Nya modeller för bättre automatisk hantering av molnapplikationer

Johan Ruuskanen

Institutionen för reglerteknik

Populärvetenskaplig sammanfattning av doktorsavhandlingen *Dynamical Modeling of Cloud Applications for Runtime Performance Management*, november 2022. Avhandlingen kan laddas ner från `www.control.lth.se/publications`.

Ett datormoln kan överskådligt beskrivas som en samling datorer där delar av den samlade datorkraften snabbt kan lånas ut till olika aktörer. Detta förhållandevis simpla koncept innebär att aktörerna kan köra mjukvara på den lånade datorkraften, utan att behöva spendera dyra pengar och tid på att köpa in och hantera sina egna datorer. Detta är speciellt betydelsefullt för aktörer med begränsade resurser, såsom nystartade företag. Om mjukvaran som körs i molnet löpande hanterar förfrågningar från olika användare brukar den kallas för en *molnapplikation*. Som ett exempel kan molnapplikationen vara en sökmotor och användarna personer som vill veta mer om diverse ämnen. Personerna slår in sina sökord i exempelvis en webbläsare, vilken i sin tur skickar en förfrågan till molnapplikationen, som svarar med sökresultaten. Moderna molnapplikationer består oftast av flera mindre delar mjukvara där varje del körs på sin egen lånade datorkraft, se illustrationen till höger. Denna uppdelning har flera fördelar, bland annat blir det lättare att underhålla och uppdatera applikationen och se till att de olika delarna får tillgång till rätt typ av hårdvara.

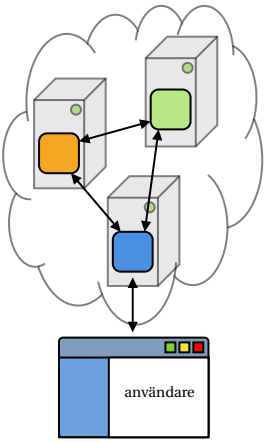

Illustration av en molnapplikation.

Det är viktigt att molnapplikationen har tillräckligt bra prestanda för att användarna ska vara nöjda. Detta kan vara svårt att precisera. Ofta anger man olika gränsvärden för diverse prestandamått som ska vara uppfyllda. Vanliga mått är till exempel tiden det tar för användarna att få ett svar och hur stor del av tiden som applikationen går att nås. Förutom att effektivisera själva koden i mjukvaran, så kan molnapplikationens prestanda påverkas genom att ändra *hur* mjukvaran körs. Här finns det många olika möjligheter, till exempel kan mer datorkraft

hyras till tungt belastade delar för att snabba på svarstiden. Dock är molnapplikationens prestanda varierande, bland annat belastas den olika över tid då antalet användare ofta förändras. För att kunna tillmötesgå prestandakraven utan att slösa resurser på att hyra datorkraft i onödan, måste hanteringen av hur molnapplikationen körs ske löpande – helst även automatiskt för att snabbt kunna anpassa sig till nya förhållanden. Bra beslut i denna hantering bygger på att man vet hur olika belastningar och körningsändringar påverkar prestandamåtten, och ett sätt att öka förståelsen är att konstruera en matematisk modell över hur dessa är sammankopplade.

I den här avhandlingen studeras sådana matematiska modeller. Fokus läggs på så kallad *köteori*, vilket är vanligt förekommande i modellering av molnapplikationer. De olika delarna av molnapplikationen representeras av köer, där förfrågningarna måste samsas om åtkomsten till en begränsad resurs. Från en sådan kömodell är det möjligt, men generellt sätt tidskrävande, att utvärdera viktiga prestandamått. För vissa typer av kömodeller finns det dock metoder för att approximativt snabbt utvärdera måtten. I avhandlingens första del studeras flödesmodeller för detta ändamål, där ankomst och avgång av förfrågningar till en kö representeras som ett in- och utflöde. Existerande resultat för utvärdering av stora könätverk utvidgas, och nya sätt för att öka precisionen och för att ta fram approximationer av olika statistiska mått på svarstider introduceras.

Dessa flödesmodeller används sedan för att ta fram en enkel men allmän modell för molnapplikationer bestående av många små mjukvarudelar. Modellen fångar även märkbara nätverksfördröjningar mellan delarna, vilka kan uppstå om de till exempel är placerade på olika datormoln. Tack vare sin enkelhet kan modellen snabbt tas fram och uppdateras direkt från mätdata av en körande molnapplikation, vilket är viktigt om den ska användas för att hantera körningen. Experiment visar sedan att modellen, trots sin enkelhet, i många fall kan fånga viktiga prestandamått med hög precision. Baserat på denna modell introduceras sedan en ny metod för att minska körningskostnaden av en molnapplikation utan att bryta mot satta prestandakrav, genom att hantera balanseringen av förfrågningar över applikationens olika delar. Metoden bygger på så kallad *automatisk differentiering*, vilket innebär att det i princip är möjligt att godtyckligt definiera kostnaden och prestandakraven så länge de kan uttryckas i modellen.

Slutligen studeras konceptet *kloning* av förfrågningar, vilket innebär att en förfrågan kopieras och skickas till olika kopior av molnapplikationen. Svaret användaren får är sedan den första kopian som blir klar. På grund av inneboende osäkerhet kring hur snabbt en förfrågan hanteras, kan kloning i vissa fall ge ökad prestanda. I avhandlingen introduceras ett nytt allmänt sätt att modellera kloning över kömodeller, vilken kräver mycket få antaganden om de inblandade köerna. Dock kräver kloningsmodellen vissa andra orealistiska antagande, och hur modellens precision påverkas av att släppa på dessa studeras för en typ av kö vanlig inom modellering av molnapplikationer.