

Navn:	Studienummer:
Thomas Carstensen	201807833
Markus Kristian Simonsen	201808680
Matias Rosenkilde Tvorup	201807748
Frederik Christ Vestergaard	201708560

Problemer med vores schema

Hvis en bruger er populær og meget aktiv på det sociale medie ved at ofte lave opslag på sin "wall" og i sine cirkler, så er schemaet problematisk fra et ressourcensynspunkt, da mange posts med mange kommentarer hurtigt ville fylde de 16 MB, som er begrænsningen for hver MongoDB samling. Alternativet kunne være, at hver post havde sin egen samling af kommentarer, som så får sit eget dokument eller flere dokumenter. Posts ville så have en eller flere referencer til samling(er) med kommentarer.

Et andet problem med vores schema er at posts er embedded i hhv. users og circles. Hvis man skal query efter posts som en user har lavet, så returneres alle posts som useren har lavet, hvorefter der skal filtreres i vores c# kode. Hvis vi havde refereret til en samling, kunne man have filtreret sin query til kun at returnere det man ønskede. Dette ville også være nemmere at query for specifikke ting.

Problemer ved sharding

Sharding kræver meget præcis/forsigtig planlægning, udførsel og vedligeholdelse. Sharding er godt til udvidelse og skalering, men dette kræver også at database er designet til at kunne skaleres. Hvis databasen ikke er designet til dette, vil sharding være problematisk og kunne resultere i en række af problematiske scenarier.

Et problem kunne være, at shards kunne blive for små, hvilket kunne resultere i, at det meste data ender på nogle enkelte shards i stedet for at blive spredt over hele netværket balanceret.

Et andet problem kunne være, at man at fylder en shard eller en node med for meget data, hvilket ville kunne overbelaste de tilgængelige resourcer. Særligt med et social network med en populær profil og tusindvis af posts, hver med tusindvis af kommentarer. Dette ville hurtigt kunne overbelaste en shard og/eller en node.

Hvis balanceringen af shards og data sets til shards ikke er gjort ordentlig, kan man opleve ubalance mellem shards, hvilket resultere i, at nogle shards fyldes hurtigere end andre. Når dette sker, vil andre, mindre belastede dele af netværket, være nødt til at vente på den overbelastede shard, som tager længere tid om at query da dens RAM er fyldt.