Parallel Computing: Concepts and keywords

Jesper Larsson Träff

January 8, 2018

This checklist lists concepts and keywords from the bachelor lecture (VU) on Parallel Computing (184.710) roughly in order of appearance. The list can be used to check your understanding of the material; all keywords can come up during the examination.

Models, Performance

"Free lunch"
Single-core performance
SPEC benchmarks
Deep pipelining
Superscalar execution
Data parallelism
Out-of-order execution
Branch prediction
Cache
Instruction set
Parallel architecture
Architecture model
Programming model
Performance portability
Parallel computing (definition)
Distributed computing (definition)
Concurrent computing (definition)
Coordination
Synchronization
Scheduling
Mapping
Load balancing
Communication
Explicit parallelism
Automatic parallelization
Matrix-vector multiplication
Matrix-matrix multiplication
LU factorization
Stencil computation
5-point stencil
Data distribution
Merging or ordered sequences
Prefix-sums problem
Sorting (mergesort, Quicksort, countingsort, bucketsort)
Architecture model
Execution/cost model
RAM (Random Access Machine)
"von Neumann bottleneck"
Bus
Vector computer
Shared-memory model
PRAM (Parallel Random Access Machine)
EREW (Exclusive Read Exclusive Write) PRAM
CREW (Concurrent Read Exclusive Write) PRAM
CRCW (Concurrent Read Concurrent Write) PRAM
CRCW COMMON/ARBITRARY/PRIORITY conflict resoltion
PRAM fast maximum
UMA (Uniform Memory Access)
NUMA (Non-Uniform Memory Access)
Distributed memory model
Communication network
Atomic operation
Cellular automaton
Flynn's taxonomy
SISD (Single Instruction, Single Data)
MISD (Multiple Instruction, Single Data)
SIMD (Single Instruction, Multiple Data)
MIMD (Multiple Instruction, Multiple Data)
Programming model
SPMD (Single Program, Multiple Data)
PGAS (Partitioned Global Address Space)

OpenMP
MPI (Message Passing Interface)
Cilk
Speedup, $S_p(n)$
$T_{\text{seq}}(n)$
$T_{\text{par}}(p, n)$
Work
Measured time
Worst case complexity
Problem complexity
Best possible/best known algorithm
Absolute speedup
Relative speedup
$T\infty(n)$
Parallelism
Perfect speedup
"Embarrassingly parallel"
"Pleasantly parallel"
Load imbalance
Simulation (argument)
SimGrid
"Modest potential"
Cost
Cost optimality
Work optimality
Overhead
Idle time
Granularity
"Fine grained" parallel computation
"Coarse grained" parallel computation
Sequential fraction
Parallel Fraction
Amdahl's law
Sequential bottleneck
Scaled speedup
Scalability
Strong scaling
Weak scaling
Parallel efficiency
Iso-efficiency function

Parallelization patterns

Parallelization pattern/paradigm
Loop parallelism
"Parallel loop"
Independent iterations
Forwards dependency
Backwards dependency
Nested parallelism
Barrier synchronization
Map
Scatter
Gather
Reduce
Map-Reduce
"Seven Dwarwes" Berkeley patterns
Stencil pattern
Data distribution
Block data distribution
Cyclic data distribution
Block-cyclic data distribution
Domain decomposition
Task parallelism
Functional parallelism
Pipeline, linear
Pipeline, complex
Pipeline performance
Pipeline balancing
Master-worker pattern
Distributed work pool
Work-dealing
Work-stealing
Termination detection
Acyclic task graph (Directed Acyclic Graph, DAG)
Immediate dependency
Dependent tasks
Independent tasks
Schedule, optimal, Makespan
Scheduling problem
Fork-join parallelism
Input node(s), output node(s)
Topological order
Heaviest path
Work law
Depth law
Work-time presentation
Merging ordered sequences in parallel
Stable merge
Lexicographic ordering
Rank, ranking
Ranking, binary search
Load imbalance, bad segments
Co-ranks, co-ranking
Oblivious merging
Bitonic sequence

Bitonic split
Bitonic merge
Comparator network
Sorting networks, depth, size
"Cole's parallel merge sort"
High-Performance Computing (HPC)
Thinking Machines CM2, CM5
Intel iPSC
MasPar
5th Generation project, Star Wars
"Grand Challenge Problems"
"Lost decade"
SiCortex, Kautz network
Moore's "law" (popular version)
Moore's "law" (what Moore observed)
Top500 list, `www.top500.org`
High-Performance LINPACK (HPLINPACK)
LU-factorization
FLOPS (Tera-, Peta-, Exa-, Zetta-, ...)
K Computer, Road Runner, Earth Simulator
Vienna Scientific Cluster (VSC)
Empirical "law": observation, forecast, prediction, prophecy, dictate
Hardware-oriented efficiency
Peak performance
Roofline model
HPCC Benchmarks (DGEMM, STREAM, PTRANS, FFT, random access)
Green 500 list
Graph 500 list
Superlinear speedup
Observation, experiment, hypothesis
Experimental setup, experiment design
Reproducibility
Statistical testing
Memory hierarchy
Superlinear speedup by memory hierarchy effects
Superlinear speedup by non-obvious algorithmic differences
Searching, randomization
Guided tree search
Reduction problem
Prefix sums problem (inclusive, exclusive)
Collective operation pattern
Reduction-to-one, reduction-to-all, reduction-with-scatter
Scan (inclusive, exclusive)
Array compaction
Load balancing
Quicksort partitioning
Binomial tree, binomial coefficient
Prefix sums algorithms (recursive, iterative)
Hillis-Steele algorithm
$2n - 2 \leq s + t$ trade-off for prefix sums
Blocking technique
List ranking problem
Parallel breadth first search (BFS)
Depth first search (DFS)

pthreads, OpenMP

Memory consistency
Program order
Interleaving (of instructions)
Cache
TLB (Translation Lookaside Buffer)
Write buffer
Cache coherency problem
Memory consistency modeling problem
Cache line
Cache block
Temporal locality
Spatial locality
Cache hit
Cache miss
Write allocate
Write-through cache
Write-back cache
Directly mapped cache
Fully associative cache
Set-associative cache
Cold (compulsory) cache miss
Capacity miss
Conflict miss
Eviction policy
LRU (Least Recently Used)
LFU (Least Frequently Used)
Sequential consistency
Cache coherency definition
Cache coherence protocols
Update based, Invalidation based
Snooping (bus based)
Directory based
False sharing
Relaxed consistency models
Memory fence/barrier

Prefetching
Multithreading
SIMD extensions (MMX, SSE, AVX)
Thread model
POSIX (Portable Operating Systems Interface for uniX)
`pthreads`
Fork-join parallelism
Peer threads
Spawn
Join
Race condition
Semaphores, locks, monitors, atomic instructions
Critical section
Condition variable
Wait, Signal, Broadcast
Deadlock freedom
Starvation freedom
Bounded waiting
Fairness
Thread safety
State, static variables
Stratified locks
Backoff
Readers-writers lock
Spurious wakeup
Spin lock (pragmatics)
Load imbalance, prime numbers
Shared counter
Fetch-and-increment
Progress guarantee
Wait-free algorithms
Lock-free algorithms
Atomic instructions
Test-and-set
Compare-and-swap
Work-stealing: randomized, deterministic
C11, C++11 threads
OpenMP
Work-sharing (constructs)
Loop, sections, single, master construct
Implicit barrier
OpenMP clauses
Canonical form loop
Parallel loop schedule (static, dynamic, guided)
Independence conditions
Bernstein's conditions
True (flow) dependency
Anti-dependency
Output dependency
Loop carried dependency
Reduction
Ordered iteration

Cilk

Spawn node, spawn edge
Sync edge
Fully strict computation
Finish edge
Strict computation
Single control variable canonical loop form
Cut-off unit
Mergesort, Quicksort
Divide-and-conquer merge
Depth, work analysis

MPI

Communication driven model
Data distribution driven model
Network topology
Communication capabilities
One-ported
k-ported
Uni-directional
Bi-directional
Telephone model
Send-receive bi-directional model
Routing
Switching
Latency-bandwidth model
Network contention
Direct network
Indirect network
Diameter
Degree
Bisection width
Linear array, ring topology
Tree topology
Fat-tree topology
Torus, mesh topology
Hypercube topology
Kautz network
Fully connected network

Broadcast lower bound, fully connected network
Diameter bound
Deterministic, oblivious routing
Adaptive routing
Minimal routing algorithm
Transmission cost model
Pipelining
Hybrid architectures, heterogeneous networks
Message-passing model
Shared nothing
CSP (Communicating Sequential Processes)
MPI communication models
Point-to-point communication
One-sided communication
Collective communication
MPI-IO
Dynamic process management
MPI Standard
MPI Forum
Error handling, fault tolerance
MPI error codes
MPI bindings
Communication domain, communicator
Process group
MPI object
Encapsulation, library building
Communicator splitting
Point-to-point semantics
Blocking communication
Non-blocking (immediate) communication
Non-local completion
Non-overtaking, ordered communication
Message envelope
Communication context
Message non-determinism, wildcards
Unsafe communication
Safe programming
Send-receive communication
Enforcing completion
Testing completion
Communication progress, progress engine
MPI datatypes
Derived (user-defined) datatypes
Contiguous, vector, index, struct types
Origin, target process
Communication window
Communication epoch
Exposure epoch
Access epoch
Active synchronization
Fence synchronization
Generalized, pairwise synchronization
Passive synchronization
Distributed memory binary search
Distributed memory merging
Cartesian communicator
Collective communication operation
Collective operation complexity
MPI collectives
Symmetric vs. non-symmetric (rooted) collectives
Regular vs. irregular collectives
Distributed matrix-vector multiplication
Distributed memory Quicksort
Distributed memory counting (bucket) sort