

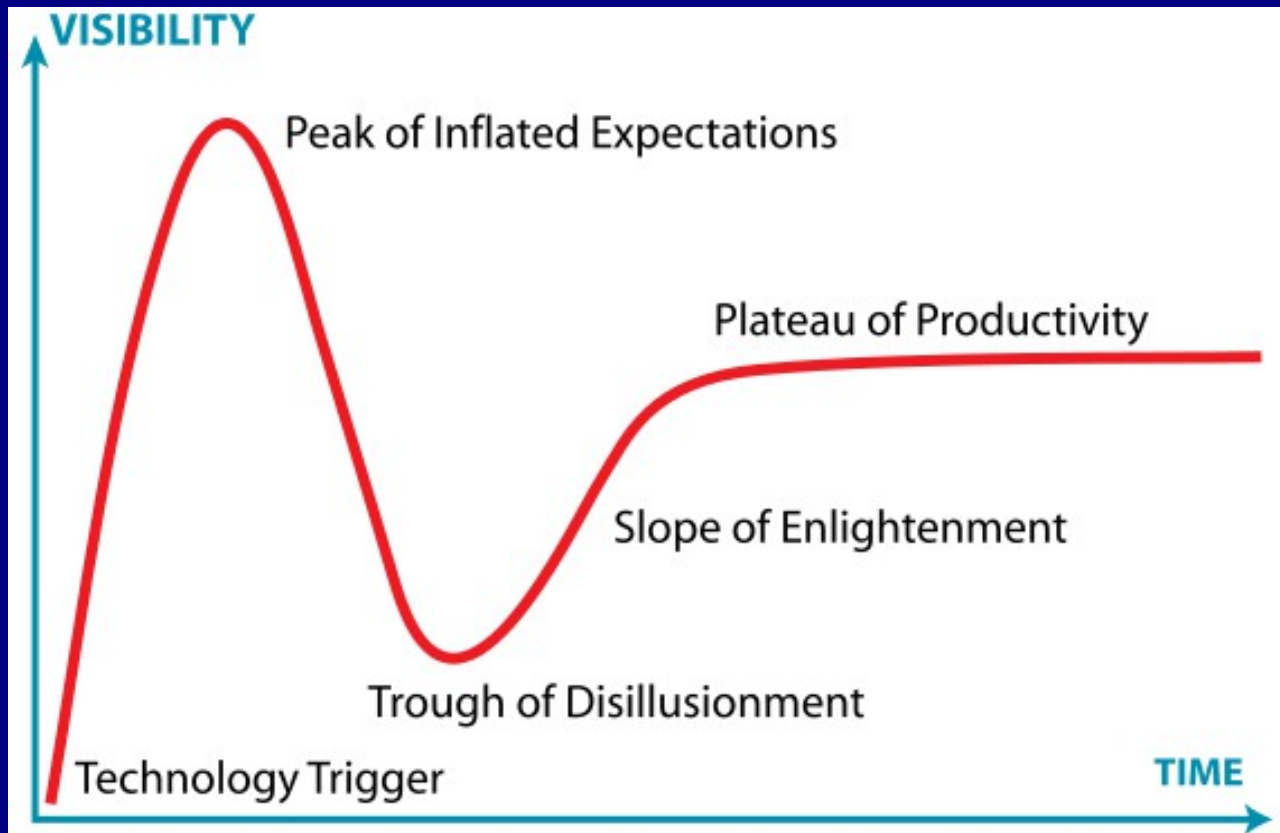
NoSQL & Big Data

Sławomir Świętoniowski

Stanowiska względem baz NoSQL

- Opinie na temat rodziny technologii NoSQL są mocno spolaryzowane:
 - Wielbiciele
 - wykorzystują aktywnie, testują, wdrażają
 - Przeciwnicy
 - skupiają się na słabościach i niedoskonałościach
 - Sceptycy
 - ignorują, czekają na ustabilizowanie technologii...

„Hype Cycle”



Źródło: http://en.wikipedia.org/wiki/Hype_cycle

Definicja baz NoSQL

- Literalnie No SQL, w praktyce powinno być raczej No RDBMS (był nawet pomysł by było to *NonRel*), ogółem **zbiór technologii bazodanowych, które starają się mierzyć z czterema współczesnymi problemami przy przetwarzaniu danych (*Volume, Velocity, Variability, Agility*)**.
- Bazy nierelacyjne istnieją od lat (a nawet istniały przed bazami relacyjnymi), **ich aktualny rozwój jest ściśle powiązany z rozwojem Internetu i problemami jakie pojawiły się przy przetwarzaniu gigantycznych ilości danych jakie wraz z nim zaczęły się pojawiać, w założeniu mają pokonywać ograniczenia modelu relacyjnego**.
- W przypadku gigantycznych zbiorów danych **skalowalność baz relacyjnych zaczyna być problemem**. Trudno w ich wypadku stosować skalowanie poziome (ang. *horizontal*) a preferowane skalowanie pionowe (ang. *vertical*) jest kosztowne.
- Więcej informacji na temat definicji tutaj: <http://nosql-database.org/>

Ograniczenia modelu relacyjnego

- Z góry określona struktura danych, którą modelujemy na samym początku.
- Dane są kompletne, powiązania między danymi nie ulegają istotnym zmianom, z góry określamy indeksy, które optymalizują nam wyszukiwanie potrzebnych informacji.
- Model nie skaluje się aż tak dobrze, w przypadku „Big Data” zaczynają się pojawiać problemy.
- Można częściowo pokonywać te ograniczenia, ale w istocie jest to odejście od modelu relacyjnego: <http://backchannel.org/blog/friendfeed-schemaless-mysql>

Podejścia do skalowalności

- **Scaling-up** – skalowalność pionowa, zwiększanie mocy obliczeniowej przez rozbudowę pojedynczej maszyny, relatywnie prosta, ale od pewnego momentu (skali) bardzo kosztowna.
- **Scaling-out** – skalowalność pozioma, zwiększenie mocy obliczeniowej przed dodawanie kolejnych maszyn (przetwarzanie równoległe/rozproszone), komplikuje projekt i implementację, ale daje praktycznie nieograniczone możliwości rozbudowy.

Przetwarzanie danych i Big Data

- Pojęcie „Big Data” może być interpretowane różnie, jedno z bardziej racjonalnych określeń, to stwierdzenie, że **mamy z nim do czynienia, jeżeli ilość danych do przetworzenia przekracza możliwości pojedynczej maszyny.**
- Mniej więcej taką definicję można znaleźć na Wikipedii: http://en.wikipedia.org/wiki/Big_data
- Ciekawa jest wizja przyszłości jeżeli chodzi o ilość przetwarzanych danych, która jest prezentowana w „*The Digital Universe Decade – Are you ready?*”: <http://www.emc.com/collateral/demos/microsites/emc-digital-universe-2011/index.htm>
- Ale są też głosy, że całe to pojęcie to tylko chwilowa fanaberia i marketing: <http://www.simple-talk.com/sql/database-administration/big-data-is-just-a-fad/>

Podejście wykorzystujące bazy NoSQL

- Można tu mówić o częściowym pokonywaniu ograniczeń modelu relacyjnego, ale, dzieje się to kosztem:
 - braku lub ograniczenia wsparcia dla transakcji i mechanizmów sprawdzających spójność danych
 - utrudnieniem odpytywania o potrzebne informacje
 - nieco mniej elastycznym podejściem do indeksowania

Geneza baz NoSQL

- Google
 - 2003 – „*The Google File System*”: <http://research.google.com/archive/gfs.html>
 - 2004 – „*MapReduce: Simplified Data Processing on Large Clusters*” <http://research.google.com/archive/mapreduce.html>
 - 2006 – „*Bigtable: A Distributed Storage System for Structured Data*” <http://research.google.com/archive/bigtable.html>
 - 2006 – „*The Chubby lock service for loosely-coupled distributed systems*” http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/archive/chubby.html
- Amazon
 - 2007 – „*Amazon's Dynamo*” http://www.allthingsdistributed.com/2007/10/amazons_dynamo.html

Bazy NoSQL i wsparcie dla Big Data

- Poprzez zapewnienie efektywnego **przechowywania i pobierania gigantycznych ilości danych**, zapewniając dużą odporność na awarie.
- Wykorzystanie **przetwarzania równoległego**, możliwie **największą skalowalność**, pożądane jest uzyskanie stałego czasu odpowiedzi przy zmiennym obciążeniu.
- Wsparcie dla przechowywania i przetwarzania danych, których **struktura nie jest z góry określona** i może się znacząco zmieniać w czasie.

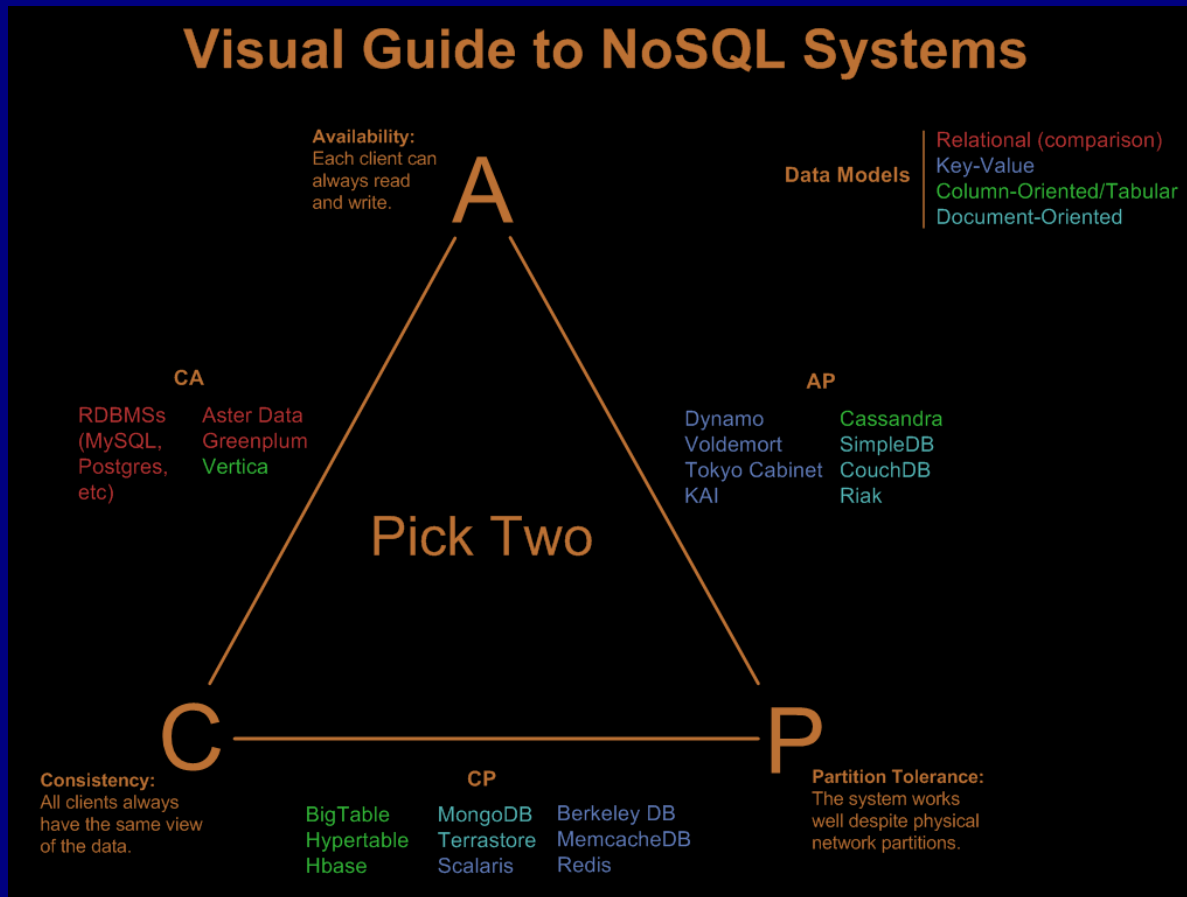
MapReduce

- Opatentowany przez Google model programowania równoległego wykorzystywanego do przetwarzania ogromnych zbiorów danych z wykorzystaniem klastrów komputerów.
- Źródłem jest programowanie funkcjonalne w którym występują pojęcia funkcji **map** i **reduce** (fold): <http://en.wikipedia.org/wiki/MapReduce>
- Powyższe podejście stało się inspiracją dla wielu projektów Open Source i w efekcie baz NoSQL.

CAP Theorem i BASE

- CAP Theorem – teoria zgodnie z którą nie możemy w systemie rozproszonym uzyskać jednocześnie: spójności (ang. *consistency*), dostępności (ang. *availability*) i tolerancji dla partycji (ang. *partition tollerance*).
- BASE (ang. *Basically Available, Soft state, Eventual consistency*) – alternatywa dla modelu ACID, oferująca „ewentualną” spójność.
- I jedno i drugie jest bardzo ważne w kontekście rozwiązań NoSQL.
- Ciekawa prezentacja na ten temat:
<http://www.slideshare.net/larsga/nosql-databases-the-cap-theorem-and-base>

CAP Theorem i bazy NoSQL



Źródło: <http://blog.nahurst.com/visual-guide-to-nosql-systems>

Klasyfikacja baz NoSQL

- „Bazy kolumnowe” (ang. *sorted ordered column-oriented stores*):
 - HBase, Hypertable, Cloudata
- „Bazy klucz-wartość” (ang. *key/value stores*):
 - Oracle Berkeley DB, EHCACHE, Memcached, Membase, Kyoto Cabinet, Redis, Cassandra, Voldemort, Riak
- „Bazy dokumentowe” (ang. *document databases*):
 - MongoDB, CouchDB
- „Bazy grafowe” (ang. *graph databases*):
 - Neo4j, FlockDB
- Bazy obiektowe (ang. *object databases*):
 - Versant Object Database, db4o
- Bazy XML (ang. *xml databases*):
 - BaseX, eXist
- Powyższe jest dość luźną klasyfikacją, gdyż definicja NoSQL jest zbyt mało precyzyjna, nieco inną klasyfikację można znaleźć np. tutaj: <http://en.wikipedia.org/wiki/NoSQL>
- Bardziej wyczerpująca lista produktów klasyfikowanych jako NoSQL może zostać odnaleziona w tym miejscu: <http://nosql-database.org/>

„Bazy kolumnowe”

- Definicja: http://en.wikipedia.org/wiki/Column-oriented_DBMS
- Przechowywanie danych w kolumnach nie w wierszach.
- Efektywne wykorzystanie przestrzeni dyskowej, szybkie pobieranie danych.
- Nieco więcej o zaletach:
 - <http://horicky.blogspot.com/2010/10/bigtable-model-with-cassandra->

HBase (Apache Hadoop)

- <http://hbase.apache.org/>
- <http://hadoop.apache.org/>
- <http://radar.oreilly.com/2012/01/microsoft-big-data.html>
- Wykorzystujący:
 - <http://wiki.apache.org/hadoop/PoweredBy>
 - Amazon, EBay, Facebook, ...
- Instalacja:
 - <http://alans.se/blog/2010/hadoop-hbase-cygwin-windows-7-x64/>



„Bazy klucz-wartość”

- Prezentacja: <http://www.slideshare.net/marc.seeger/keyvalue-stores-a-practical-overview>
- Można je porównać do prostych tablic haszowanych, mamy klucz poprzez który odwołujemy się do konkretnej wartości.
- Potencjalnie najbardziej skalowalne.
- Bardzo prosty (w efekcie ograniczony) model danych.
- Ciekawy dokumentach o tego rodzaju bazach:
<https://www.simple-talk.com/cloud/data-science/data-science-lab/>

Cassandra



- <http://cassandra.apache.org/>
- Wykorzystujący:
 - <http://www.datastax.com/cassandrausers>
 - Facebook, Twitter, Cisco Webex, Digg, ...
- Instalacja:
 - <http://coderjournal.com/2010/03/cassandra-jump-start-for-the-windows-developer/>
- Programowanie:
 - <http://www.ridgway.co.za/archive/2009/11/06/net-developers-guide-to-getting-started-with-cassandra/>

„Bazy dokumentowe”

- Definicja: [http://
en.wikipedia.org/wiki/Document-oriented_database](http://en.wikipedia.org/wiki/Document-oriented_database)
- Pozwalają na przechowywanie „dokumentów” o praktycznie dowolnej strukturze.
- Ciekawy opis pracy z bazami dokumentowymi:
<https://www.simple-talk.com/cloud/data-science/data-science/>
L

MongoDB

- <http://www.mongodb.org/>
- Wykorzystujący:
 - <http://www.mongodb.org/display/DOCS/Production+Deployments>
 - SAP, SourceForge, MTV Networks, ...
- Instalacja:
 - <http://www.mongodb.org/display/DOCS/Quickstart+Windows>
- Programowanie:
 - <http://www.mongodb.org/display/DOCS/CSharp+Driver+Tutorial>



„Bazy grafowe”

- Definicja: [http://
en.wikipedia.org/wiki/Graph_database](http://en.wikipedia.org/wiki/Graph_database)
- Główni gracze: [http://
www.readwriteweb.com/cloud/2011/04/5-graph-databases-to-watch/](http://www.readwriteweb.com/cloud/2011/04/5-graph-databases-to-watch/)
- Umożliwiają reprezentacje danych, których modelowanie w bazach relacyjnych jest utrudnione.
- Ciekawy dokument o bazach grafowych: <https://www.simple-talk.com/cloud/cloud-data/data-science/graph-databases/>

Neo4j



- <http://neo4j.org/>
- <http://www.slideshare.net/peterneubauer/neo4j-5-cool-graph-examples-4473985>
- <http://highscalability.com/neo4j-graph-database-kicks-buttox>
- Wykorzystujący:
 - <http://neotechnology.com/customers/>
 - Adobe, Cisco, ...
- Instalacja:
 - <http://docs.neo4j.org/chunked/1.4/server-installation.html>
- Programowanie:
 - <http://blog.jayway.com/2010/04/16/neo4j-net-client-over-http-using-rest-and-json/>

Bazy obiektowe

- Definicja: [http://
en.wikipedia.org/wiki/Object_database](http://en.wikipedia.org/wiki/Object_database)
- Obchodzą ograniczenie modelu relacyjnego pozwalając wprost przechowywać w bazie danych obiekty aplikacji.

Db4o



- <http://www.db4o.com/>
- Wykorzystanie:
 - <http://www.db4o.com/about/customers/success/>
 - Seagate, Postbank, ...
- Instalacja:
 - <http://www.emphess.net/2010/01/23/a-simple-standalone-server/>
- Programowanie:
 - <http://tutorial.fyicenter.com/out.php?ID=2339>

Bazy XML

- Definicja: [http://
en.wikipedia.org/wiki/XML_database](http://en.wikipedia.org/wiki/XML_database)
- Wykorzystują język XML i technologie pokrewne (XPath, XQuery, XSTL) do przechowywania i przetwarzania danych.

BaseX



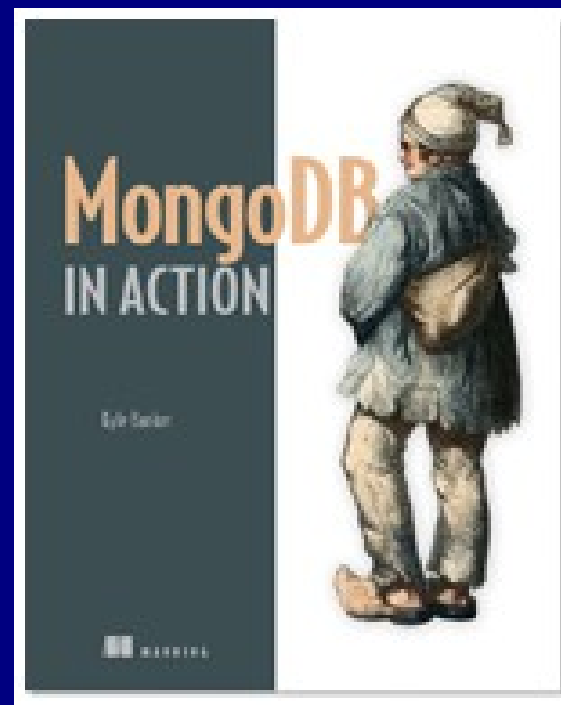
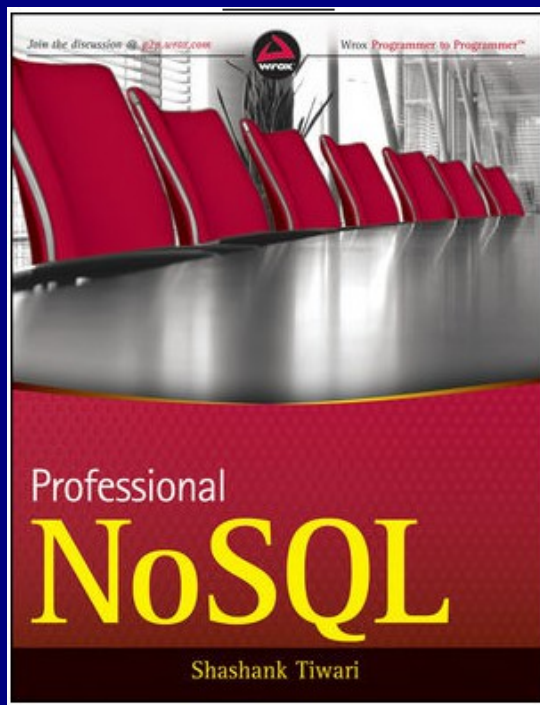
- <http://basex.org/>
- Wykorzystanie:
 - <http://basex.org/customers/>
 - Airbus, Max Planck Institute, ...
- Instalacja:
 - <http://docs.basex.org/wiki/Startup>
- Programowanie:
 - [http://docs.basex.org/wiki/Java Examples](http://docs.basex.org/wiki/Java_Examples)

Problemy z bazami NoSQL

- **Brak standardów**: tak co do modelu danych jak i języka wykorzystywanego do typowych operacji CRUD.
- Olbrzymi wybór, ale **brak zdecydowanych faworytów**.
- **Technologia nowa**, ciągle jeszcze **mało ustabilizowana**.
- Małe grono osób znających bazy NoSQL (po części konsekwencja ich różnorodności i braku standardów).
- Wspomniane wcześniej **ograniczenia dotyczące kwestii transakcji, odpytywania, indeksów**, w przypadku gdy potrzebujemy tych elementów bazy relacyjne są ciągle najlepszym wyborem.
- **Problematyczne podejście do kwestii bezpieczeństwa** (a w dużej mierze jego brak*):
[http://
www.darkreading.com/database-security/167901020/security/news/232400214/d](http://www.darkreading.com/database-security/167901020/security/news/232400214/d)

Do poczytania:

- S. Tiwari, „*Professional NoSQL*”
- K. Banker, „*MongoDB in Action*”



Do obejrzenia:

- M. Fowler, „Introduction to NoSQL”