

STUDENTER Martin Lindblom, Fredrik Strandin

HANDLEDARE Jonas Skeppstedt (LTH)

EXAMINATOR Per Andersson (LTH)

Prestandaanalys och optimering av PostgreSQL

POPULÄRVETENSKAPLIG SAMMANFATTNING **Martin Lindblom och Fredrik Strandin**

PostgreSQL är en populär databasserver som används i stora delar av industrin. Databasservern används för att lagra, bearbeta och hämta uppgifter åt t.ex. företag, organisationer och universitet som de förlitar sig på i sitt arbete. Ofta blir dock dessa databaser flaskhalsen i hur snabbt arbete kan utföras, och därför har vi analyserat och förbättrat en populär databasserver.

Vad är det då som påverkar prestanda, och vad är enkelt att påverka? Vi har i vårt arbete valt att titta på två faktorer. Den första är filsystemet - hur informationen är lagrad på hårddisken. Det finns en uppsjö med filsystem som vänder sig till olika behov, vissa är väldigt enkla, andra har extremt komplicerade funktioner för att vara säkra på att ens filer inte förstörs eller råkar försvinna. Det man ser här är att ju fler funktioner, desto långsammare blir filsystemen - i de flesta fallen. Dock fann vi ett undantag från denna regel i filsystemet *ext4*, som presterade bättre än sin föregångare *ext2*, trots nya funktioner som till exempel journalföring.

Den andra aspekten vi tittade på var kompilatorer och de optimeringar som de kan göra. En kompilator är det verktyg som översätter programmerarens kod till en binärfil - ettor och nollor som datorn kan förstå. I samband med detta kan kompilatorn göra optimeringar i programmet som gör att koden kör snabbare. Alla typer av optimeringar är dock inte bra för alla program, då olika program beter sig på olika sätt. Därför har man lagt optimeringarna i olika nivåer beroende på hur aggressivt de optimerar. Aggressiva optimeringar kan öka kodstorleken vilket kan försämra prestandan. Det finns även ett flertal olika kompilatorer som beter sig olika, där vi har valt att jämföra *GCC* och *Clang*. *GCC* är mer än 15 år äldre än *Clang*, vilket har sina för- och nackdelar. *GCC* är dåligt strukturerad vilket gör det svårt för utvecklarna att lägga till nya funktioner, men har å andra sidan använts väldigt länge och därmed

blivit mer mogen. Det senare syntes tydligt i våra resultat som visade på att *GCC* fortfarande producerar bättre och snabbare program. Vår studie visar också att PostgreSQL är ett program som presterar bäst om man inte optimerar allt för aggressivt.

Vi ville också se om vi kunde få PostgreSQL att bli snabbare genom att ändra i koden. För att göra detta utnyttjar vi att moderna datorer har buffertar mellan arbetsminnet och processorn. Dessa buffertar kallas för cacheminnet och är extremt snabba att jobba med, men små eftersom dom är dyra att tillverka. Därför kan man hjälpa datorn att välja vad som ska hämtas in i cachén närmast, så att den data man vill jobba med ligger redo att användas när den behövs. Dock är detta riskfyllt, och kräver att man har full koll på hur lång tid det tar att hämta datan. Gör man det för sent hjälper det inte, och ger konstiga effekter i cacheminnet som påverkar senare beräkningar negativt. Gör man det för tidigt så riskerar datan att skrivas över igen innan den ska användas. Efter analys av hur PostgreSQL beter sig, och hur det är skrivet, kunde vi slutligen hitta en del av koden där vi kunde ge denna typ av hjälp till datorn så att PostgreSQL blev 0.5% snabbare.

Sammanfattningsvis kan vi konstatera att PostgreSQL presterar bäst på *ext4*-filsystem, kompilerad med *GCC* på optimeringsnivå 2, samt att PostgreSQL kan fördelaktigt tipsa processorn om vad för data som ska användas. ■