INSTITUTIONEN FÖR DATAVETENSKAP | LUNDS TEKNISKA HÖGSKOLA | PRESENTATIONSDAG 2016-01-16

EXAMENSARBETE The Shortcut Index
STUDENT Anton Persson
HANDLEDARE Krzysztof Kuchcinski (LTH), Johan Svensson (Neo Technology)
EXAMINATOR Per Andersson

Snabb sökning i kopplad data

POPULÄRVETENSKAPLIG SAMMANFATTNING **Anton Persson**

Den data som skapas, hanteras och analyseras idag är allt mer "kopplad". För att hantera den sortens data krävs grafdatabaser. Vi visar i detta arbete hur vi kan göra svarstiderna 10100 gånger snabbare.

Typen av data som vi hanterar har förändrats mycket sedan slutet av 90´talet. Framför allt på grund av internet. Då använde vi datorer för att digitalisera lönelistor och svar på formulär. Denna typen av data är isolerad och har ett eget värde i sig självt. Men den data som skapas på internet ser annorlunda ut. Den är "kopplad" och mycket av informationen ligger i hur datan relaterar till sin omvärld.

Ta Facebook som exempel. Jag har en profil med lite information om mig själv. Men du är mer intresserad av vilka gemensamma vänner vi har. Dvs, våra relationer till andra människor. Jag har också skrivit kommentarer, men vad som är mer intressant är vad jag har kommenterat på. Alla dessa kommentarer, profiler och bilder är små noder av data och de relaterar till varandra med relationer. T.ex likes och vänrelationer. Dessa noder och relationer bildar tillsammans ett nätverk av data. När vi hämtar data från detta nätverk så "traverserar" vi från en nod längst med dess relationer ut till nya noder. T.ex när du öppnar min Facebookprofil så vill du även se vilka mina vänner är. Då traverserar du från min "profil"nod via "vän"relationer ut till andra "profil"noder. På så sätt kan du sätta mig i ett sammanhang och lära dig mer om mig.

För att traversera i detta nätverk av information i realtid krävs det att datan är sparad på ett smart sätt. Traditionella databaser, så kallade relationsdatabaser, använder tabeller med rader och kolumner för att spara data. Men den modellen passar dåligt för den här typen av "kopplad data". Det är här som grafdatabaser kommer in i bilden. Ordet graf är det matematiska namnet på ett nätverk av noder och relationer och det är så grafdatabaser sparar all data, som noder och relationer. Det gör det enkelt att "traversera" i informationsnätverket.

MEN! Även grafdatabaser stöter på problem när noder har extremt många relationer. Speciellt svårt är det om vi endast är intresserade av en liten del av alla "grann"noder. T.ex kanske det endast är de tio nyaste vännerna som skall visas på profilsidan. Men för att hitta de tio senaste så måste vi först plocka fram alla vänner och sedan sortera dem efter när vänrelationen skapades. Om jag har 200 vänner så innebär det en otrolig mängd onödigt arbete och försämrade svarstider för databasen.

I detta arbete visar vi hur vi kan komma runt problemet genom att hålla ett sorterat register, eller index, över de mest kritiska traverseringarna. Svarstiderna kan på så vis bli mellan 10100 gånger snabbare. Tricket för att nå denna förbättring är att vi använder oss av ett B+ träd, en datastruktur som länge har använts för indexering i databaser. Vad som är nytt är hur vi använder den för att indexera en hel substruktur av grafen istället för bara en enda nod.

Genom att minska svarstiden för täta grafdatabaser öppnar vi upp möjligheter att utföra mer avancerad analys av data i realtid. Vilket är mycket värdefullt eftersom data i sig själv inte är värt något förrän vi kan tolka den.