PyData/Sparse — Future Plans

Optimality, with simplicity

Hameer Abbasi, 26.09.2022

Optimality 🏃

What do we mean by it?

- Optimal big-O runtime and memory performance.
  - No temporaries, where avoidable
  - No iterating over the inputs/outputs, where avoidable
  - No pointer chasing
- NO optimal scheduling performance,
  - It varies from microarchitecture to microarchitecture.

Simplicity ☺️

What do we mean by it?

- You could use NumPy notation to achieve optimality **(!)**
  - One iteration of Bellman-Ford is simply: previous = np.minimum(previous, previous.T + adjacency_matrix).
  - Matmul is simply np.sum(a[…, None, :], np.swapaxes(b, (-2, -1))[…, :, None])
- Yes, it has been done before in the C++/dense world (ref: xtensor).
- And in the sparse world (ref: TACO).

What's the catch? 🪝😰

There's always a catch

- We need *some form* of a codegen at runtime.
  - This can be *libclang*/*libllvm* just linked into the library,
  - Or an actual, y'know, compiler.
  - Most of the heavy lifting has been done here too (ref: cppyy).
    - No wheels
    - No conda packages for Windows

Thanks for the ear 👂

And, pssst, reach out

- GitHub — hameerabbasi/xsparse
- Gitter — pydata/sparse