Computer Science Performance Sheet / Program Optimization TRIZ

Algorithms
Greedy
Search
Graphs
Depth first, breadth first
Shortest path

Data structures
Hash table
Sort
Array, linked list
Trees
Heaps
Stacks (FILO), queues (FIFO)

Time-Space
Lookup tables vs. recalculation
Compressed vs. uncompressed data
Compile time vs runtime
Local inefficiency vs global inefficiency
Abstraction/simplicity vs performance/bugs

Parallelism
Scheduling
Speculation, branch prediction, prefetching
Out-of-order execution, Instruction pipelines
Instruction-level, thread-level, data-level
Multiple buffering
Reductions like MatMul, sum, avg, map-reduce

Merging
Bin filling
Loop Fusion, Kernel Fusion

Division
Divide and conquer, dynamic programming
Multi-level hierarchy
Intermediary, middleware, proxy

Extra work
Sort, index, recomputation
Cache, memoize, warming
Precompute, look-up table
Compiler optimization
Exhaustive brute-force superoptimizing
JIT compile, use runtime info
Excess capacity
Redundancy
Tracing

Less work
Cache, offload, lazy
Strength reduction,
Inline, loop unroll
Remove duplicated/unnecessary work

Fidelity
Resolution, precision, size, lossy
Approximate, relaxation

Equivalence
Combine, replace, reorder
Algebraic laws
Constant folding
Strength reduction
Operate on compressed data

Global
Computational graph
Identify bottleneck
Suboptimal high-level algorithm

Local
Peephole optimization
Temporal/spatial locality
Greedy algorithm

Structure/Patterns/Dynamics/Change
Zero compression, base delta
Deduplication, shared assets
Workload behaviour
Modelling and prediction
Randomness
Virtuous cycle
Trigger compute on data change
Phase transitions
Summarize
Embedded structures, self similarity, repetition

Symmetry/Balance
Padding, fit to physical container
Data alignment, coalesced memory

Asymmetry
Load imbalance
Specialization
Hybrid or adaptive algorithm
Offload critical section to accelerator
80/20 Pareto Principal
Thread divergence

Periodic/Delayed action
Garbage collection
Batch processing
Frequency
Backpressure

Continuous
Stream processing
Real-time, deadlines
Fill idle time

Inversion
Push/pull
Trade-offs
Traverse from leaves to root

Hardware
Compute units, stall reasons
Scratch pad, explicit cache control
Hardware intrinsic
Systolic array
Processing in memory
Cache locality
Cache coherency

Constraints
Compute
Capacity
Communication
Overhead
Thermal/Power/Cooling
Algorithmic Efficiency
Parallelization
Workload Patterns
Human Factors
Security
Cost

Vertical scaling (Scale up)
Bigger, easier, expensive

Horizontal scaling (Scale out)
Parallelism, scheduling, distributed

Coordination
Priority, congestion control, quality of service
Synchronization barriers
Master, Distributed, Decentralized

Abstraction
Virtual address space

Another dimension
Transpose or reorder dimensions
Additional communication channel

Overhead
Abstraction, compatibility
Control mechanisms, journaling

Performance Goals
1. Bandwidth/Throughput/Rate
2. Latency/Runtime
3. Cost/Utilization/Power Efficiency
4. Space/Size
5. Correctness/Accuracy
6. Compatibility/Accessibility/Universality

Implementation Level
1. Design level
2. Algorithm and data structure level
3. Source code level
4. Compile level
5. Assembly level
6. Hardware level

Context
1. Implementation level
2. Source code
3. Stack trace
4. Computational graph
5. Scheduler decision making factors
6. Temporal execution of program
9. Workload statistics