Bazy danych

14. Bazy NoSQL

P. F. Góra

http://th-www.if.uj.edu.pl/zfs/gora/

2021

Bazy NoSQL:

Nierelacyjne bazy danych, zaprojektowane (między innymi) do tego, aby rozwiązywać problemy z dostępnością i spójnością jak najlepiej z punktu widzenia zakładanej funkcjonalności.

$$\text{NoSQL} \neq \text{Not SQL}$$
$$\text{NoSQL} = \text{Not Only SQL}$$

Copyright © 2018-21 P. F. Góra

Zasady BASE

Bazy OLTP podlegają zasadom ACID. Bazy NoSQL podlegają zasadom BASE:

Basically available: System zapewnia dostępność nawet w przypadku awarii (niedostepności) części węzłów.

Soft state: Stan systemu może się zmieniać w czasie, nawet jeśli system nie otrzymuje w tym czasie żadnych danych wejściowych. Dzieje się tak z uwagi na...

Eventual consistency: Węzły sieci wymieniają informacje o swoim stanie, w skutek czego, po dostatecznie długim czasie, system osiąga spójność, o ile w tym czasie system nie otrzymał żadnych nowych danych.

Copyright © 2018-21 P. F. Góra

Popularne typy baz NoSQL

Bazy klucz-wartość

1	‘Ala’
2	‘ma’
3	’kota’
4	’Ola’
5	’ma’
6	‘Asa’

Tablica — jedna z podstawowych struktur danych. Jednak kolumna to “indeks”, będący dowolnym typem całkowitoliczbowym, niekiedy wyliczeniowym. Indeksy nie mogą się powtarzać. Druga (i ewentualnie dalsze) kolumny tabeli przechowują “zawartość” tabeli. Każda kolumna ma ustalony typ, na ogół z góry określony jest rozmiar tabeli. Występuje naturalny porządek.

Copyright © 2018-21 P. F. Góra

1	’porucznik Columbo’
’Pi’	3.141569
33.2	‘to jest liczba’
15.0	3×5
Klient:Id	145603
Klient:NIP	‘677-112-42-43’
4	’Ola’
True	’prawda’

Tablica asocjacyjna — uogólnienie tablicy. Indeksem może być (niemalże) cokolwiek, typ “zawartości” nie musi być ustalony i może być zmienny. Indeksy nie mogą się powtarzać. Nie ma “naturalnego” porządku.

Copyright © 2018-21 P. F. Góra

Baza klucz-wartość jest wzorowana na tablicach asocjacyjnych: Każdemu unikalnemu obiektowi, który chcemy modelować (klientowi, pojazdowi, studentowi, ...) przypisujemy unikalny *klucz*, a raczej każdemu atrybutowi takiego obiektu przypisujemy unikalny klucz: `Klient1:Nazwa`, `Klient1:Adres`, `Klient1:NIP` itd, a następnie przechowujemy *pary* klucz-wartość, np `(Student1:Imie,'Alicja')`, `(Student1:Nazwisko, 'Kowalska')`, `(Student2:Imie,'Bogdan')`,`(Student2:Nazwisko,'Nowak')`. Nie musimy specyfikować typów wartości ani ile takich "atrybutów" będziemy przechowywać. Jest to najprostszy typ bazy NoSQL, nie obsługujący ani złączeń, ani innych operacji typowych dla SQL.

Copyright © 2018-21 P. F. Góra

Klucz musi być unikalny w przestrzeni nazw. “Przestrzeń nazw” można interpretować jako abstrakcyjną encję, a jej poszczególne wystąpienia jako wystąpienia tej encji. Możliwe są więc na przykład takie operacje (zakładam, że *jakiś* język programowania je obsługuje):

```
Klient:037:Nazwa = 'SpaceX'
Klient:224:Nazwa = 'Amazon'
```

Kiedy to się może przydać? Jeśli program ma przez pewien, na ogół z góry nieznany czas, przechowywać dane dotyczące różnych obiektów i manipulować nimi. Może to być na przykład zawartość przypisanego danemu klientowi koszyka w e-sklepie, dane adresowe kilku kontrahentów, któremu program ma wystawić faktury, dane katalogowe książek wypożyczanych z biblioteki itp,

Copyright © 2018-21 P. F. Góra

itp. Ważne, że *na potrzeby tego programu* nie będzie się przeprowadzać złączeń, skomplikowanych wyszukiwań i podobnych klasycznych operacji bazodanowych.

Baza klucz-wartość często jest zasilana z klasycznej bazy OLTP lub też dane są wprowadzane “ręcznie”, przez działający program, nie ma natomiast potrzeby zapisywania tych danych w bazie OLTP lub też spowolniłoby to działanie programu.

Jeśli pracujemy jednocześnie z wieloma przestrzeniami nazw (encjami), dla przyspieszenia działania programu, klucze czesto się haszuje, a następnie wyszukuje w tablicy haszy.

Copyright © 2018-21 P. F. Góra

Bazy dokumentów

Baza dokumentów — zapewne najbardziej popularny typ baz NoSQL. Podobna do baz klucz-wartość, ale “wartościami” są dokumenty, zapisane w formacie JSON, XML lub jakimkolwiek innym. Schemat każdej encji może być inny. Główną zaletą w stosunku do baz klucz-wartość jest to, że atrybuty w pewien sposób ze sobą powiązane są traktowane łącznie, jako jeden obiekt. W bazach klucz-wartość, dzięki stosowaniu znaczących* kluczy, *domyślamy się*, że wartości przyporządkowane kluczom “Klient:124:NIP” i “Klient:124:REGON” sa jakoś ze sobą związane, ale nic tego logicznie nie wymusza. W bazie domumentów odpowiednie wartości byłyby zapewne różnymi atrybutami tego samego obiektu typu JSON.

*W odróżnieniu od abstrakcyjnych.

Copyright © 2018-21 P. F. Góra

Bazy rodzin kolumn

Bazy rodzin kolumn, zwane także *wide column database* są bazami kolumnowymi: Dane przechowywane są w porządku kolumnowym, nie wierszowym. Wszystkie dane w jednej kolumnie są jednego typ. Kolumn może być bardzo wiele (setki, tysiące, niekiedy dziesiątki tysięcy), kolumna może liczyć sobie tysiące–setki tysięcy–miliony pozycji, ale niektóre (w ogólności: wiele) pozycje odpowiadające poszczególnym wierszom w danej kolumnie mogą być puste.

O bazie rodzin kolumn można myśleć jako o bardzo dużej, liczącej *wiele* wierszy i kolumn zdenormalizowanej tabeli, odpowiadającej na przykład wymiarowi w hurtowni danych.

Pierwowzorem baz rodzin kolumn była Google Big Table i do dziś wielkie firmy nowych technologii utrzymują swoje serwisy w opaciu o bazy rodzin kolumn.

Copyright © 2018-21 P. F. Góra

Bazy rodzin kolumn znajdują główne zastosowanie w sektorze VLDB, *Very Large DataBase*.

Zawartość baz kolumnowych indeksowana jest za pomocą nazwy kolumny, numeru (identyfikatora) wiersza i, bardzo często, *timestamp*.

Operacje kolumnowe są szybkie, ale operacje wierszowe, zwłaszcza obejmujące więcej, niż jeden wiersz, są wolne. Złączenia są praktycznie niemożliwe, dlatego bazy kolumnowe w pewnym sensie odpowiadają zdenormalizowanym tabelom baz relacyjnych.

Copyright © 2018-21 P. F. Góra

Bazy grafowe

Bazy grafowe służą do przechowywania informacji o grafach (węzłach i krawędziach), a więc modelują powiązania pomiędzy obiektami rzeczywistości, którą baza modeluje. Baza grafowa pozwala na *bezpośrednie* odwzorowywanie zależności pomiędzy encjami: wierzchołki są encjami (znowu, na przykład typu JSON) i są połączone krawędziami. Dzięki temu można przeprowadzać złożone wyszukiwania bez konieczności przeprowadzania złączeń.

W bazach SQL (relacyjnych) *daje się* reprezentować grafy: różne tabele współdzielą pewien atrybut (klucz obcy). Bazy SQL są jednak raczej nieefektywne przy reprezentowaniu złożonych grafów, zwłaszcza takich, które nie są drzewami.

Copyright © 2018-21 P. F. Góra

Poważne powody używania baz NoSQL

- Bardzo duże wolumeny danych.
- Bardzo duży spodziewany ruch.
- Zmienny, trudny do przewidzenia schemat poszczególnych “krotek”.

Niepoważne powody używania baz NoSQL

- Modny framework lansuje jakąś konkretną bazę NoSQL.

Jeżeli tworzymy bazę, w której będziemy przechowywać $\sim$ kilkanaście tysięcy “krotek” (lub mniej), o strukturze, którą znamy lub możemy łatwo przewidzieć, stosowanie baz NoSQL *jest błędem projektowym*. Jeżeli baza ma zapewniać transakcyjność lub chociażby spójność danych, baza SQL (relacyjna) **jest jedynym możliwymy wyborem**.

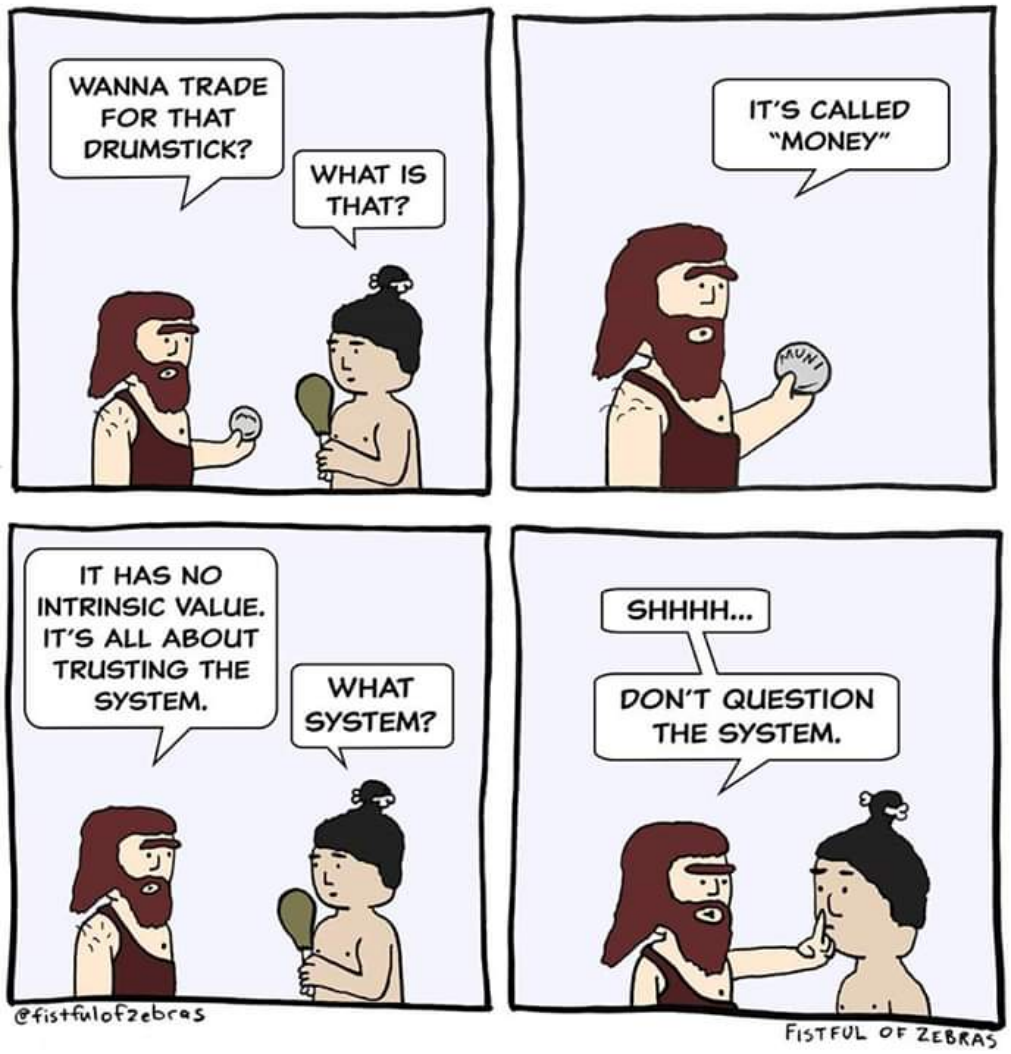
Copyright © 2018-21 P. F. Góra

Blockchain — motywacja

Dwa problemy z pieniądzem fiducjarnym:

1. *Double-spending:* jak zapewnić, że **te same** pieniądze nie zostaną wydane dwukrotnie?
2. *Zaufana trzecia strona:* Czy naprawdę możemy (i chcemy) zaufać, oddając przy okazji część kontroli?

Copyright © 2018-21 P. F. Góra



Copyright © 2018-21 P. F. Góra

Blockchain — historia

Blockchain i pierwsza kryptowaluta, Bitcoin, zostały zaproponowane w 2008 przez Satoshi Nakamoto. Zostały wprowadzone w życie w 2009.

Nikt nie wie, kim jest Satoshi Nakamoto ☺

Oryginalny artykuł Satoshi Nakamoto można przeczytać tutaj.

Copyright © 2018-21 P. F. Góra

Abstract

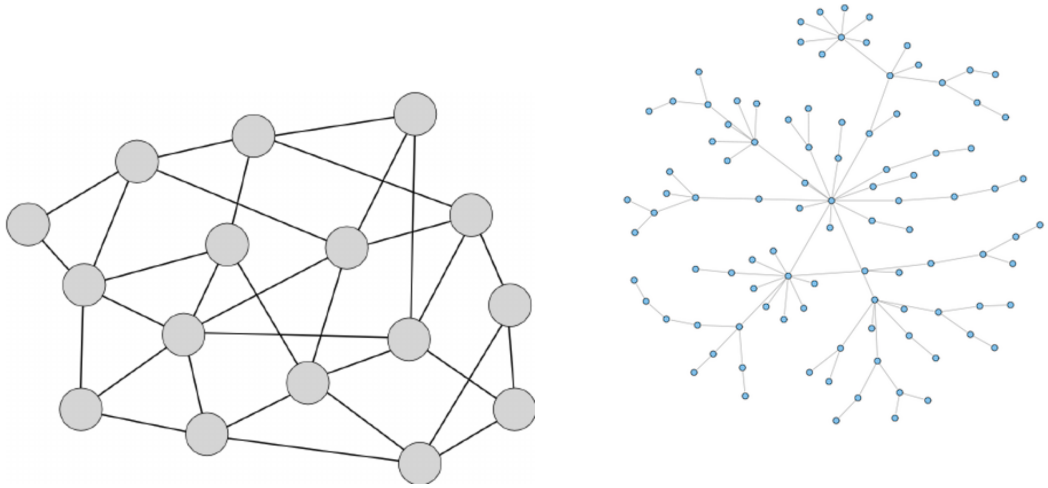
A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

Copyright © 2018-21 P. F. Góra

Z punktu widzenia baz danych. . .

Blockchain jest specjalistyczną, rozproszoną, zdecentralizowaną, nietransakcyjną bazą danych w modelu peer-to-peer (P2P). Blockchain to publiczny i jawny rejestr “transakcji” lub “operacji księgowych” (*public ledger*), zapewniający bezpieczeństwo i niezmienność raz zapisanych operacji na drodze kryptograficznej.

Copyright © 2018-21 P. F. Góra



W sieci P2P komputery są połączone i mogą współdzielić zasoby bez konieczności odwoływania się do zewnętrznego serwera.

Po lewej — idealna sieć P2P. Po prawej — sieć bezskalowa, w której widoczne centra (*hubs*) wyróżniają się ze względu na liczbę połączeń, ale nie ze względu na *formalną* rolę, jaką pełnią w sieci.

Copyright © 2018-21 P. F. Góra

Elementy blockchain

Transakcje oznaczają elementarne operacje obsługiwane przez dany blockchain. *Transakcją* może być wymiana jakichś tokenów — *na przykład* kryptowalut — w zamian za pewne inne dobra lub usługi, ale gdzie indziej transakcją może być zapisanie jakiegoś pliku, zrealizowanie inteligentnego kontraktu (*smart contract*), potwierdzenie operacji, która miała miejsce w świecie rzeczywistym, dokonanie zmian w pewnym rejestrze itp. Każda transakcja podpisywana jest za pomocą klucza publicznego. Transakcje są jawne, choć użytkownicy mogą pozostawać anonimowi. Węzły dokonujące transakcji informują o jej dokonaniu wszystkie węzły, z którymi są połączone. Każda poprawna transakcja zapisywana jest w buforze transakcji. Obsługa transakcji na ogół nie jest darmowa.

Copyright © 2018-21 P. F. Góra

Blok zawiera określoną liczbę transkacji **oraz hash** poprzedniego bloku, co w chwili tworzenia oznacza ostatni, najmłodszy blok z blockchain, a także *nonce*, czterobajtową, unikalną liczbę, której wartość ustala się w procesie zatwierdzania bloku.

Zatwierdzenie bloku oparte jest na rozwiązaniu pewnego problemu kryptograficznego. Węzeł zatwierdzający blok pobiera transakcje z bufora, tworzy nowy blok i próbuje rozwiązać problem kryptograficzny. W sieci Bitcoin jedynym znanym sposobem jest rozwiązanie siłowe, stąd wymaga to dużych (właściwie należałoby powiedzieć: **bardzo dużych**) mocy obliczeniowych. Jest to tak zwany *Proof of Work* (PoW). (Możliwe są także inne sposoby zatwierdzania nowych bloków, takie jak *Proof of Stake*. PoS nie wymaga tak

Copyright © 2018-21 P. F. Góra

potężnych mocy obliczeniowych, jak PoW. Niektóre “alternatywne” kryptowaluty oparte są o PoS.) Po rozwiązaniu problemu, węzeł, który tego dokonał, dołącza ten blok do swojej kopii blockchain i rozsyła ten blockchain do pozostałych węzłów sieci.

Górnicy (kopacze, *miners*) to wyspecjalizowane węzły zajmujące się zatwierdzaniem bloków. Specjalizacja jest potrzebna z uwagi na wymaganie posiadania wielkich mocy obliczeniowych. Górnicy mogą (chociaż nie muszą) otrzymywać wynagrodzenie za skuteczne zatwierdzenie każdego bloku.

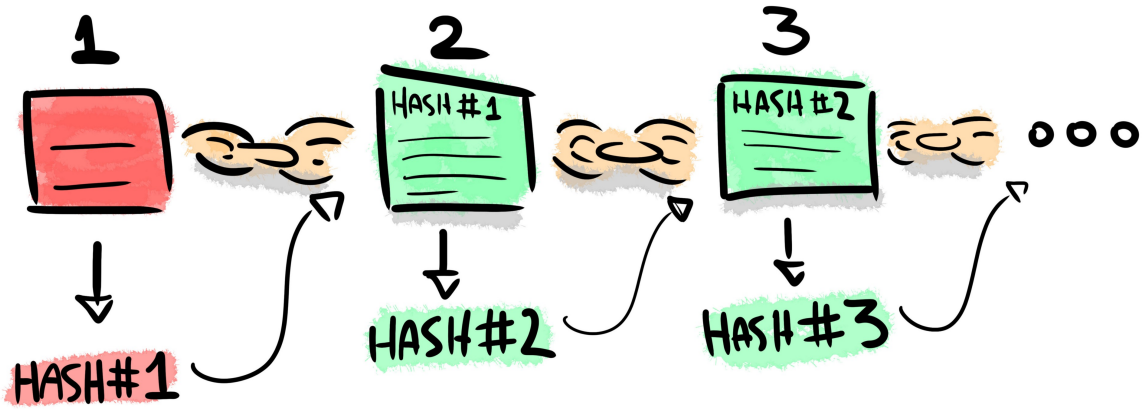
Dołączanie nowego bloku Każdy węzeł przechowuje swoją kopię całego rejestru (czyli łańcucha bloków, czyli blockchain). Węzeł, który otrzyma nowy

Copyright © 2018-21 P. F. Góra

łańcuch, po pierwsze sprawdza, czy najnowszy blok został utworzony poprawnie (sprawdzenie, w przeciwieństwie do samego dołączenia, jest łatwe i tanie), a następnie z dwu posiadanych egzemplarzy blockchain (dotychczasowy i nowootrzymany) wybiera *dłuższy*. Zatwierdzanie bloków przez górników, weryfikacja przez pozostałe węzły i zasada utrzymywania najdłuższego łańcucha stanowią mechanizm osiągania konsensusu w sieci blockchain.

Bloki osierocone (*orphaned block*). Może się zdarzyć, że na skutek opóźnień czasowych lub partycjonowania sieci blockchain zbifurkuje, rozszczepi się. Wówczas węzły będą akceptować dłuższą gałąź. Bloki z krótszej gałęzi zostają "osierocone". Są to poprawnie zweryfikowane bloki, ale ponieważ nie mogą zostać dołączone do blockchain, zawarte w nich transakcje wracają do bufora transakcji.

Copyright © 2018-21 P. F. Góra



Ponieważ hasz całego bloku zależy od zawartości bloku **oraz** od hasza do jego poprzednika, nie da się zmienić bloku (czyli sfałszować lub usunąć już zatwierdzonych transakcji) bez konieczności ponownego zatwierdzenia wszystkich następnych bloków w łańcuchu, co jest praktycznie niewykonalne w szybkim czasie z uwagi na monstrualną moc obliczeniową potrzebną do wykonania takiego zadania.

Copyright © 2018-21 P. F. Góra

Bitcoin: wynagrodzenie górników

W sieci Bitcoin górnicy za zatwierdzenie każdego bloku otrzymują wynagrodzenie: kreowane są **nowe** bitcoiny, któe otrzymuje górnik zatwierdzający nowy blok. Ponieważ z założenia nie może być więcej, niż 21 mln ₿, nagroda za zatwierdzenie bloku zmniejsza się o połowę po każdych wygenerowanych 210 000 blokach. Jeden blok generowany jest co około 10 minut.

- W 2009 nagroda za wygenerowanie bloku wynosiła 50₿.
- Od 2012 nagroda wynosiła 25₿.
- Od 2016 nagroda wynosi 12.5₿.
- Od 11 maja 2020 nagroda wynosi 6.25₿.
- Będzie jeszcze jedno obniżenie nagrody.

Copyright © 2018-21 P. F. Góra

- Po wygenerowaniu **wszystkich możliwych bitcoinów**, nowe bitcoiny przestaną być generowane, a górnicy będą wynagradzani z drobnych opłat transakcyjnych.

Aktualny stan tego procesu można śledzić na stronie `https://www.bitcoinblockhalf.com/` (warto ją co kilka minut odświeżyć).

1 ₿ jest obecnie wart ponad §31 000 (wcześniej w styczniu 2021 bitcoin osiągnął swój najwyższy kurs, przekraczając $33 000). Na początku 2019 bitcoin wart był około §3 000, a rok temu niespełna §9 000. Wydobywanie bitcoinów, które miało być rodzajem *community service*, z uwagi na swoją opłacalność stało się celem samym w sobie. Szacuje się, że obecnie około 60% bitcoinów w obiegu znajduje się w posiadaniu górników.

Szacuje się, że Satoshi Nakamoto — kimkolwiek jest — posiada pomiędzy 200 000 a 600 000₿.

Copyright © 2018-21 P. F. Góra

Z punktu widzenia twierdzenia CAP...

Blockchain jest systemem typu AP: Ma zapewniać dostępność i odporność na podział sieci kosztem spójności danych: Dopuszcza się pojawianie bifurkacji, eliminowanie których jest elementem osiągania konsensusu. Blockchain podlega zasadom BASE i gdyby zaprzestano dodawania nowych bloków (dokonywania nowych transakcji), osiągnąłby spójność po odpowiednio długim czasie. W praktyce większość działających blockchains jest bardzo bliska stanu spójności.

Copyright © 2018-21 P. F. Góra