

IMT 3003 Ukeoppgaver 2. Mars

Å besvare en oppgave betyr å gjennomføre den på de virtuelle maskinene og å dokumentere det i en lab-rapport som leveres via fronter. Noen av disse oppgavene vil bli del av en obligatorisk innlevering, men det er mulig å levere dem allerede nå for å få tilbakemelding.

Hint: Kommandoer som kan være til hjelp for å jobbe med logfiler: grep, cut, sed, head, tail

1. (Del av oblig 2) I mappen “gamedata” under “Undervisningsmaterieell” i blackboard finner dere tre filer: tf2_50k.dat, tf2_3k_begin.dat, tf2_3k_end.dat. Den første filen inneholder 50000 målinger (tatt hvert 5. minutt) av antall spillere på Steam.com som spiller Team Fortress 2. I blackboard finner dere også LibreOffice regnearket som hjelper til med analysen.

1. Undersøk filen tf2_50k.dat og beskriv fordelingen du ser og den deskriptive statistikken. Beskriv den øvre 5% percentilen (95% - 100%) av dataene for både 1%,5% og 10% fordelingene. (Du trenger ikke være helt nøyaktig)
2. Den forrige filen inneholder data fra en lang periode (ca. 173 dager) For å undersøke om det har vært noe endring i antall spillere, undersøk filene tf2_3k_begin.dat og tf2_3k_end.dat som kommer fra begynnelsen og slutten av den store filen. Finner dere noen forskjeller?
3. Hvordan kan man bruke denne informasjonen til å effektivisere og optimalisere driften av spillservere?

2. (Del av oblig 2) Vi ønsker å se nærmere på forutsigbarheten til bookface aktiviteten og trenger derfor å samle inn data om oppførselen. Senere kan vi bruke de dataene til å begynne å forutse hva som skjer.

Bruk det du kan fra tidligere oppgaver til å lage et script som kan skrive ut tidspunkt og rate på inkommende forbindelser akkurat nå. Dette er det vanligvis HAproxy som vet og man kan bruke stats-siden og curl til å hente denne infoen sammen med litt shell-scripting.

For at analysen av dataene skal bli så enkel som mulig senere, er det best om tidspunktet skrives ut i “sekunder siden UNIX epoken”, som er et standardmål alle UNIX/Linux maskiner har. Dette får du på denne måten:

```
date +%s
```

Kjøringen av scriptet burde f.eks se slik ut:

```
./webrate.sh  
1519974932,23
```

Det forteller oss at på tidspunkt 1519974932 var det en rate på 23 nye forbindelser per sekund. Dersom du kjører dette scriptet i en cron-oppføring med f.eks hvert femte minutt interval og legger hver datalinje til en fil, så har du det du trenger.

Man legger til nye linjer til en fil slik:

```
/root/webrate.sh >> /root/webrate_data.csv
```

4. Installer bonnie++ på db1. Kjør en benchmark på av den og beskriv resultatet.

(ADVARSEL: resultatene kan avvike sterkt avhengig av tidspunktet på døgnet og om andre kjører tester samtidig. De kan ta en stund, så det lønner seg å kjøre dem i en screen session så du slipper å vente på dem. Det er også en mulighet for at bonnie++ kan ta ned de virtuelle maskinene. Vent gjerne til slutten av dagen eller om kvelden før dere setter i gang, siden belastningen vil affektere samtlige virtuelle maskiner.)

5. Gjenta samme oppgave som over, men denne gangen med hdparm.