

Index

A

Application programming interface (API), 34, 77, 161
Argument list, 113, 217, 284, 430
Associative cache, 23
`atomic` block, 264
`atomic directive`, 259–261, 263, 264, 278, 283, 284
`atomicAdd`, 450
Autonomous processors, 35, 80

B

Barrier
 in Pthreads, 190
 interface for Pthreads, 193
`barrier` directive, 280, 282
Binary semaphore, 185, 186
Bitonic sort, 338, 347, 348, 358, 360, 435–437
 CUDA, 346, 348
 parallel, 345, 347, 349
 program, 360
 serial, 338, 340, 346–348, 357
 serial implementation, 345
Block
 cache, 22, 202, 212, 266, 280
 code, 4, 53, 55, 81, 82, 146, 174, 186, 207, 211, 221, 226, 260, 264, 271, 274, 277, 281, 282, 289, 373
 cyclic, 119, 360
 distribution, 118, 120, 122, 123, 149, 371, 381, 446
 mapping, 381
 memory, 19, 26, 34, 59, 95, 97, 113, 160, 202, 213, 266, 293
 parallel, 233–236, 243, 244, 259, 270, 281, 284
 partition, 117, 146, 360, 370, 371, 374, 376, 377, 396, 444, 445
 partitioning, 146, 237, 249, 279, 379, 383, 445
 schedule, 379, 445
 single thread, 346, 347, 352, 358, 397, 408, 430, 437, 441
 submatrices, 455
 sum, 332
 threads, 299, 301, 305, 321, 322, 324, 326, 328–335, 337, 347, 348, 393, 395, 397, 407, 408, 430, 432, 434–437, 440
Blocked threads, 180, 190, 191, 261, 278
Broadcast function, 113

C

C compiler, 8, 92, 165, 221, 222, 295, 296
Cache
 block, 22, 202, 212, 266, 280
 coherence, 45, 46, 81, 203, 205, 212, 265, 266, 269, 280
 controller, 47, 81, 206, 269
 data, 212, 281
 design, 23
 hierarchy, 22
 hit, 22, 78
 issues, 379
 line, 22–25, 33, 47, 48, 81, 202, 205–207, 212, 218, 266, 269, 270, 280, 285, 395, 446
location, 23
mappings, 23
memory, 160, 202, 222, 265, 266
misses, 22, 25, 78, 217, 284, 370, 379, 446
performance, 75, 204, 268, 371, 445
profiler, 204, 217, 268, 284
profiling, 217, 284
Central processing unit (CPU)
 caches, 21, 22, 24, 25, 27, 45, 47, 78, 79
 memory, 293
 registers, 78
Child threads, 226, 277
`clock` function, 67
Collective communications, 90, 108, 110, 112, 113, 116, 123, 145, 150, 383
Collective communications nonblocking, 455
Command line, 76–78, 164, 168, 210, 214, 223–226, 254, 296, 298, 299, 305, 308, 351, 382, 437
Communication
 channels, 102
 function, 126, 152
 mechanism, 98
 MPI, 118
 network, 37
 overhead, 371
 pattern, 113
Compiler
 CUDA, 77, 295, 296
 optimizations, 176, 211, 213
Consumer threads, 186, 256
Contiguous
 blocks, 24, 441
 instructions, 78
 memory, 367

memory locations blocks, 21
 sublist, 360
CUDA
 API, 295, 302, 456
 bitonic sort, 346, 348
 capability, 393, 398
 compiler, 295, 296
 compiler `nvcc`, 77, 78, 296
 function, 307, 309, 350
 function `cudaMalloc`, 309
 function `cudaMemcpy`, 311
 global memory, 395
 heap sort, 434
 hello, 295
 implementation, 315, 394, 407, 408, 431, 434, 447
 kernel, 296, 298, 299, 304, 312, 317, 329, 330, 337, 350
 library, 318, 319
 memory allocation, 307
 parallelization, 441
 sample sort, 432, 434, 438
 solver, 390, 393
 thread, 296, 316, 434, 456
 thread blocks, 347, 407
 trapezoidal rule, 314, 319, 320, 331
 vector addition program, 311
 version, 314, 329
`cuda_hello` program, 355
`cudaDeviceSynchronize`, 296, 298, 312, 314, 353, 355
`cudaError_t`, 307, 351
`cudaFree`, 308, 312
`cudaMemAttachGlobal`, 307
`cudaMemcpy`, 312
`cudaMemcpyDeviceToHost`, 311
`cudaMemcpyHostToDevice`, 311
`cudaMemcpyKind`, 311
 Cyclic distribution, 122, 149, 157, 371, 383, 388, 446, 447

D

Data cache, 45
 Data parallel programs, 80
 Data parallelism, 441
 Datapaths, 31, 32, 34, 292, 293, 350
 Deadlock, 215, 265, 278
 Default schedule, 250, 252, 255, 379
 Destination thread, 256, 257
 Digital signal processor (DSP), 291, 295
 Directive
 barrier, 280, 282
 parallel, 225–227, 232, 233, 235, 237, 238, 247, 248, 259, 271, 277, 373, 374
 single, 271, 272, 281, 289, 373, 374, 445

E

Embarrassingly parallel, 51, 81, 408, 409
 implementations, 407
 program, 51
 Erroneous parallelization, 377
Error
 codes, 93, 168, 214
 message, 165
 Exclusive prefix sums, 403, 405, 406, 413, 420, 422, 430–432, 435, 441

F

Fibonacci numbers, 240, 271, 273
 Field programmable gate array (FPGA), 291, 295

G

`gcc` compiler, 77, 214, 239
 General Purpose Programming on Graphics Processing Unit (GPGPU), 291, 349
 programming, 295, 350
 programs, 291, 295
Graphics Processing Unit (GPU), 10, 13, 32, 34, 61, 291, 292, 295, 302, 390, 435, 456, 457
 cores, 77
 memory, 58, 293, 322
 Nvidia, 10, 291, 292
 performance, 61, 70
 threads, 61, 77

H

Heterogeneous programming, 58, 350
 Homogeneous MIMD
 program, 83
 systems, 81–83
 systems performance, 61

I

Idle
 thread, 169
 time, 67, 129, 147
 Infinite loop, 214, 238
 Interblock synchronization, 430
 Interleaved memory, 33
 Internode communication, 60
 Intertask communication, 369

K

Kernel code, 318, 431, 449

L

Linear speedup, 62, 70, 82, 83, 87, 133, 134, 181
 Lock function, 53, 198, 261
 Locked
 mutex, 185
 semaphore, 211

- L**
- Locking, 181, 197, 199, 201
 - Loop
 - body, 241, 278
 - iterations, 249, 252, 270, 381
 - parallel, 249, 289
 - serial, 6, 249, 251, 316
 - variable, 172, 238, 244, 342, 348, 380
 - variable `thread`, 165
- M**
- Machine instructions, 319
 - Master thread, 51, 61, 226, 259, 261, 376, 445
 - Memory
 - access, 22, 34, 36, 62, 75, 79, 160, 206, 212, 216, 252, 269, 280, 439
 - access latency, 293
 - addresses, 26
 - bandwidth, 33
 - banks, 85
 - block, 19, 26, 34, 59, 95, 97, 113, 160, 202, 213, 266, 293
 - cache, 160, 202, 222, 265, 266
 - chip, 21
 - consumption, 411
 - contiguous, 367
 - device, 293, 307, 352
 - hierarchy, 84
 - local, 47, 58, 123, 171
 - modules, 38
 - private, 8, 31, 55
 - private block, 80
 - requirements, 380, 429, 439, 441
 - single word, 212, 280
 - system, 35, 85
 - transfers, 309
 - usage, 451, 452
 - Message-passing interface (MPI), 8
 - collective communication, 112, 129, 146, 381, 408, 442
 - communication, 118
 - communicator, 393
 - Microprocessor, 1
 - Monolithic processors, 3
 - `MPI_Allreduce`, 145
 - `MPI_Gather`, 119, 121, 123, 146, 149, 151, 152, 421, 425, 442
 - `MPI_Gatherv`, 151, 421, 425, 442
 - `MPI_Recv`, 97–100, 110, 112, 115, 129, 140, 145, 148, 152, 153
 - `MPI_Reduce`, 111–113, 130, 145, 146, 150, 154
 - `MPI_Reduce_Scatter`, 156
 - `MPI_Send`, 95–97, 99, 100, 110, 115, 140, 145, 148, 152, 153, 387, 427, 443
 - `MPI_Sendrecv`, 141–143, 148, 387, 426, 428
 - `MPI_Sendrecv_replace`, 148, 387, 447
 - `MPI_Wait`, 428, 443
- M**
- Multicore processors, 3, 36, 37, 49, 60
 - Multilevel caches, 22
 - Multiple
 - cores, 3, 6, 11–13, 49, 60, 82
 - data stream processors, 292
 - datapaths, 31, 292
 - issue processors, 29, 79
 - processes, 49, 55, 60–62, 68, 105, 107, 148, 371, 446
 - processors, 1, 6, 46, 78
 - warp sums, 329
 - Multiple instruction multiple data (MIMD), 8
 - parallel implementations, 408
 - parallel program performance, 66
 - performance, 70
 - programs, 61, 67, 70, 75, 77
 - systems, 9, 10, 34, 35, 60, 61, 66, 70, 74, 77, 80, 407, 441
 - threads, 434
- M**
- Multithreaded
 - linked lists, 194, 198
 - tokenizer, 208
 - Mutex
 - function, 53
 - implementation, 201
 - program, 181
 - solution, 182
 - version, 214
 - Mutual exclusion, 53, 54, 62, 63, 81, 178, 219, 261, 263, 264, 278
 - Mutual exclusion lock, 53, 81
 - Mutually exclusive access, 211
- N**
- Nested
 - locks, 262
 - loops, 24, 25, 84
 - Nonblocking
 - collective communications, 455
 - communications, 428
 - operations, 428, 456
 - versions, 456
 - `nowait` clause, 374, 445
 - Null pointer, 184, 225
 - `num_threads` clause, 226, 277
 - Nvidia
 - GPUs, 10, 291, 292
 - processors, 34
- O**
- `omp_get_num_threads`, 227

`omp_get_thread_num`, 227, 278, 283
`Omp_get_wtime`, 281, 282

`omp_lock_t`, 262
`OMP_NUM_THREADS`, 410
`omp_set_lock`, 263
`omp_unset_lock`, 263, 278

OpenCL program, 291

OpenMP

- architecture, 456
- compilers, 240
- construct, 279
- directives, 225, 259, 277, 279, 286, 440, 441
- header file, 224
- implementation, 247, 248, 374, 380, 389, 410, 415, 445, 449
- parallel region, 410, 415
- reduced solver, 379
- simple lock, 262
- simulations, 390
- solver, 389, 390, 452
- specification, 222, 263
- standard, 225, 226, 236, 456
- tasks, 273
- threads, 226, 390
- trapezoidal rule program, 231

Operating system (OS), 78

P

Parallel

- algorithm, 139, 361, 423, 432, 440, 457
- APIs, 361
- architectures, 33
- bitonic sort, 345, 347, 349
- deterministic sample, 417
- efficiency, 63, 133, 134
- execution time, 129
- Fibonacci program, 273
- hardware, 11, 17, 31, 49, 79, 85, 457
- mergesort, 156
- MIMD programs, 70
- MIMD systems, 77
- overhead, 62–64, 71, 131, 147, 148
- performance, 133
- prefix sums, 435, 437
- processes, 147
- processors, 10
- program design, 71, 83
- program performance, 90
- region, 271
- runtime, 63, 129
- sort, 408, 409, 413, 423
- sorting algorithm, 135
- systems, 1, 3, 8, 17, 31, 38, 49, 62, 65, 80, 82, 83, 85, 371, 457
- trapezoidal rule, 104, 154

trapezoidal rule program, 107
vector, 117

parallel

- block, 233–236, 243, 244, 248, 259, 270, 281, 284
- directive, 225–227, 232, 233, 235, 237, 238, 247, 248, 259, 271, 277, 373, 374

parallel for

- block, 243, 244
- directive, 237, 238, 240, 241, 243–245, 247–250, 253, 254, 274, 278, 380, 440

Parallelism, 1, 3, 8, 15, 30–32, 49, 50, 79, 81, 136, 137, 246, 434, 441, 456

Parallelism SIMD, 32, 34, 436

Parallelization, 51, 65, 71, 73, 86, 150, 241, 256, 272, 279, 288, 372, 375, 408, 439, 441

CUDA, 441

SIMD, 439

Parallelizing

- sample sort, 407
- simple loops, 32
- vector addition, 116

Parent thread, 226, 273

Pascal GPUs, 432

Performance

- application, 49
- cache, 75, 204, 268, 371, 445
- characteristics, 253
- GPUs, 61, 70
- MIMD, 70
- tests, 433

POSIX, 184, 185, 193

- library, 128
- threads, 160, 161

Prefix sums, 150, 151, 404, 406, 409, 430–432, 435, 437

- parallel, 435, 437
- serial, 432, 435–437

Preprocessor

- directives, 185, 222, 235, 277
- instructions, 222

Private

- address, 55
- copy, 243
- memory, 8, 31, 55
- scope, 233, 243, 244, 279
- storage, 218, 285
- threads, 324
- values, 173
- variable, 51, 58, 173, 178, 203, 213, 233–236, 243, 266

Processors

- conventional, 3, 8, 13
- multiple, 1, 6, 78
- Nvidia, 34

- parallel, 10
 SIMD, 292
 vector, 32, 33
- Producer threads, 186, 256
- Program
 counter, 18, 20, 51, 160, 225, 277
 execution, 127, 295, 371
 memory usage, 323
 performance, 147
 SPMD, 94
 vector addition, 309, 315
- Programmable
 logic blocks, 295
 stages, 34
- `pthread_cond_broadcast`, 191, 192
`pthread_cond_signal`, 191, 192
`pthread_cond_wait`, 191, 192
`pthread_create`, 164–167, 187, 210, 214
`pthread_detach`, 168, 214
`pthread_join`, 167, 210
`pthread_mutex_lock`, 180, 181, 184, 192, 211, 215, 456
`pthread_mutex_t`, 178
`pthread_mutex_unlock`, 180, 184, 192, 211
`pthread_t` data structure, 164
`pthread_t` object, 164, 165, 167
`PTHREAD_MUTEX_ERRORCHECK`, 456
`PTHREAD_MUTEX_RECURSIVE`, 456
- Pthreads
 API, 161
 barriers, 193, 216, 417
 condition variable, 381
 functions, 164, 165, 168, 181, 199
 global sum, 177
 header file, 163
 hello, 214
 library, 161, 210, 221, 416
 objects, 188, 211, 216, 416
 standard, 164, 178, 381
 version, 380, 381
- Q**
`qsort` library function, 349
- R**
 Race condition, 53, 74, 81, 174, 190, 210, 211, 230, 278, 316, 318, 327, 332, 334, 345, 372–374, 376, 378, 381
- Registers
 CPU, 78
 vector, 32
- Remote memory, 57, 58
- Remote memory access, 57
- Ring pass, 141, 385, 390, 440, 447
- S**
`sem_post`, 211
`sem_wait`, 185, 189, 190, 211, 451, 456
 Semaphores, 54, 178, 182, 184–187, 189, 190, 207, 211, 451, 455, 456
- Serial
 algorithm, 150, 228, 288, 371, 418, 432, 439, 440
 bitonic sort, 338, 340, 346–348, 357
 bitonic sort pseudocode, 357
 calculation, 435
 code, 4, 82, 128, 129, 147, 172, 211, 241, 280, 316, 346, 347, 377, 434, 435, 439
`for` loop, 304, 315, 318, 319
 hardware conventional, 17, 78
 prefix sums, 432, 435–437
 quicksort, 143, 449
 sample sort, 399, 400, 407, 431
 solver, 379, 389, 394, 444
 sort, 409, 423
 sorting algorithm, 135, 138, 143, 287, 433, 441
 systems, 17, 78
 trapezoidal rule, 103, 154, 229, 237, 319, 320
 vector addition, 304, 305
 vector sum, 304
- Serialization, 54
- Single
 core, 8, 20, 34, 49, 61, 65, 77, 80, 130, 167, 320, 338, 433, 438
 instruction stream, 31, 80, 292
 processor, 1, 46, 65, 79, 86, 138, 143
 processor performance, 3
 processor systems, 1, 87
 thread, 82, 172, 178, 192, 208, 226, 228, 229, 233, 259, 271, 274, 342, 409, 412, 413, 436, 347, 352, 358, 397, 408, 430, 437, 441
 block bitonic sort, 437
`single` directive, 271, 272, 281, 289, 373, 374, 445
- Single instruction multiple data (SIMD), 350
 cores, 9, 34, 390
 parallelism, 32, 34, 436
 parallelization, 439
 processors, 292
 systems, 9, 10, 17, 31, 32, 34, 35, 80, 85
 threads, 433, 434
- Single instruction multiple thread (SIMT), 293
- Single program multiple data (SPMD), 50, 81, 94, 145
 programs, 50, 81, 94
- Snooping cache coherence, 46, 47
- Solaris threads, 161
- Sorted
 keys, 135
 linked list, 193

list, 135, 138, 143, 338, 399, 418, 425, 449
 sublists, 347, 424, 435, 448, 449
 Source code, 31, 68, 79, 83, 127, 129, 163, 222,
 224, 260, 262, 293, 296
 Speedup, 61–63, 65, 66, 70, 82, 133, 147, 181, 250,
 255, 398, 434
 Static mutex initialization, 178
`stderr`, 60, 61, 76, 82, 105, 145
`stdin`, 7, 60, 61, 76, 82, 107, 145, 146, 160, 170,
 225, 289
`stdout`, 56, 60, 61, 76, 82, 95, 105, 106, 145, 160,
 224, 225, 227, 289
 Streaming multiprocessor (SM), 292, 299, 350
 Streaming processor (SP), 293, 299, 350
 Strided memory access, 33
`strtok`, 55, 207, 209, 210, 274–276
`strtok_r`, 209, 210, 219, 276, 286
 Subsidiary threads, 20, 168
 Superlinear speedup, 87, 154
 Superscalar processors, 31
 Suspended thread, 34, 192, 211

T

`taskwait` directive, 272, 273, 281

Threads
 affinity, 456
 behavior, 222
 block, 299, 301, 305, 321, 322, 324, 326,
 328–335, 337, 347, 348, 393, 395, 397,
 407, 408, 430, 432, 434–437, 440
 block size, 325, 330
 blocked, 180, 190, 191, 261, 278
 creation, 169, 187
 divergence, 352, 353, 395, 434
 function, 165, 170–173, 177, 178, 207, 210
 GPUs, 61, 77
 MIMD, 434
 OpenMP, 226, 390
 overhead, 178
 parallel, 280
 placement, 167, 456
 pool, 169, 187
 private, 324
 procedure, 165
 safe, 55, 82
 safety, 54, 55, 82
 SIMD, 434
 slowdown, 31

specifications, 161
 stalls, 79
 startup, 168
 switching, 30
 synchronization, 54, 222, 316, 441
 Tokenizer program, 210, 276
 Translation programs, 3, 13
 Trapezoidal rule, 100–105, 228, 229, 234, 314, 315,
 317, 319, 320, 327–330, 336, 337
 CUDA, 314, 319, 320, 331
 parallel, 104, 154
 program, 103, 105, 108, 110, 113, 125, 145,
 148, 153, 154, 233, 253, 283, 328, 360
 program OpenMP, 231
 program parallel, 107
 serial, 103, 154, 229, 237, 319
 Trapezoids, 102, 125, 145, 229, 232, 318, 356

U

Unified memory, 307, 309–313, 352
 Unified memory locations, 352
 Unlock function, 53, 202, 261
 Unlocked mutex, 185
 Unlocking, 181, 201
 Unnamed `critical` directives, 263, 283

V

Vector
 addition, 31, 88, 116, 117, 119, 302, 312
 program, 309, 315
 serial implementation, 117
 instructions, 33
 parallel, 117
 processors, 32, 33
 registers, 32
 Virtual memory, 21, 25, 27, 79, 205, 268
 von Neumann architecture, 17–20

W

Warp
 shuffle, 322, 324–329, 331, 333, 336, 337, 353,
 354
 shuffle version, 336
 sums, 331–336
`warpSize`, 325, 328, 329
 Windows threads, 161
 Worker thread, 51, 187