

Optimierung der Codeausführung in einer nutzerzentrischen Datenbankquery mit ORM Relationen

Yann Berton

9. März 2024

Zusammenfassung

Dieses Dokument analysiert die Verbesserung der Ausführungsgeschwindigkeit eines spezifischen Teils eines Projektverwaltungssystems durch die Optimierung des Codes. Der ursprüngliche Code diente dazu, die neuesten Projekte der Benutzer in einer aktiven Gruppe zu ermitteln, was jedoch ineffizient umgesetzt wurde. Durch eine Überarbeitung des Abfrageverfahrens konnte eine signifikante Reduzierung der Ausführungszeit erreicht werden.

1 Einführung

Das Ziel des ursprünglichen Codes war es, innerhalb eines Projektverwaltungssystems die aktuellsten Projekte der Benutzer zu ermitteln. Hierfür wurden zuerst alle Benutzer abgefragt und anschließend in einer Schleife für jeden Benutzer das jeweils neueste Projekt gesucht. Diese Methode erwies sich als zeitintensiv und ineffizient.

2 Vergleich der Code-Versionen

Der ursprüngliche Ansatz nutzte eine Schleife, um durch alle Benutzer zu iterieren und das neueste Projekt jedes Benutzers separat zu ermitteln. Dies führte zu einer hohen Anzahl von Datenbankabfragen und somit zu einer längeren Ausführungszeit.

Im optimierten Code wurde ein Subquery-Ansatz verwendet, um die ID des neuesten Projekts jedes Benutzers zu ermitteln und diese in einer einzigen

Abfrage vorzuholen. Anschließend wurden die Projekte mit den zugehörigen IDs in einem Dictionary gespeichert und den Benutzern zugeordnet, was die Anzahl der Datenbankabfragen erheblich reduzierte.

3 Flussdiagramme

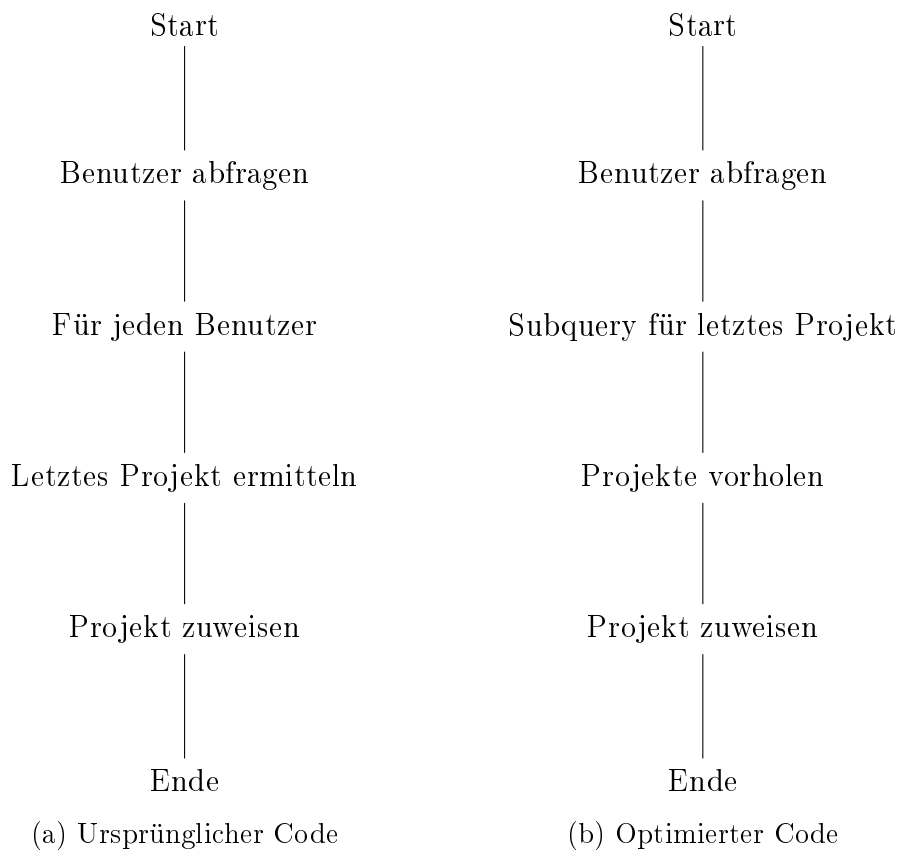


Abbildung 1: Flussdiagramme der Codeausführung

4 Ausführungszeit

Die Ausführungszeit des Codes wurde vor und nach der Optimierung gemessen. Die Ergebnisse sind im folgenden Diagramm dargestellt. Die Geschwindigkeitsverbesserung durch die Optimierung des Codes beträgt etwa 97.64%

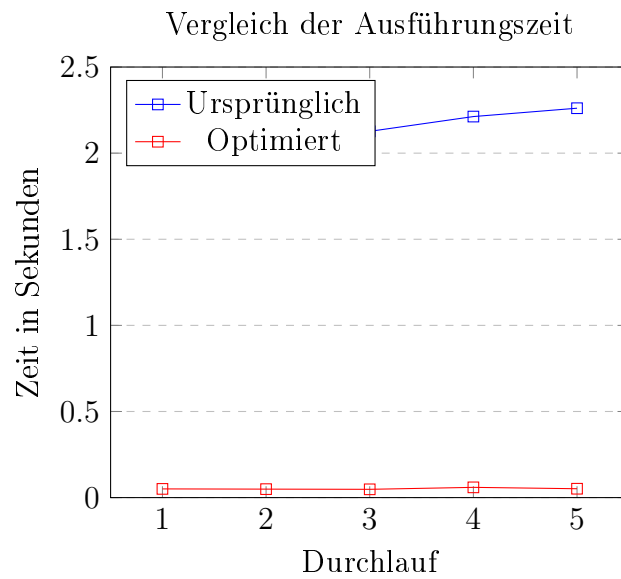


Abbildung 2: Vergleich der Ausführungszeiten vor und nach der Optimierung

5 Fazit

Die Optimierung des Codes führte zu einer deutlichen Reduzierung der Ausführungszeit, indem ineffiziente Schleifen durch einen effektiveren Subquery-Ansatz ersetzt wurden. Dieses Beispiel zeigt, wie wichtig die Überprüfung und Optimierung von Datenbankabfragen in der Softwareentwicklung ist.