introduction to parallel computing – course outline

The target autdience for CSE 5441 consists of both students who are interested in obtaining performance gains for advanced scientific computing applications as well as students who have an interest in compiler technology. While it is not a compiler class, we study how to influence compiler behavior, as well as develop an understanding of some of the considerations compiler developers must have in mind when automating performing optimizations.

CSE 5441 is not an Electrical Engineering class. While we study cache and cache behavior we will use a simplified model to retain focus on programming techniques. We will not attempt to design cache circuitry, but will develop an understanding of many of the issues that cache designers must consider and some of the trade-offs between various general cache designs.

We will begin with programming techniques that, based upon their expected cache behavior, can benefit both serial and parallel programs and then study four different APIs for parallel programming.

1 introduction

- background and justification for study
- power density
- Moore's "Law"
- hyperthreading, pipelining

2 high-performance single-core methods

2.1 cache management

section goal: to understand different cache implementations and their impact on performance, to write programs which make effective use of cache memory.

- is all memory the same? - the need for cache memory
- cache architectures
  - single -vs- multi-level
  - direct-mapped -vs- associative cache
- locality of reference
- Amdahl's Law
- thrashing
- cache write strategies
- cache performance metrics

programming project 1: a simplified scientific computing application

2.2 loop analysis and transformations

section goal: to understand the impact of loop structure on performance. to develop familiarity with methods to compare the performance of various loop structures.

- access stride and spatial locality
- loop permutations
- loop unrolling
- blocking and tiling

2.3 data dependence analysis

section goal: to understand methods for analyzing the correctness of loop transformations.

- flow dependence, anti-dependence and output dependence
- iteration space
- dependence vectors, distance and direction vectors

3 parallel programming methods

3.1 global shared memory architecture - Posix Threads (pthreads)

section goal: implement parallel applications with explicit thread creation/destruction.

- Single Instruction Multiple Data (SIMD)
- parallel regions
- barriers and synchronization
- critical regions and atomicity

programming project 2: disposable and persistent threads

midterm exam

3.2 shared memory architecture - OpenMP

section goal: implement pragma-based multi-threaded solutions.

- pragmas
- false sharing
- OpenMP syntax
- OpenMP data sharing

programming project 3: controlling threads via pragmas

3.3 Graphics Processing Unit - CUDA

section goal: experience applying Graphics Processing Units as MP accelerators for numerical analysis

- GPU architecture
- CUDA programming model
- thread hierarchies
- thread scheduling
- GPU memory model
- GPU thread synchronization

programming project 4: GPU as a scientific computing accelerator

3.4 message passing - MPI

section goal: to develop an understanding of communications-based MP at the process level.

- reductions
- MPI basic concepts and organization
- MPI data types
- MPI communications methods
- blocking and non-blocking communications, deadlock
- MPI data movement
- MPI synchronization
- current MPI restrictions and limitations

programming project 5: distributed computing

4 vectorizing compilers

section goal: an appreciation of the benefits and limitations of automatic compiler vectorization.

- Single Instruction Multiple Data (SIMD)
- dependence graphs
- vectorization intrinsics

final exam

CSE 5441: Proposed Schedule

Every semester is a little different and not everything always goes according to plan. This information is provided for your planning purposes and indicates our intended schedule. Specific dates may vary with updates to this schedule being announced in class.

DATE	LECTURE TOPIC	PROGRAMMING ASSIGNMENTS
8/22	introduction	
8/24	direct-mapped cache	
8/29	direct-mapped cache	
8/31	direct-mapped cache	
9/5	associative cache	
9/7	associative cache	
9/12	loop analysis	lab 1 assigned
9/14	loop analysis	
9/19	loop analysis	
9/21	POSIX threads	
9/26	no lecture - career fair	lab 1 due 11:59pm
9/28	POSIX threads	
10/3	POSIX threads	lab 2 assigned
10/5	review	
10/10	MIDTERM EXAM (in class)	
10/12	AUTUMN BREAK – no classes	
10/17	Ohio Supercomputer Center	
10/19	OpenMP	lab 2 due 11:59pm
10/24	OpenMP	lab 3 assigned
10/26	parallel performance monitoring	
10/31	CUDA	
11/2	CUDA	lab 3 due 11:59pm
11/7	CUDA	lab 4 assigned
11/9	Message Passing Interface	
11/14	Message Passing Interface	
11/16	Message Passing Interface	lab 5 assigned
11/21	no lecture - programming day	lab 4 due 11:59pm
11/23	THANKSGIVING – no classes	
11/28	vectorizing compilers	
11/30	review	lab 5 due 11:59pm
12/5	FINAL EXAM (in class)	