Evaluating the Nix Evaluator

Why Nix Performance Sometimes… Doesn’t

Connor Baker

2025-03-07

Planet Nix

Topics covered

- Benchmarking setup
- Nix evaluation performance over time
- Suggested areas for improvement

Assumptions 🔖

Nix evaluation performance

Nix evaluation performance

1. Can improve?

Nix evaluation performance

1. Can improve?
    - Historically, yes!

Nix evaluation performance

1. Can improve?
   - Historically, yes!

2. Should improve?

Nix evaluation performance

1. Can improve?
   - Historically, yes!

2. Should improve?
   - It depends!

Benchmarking setup 🕐

Benchmarking is **difficult**.

What do we want to measure?

- Space
- Time

Why do we want to measure it?

- TODO

How will we measure it?

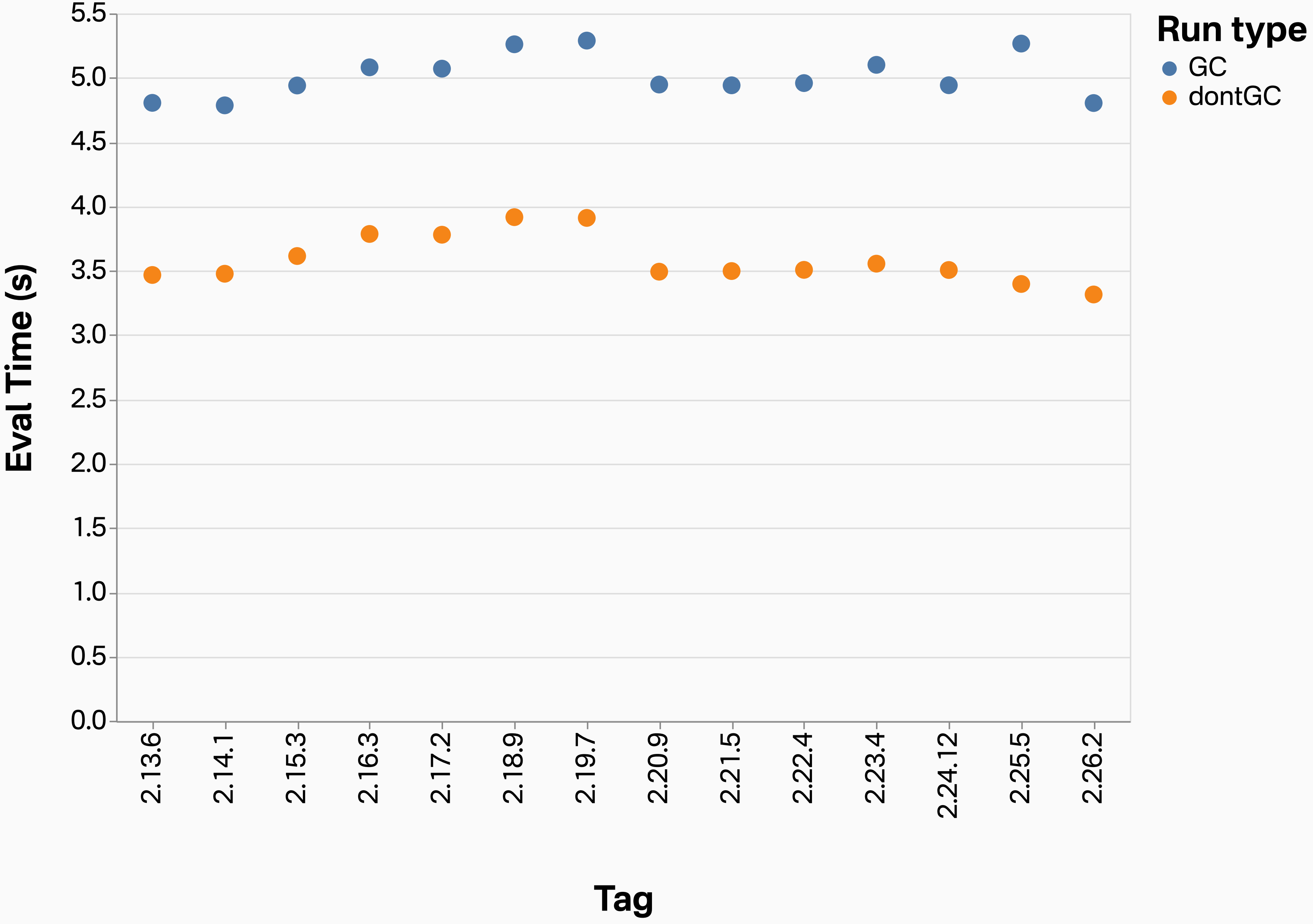
- TODO

Examples 📊

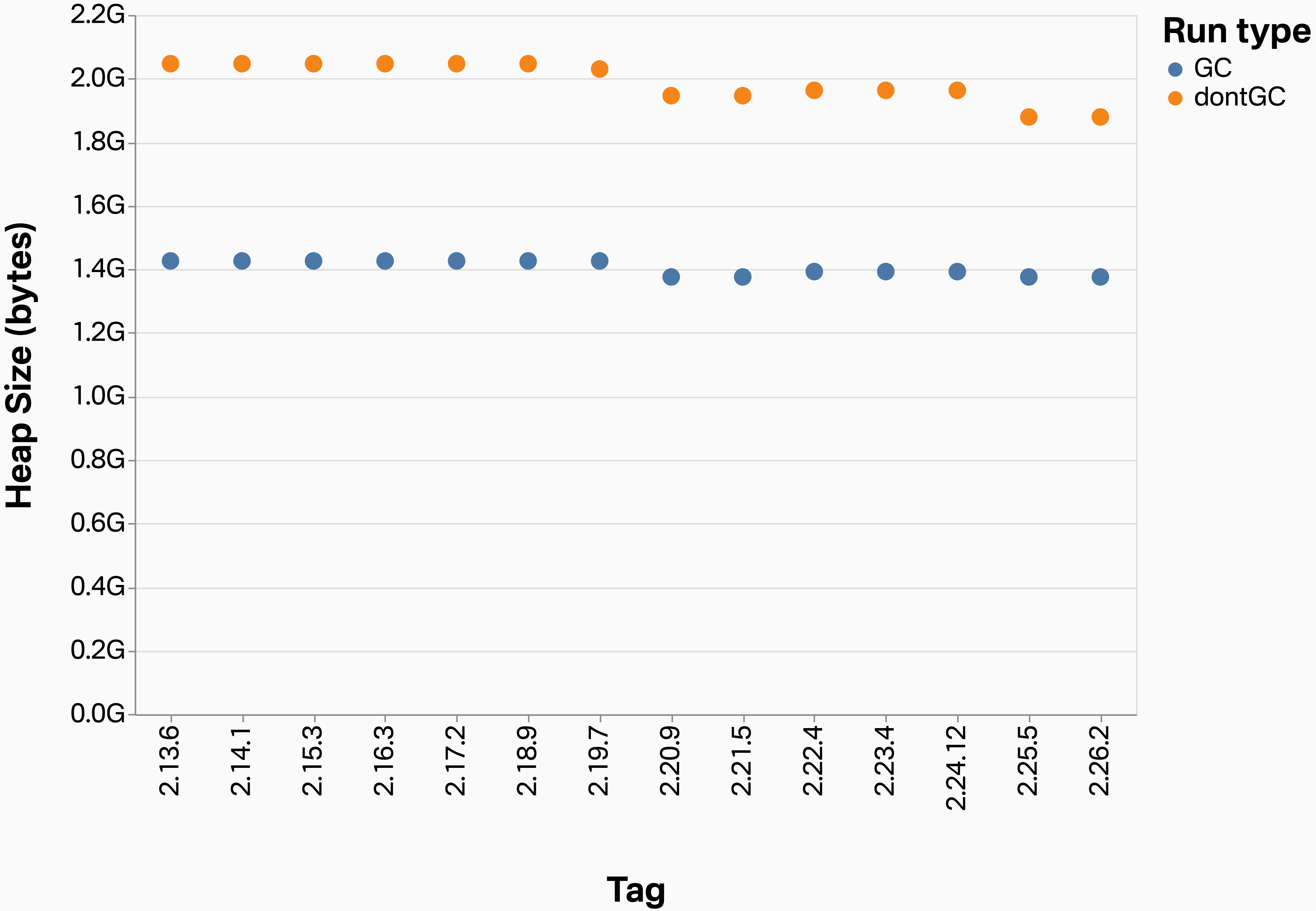
Setup

- Intel i9-13900K (locked to 3 GHz) with 96 GB DDR5
- Four-way ZFS RAID0 with integrity protections disabled
- Each benchmark uses 20 runs
- Median values are plotted

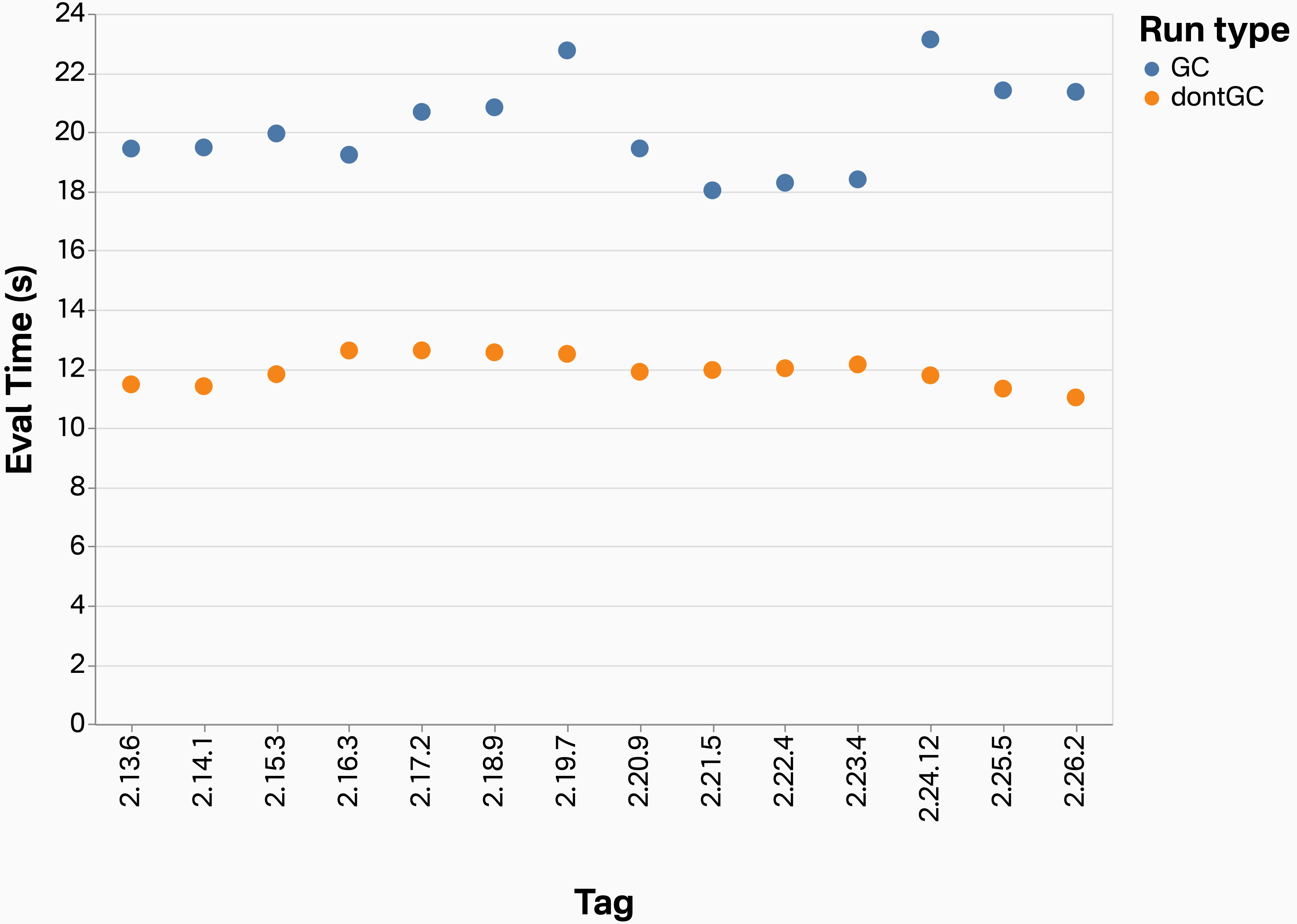
firefox-unwrapped eval time



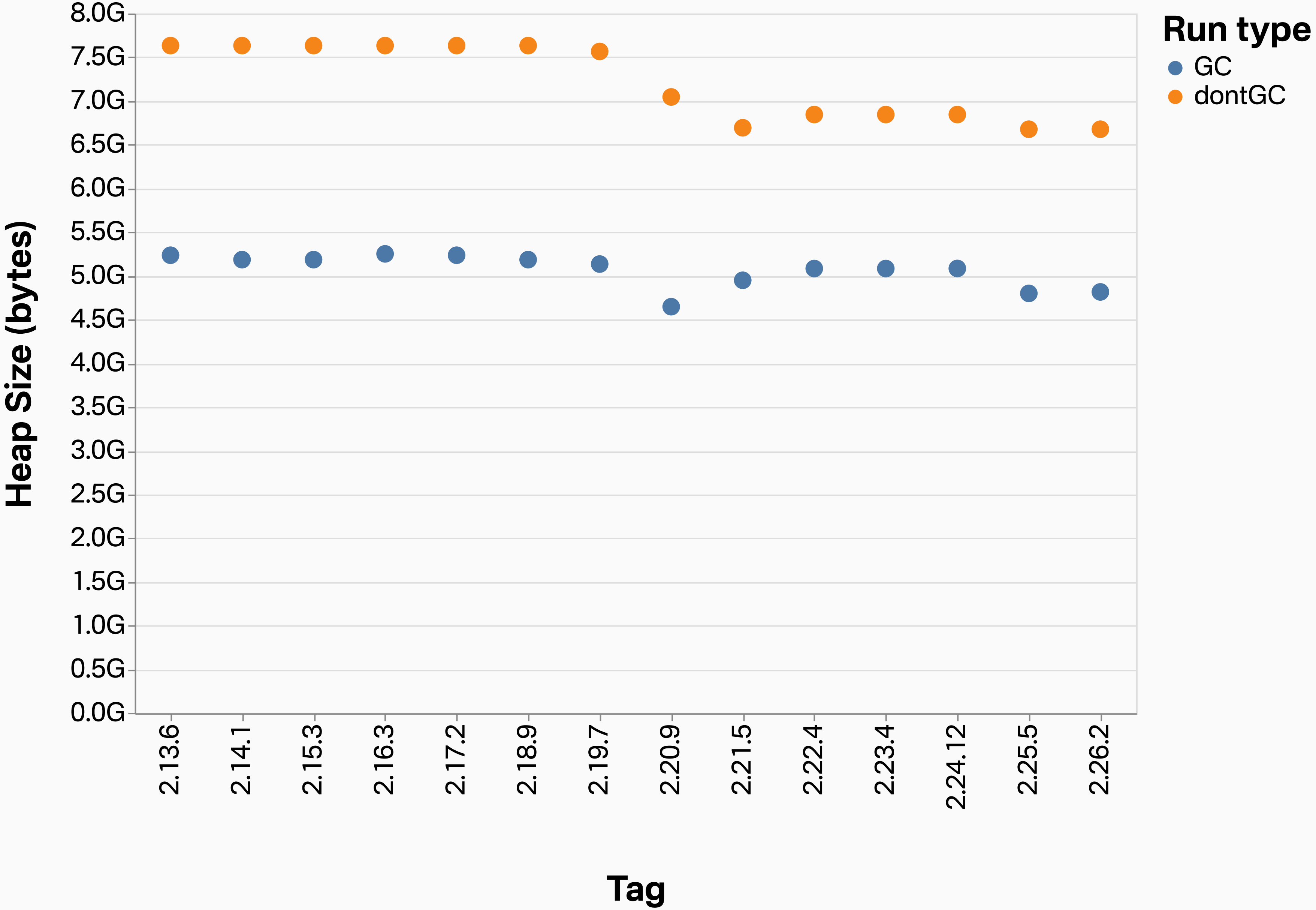
firefox-unwrapped eval space



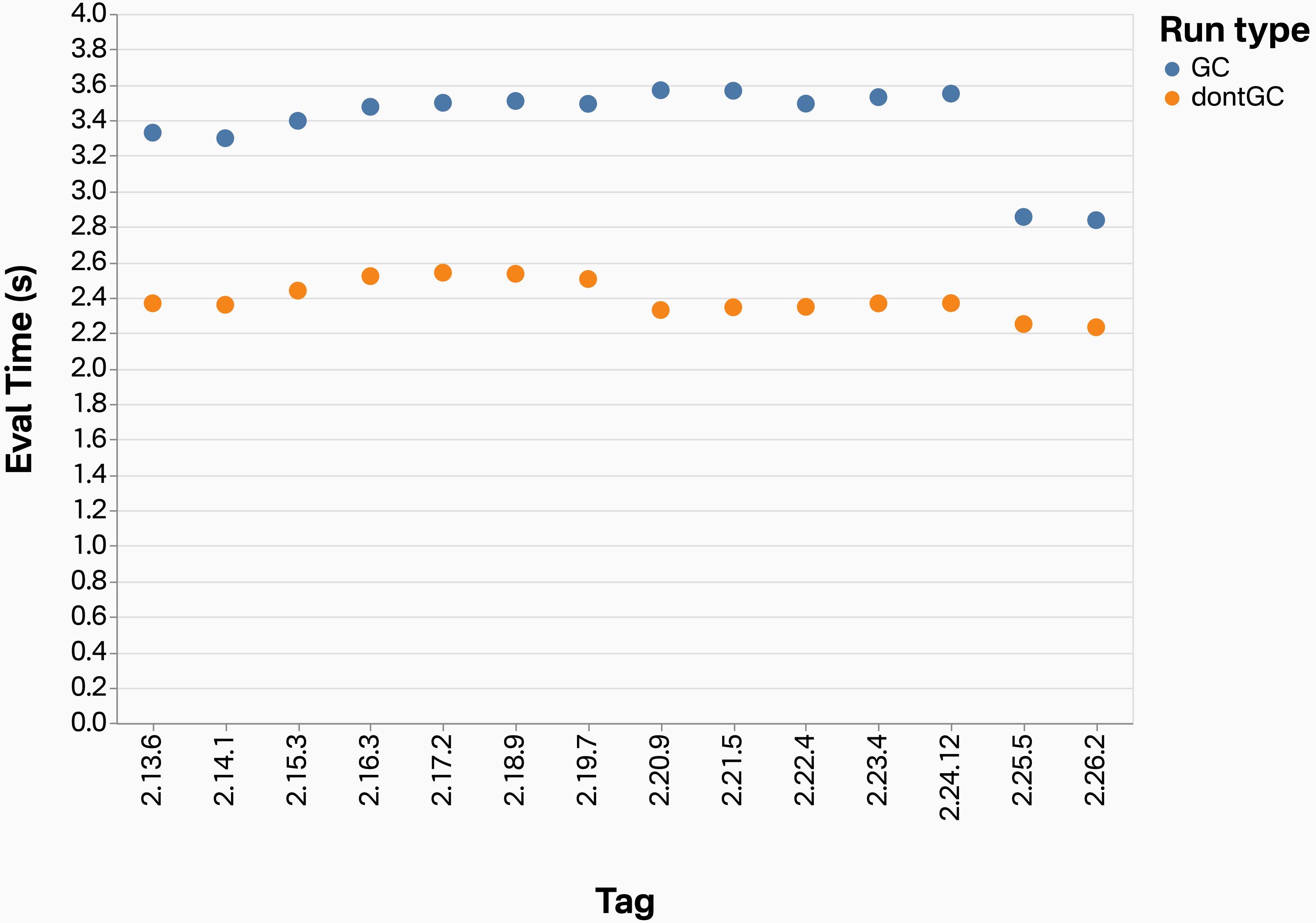
release-attrpaths-superset.names eval time



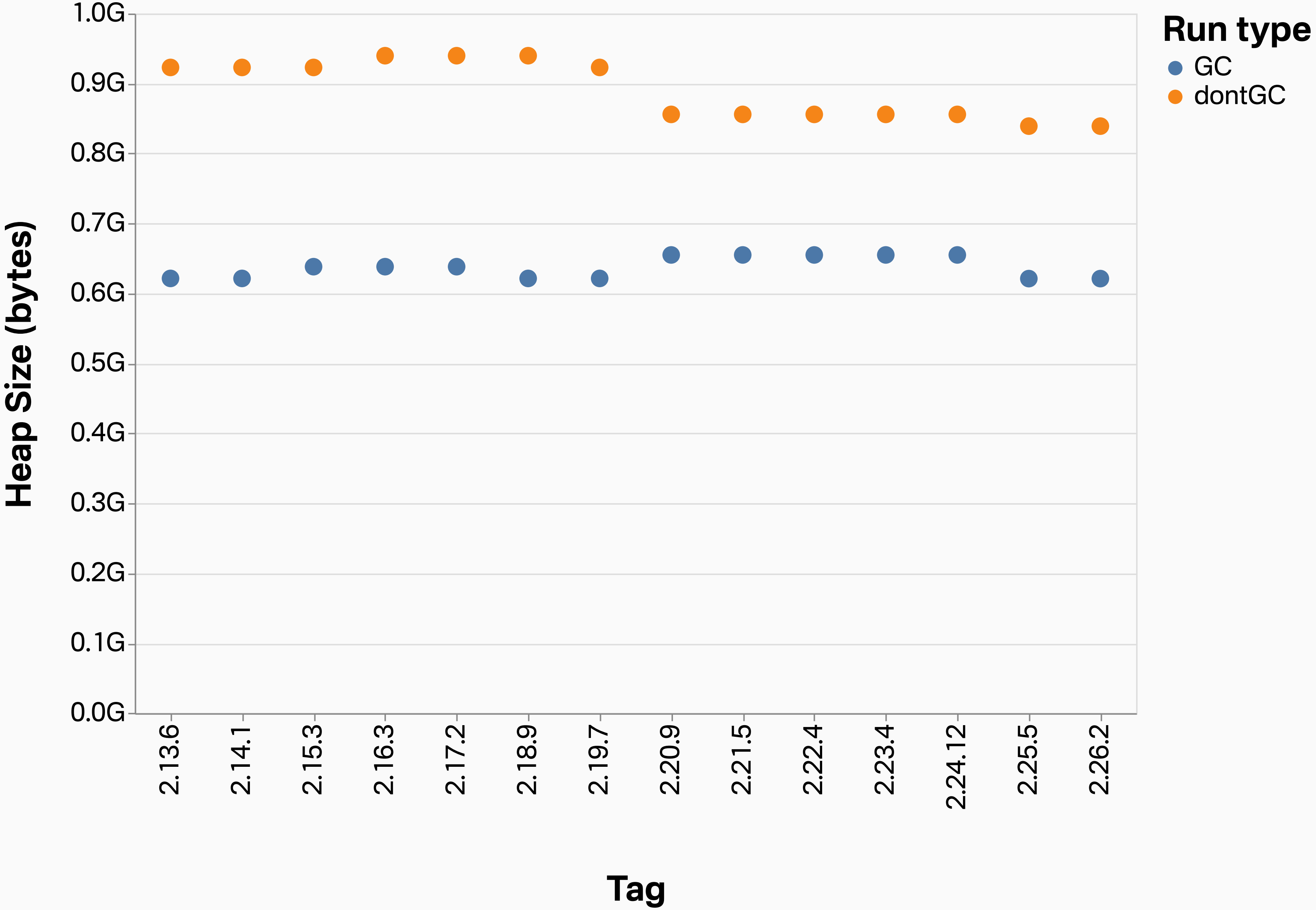
release-attrpaths-superset.names eval space



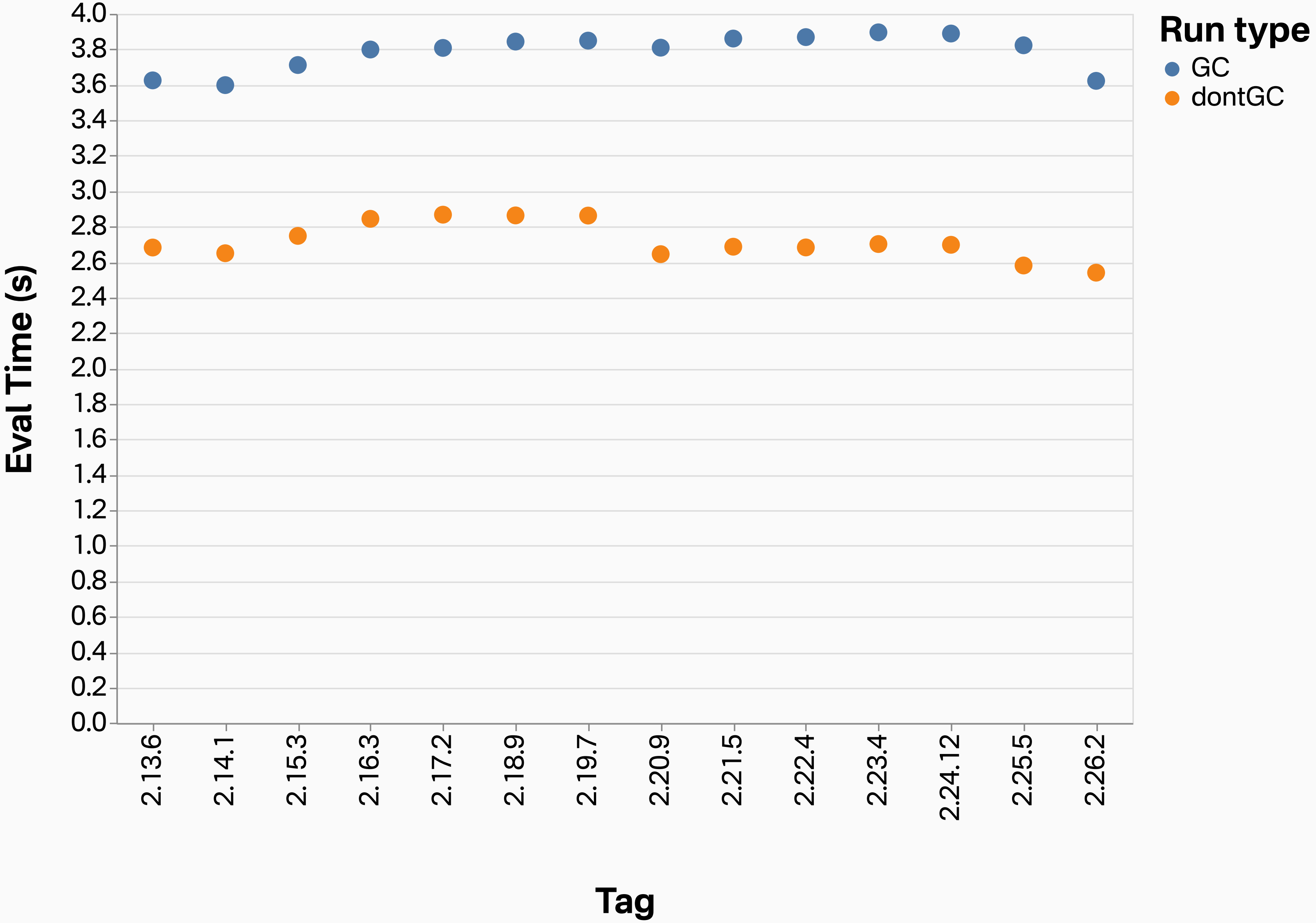
closures.smallContainer.x86_64-linux eval time



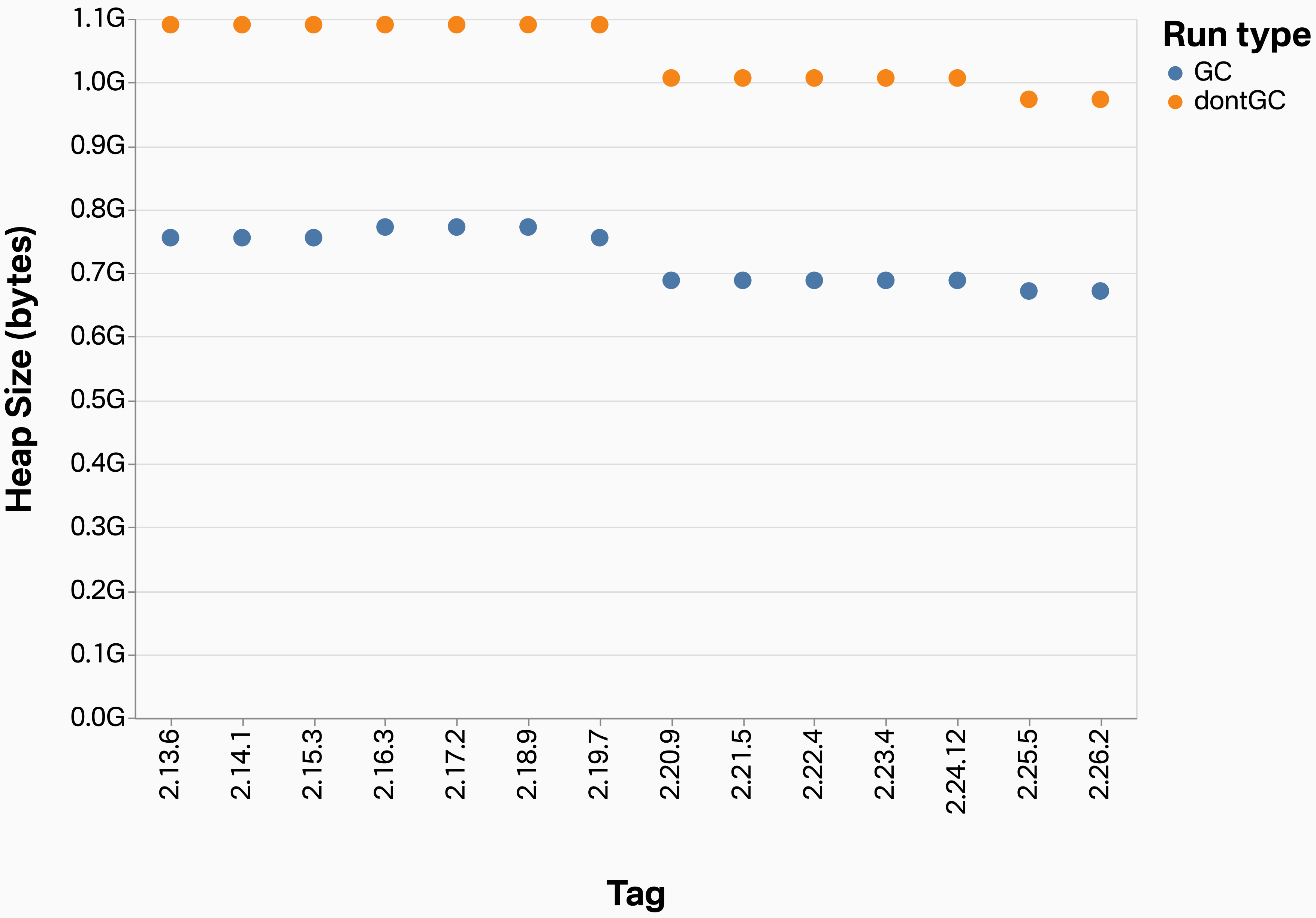
closures.smallContainer.x86_64-linux eval space



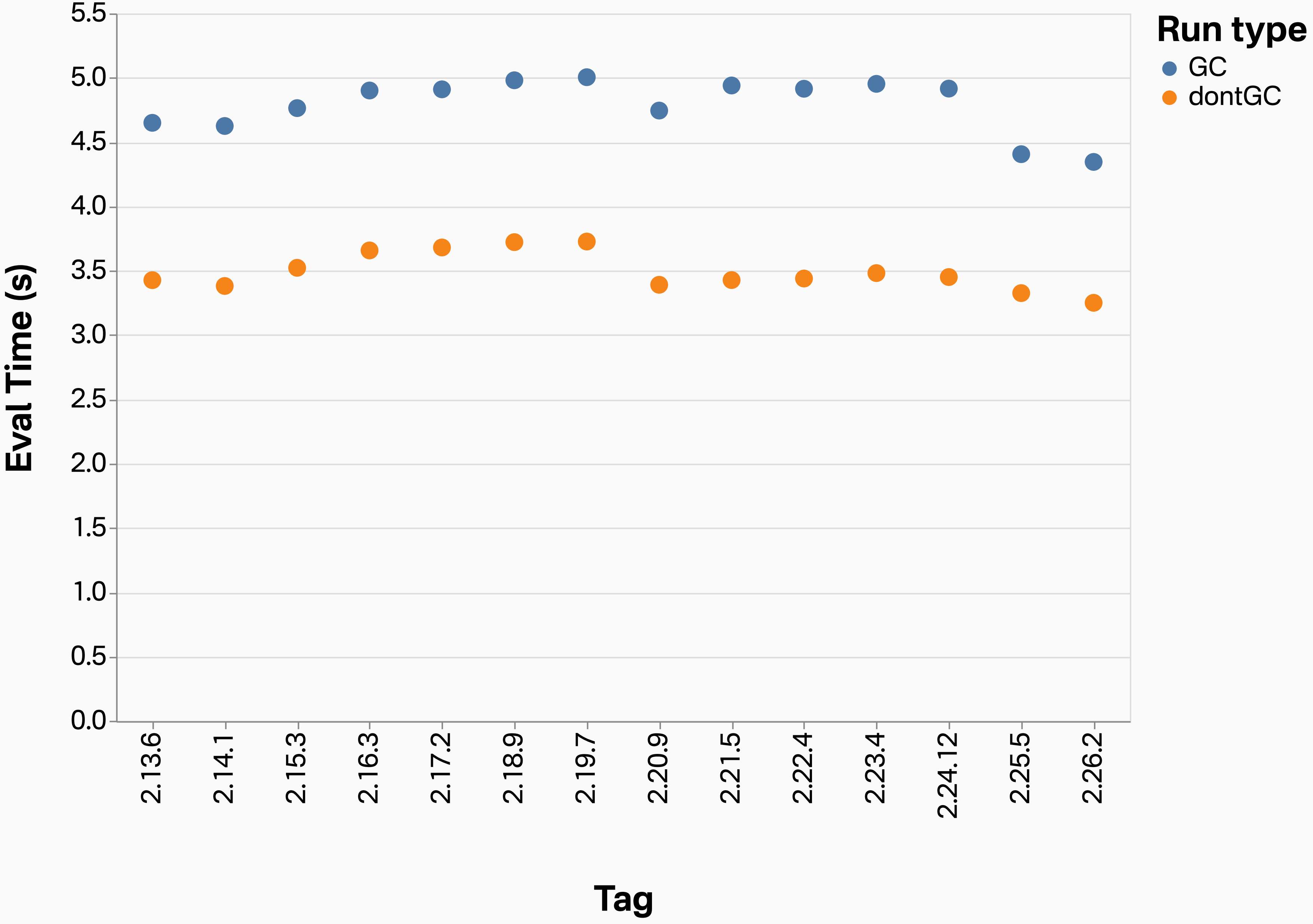
closures.lapp.x86_64-linux eval time



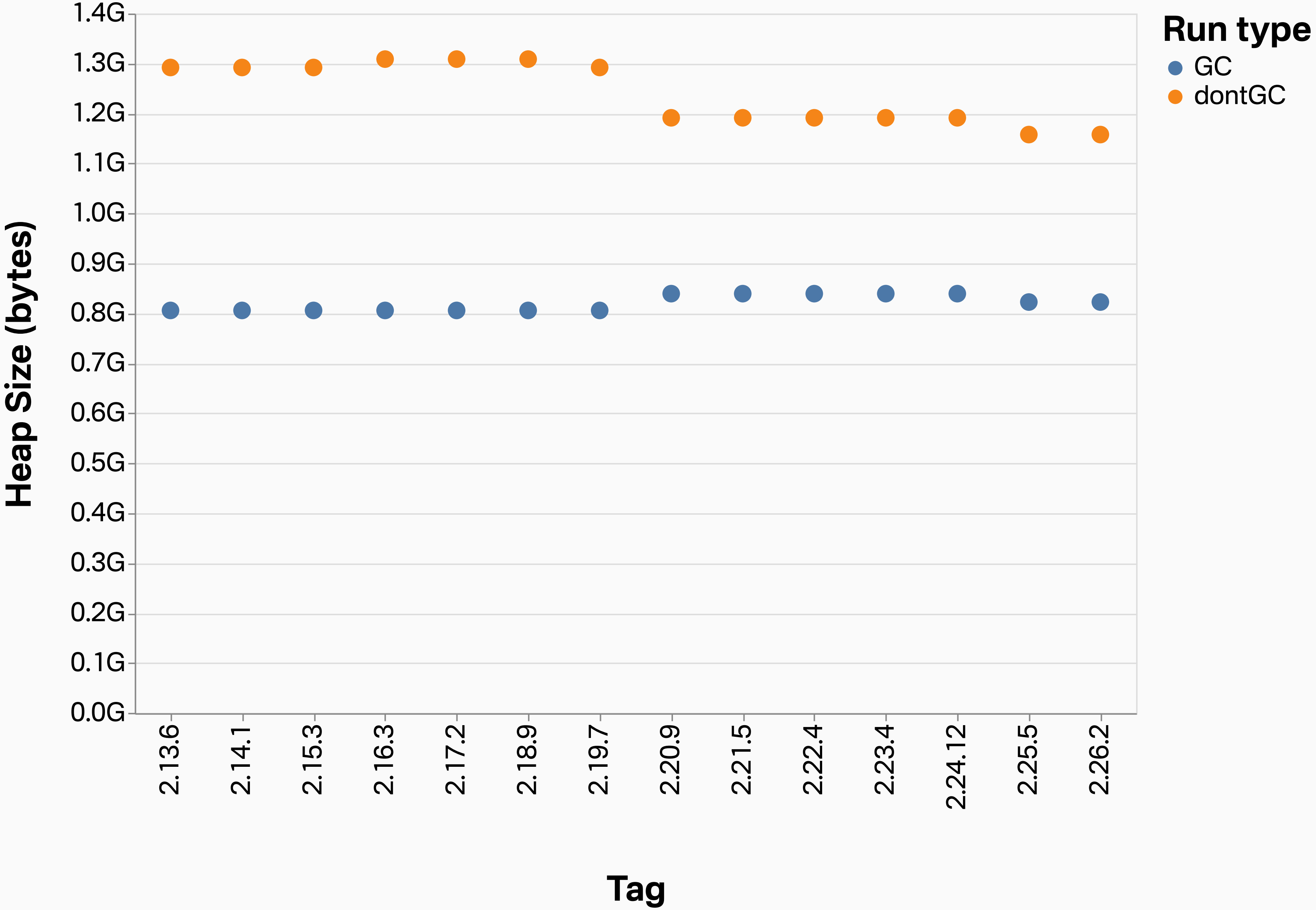
closures.lapp.x86_64-linux eval space



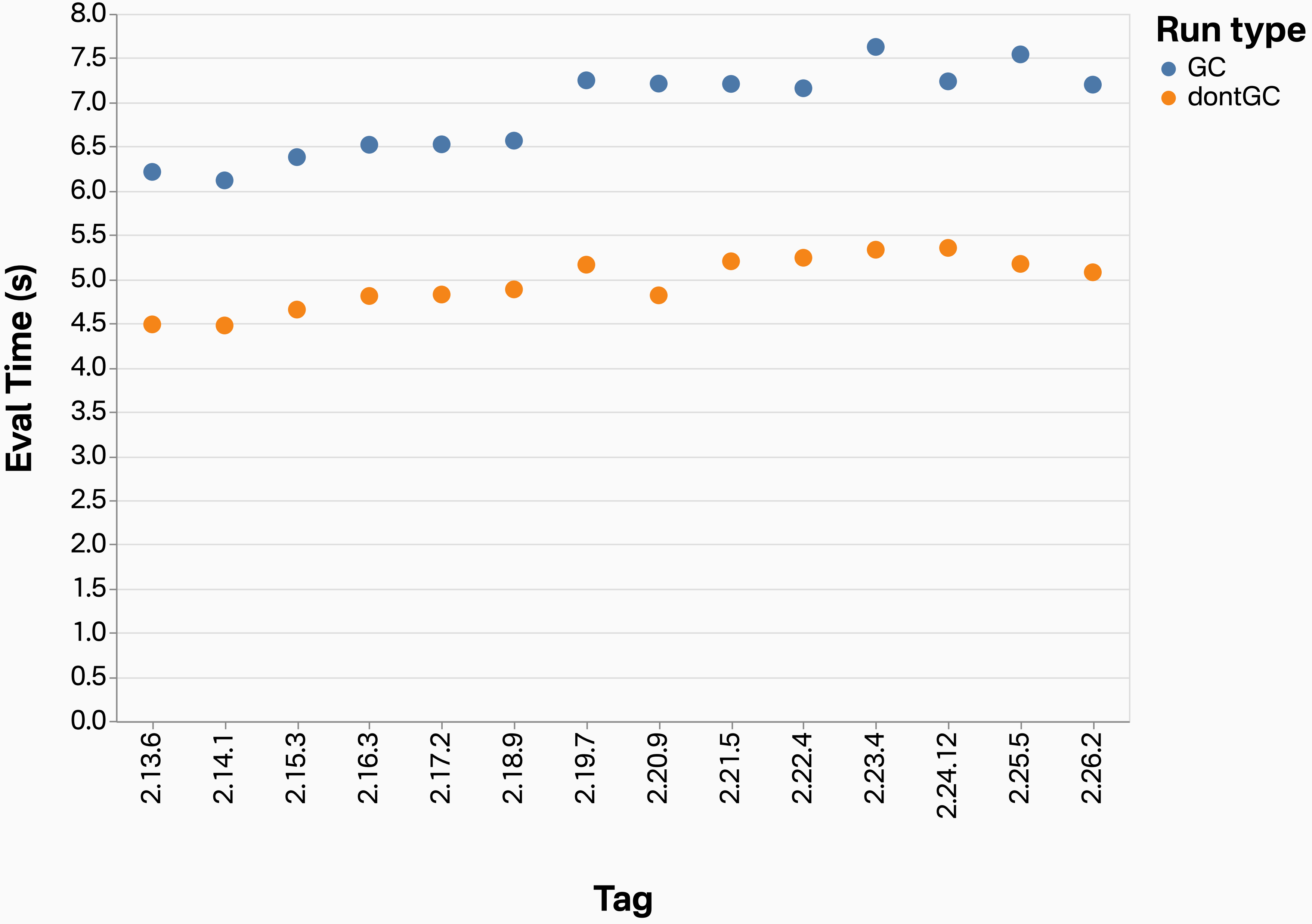
closures.kde.x86_64-linux eval time



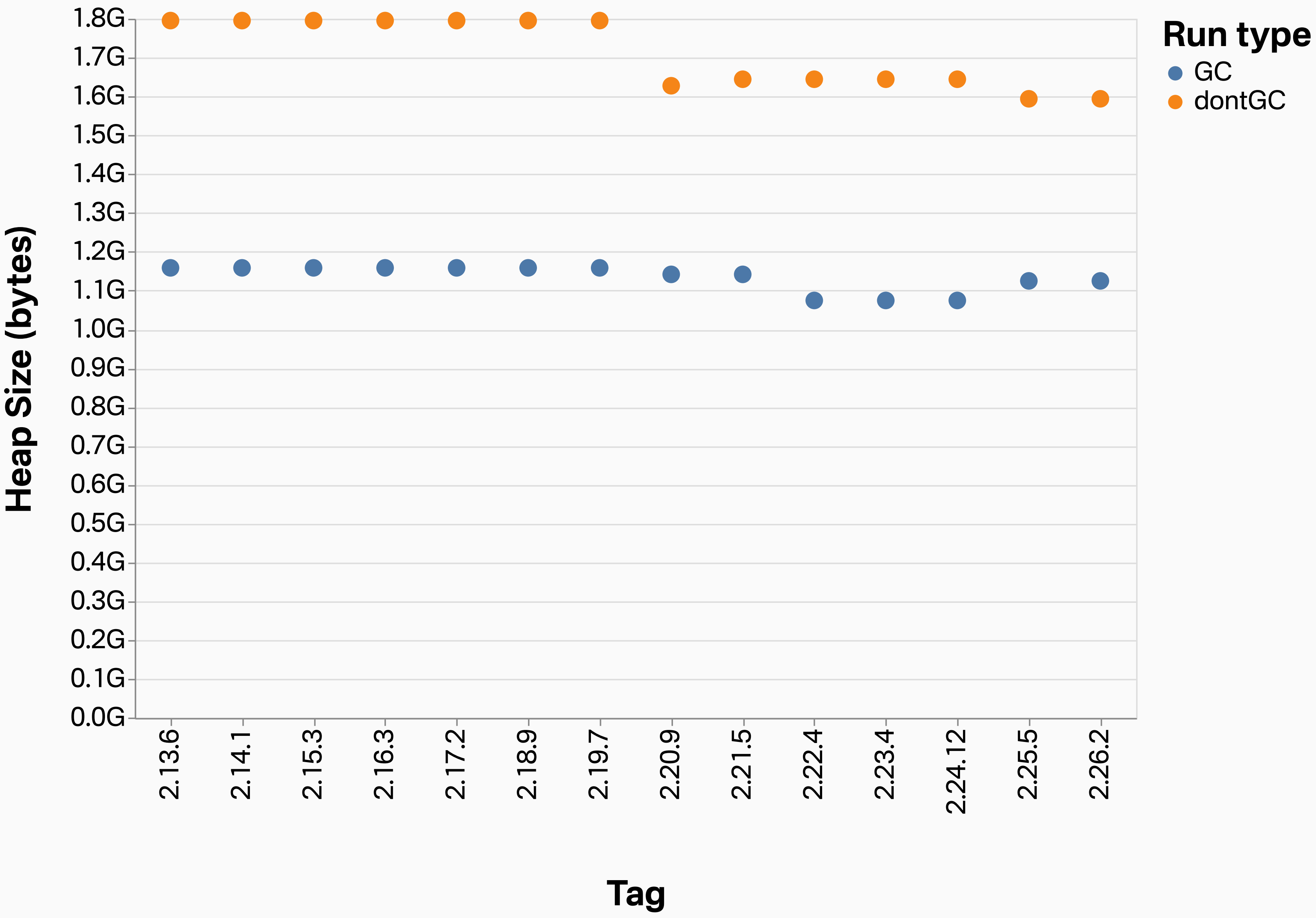
closures.kde.x86_64-linux eval space



iso_gnome.x86_64-linux eval time



iso_gnome.x86_64-linux eval space



Nix evaluation performance trends

- Charts for evaluation performance over time
- Discuss axes on which evaluation can be expensive
  - Evaluator implementation
  - Nix data structures
  - Nix expressions

What’s with all the garbage?

- TODO: benchmarks without GC running and without Boehm entirely
- Transition to looking at the actual implementations

Evaluator structures 🧬

Value

- Padding, etc.

List

- Special-cased for lists of size 0, 1, and 2, which can fit in a Value
- Implemented as a C-style array, so great data locality

Attribute set

- TODO: has it changed? I remember there being two arrays (one for names, one for values), but now it seems to be a vector of tuples.

Improvements 🔮

Suggested improvements should be **orthogonal** to those an **optimizing** or **parallel interpreter** would provide.

Persistent data structures

- TODO
- I mean, functional programming language with immutable values so why not benefit from sharing?
- Describe Immer library

Shrinking structures

- TODO: Link to branch I have with these changes