

Prestanda- och kostnadsoptimerade molnarkitekturer

Författare/Authors: Daniel Raniz Raneland

Handledare/Supervisor: Vadim Feldman (Flygprestanda AB)

Examinator/Examiner: Mathias Haage (LTH)

POPULÄRVETENSKAPLIG SAMMANFATTNING

Att flytta sina servrar från källaren till molnet blir mer och mer populärt, man behöver inte längre oroa sig för att köpa in och underhålla fysiska datorer och man kan enkelt utöka sin kapacitet utan att behöva vänta på en budbil. När Flygprestanda AB flyttade sin serverlösning från kundens källare till Amazons EC2 öppnades möjligheter för att effektivisera resursanvändningen samtidigt som en del tidigare okända prestandaflaskhalsar uppvisade sig.

Flygprestanda AB har tidigare erbjudit sin klient-/serverlösning, FOCS, till kunder som ett installationsprogram de kan köra på sina egna servrar och klientdatorer men efter att flera mindre företag som inte har möjlighet att drifva sina egna servrar uttryckt önskemål om att köpa en komplett lösning med serverhårdvara inkluderad bestämdes det att en centraliserad lösning skulle sättas upp på Amazon Web Services där Flygprestanda har full kontroll över både (virtuell) hårdvara och mjukvara. Samtidigt ska även den traditionella modellen fungera för kunder som har möjlighet att drifva sina egna servrar.

Jämfört med en isolerad lösning som körs på plats hos kunden erbjuder en centraliserad lösning som körs på Amazons EC2 möjligheter dela gemensamma resurser mellan flera olika FOCS servrar

för att uppnå en högre effektivitet. Värt att notera här är att FOCS är utvecklat på ett sätt som inte tillåter att en serverinstans servrar flera olika kunder. Däremot är en del av de tjänster som FOCS utnyttjar för beräkningar helt tillståndslösa och kan därmed brytas ut och användas som delade resurser.

Att dela resurser mellan flera olika servrar öppnar möjligheter för att antingen spara pengar genom att använda totalt färre resurser eller genom att erbjuda bättre och snabbare service för samma kostnad. Genom att använda så kallad autoskalning så kan man kombinera dessa och matcha den beräkningskapacitet som finns tillgänglig med den som krävs och därmed se till att man inte betalar för mer beräkningskapacitet än vad som behövs.

När klienten testkördes mot en ny-

uppsatt FOCS server på Amazon EC2 upptäcktes en del prestandaproblem som efter undersökning med hjälp av instrumentering av källkoden visade sig bero på både den längre fördröjningen mellan klient och server för en server som befinner sig i ett datacenter på Irland gentemot en server på samma lokala nätverk samt på grund av den nätverksbaserade "hårddisk" som valt att användas på molnservern på grund av att det gav enklare administration av servern.

Dessa prestandaproblem löstes genom att minska antalet anrop mellan klient och server genom att slå ihop flera mindre anrop till ett större och genom att lagra alla I/O-intensiva resurser på det lagringsutrymme som finns tillgängligt lokalt på varje virtuell server i EC2, detta innebar dock en längre uppstarttid för servern eftersom dessa resurser måste kopieras över nätverket innan de är färdiga att användas.

I slutändan uppnåddes en centraliserad servermiljö körandes på EC2 med flera beräkningsintensiva resurser delade mellan flera servrar. Någon autoskalning sattes däremot aldrig upp eftersom kapacitetskraven på de delade resurserna inte var tillräckligt stora för att kräva mer än en server per resurs.