Announcements

• Reading assignments are on the website:
  • E-mail 2-4 questions on each paper to bhatele@cs.umd.edu the night before.

• Assignment 1 on MPI is posted and is due on September 23
Summary of last lecture

- Collective operations
- MPI communicators
- Non-blocking point-to-point operations
- Timing MPI programs
OpenMP is a language extension that enables parallelizing C/C++/Fortran code via compiler directives and library routines

- Compiler converts code to multi-threaded code

- Meant for certain kinds of programs/computational kernels
  - Parallelism can be specified for regions and loops

- Fork/join model of parallelism
Fork-join parallelism

Parallel Task I Parallel Task II Parallel Task III

Master Thread

https://en.wikipedia.org/wiki/OpenMP
Hello World in OpenMP

#include <stdio.h>
#include <omp.h>

int main(void) {
    #pragma omp parallel
    printf("Hello, world.\n");
    return 0;
}

Compiling: gcc -fopenmp hello.c -o hello

export OMP_NUM_THREADS=2
A simple OpenMP program

```c
int main(int argc, char **argv)
{
    int a[100000];

    #pragma omp parallel for
    for (int i = 0; i < 100000; i++) {
        a[i] = 2 * i;
    }

    return 0;
}
```
Questions

What is cache coherence? What are the benefits and costs in terms of scalability? What are the protocols for maintaining cache coherence?

What are the advantages of a shared memory model? Of OpenMP compared to an MPI parallelization model?

- Speed up of OpenMP vs MPI?
- If MPI works comparably efficiently, then why do we want to learn OpenMP? Is it for the simpler code detail of implementation?

Is Fortran still a frequently used language for parallel programming now?

Is there any general advice or “rules of thumb” on how we reduce the synchronization requirement in an OpenMP program?
Questions

The Ongoing Evolution of OpenMP

• How has OpenMP evolved through its various iterations? What are the guiding principles that have remained the same? Why?

• How does OpenMP try to address different hardware, available devices and accelerators (vectorization, GPUs, etc.)

• What is the difference between prescriptive and descriptive semantics? What is OpenMP’s approach to this.

• How well does OpenMP work with C++? Would like to know what to avoid when programming with OpenMP + C++.

• How does memory hierarchy affect implementations of memory management in OpenMP?
Questions?

Abhinav Bhave

5218 Brendan Iribe Center (IRB) / College Park, MD 20742

phone: 301.405.4507 / e-mail: bhatele@cs.umd.edu