

Wprowadzenie do Big Data

Właściwości Big Data

Big Data charakteryzuje się pięcioma kluczowymi "V", które opisują główne cechy tych złożonych zbiorów danych.

Objętość (Volume)

Określa wielkość generowanych i przechowywanych danych.

Przykład danych:

- Archiwa danych (opinie) z mediów społecznościowych, takie jak tweety, posty na Facebooku.
- Dane pogodowe

Wymaga skutecznego przechowywania i przetwarzania na dużą skalę.

Różnorodność (Variety)

Określa typy i formaty danych, zarówno strukturalnych, jak i niestructuralnych.

Przykład danych:

- Tekst
- Obrazy
- Wideo
- dane z sensorów
- Dane pacjentów w formie tabelarycznej, obrazów, notatki lekarskie
- Dane genetyczne
- logi transakcji

Wymaga zaawansowanych technik integracji i przetwarzania, aby móc korzystać z różnych typów danych.

Prędkość (Velocity)

Określa szybkość, z jaką dane są generowane i muszą być przetwarzane.

Przykład danych:

- dane z czujników IoT przesyłane w czasie rzeczywistym
- systemy zarządzania ruchem drogowym analizują dane z kamer, czujników na drogach, GPS w pojazdach
- dane z giełdy

Wymaga technologii przetwarzania strumieniowego i możliwości analizy w czasie rzeczywistym.

Wartość (Value)

Określa użyteczność i wartość informacji, jakie można uzyskać z danych.

Przykład danych:

- Informacje pozyskane z analizy dużych zbiorów danych klientów, które pomagają w personalizacji oferty.

Dane same w sobie nie są wartościowe, dopóki nie zostaną przetworzone i zanalizowane w celu uzyskania przydatnych wglądów.

Wiarygodność (Veracity)

Określa niepewność związana z danymi, w tym ich dokładność i wiarygodność.

Przykład danych:

- Dane pochodzące z niezetelnych źródeł mediów społecznościowych
- Banki i instytucje finansowe analizują transakcje w czasie rzeczywistym, aby wykryć oszustwa (Fraud detection).
- Fałszywe alarmy, niekompletne informacje.

Wymaga mechanizmów weryfikacji i oceny jakości danych, aby zapewnić dokładność analiz.

Systemy Rozproszone

Systemy rozproszone to grupa komputerów, które współpracują ze sobą, aby zapewnić wspólną funkcjonalność lub usługę.

Skalowalność

Zdolność systemu do obsługi rosnącej ilości pracy poprzez dodawanie zasobów.

- Pozioma - dodawanie więcej maszyn
- Pionowa - dodawanie większych maszyn
- Autoscaling?

Tolerancja na błędy

Właściwość systemu pozwalająca na kontynuację działania bez przerwy w przypadku awarii jednego lub więcej jego komponentów.

Strategie zapewniania tolerancji na błędy:

- Redundancja
- Replikację danych

Modele spójności

Opisują sposób, w jaki system zarządza aktualizacjami danych, aby zapewnić spójność informacji.

Rozróżniamy spójność:

- silną,
- słabą,
- ostateczną

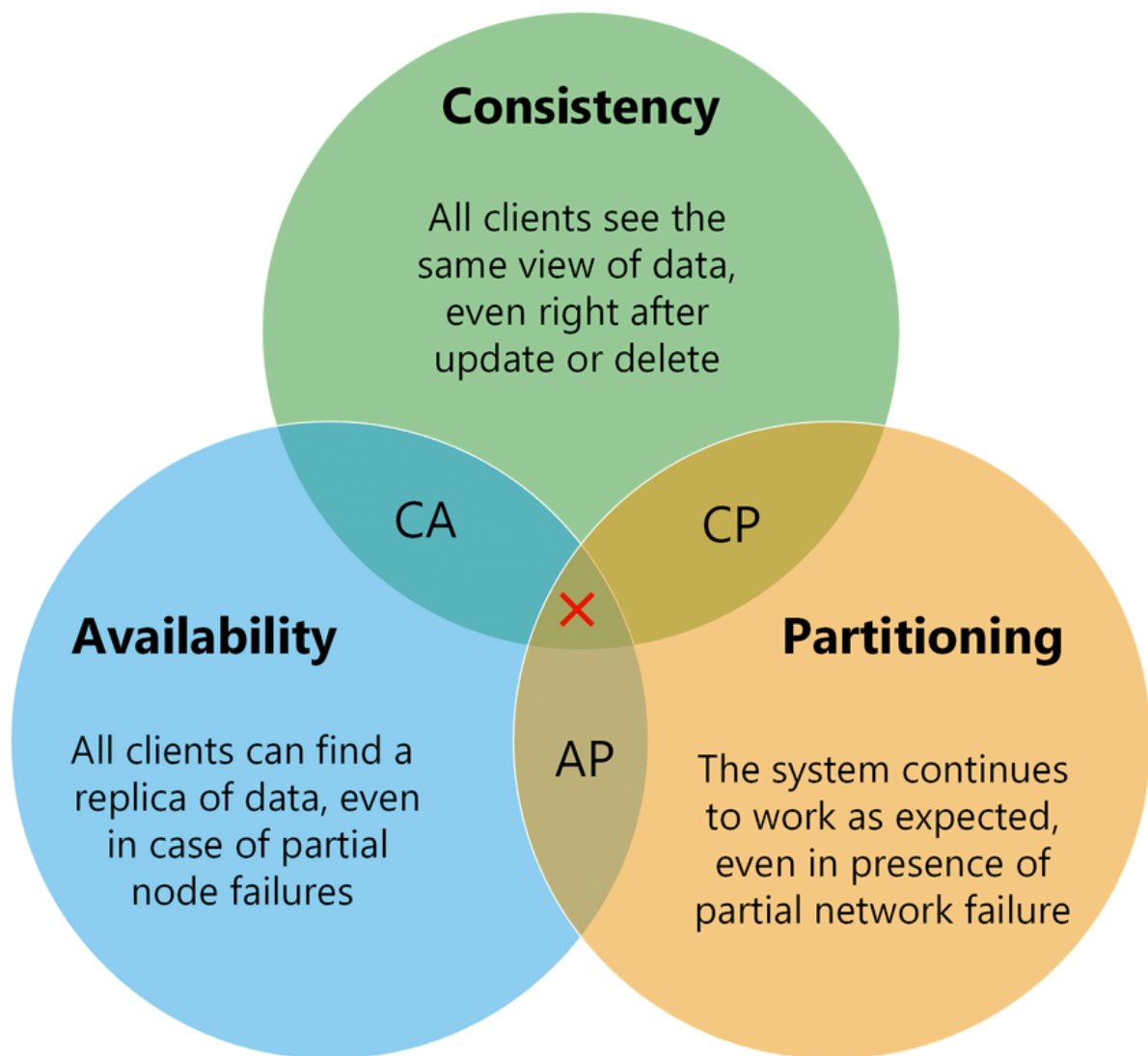
Wpływ na wydajność i skalowalność systemów?

CAP Theory

Teoria **CAP**, znana także jako Twierdzenie CAP, opisuje trzy podstawowe gwarancje, które mogą być zapewnione przez systemy rozproszone:

- Consistency (Spójność)
- Availability (Dostępność)
- Partition Tolerance (Tolerancja na podział)

Teoria stwierdza, że system rozproszony może zapewnić co najwyżej dwie z tych trzech gwarancji jednocześnie.



Rozumienie teorii CAP jest przydatne przy projektowaniu i wyborze odpowiednich technologii dla systemów rozproszonych, szczególnie w kontekście wymagań biznesowych i technicznych aplikacji.

C - Consistency (Spójność)

Definicja

Wszystkie węzły w systemie rozproszonym widzą te same dane w tym samym czasie. Po zaktualizowaniu danych, każde zapytanie do systemu zwróci nową wartość.

W praktyce

Gdy aktualizujesz swój status na platformie społecznościowej, oczekujesz, że każdy, kto go zobaczy, zobaczy tę samą aktualizację niezależnie od tego, z którego serwera korzystają.

Wyzwania

Utrzymanie spójności może być trudne w dużych, rozproszonych systemach, szczególnie przy równoczesnym zapewnieniu wysokiej dostępności i tolerancji na podział.

A - Availability (Dostępność)

Definicja

Każde żądanie otrzymuje odpowiedź o sukcesie lub porażce, ale system nigdy nie ignoruje żądania. System jest zawsze gotowy do obsługi zapytań od użytkowników.

W praktyce

Nawet podczas aktualizacji oprogramowania serwera bazy danych sklep internetowy nadal przyjmuje zamówienia i pokazuje użytkownikom aktualne informacje o produktach.

Wyzwania

Zapewnienie ciągłej dostępności może być kosztowne lub technicznie trudne, zwłaszcza gdy systemy muszą być aktualizowane lub naprawiane.

P - Partition Tolerance (Tolerancja na podział)

Definicja

System nadal działa poprawnie, nawet jeśli występuje 'podział' w sieci, który uniemożliwia komunikację między dwoma fragmentami systemu.

W praktyce

W przypadku awarii sieci oddziałującej na część serwerów w klastrze baz danych, pozostałe serwery nadal mogą obsługiwać zapytania, nawet jeśli nie mogą komunikować się z serwerami dotkniętymi awarią.

Wyzwania

Projektowanie systemów tolerujących podziały często wymaga kompromisów w zakresie spójności lub dostępności, ponieważ trudno jest zapewnić wszystkie trzy cechy jednocześnie.

Przykłady

Baza danych SQL (ACID)

Skupia się na zapewnieniu spójności i dostępności, ale może napotkać trudności w utrzymaniu operacji w przypadku podziału sieci (P).

NoSQL (np. Cassandra)

Zaprojektowana do zapewnienia wysokiej dostępności i tolerancji na podział, często kosztem silnej spójności (C).

Systemy plików rozproszonych (np. HDFS)

Zaprojektowane do tolerancji na podział i spójności, mogą czasami poświęcić dostępność (np. w skrajnych warunkach sieciowych).

Przechowywanie danych

Rozwiązania przechowywania danych

NVMe SSD

Pamięci NVMe (Non-Volatile Memory Express) SSD wykorzystują magistralę PCIe do szybkiego transferu danych.

Cechy:

- Oferują znacznie wyższe prędkości odczytu i zapisu w porównaniu do SATA SSD, dzięki wykorzystaniu interfejsu PCIe
- Prędkość odczytu może sięgać nawet 3500 MB/s
- Zapisu – 3000 MB/s, znacznie przewyższając inne technologie przechowywania

SATA SSD

SATA SSD to powszechne dyski SSD używające interfejsu Serial ATA.

Cechy:

- Tańsze w produkcji niż NVMe,
- oferują dobre prędkości odczytu i zapisu, lecz są ograniczone przez interfejs SATA.
- 550 MB/s odczyt
- 520 MB/s zapis

HDD (Dysk Twardy)

HDD, czyli dyski twarde, to tradycyjne urządzenia przechowywania danych z mechaniką.

Cechy:

- Oferują dużą pojemność przechowywania przy niższych kosztach niż SSD
- są wolniejsze
- bardziej podatne na uszkodzenia mechaniczne.
- Prędkości odczytu i zapisu znacznie niższe niż w SSD, zakresie 100-200 MB/s.

Przechowywanie danych w chmurze

Przechowywanie danych w chmurze polega na wykorzystaniu zasobów obliczeniowych dostawców chmury do przechowywania danych online.

Cechy:

- Elastyczność w skali,
- dostępność z dowolnego miejsca
- model płatności za faktyczne użycie

Ograniczenia:

- Zależność od szybkości i stabilności połączenia internetowego.
- Bezpieczeństwo danych
- Prywatność danych

Konfiguracje RAID i ich wpływ na prędkość

RAID (Redundant Array of Independent Disks) to technologia łączenia wielu dysków twardych w jeden logiczny system, aby zwiększyć niezawodność i/lub wydajność.

Istnieje kilka różnych konfiguracji RAID, z których każda ma inne wpływy na prędkość i odporność na awarie.

RAID 0 (striping)

- Dane są rozdzielane równomiernie między dwa lub więcej dysków
- zwiększa prędkość odczytu i zapisu przez zrównoleglenie operacji.
- Nie oferuje jednak redundancji danych - awaria 1 dysku powoduje utratę danych

RAID 1 (mirroring)

- Dane są dokładnie kopiowane na dwa lub więcej dysków.
- Zapewnia wysoką redundancję
- prędkość zapisu może być ograniczona do najsłabszego dysku w macierzy
- Prędkość odczytu może być jednak szybsza - system może czytać dane z wielu dysków jednocześnie.

RAID 5 (striping with parity)

- Dane i informacje parzystości są rozdzielane między trzy lub więcej dysków
- zrównoważone rozwiązanie między prędkością a redundancją
- pozwala na odczyt danych z wielu dysków jednocześnie
- pozwala na odzyskiwanie danych w przypadku awarii jednego dysku

RAID 6 (striping with double parity)

- Podobnie jak RAID 5, ale z dodatkową warstwą informacji parzystości
- umożliwia wytrzymanie awarii dwóch dysków
- zapewnia wyższą redundancję kosztem nieco mniejszej prędkości zapisu

HDFS (Hadoop Distributed File System)

HDFS to system plików zaprojektowany specjalnie do przechowywania dużych zbiorów danych w środowisku rozproszonym.

Charakteryzuje się wysoką odpornością na błędy, co czyni go idealnym rozwiązaniem dla aplikacji wymagających przetwarzania i analizy big data.

Rozproszenie danych

HDFS przechowuje dane poprzez rozdzielanie plików na bloki danych (domyślnie 128 MB lub 256 MB) i dystrybuowanie ich między wiele węzłów w klastrze, co umożliwia przetwarzanie danych równoległe na wielu maszynach.

Tolerancja na błędy

System automatycznie replikuje bloki danych na kilku maszynach, zapewniając wysoką dostępność i odporność na awarie poszczególnych węzłów.

Skalowalność

HDFS został zaprojektowany do pracy z tysiącami maszyn i petabajtami danych. Jest wysoce skalowalny, co pozwala na łatwe dodawanie kolejnych węzłów do klastra bez przerywania działania systemu.

Miary systemów przechowywania

Systemy przechowywania charakteryzują się kilkoma miarami, które pozwalają na lepsze zrozumienie i optymalizację wydajności systemów przechowywania danych w kontekście big data.

IOPS (Input/Output Operations Per Second)

IOPS to miara, która określa, ile operacji wejścia/wyjścia na sekundę może wykonać urządzenie przechowywania. Wysokie wartości IOPS wskazują na lepszą wydajność w zastosowaniach wymagających intensywnych operacji na danych, np. w bazach danych.

Latencja i jej wpływ na wydajność

Latencja to czas, który upływa od momentu wysłania żądania do nośnika danych do momentu otrzymania odpowiedzi.

Niska latencja jest kluczowa dla aplikacji czasu rzeczywistego i systemów wymagających szybkiej reakcji.

Przepustowość

Przepustowość, mierzona w MB/s lub GB/s, wskazuje, jaką ilość danych można przesyłać przez system w określonym czasie. Jest to szczególnie istotne dla operacji przenoszenia dużych ilości danych, np. w przetwarzaniu batchowym.

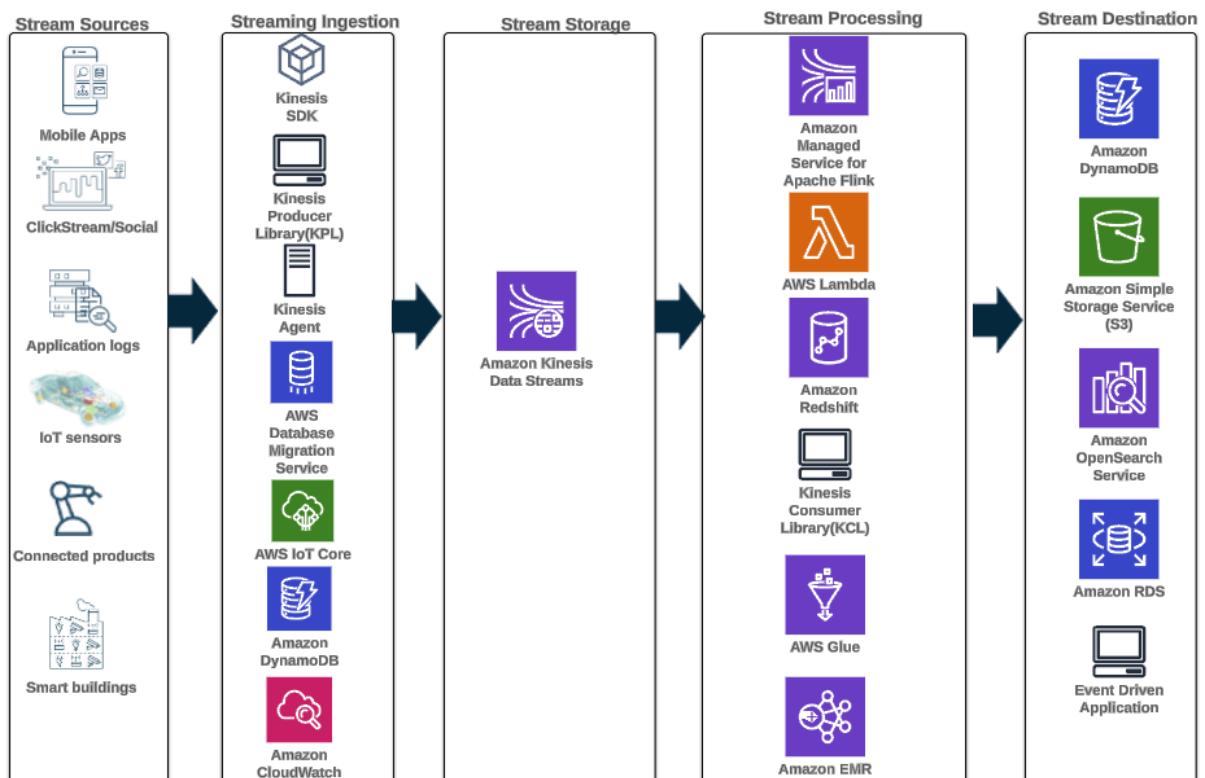
Trwałość pamięci masowych

Wytrzymałość pamięci odnosi się do całkowitej ilości danych, które mogą być zapisane na nośniku zanim jego wydajność zacznie spadać. Jest to szczególnie ważne w przypadku SSD, gdzie liczba cykli zapisu/odczytu jest ograniczona.

Skalowalność systemów przechowywania danych

Skalowalność to zdolność systemu do radzenia sobie ze zwiększającą się ilością danych. W kontekście big data, systemy przechowywania muszą być projektowane z myślą o łatwej rozbudowie, aby sprostać rosnącym wymaganiom przetwarzania danych.

Przetwarzanie Danych



Big Data Pipeline

Data Ingestion

Pierwszy etap polega na zbieraniu danych z różnorodnych źródeł, takich jak:

- logi serwerów
- media społecznościowe
- systemy IoT
- bazy danych
- pliki, itp.

Data Storage

Zebrane dane są przechowywane w odpowiednich systemach magazynujących.

Mogą to być systemy typu:

- Data Lake - przechowują surowe dane w różnych formatach
- Data Warehouse - przechowują dane już w pewnym stopniu przetworzone i strukturyzowane

Data Processing/Cleaning

Dane są przetwarzane i oczyszczane z błędów, duplikatów, czy niekompletności.

Używa się do tego celu narzędzi, które pozwalają na efektywne przetwarzanie dużych zbiorów danych:

- Apache Spark
- Apache Flink

Data Analysis

Przetworzone dane są analizowane za pomocą zaawansowanych algorytmów analitycznych i narzędzi do przetwarzania zapytań, takich jak:

- SQL na Hadoop
- Apache Hive
- Amazon Athena
- czy narzędzia do analizy strumieniowej danych

Data Visualization

Wyniki analizy są prezentowane użytkownikom końcowym za pomocą narzędzi do wizualizacji danych, takich jak:

- Tableau
- Power BI
- czy dashboards w Apache Superset.

Data Management & Security

Cały proces jest wspierany przez systemy zarządzania danymi, które odpowiadają za:

- bezpieczeństwo
- zarządzanie dostępem
- backup
- odzyskiwanie danych

Przetwarzanie ETL, ELT oraz reverse ETL

Procesy ETL, ELT oraz Reverse ETL różnią się głównie kolejnością i miejscem przetwarzania danych.

- ETL i ELT koncentrują się na przesyłaniu danych do repozytoriów analitycznych dla celów raportowania i analizy
- Reverse ETL skupia się natomiast na wykorzystywaniu tych zgromadzonych i przetworzonych danych do zasilania operacyjnych procesów.

ETL (Extract, Transform, Load)



Extract (Ekstrakcja)

- Dane są zbierane z różnych źródeł.

Transform (Transformacja)

- Dane są czyszczone
- Wzbogacane
- Transformowane do odpowiedniego formatu przed załadowaniem do systemu docelowego
- Transformacja odbywa się w oddzielnym obszarze pośrednim (takim jak obszar roboczy lub w pamięci).

Load (Ładowanie)

Po transformacji dane są ładowane do systemu docelowego, np. magazynu danych.

ELT (Extract, Load, Transform)



Extract (Ekstrakcja)

- Dane są zbierane z różnych źródeł.

Load (Ładowanie)

- Dane w surowej formie są bezpośrednio ładowane do systemu docelowego.

Transform (Transformacja)

- Transformacja danych jest wykonywana po załadowaniu
- Kluczowa moc systemu docelowego - nowoczesnego magazynu danych

Reverse ETL

Reverse ETL to proces, który może wydawać się mylący z nazwy, ponieważ sugeruje odwrócenie procesu ELT. Jednak w praktyce termin ten odnosi się do wykorzystania danych z magazynu danych lub innego systemu centralnego do zasilania innych operacyjnych systemów biznesowych lub aplikacji SaaS (Software as a Service).

Proces ten można opisać w następujący sposób:

Extract (Ekstrakcja)

- Dane są ekstrahowane z centralnego repozytorium, np. magazynu danych
- wcześniej zostały tam załadowane i przetworzone (może to być wynik procesu ELT).

Transform (Transformacja)

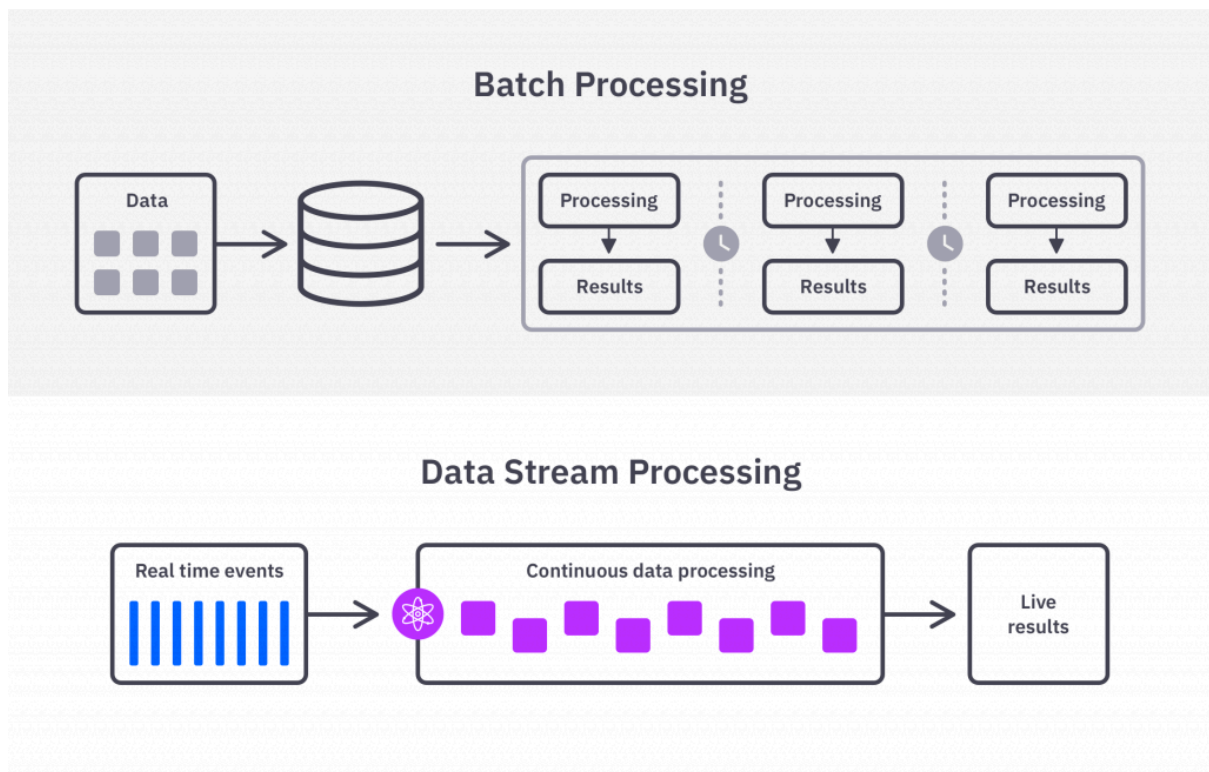
- Dane są transformowane lub przygotowywane do użycia w kontekście docelowym
- może to obejmować mapowanie danych na specyficzne schematy wymagane przez aplikacje docelowe.

Load (Ładowanie)

- Przetworzone dane są ładowane do zewnętrznych systemów operacyjnych lub aplikacji SaaS
- Wspierają procesy biznesowe lub analizy w czasie rzeczywistym.

Przetwarzanie Strumieniowe

Przetwarzanie strumieniowe danych umożliwia ciągle przetwarzanie dużych strumieni danych w czasie rzeczywistym lub niemal w czasie rzeczywistym.



Przetwarzanie w czasie rzeczywistym

Umożliwia analizę i przetwarzanie danych natychmiast po ich wygenerowaniu, bez opóźnień.

Skalowalność

Systemy przetwarzania strumieniowego są zaprojektowane do pracy z dużymi wolumenami danych, często w rozproszonych środowiskach, umożliwiając skalowanie w miarę wzrostu ilości danych.

Elastyczność

Mogą przetwarzać różne formaty danych, od prostych do złożonych, włączając w to dane niesktrukturizowane jak obrazy, dźwięk, oraz dane tekstowe.

Niskie opóźnienia

Systemy te charakteryzują się minimalnym czasem przetwarzania, co jest kluczowe dla aplikacji wymagających szybkiej reakcji, takich jak monitoring w czasie rzeczywistym czy analizy finansowe.

Trwałość

Zapewniają mechanizmy do obsługi błędów i utraty danych, takie jak replikacja i zapisywanie stanu, aby zapewnić ciągłość działania aplikacji.

Narzędzia Big Data do przetwarzania strumieniowego

Popularne narzędzia:

- Apache Kafka
- Apache Storm
- Apache Flink
- Apache Spark Streaming
- Amazon Kinesis
- Google Cloud Pub/Sub

Streams Make Customer Engagement Real-Time, Context-Aware and More Successful



Gwarancje przetwarzania wiadomości

W przetwarzaniu strumieniowym danych, fraza "at least once, exactly once, etc." odnosi się do gwarancji przetwarzania wiadomości w systemach przetwarzania strumieniowego.

Określają one, jak system radzi sobie z potencjalnymi błędami i zapewnia integralność danych podczas przetwarzania.

At Most Once (Co najwyżej raz)

Cechy:

- Każda wiadomość może być przetworzona co najwyżej raz.
- Jeśli wystąpi błąd, wiadomość może zostać utracona.

Zastosowanie:

- Ta gwarancja minimalizuje ryzyko duplikacji
- zwiększa ryzyko utraty danych
- Może być stosowana w systemach, gdzie szybkość przetwarzania jest ważniejsza niż integralność każdej pojedynczej wiadomości.

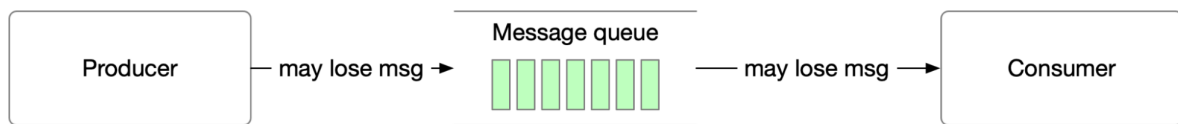


Figure 1 At-most once

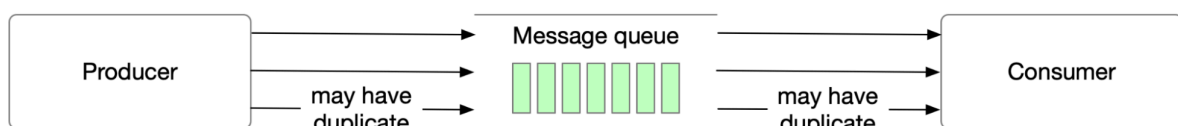


Figure 2 At-least once

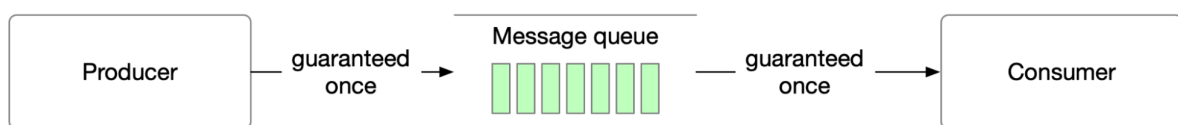


Figure 3 Exactly-once

At Least Once (Przynajmniej raz)

Cechy:

- Każda wiadomość jest przetwarzana co najmniej raz.
- W przypadku wystąpienia błędu, system ponowi próbę przetworzenia wiadomości, co może prowadzić do duplikacji.

Zastosowanie:

- Gwarantuje, że żadna wiadomość nie zostanie utracona
- może wprowadzić duplikaty, które muszą być obsłużone na etapie późniejszym.

Exactly Once (Dokładnie raz)

Cechy:

- Każda wiadomość jest przetwarzana dokładnie raz
- eliminuje ryzyko utraty lub duplikacji danych.

Zastosowanie:

- Jest to najbardziej pożądana gwarancja
- zwłaszcza w krytycznych dla biznesu aplikacjach, gdzie duplikaty lub utrata danych są nieakceptowalne
- Realizacja tej gwarancji jest jednak technicznie trudniejsza i może wymagać dodatkowych mechanizmów koordynacji i zarządzania stanem.

Systemy Baz Danych

Systemy Baz Danych SQL

Relacyjne

Bazy danych SQL są relacyjnymi bazami danych:

- dane są przechowywane w tabelach
- mogą być ze sobą powiązane za pomocą relacji

Schemat

Wymagają z góry zdefiniowanego schematu, który określa strukturę danych, w tym:

- tabele
- kolumny
- typy danych

Zapytania

Używają języka SQL (Structured Query Language) do:

- tworzenia
- pobierania
- Aktualizacji
- usuwania danych

co umożliwia wykonywanie skomplikowanych zapytań i analiz.

Transakcje ACID

Obsługują transakcje, co zapewnia spójność danych nawet w przypadku wystąpienia błędów lub przerwania operacji:

- Atomicity
- Consistency
- Isolation
- Durability

Przykłady:

- MySQL
- MariaDB
- PostgreSQL
- Oracle Database
- Microsoft SQL Server
- Sqlite

Systemy Baz Danych NoSQL

Nierelacyjne

Bazy danych NoSQL nie są zorganizowane jako tradycyjne relacyjne bazy danych, co pozwala na większą elastyczność w przechowywaniu różnorodnych typów danych.

Schemat

Zazwyczaj nie wymagają z góry zdefiniowanego schematu, co pozwala na łatwe przechowywanie danych o zmiennym formacie.

Typy

Bazy klucz-wartość

Przechowują dane jako kolekcje par klucz-wartość

Przykład:

- Redis
- Amazon DynamoDB

Bazy dokumentowe

Przechowują dane w formacie dokumentów, takich jak JSON, BSON.

Przykład:

- MongoDB
- Couchbase

Bazy kolumnowe

Optymalizowane do szybkiego odczytu i zapisu dużych zbiorów danych.

Przykład:

- Cassandra
- HBase

Bazy grafowe

Skoncentrowane na przechowywaniu relacji między obiektami.

Przykład:

- Neo4j
- Amazon Neptune
- TigerGraph

Skalowalność

NoSQL jest często bardziej skalowalny horyzontalnie, co oznacza, że można łatwo dodać więcej serwerów do obsługi większych ilości danych.

Zastosowania

Idealnie nadają się do dużych zbiorów danych, które nie są dobrze strukturyzowane lub których struktura często się zmienia.

Porównanie SQL i NoSQL

	SQL	NoSQL
Skalowalność	skalowany pionowo: kosztowne i trudniejsze w zarządzaniu	skalowany horyzontalnie: bardziej elastyczne i mniej kosztowne
Schemat	schema on write, stały	Schema on read, elastyczny
Zapytania	umożliwia wykonanie skomplikowanych zapytań	prostszy, może nie wspierać złożonych zapytań SQL
Transakcje	pełne wsparcie dla transakcji ACID	wsparcie transakcji może być ograniczone lub realizowane inaczej

Architektury Platform Danych

