병렬프로그래밍

개설학기	2018학년도 2학기
소　　속	컴퓨터학부
교 수 명	백낙훈
학　　점	3
수업목표	- Focusing on the capability of parallel programming capability. - Also Focusing on the capability of CUDA, OpenCL, OpenMP programming.

주차	주차별 학습내용	차시(모듈)	차시별 학습내용	학습목표	학습목차	학습자료	PPT 슬라이드 수
1	Introduction to the course. History of GPU Compuing	1	cuda-prelude	1. Course Objectives 2. Lecturer 3. Textbooks 4. 수업의 구성	1. Course Objectives 2. Lecturer 3. Textbooks 4. 수업의 구성	참고문헌, ppt 교안	17
		2	trends	1. FLOPS 2. Super Computer Trends 3. Personal Super-Computing 4. Energy Efficient 5. Architectures	1. FLOPS 2. Super Computer Trends 3. Personal Super-Computing 4. Energy Efficient 5. Architectures		25
		3	intro	1. CPU and GPU 2. Moore's Law 3. Massively Parallel Processing	1. CPU and GPU 2. Moore's Law 3. Massively Parallel Processing		27
2	Data Parallelism and CUDA C Data Parallel Execution Model	1	history	1. 3D Graphics Pipeline 2. hardware Development 3. Early GPGPU	1. 3D Graphics Pipeline 2. hardware Development 3. Early GPGPU	참고문헌, ppt 교안	35
		2	cuda-install	1. before installation 2. CUDA-capable GPU 3. CUDA installation steps	1. before installation 2. CUDA-capable GPU 3. CUDA installation steps		23
		3	memcpy	1. simple CUDA model - host, device 2. CUDA memory copy	1. simple CUDA model - host, device 2. CUDA memory copy		25
		1	errorCheck	1. host-device memory copy example 2. CUDA function rules	1. host-device memory copy example 2. CUDA function rules		27

3	CUDA Memories Performance Consideration	2	CPU-kernel	1. CUDA programming model 2. CUDA function declarations 3. vector addition example	1. CUDA programming model 2. CUDA function declarations 3. vector addition example	참고문헌, ppt 교안	25
		3	CUDA-kernel	1. CUDA programming model 2. CUDA function declarations 3. vector addition example	1. CUDA programming model 2. CUDA function declarations 3. vector addition example		25
4	Floating-Point Considerations Parallel Patterns	1	kernel-launch	1. process and thread 2. CUDA programming model 3. kernel launch	1. process and thread 2. CUDA programming model 3. kernel launch	참고문헌, ppt 교안	20
		2	matrix-add	1. CUDA thread hierarchy 2. matrix addition example 3. host implementation	1. CUDA thread hierarchy 2. matrix addition example 3. host implementation		26
		3	matrix-mult	1. matrix multiplication example 2. host implementation	1. matrix multiplication example 2. host implementation		26
5	Parallel Patterns, again	1	thread-and-gpu	1. CUDA hardware - G80 architecture - Pascal architecture	1. CUDA hardware - G80 architecture - Pascal architecture	참고문헌, ppt 교안	28
		2	tiled-matrix-mult	1. Review for the problem 2. Thread Layout – single block	1. Review for the problem 2. Thread Layout – single block		28
		3	device-query	1. Review for the kernel function 2. Detailed Source Codes	1. Review for the kernel function 2. Detailed Source Codes		30
6	Application Case Studeis	1	time-funcs	1. Get the current time 2. sleep( ) function 3. QueryPerformanceCounter(	1. Get the current time 2. sleep( ) function 3. QueryPerformanceCounter(	참고문헌, ppt 교안	25
		2	memHier1	1. von Neumann Architecture - instruction cycles 2. History of Parallelism	1. von Neumann Architecture - instruction cycles 2. History of Parallelism		28
		3	memHier2	1. CUDA memory hierarchy - registers - shared memory	1. CUDA memory hierarchy - registers - shared memory		25
7	Parallel Programming and Computational Thinking	1	adjDiff1	1. Example: Adjacent Difference 2. Host version	1. Example: Adjacent Difference 2. Host version	참고문헌, ppt 교안	33
		2	adjDiff2	1. Example: Adjacent Difference 2. Shared Memory Use	1. Example: Adjacent Difference 2. Shared Memory Use		25
		3	sharedMatMult	1. Example: matrix multiplication 2. Tile load to shared memory	1. Example: matrix multiplication 2. Tile load to shared memory		26

8	Introduction to OpenCL Platfrom, Context, Device	1	sharedMatMult	1. Example: matrix multiplication 2. Tile load to shared memory	1. Example: matrix multiplication 2. Tile load to shared memory	참고문헌, ppt 교안	29
		2	float	1. IEEE 754 floating point standards 2. normalized representation	1. IEEE 754 floating point standards 2. normalized representation		24
		3	dram	1. Dynamic RAM (random access memory) 2. DRAM organization	1. Dynamic RAM (random access memory) 2. DRAM organization		27
9	OpenCL Programming OpenCL Built-in functions	1	memCoalescing	1. Memory alignment 2. Memory access examples 3. Stride considerations	1. Memory alignment 2. Memory access examples 3. Stride considerations	참고문헌, ppt 교안	24
		2	bankConflicts	1. 2D and 3D array cases 2. shared memory handling 3. banked memory	1. 2D and 3D array cases 2. shared memory handling 3. banked memory		23
		3	occupancy	1. thread scheduling 2. occupancy 3. resource limits	1. thread scheduling 2. occupancy 3. resource limits		21
10	EGL and Shaders.	1	control	1. Streaming Multiprocessor has only one control logic 2. control flow	1. Streaming Multiprocessor has only one control logic 2. control flow	참고문헌, ppt 교안	24
		2	atomic1	1. Race Conditions 2. Atomics 3. AtomicCAS operation:	1. Race Conditions 2. Atomics 3. AtomicCAS operation:		25
		3	atomic2	1. Atomics 2. Histogram Example 3. host version	1. Atomics 2. Histogram Example 3. host version		24
11	OpenGL ES Shader Language	1	reduction1	1. Reduction Problem 2. Sequential Reduction Algorithms	1. Reduction Problem 2. Sequential Reduction Algorithms	참고문헌, ppt 교안	26
		2	reduction2	1. Reduction Problem 2. A Sum Reduction Example 3. Host Version	1. Reduction Problem 2. A Sum Reduction Example 3. Host Version		26
		3	reduction3	1. Reduction Problem Again 2. Parallel Reduction 3. Sequential Addition	1. Reduction Problem Again 2. Parallel Reduction 3. Sequential Addition		24
12	Program and Kernel Buffer and Sub-buffer	1	reduction4	1. Reduction Problem Again 2. Parallel Reduction 3. Sequential Addressing	1. Reduction Problem Again 2. Parallel Reduction 3. Sequential Addressing	참고문헌, ppt 교안	31
		2	streaming	1. streaming operation = definition 2. host version	1. streaming operation = definition 2. host version		28
		3	asyncCopy	1. synchronous streaming case 2. asynchronous streaming	1. synchronous streaming case 2. asynchronous streaming		17
13	Image, Sampler, Event Interaction with OpenGL	1	sorting	1. Sorting Algorithms 2. Sorting Networks 3. comparators	1. Sorting Algorithms 2. Sorting Networks 3. comparators	참고문헌, ppt 교안	24
		2	bitonicSort	1. Bitonic Sort 2. Bitonic Sort Network 3. Implementations	1. Bitonic Sort 2. Bitonic Sort Network 3. Implementations		29

		3	search	1. Search for Unsorted List - Sequential Search 2. Search for Sorted List	1. Search for Unsorted List - Sequential Search 2. Search for Sorted List		24