NTNU
Kunnskap for en bedre verden

Institutt for datateknikk og informasjonsvitenskap

Eksamensoppgave i TDT4305 Big Data-Arkitektur

Faglig kontakt under eksamen: Kjetil Nørvåg/Heri Ramampiaro
Tlf.: 73596755/73591459

Eksamensdato: 28. mai 2016
Eksamenstid (fra-til): 09.00-13.00
Hjelpemiddelkode/Tillatte hjelpemidler: D: Ingen trykte eller håndskrevne hjelpemiddel tillatt.
Bestemt, enkel kalkulator tillatt.
Annen informasjon:

Målform/språk: Bokmål
Antall sider (uten forside): 4
Antall sider vedlegg: 0

Kontrollert av:

_____

Dato Sign

Informasjon om trykking av eksamensoppgave	
Originalen er:	
1-sidig X	**2-sidig** ☐
sort/hvit X	**farger** ☐

Merk! Studenter finner sensur i Studentweb. Har du spørsmål om din sensur må du kontakte instituttet ditt. Eksamenskontoret vil ikke kunne svare på slike spørsmål.

Oppgave 1 – Big Data – 5 %

a) Når man skal forklare Big Data snakker man ofte om de tre (eller flere) V'ene. Forklar de tre viktigste av disse.

Oppgave 2 – Hadoop – 20 % (alle deler teller likt)

a) Hva var viktige mål for Hadoop File system (HDFS), og hva er HDFS *ikke* egnet til?

b) Beskriv arkitekturen til HDFS (bruk gjerne figur). Beskriv hvordan filer er lagret og node-typer.

c) Forklar hva som skjer når en klient skal lese en fil som er lagret i HDFS (inkl. interaksjon mellom noder).

d) Forklar utføring av en applikasjon på YARN, inkl. beskrivelse av node-typer og prosesser. Forklar gjerne med figur.

Oppgave 3 – MapReduce og Spark– 10 % (alle deler teller likt)

Anta at man har en fil PersonInfo.txt som inneholder informasjon om navn, alder og lønn, dvs. format som dette:

Kari 45 450000
Ola 30 200000
Kate 30 500000
Pål 45 550000

Vi ønsker å finne gjennomsnittsinntekt for hver alder, dvs. resultat som dette (trenger ikke å være sortert):

45 500000
30 350000

a) Vis med pseudokode for *mapper* og *reducer* hvordan dette kan gjøres i MapReduce. Anta for enkelhet skyld at *value* til map er en post med feltene *age* og *salary*, dvs. bruk følgende som utgangspunkt:

```
public void map(key(name), value(age,salary))
public void reduce(key, Iterable values)
```

b) Vi ønsker nå å finne maks-lønn for hver alder ved hjelp av Spark. Anta at vi allerede har lest filen inn i en RDD av par (key,value)=(age,salary), dvs. RDD[(int,int)]. Vis hvilke(n) transformasjon(er) som må gjøres for å få en resulterende RDD der (key,value)=(age,maxSalary). Hint: viktige transformasjoner og handlinger ("actions") i Spark inkluderer `map, flatMap, filter, distinct, union, collect, count, countByValue, reduce, reduceByKey, groupByKey, values, sortByKey,` og `countByKey.`

Oppgave 4 – NoSQL – 15 % (10 % på a, 5 % på b)

a) Vi har en studentdatabase med følgende tabeller i et relasjonsskjema.

```
Student(SNo, Name, Email)
Exam(ENo, CourseName, EDay, EMonth, EYear, Duration)
ExamResult(ExamNo, StudentNo, Grade)
```

Her er det interessant å finne hvilke eksamener en spesiell student har tatt (karakterutskrift). Det er også interessant å finne ut hvilke studenter som har tatt en spesiell eksamen (sensurliste). Hvordan ville du lagre dette skjemaet i HBase når du bruker designprinsippet DDI (*denormalization, duplication, intelligent keys*) og har bruk for de to forskjellige spørringene som antydet over?

b) Gi en beskrivelse av hvordan «sharding»/partisjonering foregår i både MongoDB og i Apache HBase (også kalt «auto-sharding»).

Oppgave 5 – Datastrømmer (streaming data) – 30 % (Alle deler teller likt)

Du skal analysere hvor mange ganger et tema om amerikansk valg og valgkamp blir nevnt i meldinger i sosiale media som Twitter.

a) Drøft karakteristikkene og/eller utfordringene med datastrøm. Nevn to andre eksempler hvor man er nødt til å håndtere en datastrøm (i tillegg til Twitter og eller sosiale media generelt).

b) Vi skiller mellom to typer spørringer når det gjelder datastrøm. Forklar hva disse er. Bruk eksempler til å støtte forklaringen din.

c) Se for deg at du skal finne ut hvor stor andel av meldingene er relatert til temaet ”valg” og ”valgkamp” i en gitt begrenset tidsperiode. Til dette formålet velger vi å bruke glidende-vindu-prinsippet (“sliding window”). Anta at dette vinduet har en størrelse på 1000 twitter-meldinger (dvs. Tweets). Vis hvordan du går fram for å beregne denne andelen.

d) Kan problemet over sees på som en variant av “bit counting”? Begrunn svaret ditt.

e) Bruk “bloom filter”-prinsippet til å fylle ut tabellen nedenfor

Strømelement	**Hash-funksjon - h_1**	**Hash-funksjon - h_2**	**Filtrere Innhold**
			00000000000
39 = 10 0111			
214 = 1101 0110			
353 = 01 0110 0001			

Hint: bruk *h(x)= y mod 11*, der *y* er hentet henholdsvis fra oddetalls-bits fra *x* eller partalls-bits fra *x*.

f) Anta at vi vil analysere de 11 siste meldingene som er kommet inn. Generelt på twitter, vil mange av meldingene bli sendt på nytt av samme bruker for å markere sitt synspunkt. Andre brukere vil "re-tweete" for å få flere til å få med seg meldingene. Forklar hvordan vi kan bruke bloom-filtre for å filtrere bort slike meldinger. Gjør de antakelsene du finner nødvendig.

g) Anta nå at når de 11 meldingene har kommet inn har vi fått en strøm av data som ser ut som dette: 10100101010. Kan vi ha sett meldingen som kan representeres ved y = 1111011 før? Begrunn svaret ditt.

Oppgave 6 – Anbefalingssystem (recommender systems) – 20 % (6% på a.i, 4% på a.ii, 10% på b)

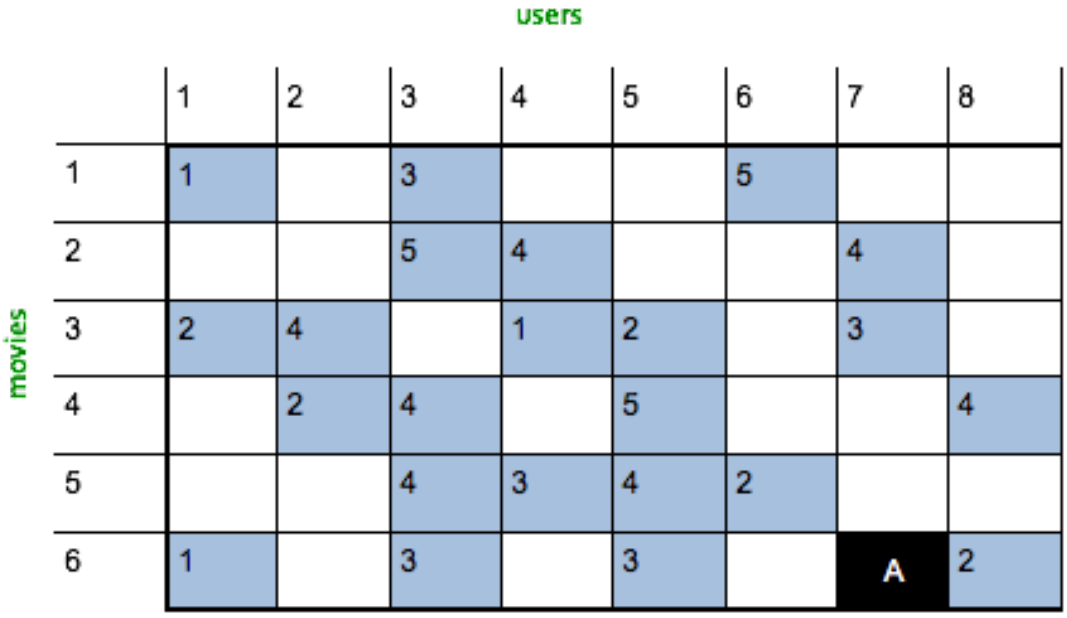
Du er nyansatt i et nytt firma som vil spesialisere seg på strømming av film. En av oppgavene dine er å utvikle gode anbefalingsalgoritmer og metoder.

a) En del av metoden du foreslår går ut på å gi brukeren mulighet til å "rate" filmene for så bruke dette til å finne ut hvilke filmer systemet deres skal anbefale senere. Anta at brukerne deres har "rated" følgende 10 filmer med 3 eller flere stjerner:

```
Jurasic Park (Fantasi/SciFi), Harry Potter (Fantasi/Adventure),
ET (SciFi), Lord of the Rings (Fantasi/Adventure), Alien (SciFi),
Terminator (SciFi), 101 Dalmatians (Adventure/Family), Titanic
(Romantic), Sleepless in Seattle (Romantic) og Mr. Bean (Comedy).
```

i. Forklar hvordan du vil gå fram for å anbefale neste film til denne brukeren. Gjør de antakelsene du finner nødvendig.
ii. Ville du brukt innholdsbasert ("content-based") anbefalingsmetode eller "collaborative filtering"? Begrunn svaret ditt.

b) Anta følgende brukerratingstabell.



	1	2	3	4	5	6	7	8
1	1		3			5		
2			5	4			4	
3	2	4		1	2		3	
4		2	4		5			4
5			4	3	4	2		
6	1		3		3		A	2

Bruk "***item-item collaborative filtering***"-metoden til å foreslå bruker nr. 7 sin rating av film nr. 6. Dvs. hva blir ratingverdien A? Du må vise mellomregningen.

Til denne oppgaven vil du trenge følgende formler:

Pearson Correlation similarity - likhet mellom vektor x, og vektor y:

$$sim(x,y) = \frac{\sum_{s \in S_{xy}} (r_{xs} - \overline{r_x})(r_{ys} - \overline{r_y})}{\sqrt{\sum_{s \in S_{xy}} (r_{xs} - \overline{r_x})^2} \sqrt{\sum_{s \in S_{xy}} (r_{ys} - \overline{r_y})^2}}$$

der r_{xs} er bruker s sin rating på film x og $\overline{r}_x$ (overline) er gjennomsnitt av alle rating-ene på film x.

Vektet gjennomsnitt (weighted average) for en brukers ratinger:

$$r_{ix} = \frac{\sum_{j \in N(i;x)} s_{ij} \cdot r_{jx}}{\sum s_{ij}}$$

r_{ix} er her bruker x sin rating på film i, mens s_{ij} er likhet (similarity) mellom ratingene til film i og j