

EXAMENSARBETE Navigating Failures in Distributed Systems:

A Comparative Study of Failure Detection Algorithms

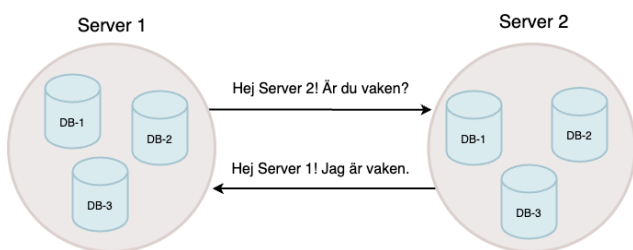
STUDENTER Hannes Brinklert, Johan Åkerman**HANDLEDARE** Jonas Skeppstedt (LTH), Aleksey Karasavov (Neo4j)**EXAMINATOR** Michael Doggett (LTH)

Hur hittas fel i distribuerade system?

POPULÄRVETENSKAPLIG SAMMANFATTNING **Hannes Brinklert, Johan Åkerman**

Distribuerade system används idag allt mer för att köra applikationer som ställer högre krav på prestanda och pålitlighet. Denna studie har undersökt hur Neo4js världsledande grafdatabaser kan bli mer pålitliga med hjälp av fel-detekteringsalgoritmer.

Distribuerade system består av geografiskt åtskilda noder som är ihopkopplade med länkar över ett nätverk. Vad noderna i fråga faktiskt är varierar på användningsområdet. I Neo4js fall är varje nod en server som kör en eller flera databaser. En av de större ingenjörsutmaningarna med distribuerade system är samarbete då alla noderna har sin egen syn på systemet som en helhet. Fel-detekteringsalgoritmer används för att lösa detta problem genom att hitta noder som inte fungerar som de ska, se exempel nedan.



I detta examensarbete har vi under fem månader samarbetat med företaget Neo4j som ligger i framkant inom grafdatabaser. Mer specifikt så har vi undersökt två alternativa vägar framåt för att förbättra Neo4js fel-detektorer som i sin tur ökar pålitligheten av databaserna som används av 75% av Fortune 100 företag. Det första alternativet bestod av att optimera valet av parametrar i Lighthouse, företagets redan existerande algoritm. Det andra alternativet var att ersätta Light-

house med en ny algoritm med namn SWIM. Den största skillnaden mellan Lighthouse och SWIM är att SWIM tar en nod hjälp av sina grannar och skickar mindre och mer relevanta meddelanden.

Den ursprungliga Lighthouse algoritmen användes som en referenspunkt och jämfördes mot en parameter optimerad version av både Lighthouse och SWIM. De tre konfigurationerna utvärderades i ett simulerat nätverk som utsattes för tre olika typer av fel: krasch, länk och fördröjningsfel.

Under simuleringen utvärderades de tre algoritmerna utifrån tre perspektiv: snabbhet, träffsäkerhet och belastning på nätverket. Resultatet visar att den ursprungliga Lighthouse algoritmen är skalbar och pålitlig men långsam. Jämfört med den etablerade referenspunkten så är den parameter optimerade Lighthouse algoritmen 61.6% snabbare på att hitta fel men något mindre träffsäker (5%). Samtidigt är den nya SWIM algoritmen något snabbare än referenspunkten (26.5%) men betydligt mindre träffsäker (15.4%). En viktig observation är att SWIM utsätter nätverket för betydligt mycket mindre trafik vilket blir särskilt uppenbart i distribuerade system med fler noder. Den något överraskande slutsatsen som kunde dras från resultatet är att optimering av algoritmernas parametrar har större påverkan än själva valet av algoritm.