

Evaluating the Nix Evaluator

Why Nix Performance Sometimes... Doesn't



Connor Baker

2025-03-03

Planet Nix

1. *Assumptions*

1. Assumptions

1. Can improve

1. *Assumptions*

1. Can improve
2. Should improve

1.1 Nix evaluation performance can be improved

- Low-hanging fruit

1.1 Nix evaluation performance can be improved

- Low-hanging fruit
- Some improvements are easy to find

1.1 Nix evaluation performance can be improved

- Low-hanging fruit
- Some improvements are easy to find

1.2 Nix evaluation performance should be improved

- Always tradeoffs to be made

1.2 Nix evaluation performance should be improved

- Always tradeoffs to be made
 - time \leftrightarrow space

1.2 Nix evaluation performance should be improved

- Always tradeoffs to be made
 - time \leftrightarrow space
 - portability \leftrightarrow performance

1.2 Nix evaluation performance should be improved

- Always tradeoffs to be made
 - time \leftrightarrow space
 - portability \leftrightarrow performance
 - maintainability \leftrightarrow implementation complexity

1.2 Nix evaluation performance should be improved

- Always tradeoffs to be made
 - time \leftrightarrow space
 - portability \leftrightarrow performance
 - maintainability \leftrightarrow implementation complexity
- TODO: Discuss how this talk addresses concerns orthogonal to an optimizing interpreter

2. Benchmarking setup

2.1 What do we want to measure?

- TODO

2.2 Why do we want to measure it?

- TODO

2.3 How will we measure it?

- TODO

3. 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

4. Survey of data structures in the evaluator

- TODO

5. Proposed improvements

- TODO