

Big Data i spelbranchen

ett projekt med Hadoop och open source i fokus

Kunden

Företaget arbetar med onlinespel och utvecklar många olika spel för över 100 spelbolag, exempelvis Casinon som Casinostugan och LeoVegas. Spelarna som är spridda över hela världen ska ha tillgång till onlinespel vilket ställer höga prestanda- och tillgänglighetskrav. Som ett exempel hade man 21,4 miljarder gaming transaktioner under 2014.



44 procent av alla internetanvändare spelar någon gång spel på internet för nöjes skull, även om det inte alltid spelas om pengar. 31 procent gör det varje vecka och 17 procent dagligen. Dessa fakta är tagna från 2015 men värdena har legat konstant senaste 9 åren enligt IIS, Internetstiftelsen i Sverige.

Projektets utmaningar

- 1) Att förstå hur företagets kunder använde spelets funktioner och inställningar beroende på vilken enhet de spelade ifrån.
- 2) Att förstå spelens användbarhet och användarupplevelse. Exempelvis genom frågor såsom:
 - a) Stänger spelarna av vissa funktioner beroende på var i spelet de befinner sig?
 - b) Vilket är det vanligaste sättet att hålla en enhet beroende på den fysiska storleken på den? När man knapparna enkelt med tummarna?
 - c) Hur ser heatmappen ut för miss clicks, det vill säga där kunderna klickar men det inte finns någonting att klicka på? Behöver man förtydliga att det inte är klickbart eller nyttja för att faktiskt vara klickbara?
 - d) Skiljer ovanstående frågor sig något beroende på geografi?
- 3) Systemet ska kunna hantera ca 150.000 events i sekunden från hela världen.

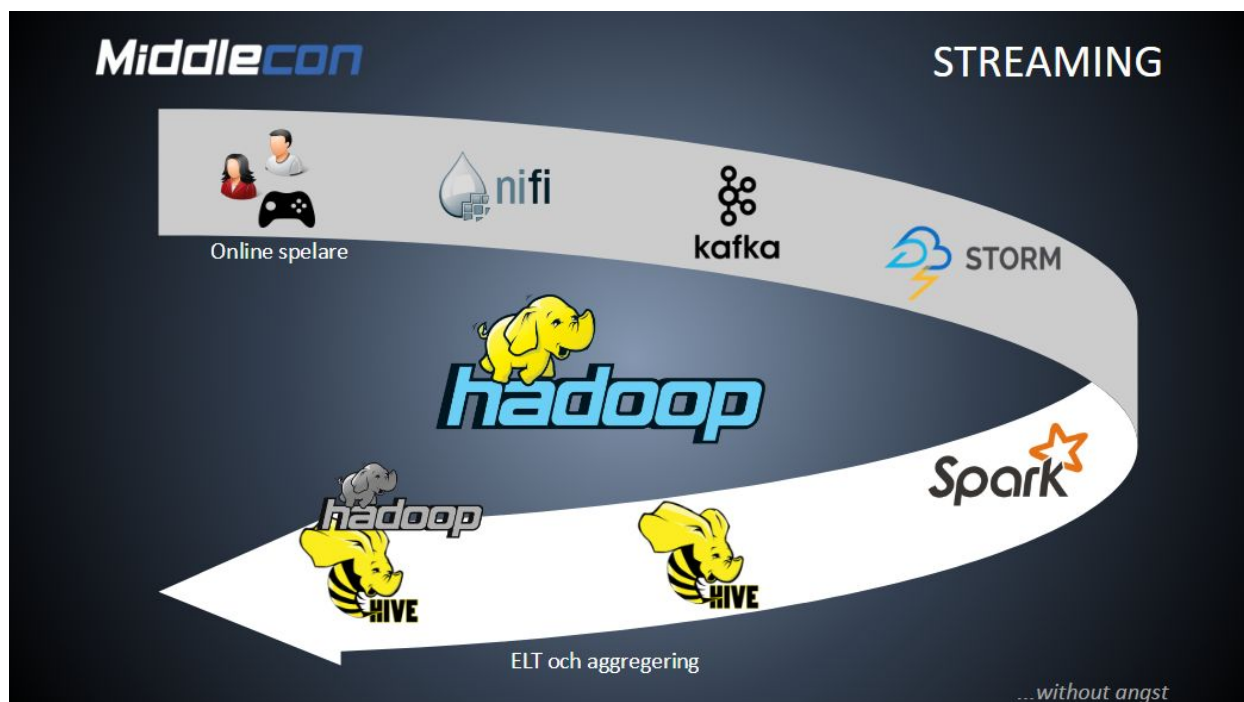
Lösningar

- Ett flertal datacenter sätts upp spridda över hela världen.
- Alla spel exekveras i en browser. En plugin skapas för att samla in information liknande:
 - Vilken enhet spelarna använder sig av?
 - Hur mycket minne det finns?
 - Vilken version av browser som används?
 - Är browsern maximerad eller inte?
 - Vilken är skärmens upplösning?
 - Operativsystemets språk i varje spelklient?
- Spelet i sig skickar information om vilka funktioner som används, vilka inställningar som görs, vilka händelser som sker i spelet och alla mus/touchskärmsklickningar som sker mm.
- Data från varje spelare skickas till närmaste datacenter för detaljerad analys.
- Processing och aggregeringar sker i varje datacenter och skickas därefter till ett centralt datacenter för self service BI samt analys på en högre nivå.

Val av Produkter

NiFi: Kunden behövde ett agilt och enkelt sätt att flytta data. Niagara Files kan göra det, samtidigt som det hjälper till med att flytta och spåra data. Det kan beskrivas som data logistik, ungefär som DHL flyttar och spårar paket. En annan fördel är att det går att konfigurera till stor del. Exempelvis med accepterade förluster kontra garanterad leverans, låg latency kontra högt throughput, dynamiska prioriteringar och flöden som modifieras i runtime. Vi valde använda REST interface, öppet från internet JSON data, med enkel validering av schema och data samt en utökning av JSON med data för plats, tid mm

KAFKA: För att klara belastningen behövdes ett Message Queue system för att hantera stora mängder av streamat data. Detta åstadkoms genom att lägga samtliga meddelande i en KAFKA kö med många som lyssnar (till skillnad från point to point i vanlig MQ). Vi såg en stor fördel med KAFKA eftersom det håller persistent data, en intressant egenskap då man enkelt kan spela upp och återskapa meddelanden och konsumera dem igen. KAFKA klarar dessutom ofta ett högre genomflöde än traditionella meddelandehanterare. En anledning till detta är att man inte håller reda på vem eller när meddelanden konsumerats, utan det är upp till mottagaren, dvs konsumenten att hålla reda på det. Därför används KAFKA ofta tillsammans med Storm, HBase och Spark för realtidsanalys och för att rendera data.



STORM: Användes som streamingprocessor av JSON data, hantering av eventuella duplikat i meddelanden från Kafka och för att spara informationen i Hadoop / HDFS. STORM används generellt för att processa stora mängder data, lite av vad Hadoop är för batch. Den är extremt snabb med möjlighet att processa miljoner rader per sekund och node.

Hadoop: Kraftfull open source programvara till låg kostnad. Hadoop erbjuder tillförlitlig, skalbar och distribuerad databehandling på allt ifrån en enda server till tusentals datorer. Det passade utmärkt i projektet då man med Hadoop kan bearbeta stora, röriga datauppsättningar för att ge inblickar och svar, vilket bidrar till att klargöra otydlig information.

Spark: Vi behövde processing av JSON meddelanden för Dynamisk ELT till Hive och lät detta ske via Spark. Spark är utmärkt för att snabbt kunna processa stora datamängder, det kör i minnet med omfattande utvecklings APIer, låter utvecklare exekvera streaming, maskinlärande eller SQL workloads som kräver snabb iterativ access till dataset.

Hive: Hive valdes som den primära SQL databasen för analys och aggregeringar av data.

- **HBase** - NoSql databas som är kolumnbaserad i stora tabeller. Den kallas även "the Hadoop database".
- **Phoenix** är som ett tillägg ovanpå HBase och levereras med JDBC drivers. Med hjälp av denna tas kundens SQL fråga och kompileras till ett antal HBase scans och ser till att dessa scans producerar ett JDBC resultat.
- Sista steget var att aggregera på en högre nivå och samla allt detta data i ett centralt Hadoopkluster för bland annat Self Service BI. Analytiker och utvecklare arbetar sedan direkt mot denna databas.

Summering

Genom Self service BI lyckades vi identifiera och snabbt analysera konsekvenser av ändringar i spelet. Analytiker och utvecklare kunde använda Self Service BI som ett verktyg för att öka kvalitén och användarvänligheten i spelen och det nu gick att utföra A/B tester för att hitta den bästa funktionaliteten för spelarna. Precis som förutsatt klarade systemet vid projektets slut av 150.000 events i sekunden i genomsnitt world wide.

Det var både ett spännande och mycket lyckat projekt där vi nådde våra målsättningar.