Data Oriented Programming

- The basic idea is that two of the most expensive operations for the CPU are typically
  - memory access (if you have a L2 cache miss)
  - allocation (its cost is unpredictable, when you need to go through the os)
- thus thinking about the management of the memory and its layout is crucial
  - no single solution is perfect you have to think about your concrete problem
  - DOD is an approach about how to think about the problem
  - there are some general strategies that depending on your problem might work
  - you have to consider the access patterns and statistical distribution of your problem

Centralize handling

- move memory management into a centralized system (system, environment,...)
  - allocation is expensive
  - managing/changing allocation pattern is easier
- Operate on bunch of elements, not on the single element
  - amortize load cost on several similar operations, ideally working on close by memory
  - global optimizations are easier to do

Minimize memory usage

- pack, use smaller types
- in some cases even compression can help
- but beware of alignement
- Bools and enums in structures
  - try to avoid if/case in the element operation
  - try to move them "outside the loop", and outside the structure, for example separating elements by type
- consider Handles/indexes instead of pointers when referring to elements

Evaluate a SoA (Struct of Arrays) instead of AoS (Arrays of struct)

- Struct of Arrays is generally better if you have several independent loops that use only parts of the structure
  - each of the loop will "cache" only the pieces it uses
- if you use all of the structure in a single loop caching it is better

References

- IT Hare, 'Infographics: Operation Costs in CPU Clock Cycles': ithare.com/infographics-operation-costs-in-cpu-clock-cycles/
- The Brain Dump, 'Handles are the better pointers': floooh.github.io/2018/06/17/handles-vs-pointers.html
- Richard Fabian, 'Data-Oriented Design': dataorienteddesign.com/dodbook/
- Andrew Kelly Practical DOD
- CppCon 2014: Mike Acton "Data-Oriented Design and C++": youtube.com/watch?v=rX0ItVEVjHc