

TD4145 Datamodellering og databasesystemer

Queryprosessering

- a) Antall blokker er 1 TB (filstørrelse)/ 32 KB (blokkstørrelse) = 33 554 432 når man antar at alle enheter(Terra,Giga,Mega,Kilo) er 1024.
Antall blokker i minne samtidig: 1 GB / 32 KB = 32767
Hvor mange pass får vi: Avrundet oppover($\log_{32767}(1024)$) = 1. Vi får da antall I/Oer $2 * 33554432$ (les og skriv hver blokk under sortering) + $1 * 2 * 33554432$ (les og skriv hver blokk under fletting, men kun ett pass) = $4 * 33554432 = 134217728$.
- b) Her får vi i gjennomsnitt 1000 I/Oer da posten kan finnes et vilkårlig sted i heapfila. Ei heapfil er en samling med blokker hvor postene ikke er sortert på noen måte. I gjennomsnitt vil posten befinne seg halvvegs ut i fila. Derfor kun 1000.
- c) Her må vi regne ut høyden på B+-treet. Vi får ca. 3000 blokker på løv nivå (nivå 0) i B+-treet, fordi B+-treet trenger 50% mer plass enn heapfila (B+-treets blokker har gjennomsnittlig 67% fyllgrad). Vi har da 67 poster per blokk på nivå 0. Vi trenger 3000 pekere (og indeksposter) på nivå 1 fordi det er 3000 blokker på nivå 0. Vi får plass til ca. 667 poster per blokk på nivå 1 fordi det er plass til 10 ganger så mange poster per blokk her enn på løv nivå., dermed må vi ha fem blokker på nivå 1, og ei blokk på nivå 2, dvs. rotblokk. Rotblokk har 5 poster. Dermed får vi tre I/Oer for direktesøket på `ansattId`. De aller fleste B+-trær har 3 eller 4 nivåer. Det er ellers OK å anta at et B+-tre har høyde 3 eller 4 i slike oppgaver (også sagt på forelesning).
- d) Her får vi 2000 I/Oer da alle blokkene i heapfila må leses. Poster i heapfila er ikke sortert på noe felt og da må alle blokkene leses.
- e) Her er det et nytt B+-tre med søkenøkkel etternavn. Vi må da på nytt regne ut høyden på B+-treet, som sannsynligvis ville bli 3 eller 4 slik som alle B+-trær er i praksis. Da får vi en traversering ned til 'Hansen', dvs. 3 blokkaksesser. Så aksesseres ca. 10% av nivå 0-blokkene "sidevegs", dvs. 300 blokker. Vi får totalt ca. 303 I/Oer. Med B+-treet organisert med søkenøkkel `ansattId`, ville vi fått traversering av hele nederste nivå i B+-treet, dvs. $3 + 3000$, dvs. 3003. Evt. kan vi ha $3 + 299$ eller $3 + 2999$ hvis vi antar 3 nivåer og at du ikke trenger å lese første løvblokk både på vei nedover og bortover ...
- f) Hvis vi gjør bruk av indeksen, får vi ca. 20000 poster som kvalifiserer og hver av dem kan gi en I/O når blokk hentes fra heapfila. I tillegg må B+-tre-indeksen traverseres. Her vil vi ha ca. 600 blokker på nivå 0 (20% av de 3000 blokkene fra c) På nivå 1 får vi 600 pekere. Hver blokk her har plass til 335 poster (5 ganger mer enn ei nivå-0-blokk fra c). Da får vi to blokker på nivå 1 og ei blokk på nivå 2. Da har vi tre blokker for nedovertraverseringen og 60 blokker for bortovertraversering. Til sammen har vi inntil 20063 blokkaksesser. Da er det bedre å skanne heapfila som totalt gir 2000 aksesser.
- g) Her kan du bruke 10 av blokkene i bufferet til Department og 1 til Employee og 1 til resultat. Da får vi totalt $10 + 2000$ I/Oer på å lese data, dvs. vi har hele Department i buffer og skanner gjennom Employee med ei og ei blokk om gangen. I tillegg får vi I/O på å skrive ut resultatblokkene også.
- h) Veldig lite kan forbedres. Hvis det hadde vært en clustered indeks på `dname`, kunne vi hentet ut kun de få blokkene som inneholder 'Accounting', og spart oss noen få I/Oer.