp 2	10mil	12.32	No
omp 4	10mil	19.71	No
omp 8	10mil	30.68	No

Explain these and how to fix the parallel version

```c
long randsum_serial(int niters){
  long sum = 0;
  for(int i=0; i<niters; i++){
    long num = rand();
    sum += num;
  }
  return sum;
}

long randsum_omp(int niters){
  long sum = 0;
  #omp parallel
  {
    for(int i=0; i<niters; i++){
      long num = rand();
      sum += num;
    }
  }
  return sum;
}
```

Review Topic: GPU Thread Sync

A

We have seen that CUDA provides the `__synchthreads()` function to synchronize threads within a thread block.
Why is this function necessary? Give an example where it would be useful.

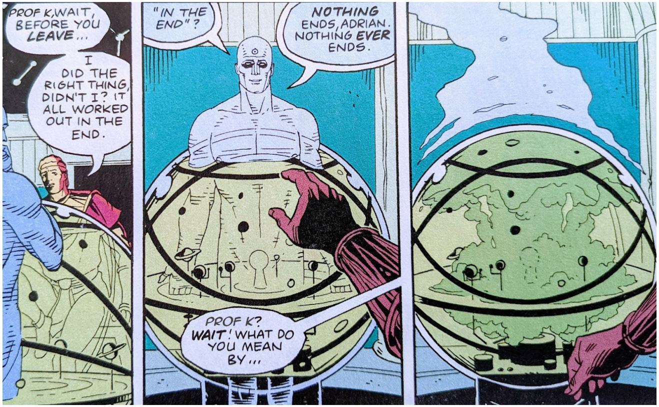
B

What are the limits to `__syncthreads()`?
Which threads does it synchronize and which threads does it not?
What alternatives are available for thread synchronization?

Review Topic Requests

- Distributed Memory Architecture
- Shared Memory Architecture
- MPI for Distributed Memory Programming
- PThreads + OpenMP for Shared Memory Programming
- CUDA for GPU Programming
- Communication Patterns
  - Broadcast, Scatter/Gather, Reductions
  - Synchronization between procs/threads
- Input/Output Problem Decomposition
- Parallel Problems/Applications
  - Sums, Min/Maxes
  - Matrix Multiplication
  - Solving Linear Systems
  - Sorting
  - Basics of Fluid Dynamics, N-Body simulation, Neural Networks, Crypto Mining

Nothing Ever Ends


PROF K, WAIT, BEFORE YOU LEAVE...
I DID THE RIGHT THING, DIDN'T I? IT ALL WORKED OUT IN THE END.
"IN THE END"?
NOTHING ENDS, ADRIAN. NOTHING EVER ENDS.
PROF K? WAIT! WHAT DO YOU MEAN BY...

By now you should realize that what you learned

- Will come up again showing whether you learned it well the first time or need another pass.
- Will change in the future and make you feel old.

Expect this and stay stay patient.

Conclusion

It's been a hell of a semester.
I'm proud of all of you.
Keep up the good work.
Stay safe. Happy Hacking.