Early Adopter: ASU - Intel Collaboration in Parallel and Distributed Computing

Yinong Chen, Eric Kostelich, Yann-Hang Lee,
Alex Mahalov, Gil Speyer, and Violet R. Syrotiuk

Ira A. Fulton Schools of Engineering
ARIZONA STATE UNIVERSITY

1st NSF/TCPP Workshop on Parallel and Distributed Computing Education (EduPar'11)
In conjunction with ***25th IEEE International Parallel and Distributed Processing Symposium***
Anchorage (Alaska), USA
Monday, May 16, 2011

Arizona State University

- Arizona State University (ASU) now has the largest campus in the U.S.A.
- The Tempe campus is one of four campuses
  - More than 51,000 students
- Focus on research and graduate education along with an analytic undergraduate education preparatory for graduate or professional school or employment

Ira A. Fulton Schools of Engineering
ARIZONA STATE UNIVERSITY

School of Computing, Informatics and Decision Systems Engineering

- One of five schools of engineering at ASU
- Enrollment:
  - 1100 undergraduate and 550 graduate students
- Includes degree programs in:
  - Computer Science (CS)
  - Computer Systems Engineering (CSE)

Our Initial Goal

- Integrate topics in parallel and distributed computing into the:
  - Computer Science (CS),
  - Computer Systems Engineering (CSE), and
  - Mathematics and Statistical Sciences (MAT)

  programs at the undergraduate and MS level
- Leverage the High Performance Computing (HPC) initiative at ASU

Courses in Early Adopter Program

Course Number	Course Name	Pilot	Enrollment
ASU 101-CSE	The ASU Experience	Spring 2011	38
CSE 310	Data Structures and Algorithms	Fall 2011	110*
CSE 430	Operating Systems	Spring 2011	36
CSE 445/598	Distributed Software Development	Spring 2011	55
CSE 494/598	Introduction to Parallel Programming	Spring 2011	17
MAT 420	Scientific Computing	Spring 2011	26

* Projected

ASU 101-CSE
The ASU Experience

- Introduce the discipline of computer science and engineering
- Illustrate benefits of parallel and distributed computing
  - Demonstrate improvements in frame rate using Intel's game demos on a single v. multiple cores

ASU 101-CSE
The ASU Experience

Topics	Bloom #	Learning Outcome
What is PDC?	K	Common issues and differences between parallel and distributed computing; history
Tasks and threads	K	Relation between number of threads and number of cores
Speed-up	K	Use parallelism to solve the same problem faster or larger problems in the same time

CSE 310
Data Structures and Algorithms

- Advanced data structures and algorithms, and algorithmic analysis
  - Data structures include heaps, hash tables, red-black trees, and graphs among others
  - Algorithms for sorting, selection, graph algorithms
  - Algorithmic techniques, e.g., divide-and-conquer, dynamic programming, and greedy algorithms

CSE 310
Data Structures and Algorithms

- To date, only sequential algorithms
- Plan to adopt recommendations in NSF/TCPP Curriculum Initiative Report for DS/A, e.g.,

Topics	Bloom #	Learning Outcomes
Parallel and distributed models and complexity	C,K	Costs of computation, asymptotics, time, space, speed-up, cost trade-offs, model based notation (PRAM)
Algorithmic paradigms	C	Divide-and-conquer, recursion
Algorithmic paradigms	A, C, K	Sorting, selection, graph algorithms

CSE 430
Operating Systems

- Operating system structures and services
- Emphasis on concurrent processes using Intel's Parallel Studio, game demos, and tools
  - mutual exclusion and synchronization, race conditions, deadlocks, threads, semaphores, concurrent programming paradigms
- Also scheduling, virtual memory, file systems, I/O and mass-storage systems, protection

CSE 430
Operating Systems

Topics	Bloom #	Learning Outcomes
Shared v. distributed memory	K	UMA and NUMA architectures, distributed memory, client server
Parallel programming notation	A, C, K	Language extensions, compiler directives/pragmas, libraries
Semantics and correctness issues	A, C, K	Tasks and threads, synchronization, concurrency defects and tools for their detection
Performance metrics and issues	A, C, K	Define/measure benchmarks, performance monitoring
Algorithmic problems	A, C, K	Asynchrony, synchronization

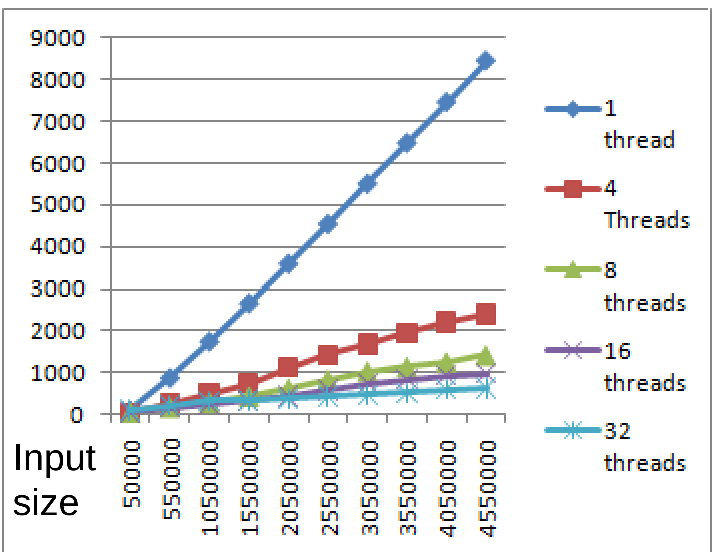
CSE 445/598 Distributed Software Development

- In service-oriented distributed systems, server applications may be invoked by multiple clients
- Multithreading with parallel computing and data synchronization using Intel's Thread Building Blocks (TBB) is discussed
- Performance analysis and case study

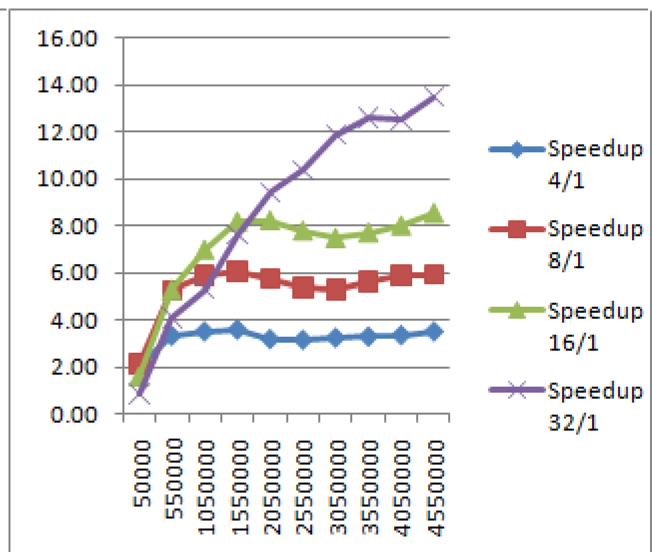
Results on Intel 32-Core MTL

Input size	1 thread	4 Threads	8 threads	16 threads	32 threads	Speedup 4/1	Speedup 8/1	Speedup 16/1	Speedup 32/1	Efficiency 4/1	Efficiency 8/1	Efficiency 16/1	Efficiency 32/1
50000	95.8	46.3	44.7	60.3	110.4	2.07	2.14	1.59	0.87	52%	27%	10%	3.2%
550000	879.4	263.2	166.6	166.6	214.9	3.34	5.28	5.28	4.09	84%	66%	33%	5.2%
1050000	1748.5	497.1	297.5	250.9	328.9	3.52	5.88	6.97	5.32	88%	73%	44%	5.5%
1550000	2658.1	738.1	439.4	325.6	349.1	3.60	6.05	8.16	7.61	90%	76%	51%	5.6%
2050000	3589.4	1119.7	625.1	436.5	380.3	3.21	5.74	8.22	9.44	80%	72%	51%	5.0%
2550000	4534.7	1430.1	845.2	583	436.5	3.17	5.37	7.78	10.39	79%	67%	49%	5.0%
3050000	5497.1	1685.9	1037	735.9	463	3.26	5.30	7.47	11.87	82%	66%	47%	5.1%
3550000	6468.9	1954.5	1150.8	839	512.6	3.31	5.62	7.71	12.62	83%	70%	48%	5.2%
4050000	7443.9	2208.7	1261.6	931.1	594.1	3.37	5.90	7.99	12.53	84%	74%	50%	5.3%
4550000	8434.8	2397.4	1428.6	988.6	625.3	3.52	5.90	8.53	13.49	88%	74%	53%	5.5%
Average						**3.24**	**5.32**	**6.97**	**8.82**	**81%**	**66%**	**44%**	**5%**

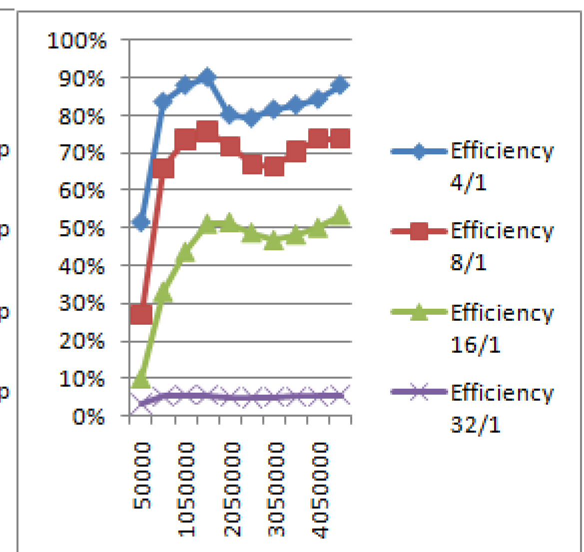
Time measured in milliseconds



Speedup



Efficiency



CSE 445/598 Distributed Software Development

Topics	Bloom #	Learning Outcomes
Distributed architectures	C	Understand the differences between distributed architectures and their impact to algorithms
Client-server architectures	A	Development and implement programs in different architectures such as thin and thick client and N-tier architectures
Control flow v. event-driven programming	C, A	Identify the needs of both programming paradigms and be able develop applications in both paradigms
Web execution model	C	Understand different Web execution models and their applications in Web and cloud computing
Multithreading	C, A	Resource sharing, synchronization, and performance impact
Parallel issues	C	Be able identify the parts of algorithms that can be execution independent of other parts and use different threads or blocks to implement them

CSE 494/598
Introduction to Parallel Computing

- Introduce the fundamentals of parallel computing and important techniques and practices in parallel programming
- Various models and applications
- A sampling of current topics in high performance computing

CSE 494/598
Introduction to Parallel Computing

Topics	Bloom #	Learning Outcomes
Architecture	C, K	Superscalar, SIMD/vector, pipelines, streams, MIMD, multicore, heterogeneous OpenMP&MPI, SMP, NUMA, message passing, atomicity, false sharing, etc.
Programming	A, C, K	Shared v. distributed memory, programming CPU v. GPU, SPMD, data parallel, parallel loop, SSE macros, OpenMP pragmas, apply BLACS and ScaLAPACK, Pthreads, TBB, etc.
Algorithms	A, C, K	Matrix multiplication, numerical integration, unfold yielding tree structures, recursive doubling, blocking communications/sync, inherent parallelism, non-determinism, power metric, locality, latency

MAT 420
Scientific Computing

- Directed at majors in applied mathematics and the physical sciences
- Emphasis on programming tools and paradigms for high-performance computing
- Topics include Fortran 95, C++ (STL), major scientific libraries, OpenMP, MPI, LAPACK, MKL, coarrays in Fortran 2008

MAT 420
Scientific Computing

Topics	Bloom #	Learning Outcomes
Multicore	A, C	Know basic OpenMP directives
Distributed memory	A, C	MPI blocking send and receive operations
LAPACK	A, C	Use multithreaded linear algebra library (e.g., Intel's MKL) for matrix multiplication and LU factorizations
Fortran 2008	A, C	Coarrays; do concurrency and sync_team constructs
Boost library	A, C	C++ Boost library templates for matrices and basic operations, including matrix multiplication and LU factorizations

Course	Evaluation Methodology
ASU 101	Multiple choice questions in final exam
CSE 310	Pilot in Spring 2011
CSE 430	Project with Intel's Parallel Studio using OpenMP pragmas, examine data locality, load balancing, acceleration, profiling, etc. Midterm and final exams
CSE 445/598	Multithreaded programming project: test on single and multicore environments; use Intel's TBB library and measure speed-up Service hosting project: explore parallelism on the server side
CSE 494/598	Four projects: • OpenMP pragmas, data locality, load balancing, acceleration, profiling, etc. • MPI collective communication and data decomposition • MPI parallel I/O, local communication and a parallel library • CUDA, GPU programming Midterm and final exams
MAT 420	Example projects: • Use of Fortran 95/2008 vector constructs to compute the Mandelbrot set • OpenMP to parallelize a PDE solver using finite differences • basic blocking MPI_Send and MPI_Recv calls to implement a PDE solver on a distributed memory cluster

Future Plans

- Many opportunities exist to integrate PDC topics into our curriculum
- Spring 2011:
  - Work on our 100-200 level courses
  - Approval of new syllabi by our undergraduate curriculum committee and program faculty

Future Plans (cont'd)

- Summer 2011:
  - Revise our 300-400 course syllabi
  - Introduce a new CSE 4xx course on Parallel and Distributed Computing
  - Develop a "data bank" of course materials, e.g.,
    - Lecture materials (e.g., slides, videos, demos)
    - Programming project ideas
    - Sample homework questions
    - Sample exam questions

Thanks to

intel

ASU School of Mathematical & Statistical Sciences

Ira A. Fulton
ASU Advanced Computing Center