releases / rust / publishing / performance

Guidedog 1.0.0-beta.16

Guidedog contributors

2026-11-27

Abstract

Build the editions of a site at the same time, as many as fit in memory, with the same site and the same reports as before, and proofs checked by Lean.

Diese Version baut die Editionen einer Website gleichzeitig. Die Guidedog-Website mit 23 Editionen und sieben PDF-Büchern entsteht in 13 Sekunden statt in etwa einer Minute und veröffentlicht dieselben Dateien.

`guidedog ssg build` baute bisher jeden Mount in jeder Sprache nacheinander, und die meisten Prozessoren blieben untätig. Jetzt baut ein Kindprozess `guidedog build` jede Edition, so viele gleichzeitig, wie in neun Zehntel des verfügbaren Speichers passen und die Prozessoren erlauben. Eine Edition reserviert den Speicher, den sie bei ihrem letzten Build brauchte, und ein Viertel mehr; `_build/ssg-projects/memory.toml` bewahrt diese Spitzenwerte. Eine Edition, die schwerer ist als der freie Platz, wird wie bisher allein gebaut, sodass ein Build nie mehr Speicher braucht als der sequenzielle Build. Linux, macOS und Windows melden den verfügbaren Speicher und den Spitzenwert jedes Builds.

Die Website, die Reihenfolge der Berichte und der erste gemeldete Fehler sind die des sequenziellen Builds. `--jobs=N` begrenzt die gleichzeitig gebauten Editionen, und `--jobs=1` baut sie nacheinander im Prozess `guidedog`.

Jeder Projekt-Build sperrt jetzt seinen Verwaltungsordner, sodass nie zwei Builds gleichzeitig in eine Ausgabe schreiben. Ein zweiter Build derselben Ausgabe bricht mit `build.busy` ab.

GDS 0020 modelliert die Planung und beweist in Lean, dass die laufenden Editionen im freien Platz bleiben, außer wenn eine allein läuft, dass jeder Lauf endet, dass das Ergebnis das sequenzielle ist, dass zwei Editionen nie dieselbe Datei schreiben und dass ein Katalog, den mehrere Editionen schreiben, vollständig gelesen wird.

GDS 0020 beschreibt das Modell, die Sätze und die Messungen. Die Befehlsreferenz beschreibt `--jobs` und `--peak-memory-file`.