Operativ Systemer 9

Lavet af: Vivek Misra

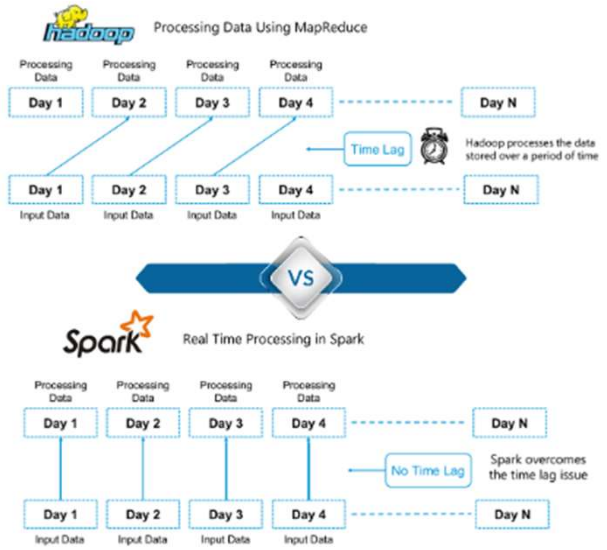
Processer

For, at kunne holde styr på de forskellige former af Operativ Systemer.

Så henviser vi venligst tilbage til Powerpoint 1.

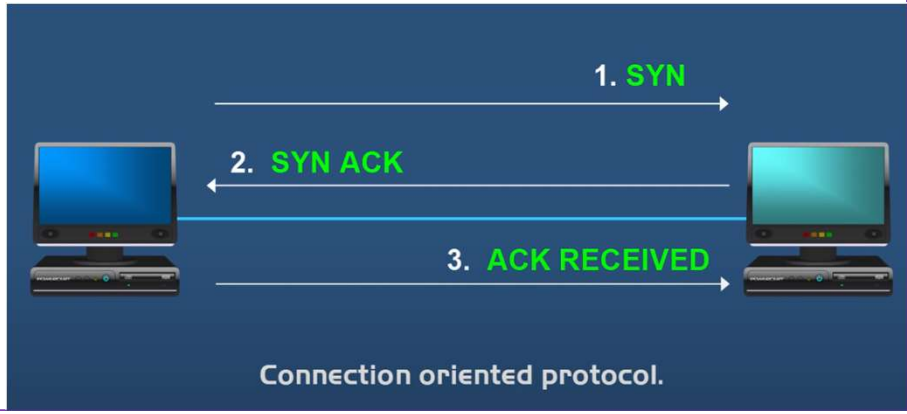
Processer (Kort Repetition)

- Batch System er der, hvor man tager noget data og giver det videre til Virksomheden.
- Denne data bliver indsat inde ved Operativ Systemet af Virksomheden.
- Ulempen var dengang, at når man indsatte data inde ved CPU'en så udførte CPU'en arbejdet til sidste ende og det krævede meget tid.
- Kigger man på Stream Processing, så er det der hvor der sker "flow" af kontinuert data og som på samme tid bliver analyseret.



TCP og UDP

- **TCP: Dette er kaldt for Transmission Control Protocol.**
- *Forestil dig, at du har noget data som du sender fra en computer og over til en anden.*
- *Når denne data bliver processeret over til den anden computer, så går den igennem TCP.*
- *TCP'ens arbejder er, at sørge for alt dataoverførslen er tjekket igennem således at igen data mangler.*
- *Det er vigtigt, at gøre obs på at TCP er en Connection Oriented Protokol som betyder, at den vil gerne have man etablere Handshake-metoden og derefter begynder kommunikation med den anden Computer.*
- *Hvis en pakke er ikke afsendt, bliver den afsendt igen!*



TCP og UDP

- **UDP: Dette er kaldt for User Datagram Protocol.**
- *UDP er ligesom TCP'en bare anderledes. En ting som er til fælles mellem de to er, at de modtager data.*
- *Men udfordringen er bare, at når man afsender noget Data vha. UDP så er der ingen garanti for at det bliver modtaget over på den anden side af Computeren.*
- *Det er derfor UDP sige for, at være en Connectionless Protcol fordi den er fuldstændig ligeglad med om dataen er modtaget eller ej.*
    - *Antag UDP for, at være Paintball hvor det handler om at slå en person ned men man misser/fejler kastet i flere omgange.*

Hvad er Protokol Stack

- Protokol Stack kan anses som en set af regler og trin i forskellige Devices som følger kommunikationen over et netværk. Det er ligesom når to mennesker snakker med hinanden, så viser de etikette. Her sørger computere for, at anvende protokoller for at kunne forstå hinanden og udveksle information på en nem måde.
- På næste side er der vist en tabel over de forskellige netværkslager, som egentligt skal forklare hvordan kommunikation mellem forskellige Devices sikre at der sker en ordentligt udveksling af information mellem enheder.

Hvad er Protokol Stack

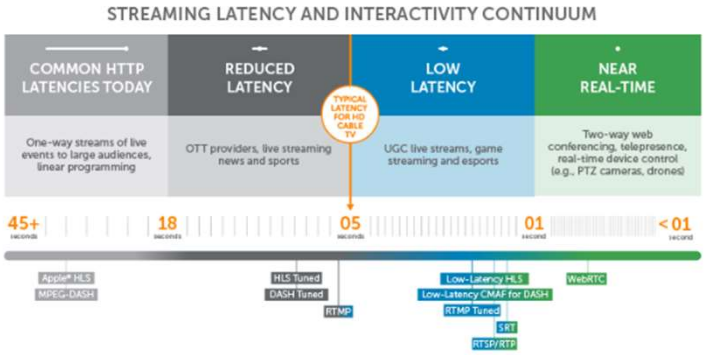
Protocol Stack Layers	Definitioner
Fysisk Lager	Det er den alger som håndterer den aktuelle forbindelse mellem to Devices gennem kabler og signaler.
Datalink Lager	Denne lager organiserer bits inde i rammer og håndterer flowet af data gennem fysisk forbindelse. Den tjekker også for Error og om dataet er leveret korrekt.
Netværkslager	Det er ligesom en Trafik-Kontrollør. Den håndterer, hvordan Dataet går fra en ende til det andet gennem forskellige netværker. Her går den igennem Routers og IP-Adresser.
Transport Lager	Her kan det ses, at Dataet er delt i mindre dele kaldet for segmenter. Den sørger for, at disse segmenter er leveret i den korrekte rækkefølge. Den håndterer også ting som trafik-congestion.
Session Lager	Denne lager håndter forbindelser og sørger for at udvekslingen af dataet sker ordentligt og sikkert. Den sætter tingene og holder styr på kommunikation på tværs af Enheder.
Presentation Lager	Denne lager er ansvarlig for, hvordan dataet er formateret, enkrypteret eller sammenpresset. Den håndter, hvordan informationen er præsenteret således at forskellige systemer kan forstå det.
Applikation Lager	Det er der, hvor den aktuelle applikation og programmer som interagerer med hinanden i Netværket. Dette inkluderer protokoller for specifikke opgaver som eksempelvis webbrowsing ved brug af HTTP., eller afsendelser af EMAIL ved SMTP eller filoverførsler ved FTP.

Publish-Subscribe Arkitekturen

- Publish Subscribe Arkitekturen forklarer egentligt, at vi har to forhold:
  Publisher: Det er den som afsender data og er ligeglad med hvem der modtager det. Det kan eksempelvis være Syddansk Universitet som offentliggøre noget på deres Youtube Kanaler og sender det videre.
- **Subscriber:** Det er ikke alle, men det er gruppe af klienter som modtager den information som er offentliggjort af Publisheren. Eksempelvis kan det være en elev fra Software Engineering som har abonneret til SDU's kanal og dermed modtager alle reklamer fra dem på Youtube.

Video on Demand Protocols

- Video on Demand (VOD) er en Protokol som afgøre en set af regler, der definerer hvordan indeholder er leveret og streamet til users-on-demand over internettet.
- Disse Protokoller sørger for en ”smood-transition”, playback og interaktion med video indhold over forskellige Enheder. Herunder er der en liste af Protokoler som indeholder VOD:
  - HTLS: Det er HTTP Live Streaming som bryder video inde i mindre segmenter.
  - DASH: Det er Dynamisk Adaptiv Streaming over HTTP som er en industri standard der arbejder ligesom HLS, og opdeler videoen inde i mindre dele.
  - RTMP: Dette er kendt som Real Time Messaging Protocol som bruges til at lave live streaming.
  - RTSP: Det er primært brugt som streaming fra server til klienter.
  - MPEG: Det er Dynamisk Adaptiv Streaming over HTTP, der er en international standard for adaptiv streaming.



Apache Kafka

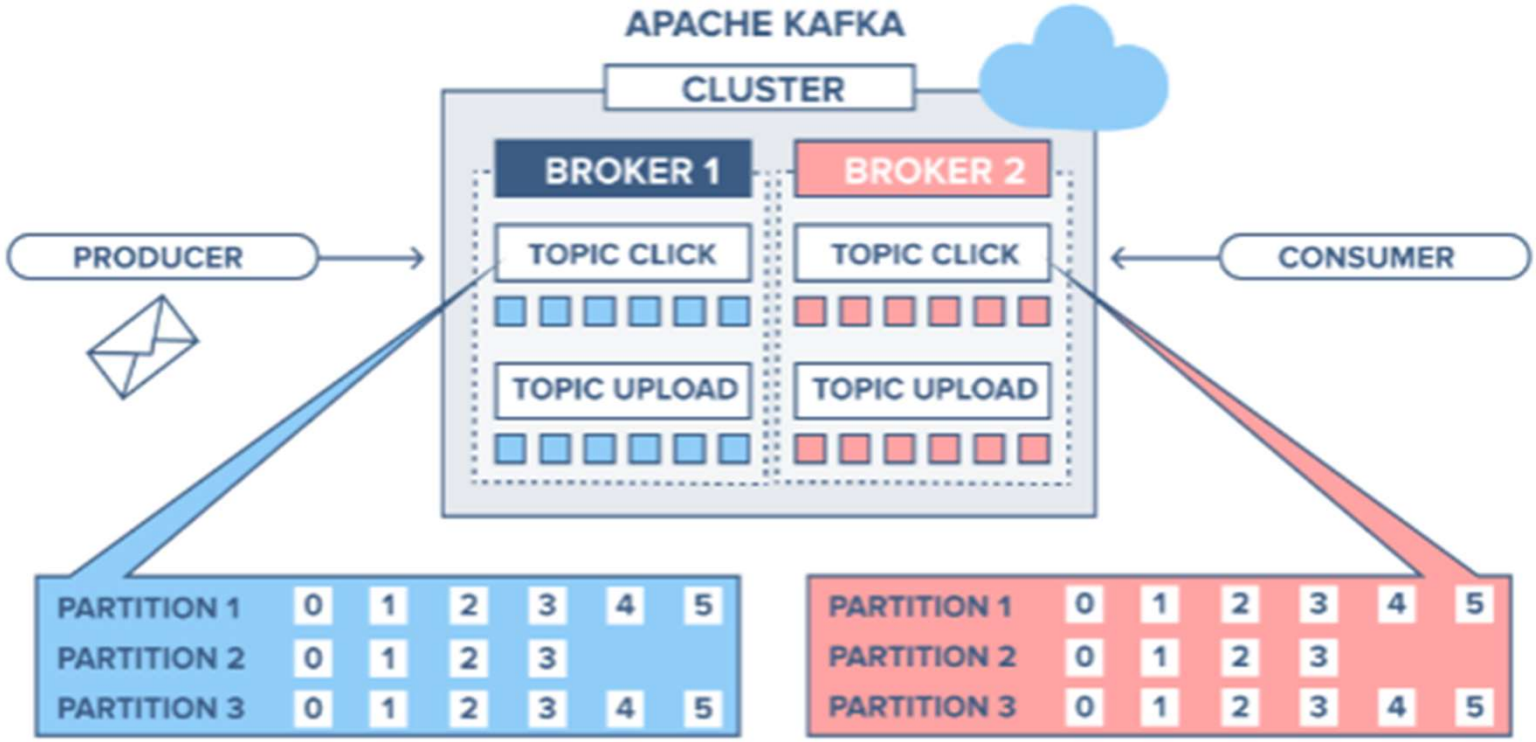
Nu kommer vi til at give en Introduktion over hvad Apache Kafka er og hvad det bruges til?

Hvad er Kafka?

- Apache Kafka er en åben source distribuerede event streaming platform der er designet til at kunne håndtere høj streaming af real-time data og streaming data pipelines. Originalt var Kafka udviklet af LinkedIn.
- Der er forskellige komponenter i Kafka:
  - Broker Arkitektur: Kafka er baseret på en distribuerede arkitektur, der inkluderer adskillige servere som er kaldt Brokers. Brokers er brugt til at styre storage, processering og distribution af data streams.
  - Topics: Data Streams i Kafka er organiseret i Topics, der fungerer som kanaler eller kategorier for beskeder. Producers offentliggør beskeder til specifikke Topics, og Consumers subscriber til disse Topics for at afhente beskeder.
  - Partitions: Hvert Topic er inddelt i Partitioner, som individuelle dele og som tillader data at være uddelt i forskellige Brokers.
  - Replikation: Kafka indeholder Fault Tolerance ved at replikere paritioner gennem flere forskellige brokers. Det sørger for, at hvis en broker fejler, så vil dataet være tilgængeligt og kan være afhentet fra de replikerede partitioner fra andre brokers. (Se det som Paritiy i RAID 5).
  - Connectiors: Det er et framework som tillader integration med forskellige data, source og sinks.
  - Streams Processing: Det er en library, som giver stream processens muligheder, som tillader udviklere til at bygge real-time applikationer til at udføre data processering.

Hvad er Kafka?

- Her viser vi et billede af Apache Kafka Strukturen:



SLUT 9

Lavet af: Vivek Misra