Performance with C++

IEEE-754 Float Numbers

as written in Numeric / Math / Linear Algebra, calculation with denormals or non-numbers slows down performance - even when not signalled.
it might be interesting to abort a calculation, e.g. a matrix/vector multiplication, with first occurence of NaN - or with one of the other conditions ..

Optimization Remarks from Compiler

Ofek Shilon held the talk Optimization Remarks - Helping the Compiler to Generate Better Code at CppCon 2022. Here his PDF slides. Here also slides of a sligthly older talk on the same topic: https://www.slideshare.net/ofekshilon/optview2-c-on-sea-2022

references from his slides:

- llvm-opt-report
  - https://reviews.llvm.org/D25262
  - https://github.com/llvm/llvm-project/tree/main/llvm/tools/llvm-opt-report
  - https://llvm.org/docs/Remarks.html
  - debian package `llvm-14-tools`
  - clang compiler switch `-fsave-optimization-record`
- opt-viewer / optview2
  - https://github.com/llvm/llvm-project/tree/main/llvm/tools/opt-viewer (origin)
  - https://github.com/OfekShilon/optview2 produces much better output
  - `opt-viewer.py –output-dir <htmls folder> –source-dir <repo> <yamls folder>`
- *Clobbered by store*
  - use compiler specific keyword `__restrict` or `HEDLEY_RESTRICT`
    - https://github.com/nemequ/Hedley
- *Clobbered by call* or *load*
  - function attribute `__attribute__((const))` or `HEDLEY_CONST`
    - `const` functions are not using any saved/global state; thus free of side-effects
    - compiler can remove repeated invokations
    - `const` is stronger than `pure`
  - function attribute `__attribute__((pure))` or `HEDLEY_PURE`
    - see https://gcc.gnu.org/onlinedocs/gcc/Common-Function-Attributes.html for `pure` and `const`
    - hedley doc is much easier to understand / discriminate:
      - https://nemequ.github.io/hedley/api-reference.html#HEDLEY_PURE
      - https://nemequ.github.io/hedley/api-reference.html#HEDLEY_CONST
  - function parameter attribute `__attribute__((noescape))` or `HEDLEY_NO_ESCAPE`
    - see https://clang.llvm.org/docs/AttributeReference.html#noescape
- Talk *Compiler-assisted performance analysis* of Adam Nemet
  - https://www.youtube.com/watch?v=qq0q1hfzidg

Sites, Blogs and Books

- Denis Bakhalov - EasyPerf!
  - https://easyperf.net/
- Johnny’s Software Lab
  - https://johnnysswlab.com/