

EXAMENSARBETE High-performance, High-Throughput memory-efficient lookup tables for schem

STUDENT Yazan Al-Aswad, Hugo Persson

HANDLEDARE Jonas Skeppstedt (LTH), Lukas Gustavsson (Neo4j)

EXAMINATOR Per Andersson (LTH)

Hur gör man för att snabbt överföra information mellan olika uppslagskatalog

POPULÄRVETENSKAPLIG SAMMANFATTNING **Yazan Al-Aswad, Hugo Persson**

Tänk dig att du ska göra uppslag i en telefonkatalog för att hitta vem du ska ringa. Tänk dig nu att denna telefonkatalog innehåller 100 miljarder människor. Hur ska du kunna hitta personen du söker på ett effektivt sätt? I vårt examensarbete undersöker vi hur moderna datastrukturer som *PtrHash* och *BuRR* kan göra sådana uppslag snabbare vid import av data till en databas.

Digitala databaser är den moderna världens informationshanterare, och i vårt examensarbete undersöker vi om datastrukturerna *PtrHash* och *BuRR* kan göra det snabbare att importera data till dem. Hur ett system lagrar sin information avgör i hög grad hur den kan användas — tänk på skillnaden mellan en historiabok och en mapp med lösa händelser.

Neo4j är en databas som specialiserar sig på att kartlägga relationer. När en användare vill sätta upp en Neo4j-databas behöver de ofta importera befintlig data, vilket kan ta upp mot 2 dagar för stora datamängder. Det ger en dålig första upplevelse och omöjliggör vissa användningsfall, t.ex. en tillfällig databas för en särskild analys.

En stor anledning till långsamheten är att det

tar mycket tid att översätta varje externt ID (en text eller siffra) till ett internt ID. Vi kombinerar därför *PtrHash* och *BuRR* till fyra potentiella implementationer av denna översättning, och jämför dem med Neo4j:s nuvarande lösning vid 1 respektive 10 miljarder nycklar.

Vid 1 miljard nycklar är vår arkitektur *Direct_PTR* runt 6× snabbare än den nuvarande. Vid 10 miljarder är den nuvarande för långsam för att jämföra med, och vår *Hybrid_BuRR_PTR* är ytterligare 11% snabbare än *Direct_PTR*.

Resultaten ger Neo4j en grund för att integrera algoritmerna i sin importprocess och bidrar samtidigt med en jämförelse mellan *PtrHash* och *BuRR* i en Java-kodbas.