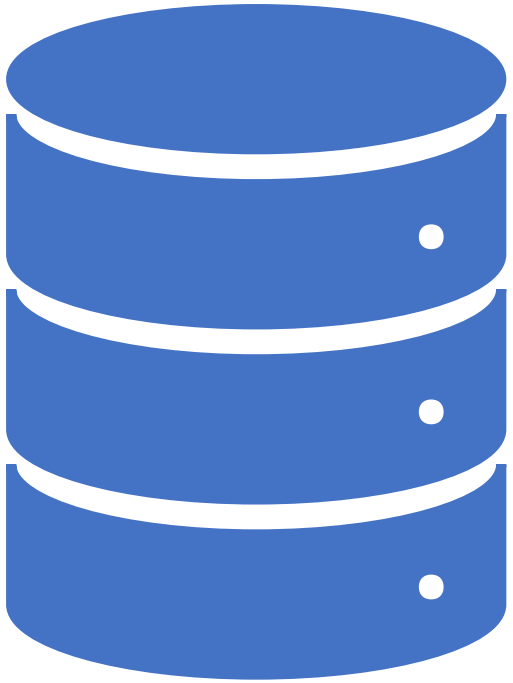


Arhitecturi Big Data

Conf.dr. Cristian KEVORCHIAN

Facultatea de Matematică și Informatică

Universitatea din București

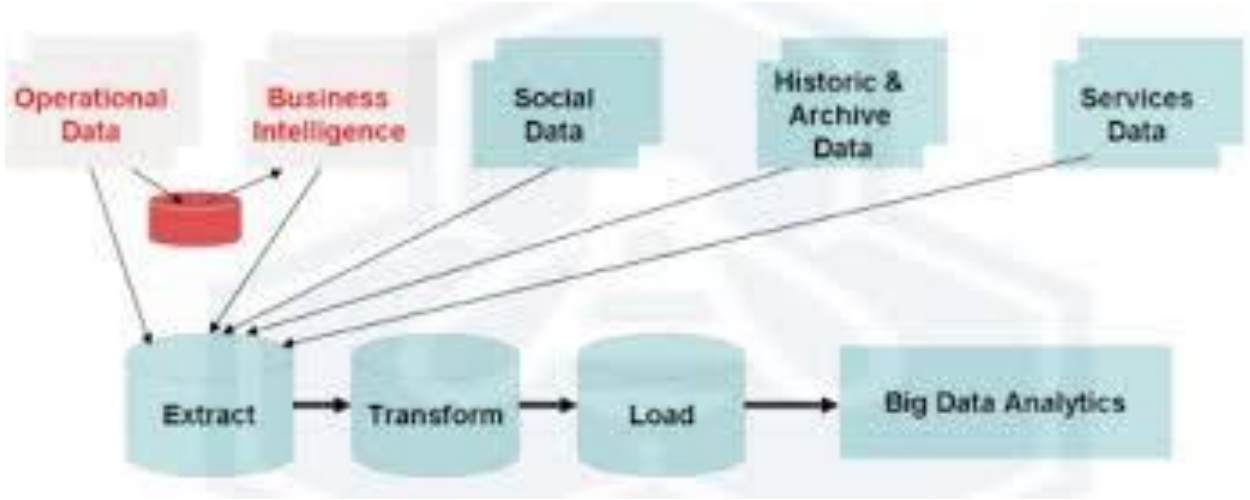
BIG DATA

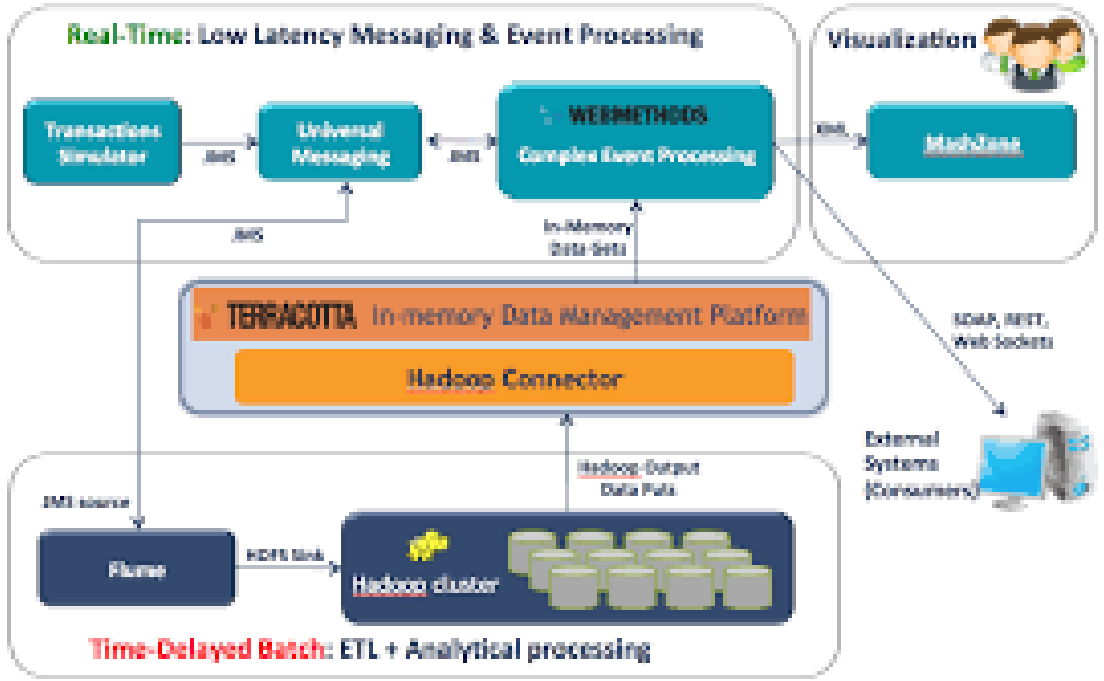
- Termenul “big data” caracterizează seturi de date de dimensiuni mari și complexitate ridicată pentru care aplicațiile tradiționale de procesare și analiză a acestora sunt inadecvate
- Încadrarea datelor în categoria ”Big Data” poate fi făcută de la sute de Gb de date la zeci de petabyte.
- Din ce în ce mai mult, acest termen vizează în principal valoarea de business generată de analize complexe, decât volumul și complexitatea datelor utilizate.
- Costul stocării a scăzut dramatic, în timp ce mijloacele prin care sunt colectate datele continuă să se diversifice.
- Unele date sunt colectate și prelucrate în timp-real, iar altele sunt prelucrate lent ca date istorice.

Soluții Big Data :

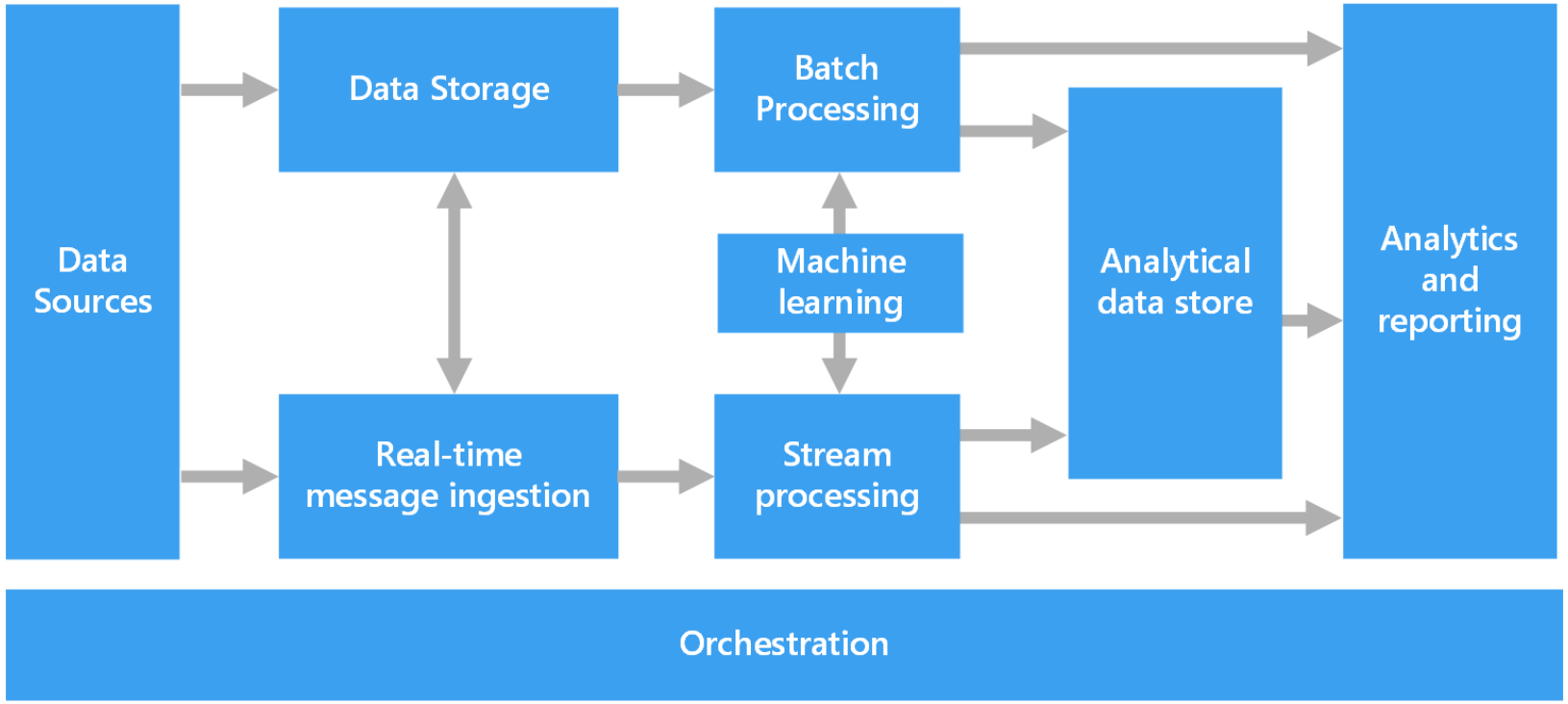
- Soluțiile Big Data implică una sau mai multe sarcini din următoarele:
  - Procesarea batch(serială) a volumelor mari de date statice .
  - Procesarea în timp-real a volumelor mari de date dinamice.
  - Explorarea interactivă a volumelor mari de date.
  - Analiza predictivă și machine learning.
- Trecerea la arhitecturi Big Data este necesară când:
  - Stochează și procesează date în volume prea mari pentru SGBD clasice
  - Transformă date nestructurate pentru analiză și raportare
  - Preia, procesează și analizează nelimitate stream-uri de date în timp real cu latențe neglijabile.

Batch vs Real-Time





Componentele unei arhitecturi Big Data



Sursa: https://docs.microsoft.com/en-us/azure/ architecture /data-guide/big-data/

Cele mai multe arhitecturi Big Data includ

- **Surse de date**: Toate soluțiile big data includ una sau mai multe surse de date:
  - Aplicații ”data store”, cum ar fi bazele de date relaționnale.
  - Fișiere statice produse de aplicații cum ar fi fisiere log ale servere-lor web.
  - Surse de date în timp-real, cum ar fi echipamentele IoT.
- **Stocarea datelor**:Datele asociate operațiilor de procesare batch sunt stocate prin intermediul unor sisteme de fișiere distribuite care pot cuprinde volume mari de date prezentate în diferite formate. O asemenea formă de stocare poartă numele de **data lake.** Aceste forme de implementare includ **Azure Data Lake Store** și containerele de BLOB-uri din **Azure Storage**

Procesarea datelor în sisteme Big Data

- **Procesare batch** – Datorită volumului mare de date, de multe ori o soluție Big Data trebuie să proceseze fișiere de date necesare execuției job-urilor batch, costisitoare din perspectiva timpului, pentru filtrare, agregare și pregătire pentru analiză. Opțiunile includ lansarea de job-uri **U-SQL** în **Azure Data Lake Analytics**, utilizând **Hive**, **Pig** sau job-uri **Map/Reduce** în clusterul **HDInsight** sau **Java**, Scala sau Python in clusterul HDInsight Spark
- **Preluarea în timp-real a mesajelor** – Daca soluția include surse de date utilizate în timp-real, arhitectura trebuie să includă o soluție de preluare și stocare în timp-real a mesajelor pentru procesarea stream-urilor. Acest segment include opțiuni cum ar fi **Azure Event Hub**, **Azure IoT Hub** și **Kafka**.

Analiza Datelor Stocate

- Multe soluții în Big Data implică pregătirea anterioară a datelor pentru analiză și apoi transferul acestora către prelucrare într-un format structurat care poate fi interogat utilizând instrumente analitice.
- Datele analitice folosite pentru a răspunde acestor întrebări pot fi prezentate ca un **model de date Kimball** asociat unui data warehouse relațional, așa cum se procedeaza în cele mai multe soluții de business intelligence (BI) tradiționale.
- Alternativ, datele ar putea fi prezentate printr-o tehnologie NoSQL cu latență redusă, cum ar fi **HBase**, sau o bază de date interactivă **Hive** care oferă o abstractizare a metadatelor peste fișierele de date din depozitul de date distribuite.
- Azure SQL Data Warehouse oferă un serviciu pentru stocarea date bazat pe cloud. **HDInsight** acceptă interactiv Hive, HBase și Spark SQL, care pot fi, de asemenea, folosite pentru a oferi date spre analiză.

Procesarea Fluxurilor de Mesaje

- După captarea mesajelor în timp-real, soluția arhitecturală trebuie să le includă o formă de procesare prin: filtrare, agregare dar și alte variante de pregătire a datelor pentru analiză.
- Datele procesate sunt apoi sunt scrise într-un canal de ieșire.
- Azure Stream Analytics oferă un serviciu de procesare al fluxurilor de mesaje bazat pe interogări SQL care rulează permanent peste fluxuri nelimitate. De asemenea, pot fi utilizate tehnologii de streaming cum ar fi Storm și Spark Streaming într-un cluster HDInsight.

Aspecte privind Compatibilitatea

Azure Service Bus ~= RabbitMQ(Pivotal, 2013)

Azure Event Hubs ~= Apache Kafka

- Ahitectura Apache Kafka este foarte diferită de Azure Service Bus. În general, dacă performanța efectivă este un **must**, atunci Kafka este soluția potrivită.
- Dacă se dorește un serviciu managed echivalent dar nu asa sofisticat, atunci Azure Event Hub sau Amazon Kinesis pot fi opțiuni de luat în considerație.

Analiza și Raportarea

- Scopul soluțiilor software în Big Data este de a oferi informații prin intermediul analizelor și a realiza livrabile sub forma unor rapoarte complexe.
- Pentru a permite utilizatorilor să analizeze datele, arhitecturile Big Data include o componentă de modelare a datelor, sub forma unui cub multidimensional OLAP sau a unui model tabular de date în **Azure Analysis Services**.
- De asemenea, democratizarea BI-ul prin utilizarea tehnologiilor de modelare și vizualizare din Power BI sau Microsoft Excel facilitează efectuarea de analize și generarea de raporte de către specialiștii în data science și analiștii de business.
- Pentru aceste scenarii, multe servicii Azure suportă notebook-uri analitice, cum ar fi **Jupyter**, care le permit acestor utilizatori să-și folosească abilitățile existente cu **Python** sau **R**. Pentru explorarea de date pe scară largă, puteți utiliza Microsoft R Server, independent sau cu Spark.

Servicii Azure pentru arhitecturi Big Data

- Azure include multe servicii care pot fi utilizate într-o arhitectură Big Data. Acestea se încadrează în două categorii:
  1. Serviciile managed: Azure Data Lake Store, Azure Data Lake Analytics, Azure Data Warehouse, Azure Stream Analytics, Azure Event Hub, Azure IoT Hub și Azure Data Factory.
  2. Tehnologiile open source bazate pe platforma Apache Hadoop, inclusiv HDFS, HBase, Hive, Pig, Spark, Storm, Oozie, Sqoop și Kafka. Aceste tehnologii sunt disponibile pe Azure în serviciul Azure HDInsight.
- Aceste opțiuni nu se exclud reciproc, iar multe soluții combină tehnologiile open source cu serviciile Azure.

Orchestrarea

- Cele mai multe soluții relative la ciclul de viață al datelor includ:
  - operații repetitive de procesare a datelor
  - includerea acestora în fluxuri de lucru în scopul transformării datelor sursă
  - transferarea datelor către diverse destinații de utilizare,
  - încărcarea datele procesate în data warehouse(Kimball) sau
  - Transferare rezultatelor direct în rapoarte sau grafice .
- Pentru a automatiza aceste fluxuri de lucru, puteți utiliza o tehnologie de orchestrare precum Azure Data Factory sau Apache Oozie și Sqoop

Tehnologie Lambda-Generalități

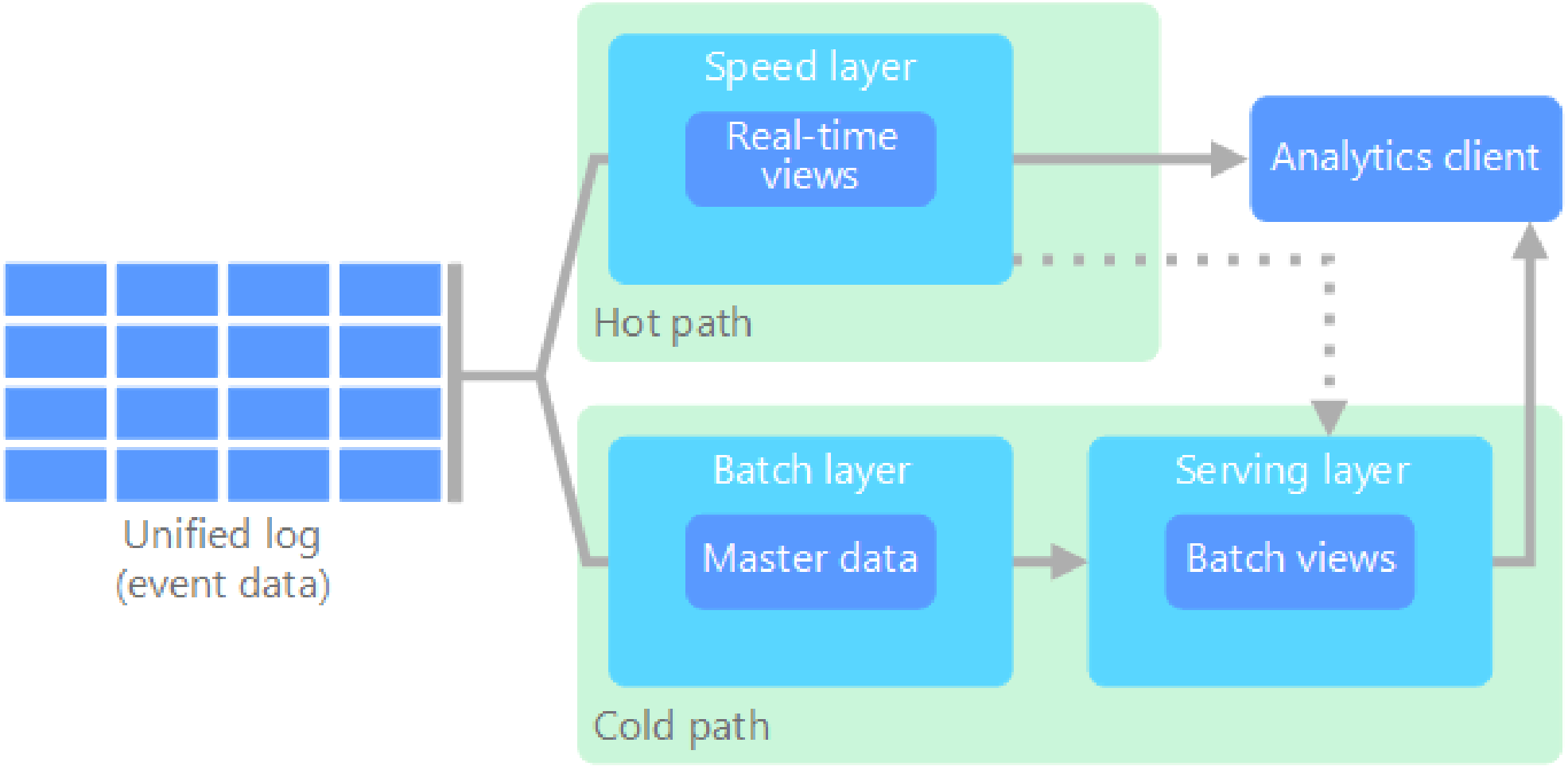
- Când operăm volume foarte mari de date, timpul de interogare poate fi foarte lung și inadecvat proceselor de business.
- Aceste interogări nu pot fi efectuate în timp real și necesită deseori tehnici de procesare paralelă cum ar fi MapReduce.
- Rezultatele sunt stocate separat de datele brute și folosite pentru interogare.
- Dezavantajul acestei abordări este faptul că introduce latență - dacă procesarea durează câteva ore, o interogare poate să întoarcă rezultate după câteva ore.
- Ideal, dorim rezultate în timp real (chiar cu o anumită pierdere de precizie) și combinăm aceste rezultate cu rezultatele din analizele istorice(batch).

Arhitecturi Lambda-Definiție

O lambda-arhitectură este o denumire generică pentru o arhitectură scalabila, tolerantă-la-erori destinată procesării datelor în scenarii de prelucrare bach cu latențe reduse. (Nathan Marz, http://lambda-architecture.net)

- Batch layer(cale rece) - stochează toate datele primite în forma lor brută și efectuează prelucrarea loturilor de date. Rezultatul acestei procesări este stocat ca un **batch view**.
- Speed layer(calea fierbinte) - analizează datele în timp real. Acest nivel al arhitecturii este proiectat pentru o latență scăzută, în detrimentul preciziei.

Arhitectura Lambda



Sursa: https://docs.microsoft.com/en-us/azure/architecture/data-guide/big-data/

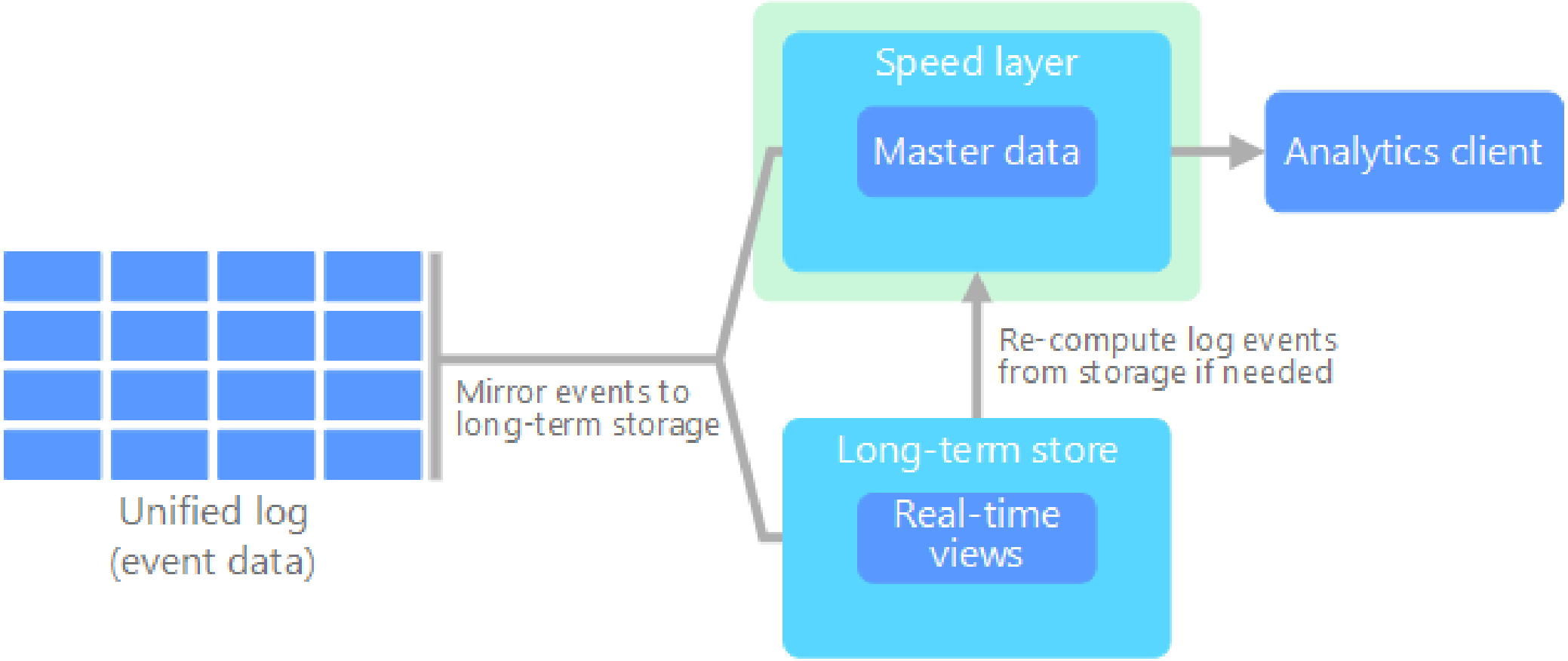
- Datele brute stocate prin ”batch layer”sunt imuabile. Datele primite sunt întotdeauna adăugate la cele existente, iar datele anterioare nu sunt suprascrise niciodată.
- Orice modificare a valorii unei date este stocată ca o nouă înregistrare a cu amprentă-de-timp. Acest lucru permite o recompunere în orice moment al ciclului de viață al datelor colectate.
- Abilitatea de a recalcula orice ”batch view” din datele originale este foarte importantă, deoarece permite crearea unor sisteme ”batch view” în evoluție.

- **Batch layer** se alimentează într-un **Serving Layer** care indexează **Batch View** pentru o interogare eficientă. **Speed Layer** actualizează **Serving Layer** cu actualizări incrementale pe baza celor mai recente date.
- Datele preluate de ”hot path” sunt restricționate de un anumit nivel al latenței impus de **Speed Layer** asa încât să poată fi prelucrate cât mai repede cu putință.
- Uneori este necesar un compromis între un anumit nivel de acuratețe a datelor în favoarea prelucrării cât mai rapide
- Datele preluate în ”cold path”, nu sunt supuse acelorași cerințe de latență redusă. Acest lucru permite calcularea cu precizii superioare a unor volume mari de date consumatoare de timp de procesare.

Arhitecturile Kappa

- Un dezavantaj al arhitecturii lambda este complexitatea sa ridicată.
- Logica de procesare apare la două ninele diferite – ”cold path” și ”hot path” - folosind diferite frameworkuri. Aceasta duce la dubla implementare a logicii de calcul dar și a complexitatea managementului arhitecturii pe ambele căi.
- Arhitectura **kappa** a fost propusă de **Jay Kreps** ca o alternativă la arhitectura lambda. Are aceleași obiective de bază ca arhitectura lambda, dar cu o distincție importantă: toate fluxurile de date se dirijază printr-o singură cale, folosind un sistem de procesare a fluxului.

Arhitectura Kappa



Kappa

- Există unele asemănări cu ”batch layer” al arhitecturii lambda, prin faptul că datele asociate evenimentelor captate sunt imuabile.
- Datele sunt preluate ca flux-uri de evenimente într-un log unificat distribuit și tolerant la erori. Aceste evenimente sunt ordonate, iar starea curentă a unui eveniment este modificată numai de un nou eveniment care este atașat.
- Similar cu speed layer-ul arhitecturii lambda, toate procesările de evenimente sunt efectuate pe fluxul de intrare și persistent ca vizualizare în timp real.

Azure Service Bus

- Azure Service Bus este un sistem generic de mesagerie bazat pe cloud computing pentru conectarea la orice categorie de aplicații, servicii și dispozitive oriunde s-ar afla.
- Datorită faptului că este un sistem de mesagerie care se integrează cu arhitecturi multi-nivel;
- În acest context arhitectura pe mai multe nivele se limitează la sensul clasic în care separarea este în cea mai mare parte de natură logică, dar și foarte utilă acolo unde separarea este și fizică.
- Este util să se genereze arhitecturi scalabile.

Scenariu

- Dacă sistemul proiectat se instalează local într-o regiune, de ex. Australia de Est și presupunem că numărul utilizatorilor înregistrează o rată de creștere foarte mare distribuiți în Oceania și Europa.
- Pentru a oferi performanță bună utilizatorilor, este de preferat ca implementarea sistemului să fie realizată în centrele de date din Europe & Australia de Est.
- Cu toate acestea, pot apărea probleme de cosistență a sistemului de baze de date. Ne putem baza pe funcția de replicare a SQL Server cu toate că acestea se pot dovedi a fi scumpe.

- În acest scenariu Service Bus, ca system de messaging între aplicații și servicii, poate fi de un real sprijin.
- Dacă plasăm funcția Service Bus + Role Worker între nivelul Service / API și nivelele de acces / bază de date; apoi în loc să se actualizeze direct baza de date, informațiile sunt transmise pe magistrala de servicii.
- Worker Role va prelua mesajul din coada de mesaje și va realiza atât actualizarea datelor din baza de date a regiunii respective (de exemplu, Europa), cât și dirijarea aceluiași mesaj în Service Bus-ul celeilalte regiuni (de exemplu, Australia de Est) - al cărui Worker Rol va prelua acest mesaj și va actualiza baza de date a regiunii Australia de Est.

Azure Cloud Service

- Azure Cloud Services reprezintă o resursă Azure clasică, inițial introdusă în cloud-ul celor de la Microsoft în 2008.
- Scopul tehnologiei a fost acela de a face posibilă scalabilitatea aplicațiilor web dar și de execuția aplicațiilor din categoria ”worker role” pe Windows.
- În timp ce Azure a avansat cu noile tehnologii de scalare pentru VM, serviciile clasice de tip ACS rămân în continuare o variantă de deloyment pentru vechile medii de lucru din Azure.

”Web Role” și ”Worker Role”

- Web Role este un rol Cloud Service in Azure care este configurat si adaptat sa ruleze aplicatii web dezvoltate în limbaje/tehnologii de programare care sunt suportate de Internet Information Services (IIS), cum ar fi ASP.NET, PHP, WCF(Windows Communication Foundation) și Fast CGI.
- Worker Role este un rol in Azure care execută task-uri la nivel de aplicații și servicii, care în general nu cer IIS. In Worker Role, IIS nu este instalat implicit. Ele sunt utilizate în principal pentru a efectua procese în background suportând împreună cu rolurile web efectuarea de task-uri, cum ar fi comprimarea automată a imaginilor încărcate, rularea scripturilor atunci când se schimbă ceva în bază de date, obținerea de mesaje noi din coadă și procese și multe altele.

Diferențe între ”Web Role” și ”Worker Role”

- Principala diferență dintre cele două constă în următoarele:
  - un ”Web Role” instalează automat și găzduiește aplicația utilizator în IIS
  - un ”Worker Role” nu utilizează IIS și rulează aplicația în mod autonom, aceasta fiind instalată și livrată prin intermediul platformei de **Servicii Azure,** ambele pot fi gestionate în același mod și pot fi instalate pe aceeași instanță Azure.
- În majoritatea scenariilor, instanțele **Web Role** și **Worker Role** lucrează împreună și sunt adesea folosite de o aplicație simultan. De exemplu, o instanță de Web Role ar putea accepta solicitări din partea utilizatorilor, apoi le transmită unei instanțe de Worker Role pentru procesare.

Web Role și Worker Role